Full paper

Distributed Service-Based Cooperation in Aerial/Ground Robot Teams Applied to Fire Detection and Extinguishing Missions

Antidio Viguria [a,*], **Ivan Maza** [b] **and Anibal Ollero** [a,b]

[a] Center for Advanced Aerospace Technologies (CATEC), 41309 Seville, Spain
[b] Robotics, Vision and Control Group, University of Seville, 41092 Seville, Spain

Received 3 October 2008; accepted 4 April 2009

Abstract
This paper presents a system for the coordination of aerial and ground robots for applications such as surveillance and intervention in emergency management. The overall system architecture is described. An important part for the coordination between robots is the task allocation strategy. A distributed market-based algorithm, called S + T, has been developed to solve the multi-robot task allocation problem in applications that require cooperation among the robots to accomplish all the tasks. Using this algorithm, robots can provide transport and communication relay services dynamically to other robots during the missions. Moreover, the paper presents a demonstration with a team of heterogeneous robots (aerial and ground) cooperating in a mission of fire detection and extinguishing.

Keywords
Distributed task allocation, market-based coordination, auctions, multi-robot, heterogeneous teams

1. Introduction

Surveillance and intervention in disaster scenarios are very valuable missions for robot teams. In fact, many missions are almost impossible to accomplish with only one robot and the cooperation of heterogeneous robots is particularly useful. This is the case of fire detection and extinguishing. Fire detection may require the patrolling of robots, such as unmanned aerial vehicles, with infrared and visual cameras or specialized fire sensors in terrains that cannot be traversed with ground robots [3]. These aerial robots are usually very constrained in the payload required for fire extinguishing. Then, the intervention of ground robots is needed for the ex-

[*] To whom correspondence should be addressed. E-mail: aviguria@catec.aero

tinguishing phase [4]. Furthermore, small low-cost aerial robots are also constrained in time of flight and autonomy. Thus, they usually need to be transported to the particular areas to be patrolled. This transportation service could also be performed by suitable ground robots that can move near to the area to be patrolled carrying autonomous helicopters. Afterwards, those aerial vehicles could take-off and patrol the area searching for the fire, and could precisely localize it by means of their onboard cameras and sensors. Also, it is important to consider that many disaster scenarios, including fire detection and extinguishing, do not have the appropriate communication infrastructure or that this infrastructure could be damaged. Then, robots providing communication relay services are also very useful. In this scenario, the team of robots requires innovative cooperation methods considering their heterogeneity and, particularly, the role of the mentioned services.

An important issue in distributed multi-robot coordination is the multi-robot task allocation (MRTA) problem that has received a lot of attention in the last decade. It deals with the way that tasks are distributed among robots and requires us to define some metrics to assess the relevance of assigning given tasks to one or another robot. Different approaches have been used to solve this problem: centralized [5, 6], hybrid [7, 8] and distributed [9, 10]. Within the distributed approaches, the market-based approach [11] has become very popular since it offers a good compromise between communication requirements and the quality of the solution. It can be considered an intermediate solution between centralized and completely distributed since it makes decisions based on inter-agent communications transmitted at different time instances. This type of algorithm is more fault-tolerant than a centralized approach and can obtain more efficient solutions than a completely distributed approach.

Market-based approaches generally assume that each task can be executed completely by a single robot. However, this could not be the case, for example, in a surveillance or disaster scenario, in which a task consisting of transmitting images in real-time could require another robot to act as a communication relay. Our approach to solve this problem is based on the concept of service. If a robot cannot execute a task by itself, it asks for help and, if possible, another robot will provide the required service. Required services are generated dynamically and are necessary to successfully complete their associated task.

It is widely accepted that one of the main advantages of multi-robot systems with respect to a stand-alone robot is their capability to perform tasks that can be impossible for a single robot. In this paper, a new task allocation protocol (S + T), designed to exploit this characteristic and based on the concept of services, is described. This protocol is based on a distributed market-based approach and could be considered as an extension of the SIT algorithm [12].

A similar idea is presented in Ref. [13], where soft temporal constraints were considered using master/slave relations, and also in Ref. [14], where the efficiency of the solution is increased considering at the same time the decomposition and allocation of complex tasks in a distributed manner. However, the potential execution loops associated with the relation between tasks and services that could lead to deadlock situations were not addressed in these works. In this paper, these deadlocks are solved by a novel distributed algorithm. Moreover, the parameters of our algorithm can be adapted to give priority to either the execution time or the energy consumption (i.e., the sum of the distances traveled by each of the robots) in the mission.

The main contribution of this paper is the application of a novel task allocation algorithm, which introduces the concept of services to enable the execution of tasks that need the cooperation of more than one robot, to a fire detection and extinguishing mission with heterogeneous robots that not only involves exploration, detection and monitoring, but also actuation in order to extinguish the fire.

This work extends the results presented in SSRR 2006 [1] and ICRA 2008 [2] where only some preliminary experiments and the distributed task allocation ($S + T$) algorithm were presented. In the present paper, we have integrated the $S + T$ algorithm within an architecture for heterogeneous robots, and a fire extinguishing demonstration with aerial and ground robots is presented.

The paper is organized as follows. The overall architecture of the multi-robot team is presented in Section 2, describing the different subsystems involved: the robot themselves, the communication system, and a monitoring and planning station. The next section is focused on the distributed task allocation system, presenting our novel algorithm called $S + T$ and describing the concept of tasks *versus* services. Also, a deadlock problem regarding the distributed execution of synchronized tasks is stated and our solution described. The performance of the $S + T$ algorithm and its different characteristics are evaluated in simulation, and the results are shown in Section 4 with missions consisting of visiting several waypoints. The whole integrated system has been tested with promising results in a demonstration with a team of heterogeneous robots (aerial and ground) cooperating in a mission of fire detection and extinguishing (described in Section 5). Finally, conclusions and future work are discussed in Section 6.

2. Multi-robot Architecture Overview

A complete system architecture has been developed in order to make the integration of heterogeneous robots easier. Disaster scenarios usually need various robots with different characteristics. This architecture is designed to reuse the common components to all the robots and minimize the time needed to incorporate a new robot to the system. A global view of the system components is presented in Fig. 1, where three main blocks can be identified:

- *Monitoring and planning station* (MPS). This provides means to the human operator for preparing plans, sending missions and monitoring the execution. It also encompasses the alarm monitoring station, which is in charge of performing autonomous cooperative perception processing [3], and specialized image processing activities (such as fire detection) providing different alarms to the

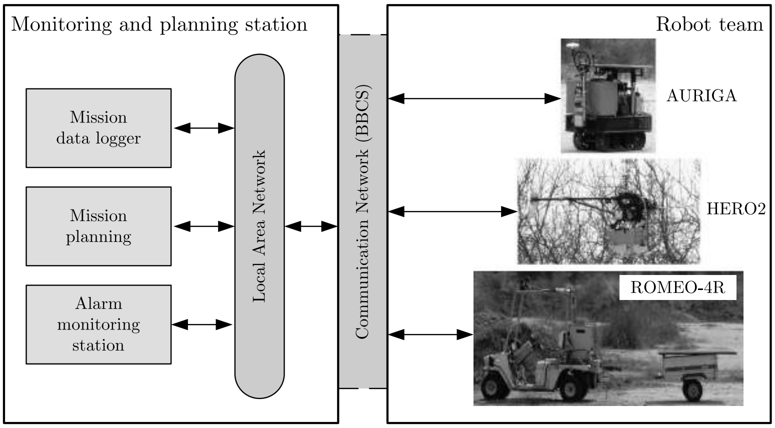


Figure 1. Global architecture illustrated with the vehicles used in the demonstration described in Section 5. The communication network is used for both the communication between the robots, and the communication between the robot team and the MPS.

operator. Finally, all the information related to each mission is saved in a database for mission debriefing purposes.

- *Communication network*. This is the support for every communication between the different components of the system. It deals with task requests/status and data transmissions, such as images or robot telemetry. It should be noted that the robots currently integrated in the architecture are using the BBCS (BlackBoard Communication System) developed by the Technical University of Berlin [15] and tested in the COMETS Project [16] funded by the IST Programme of the European Commission. It is a robust communication system implemented *via* a distributed shared memory, the blackboard (BB), in which each network node has a local copy of the BB portion it is accessing.

- *Robot team*. The software architecture of each robot (see Fig. 2) is based on hierarchical layers, with the higher levels 'decoupled' from the particular characteristics of the robot. Therefore, the high-level software modules can be reused in different types of robots with minor changes. Three layers have been developed: RAL, MML and RIL. Since this paper only deals with high-level aspects of the robot team behavior, only the RAL (see Fig. 2) will be commented on in this section. The MML layer manages the modules implemented in the RIL layer with the different robot functionalities. For example, the MML layer starts and stops the necessary modules (RIL) for each task, makes the appropriate connections between the modules, and passes to them the correct parameters.

Regarding the task allocation process, it should be pointed out that this architecture supports two different modes of operation (see Fig. 2):

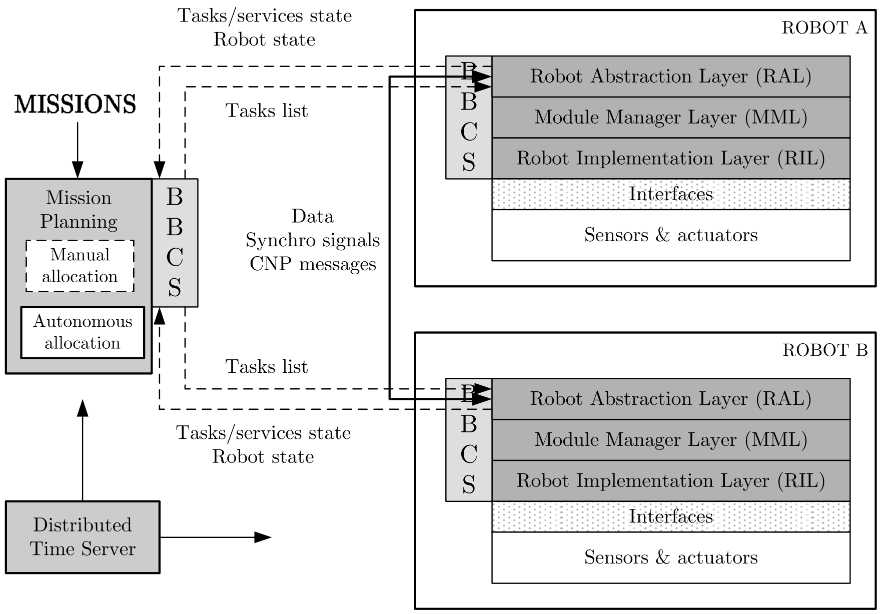


Figure 2. Robot team architecture (dashed lines correspond to the manual allocation mode and solid lines to the autonomous allocation mode). Each robot has a layered architecture with three layers: robot abstraction layer (RAL), mobile manager layer (MML) and robot implementation layer (RIL). The RAL layer is the one that deals with the task allocation and synchronization.

- *Manual allocation mode*. The human operator allocates individual elementary tasks and sequences of elementary tasks from the MPS to the robots. Each RAL manages those tasks and reports the robot status during the mission execution.
- *Autonomous allocation mode.* A Distributed Task Allocation Module (DTAM) in the RAL allows us to autonomously negotiate task allocation in a distributed way. In this mode, the MPS should only provide a list of elementary tasks to be executed by a group of robots.

The system architecture supports the use of both modes of operation during a mission execution, allowing the operator, for example, to manually allocate a task to a given robot, whereas the rest of tasks are being allocated autonomously.

When the manual allocation mode is used, the system can be considered centralized and it is very sensitive to any communication failure. On the other hand, when the autonomous allocation mode is used, the proposed system is based on a distributed architecture. Therefore, communication failures will not affect seriously the operation of the robots, allowing the execution of most of the tasks. However, it is true that the Distributed Task Allocation algorithm decreases its efficiency since robots that are out of communication coverage or with communication problems will not take part in the allocation process. In fact, a robot that experiences a complete failure of its communication system can no longer be considered part of the cooperative team and it should go back to the MPS for repairing.

The main disadvantage in the presented architecture, regarding communication losses, happens when communication failures take place during the negotiation protocol. The current DTAM assumes that no communication cut will occur during the task allocation process. However, in most of the cases, the negotiation process takes place at the beginning of the mission when the robots are not moving and no changes in the communication topology occurs, minimizing the possibility of a communication cut.

2.1. RAL

As the RAL has been designed to be mainly independent from the particular characteristics of the robot, a similar implementation can be used in different heterogeneous robots. Some modules in this layer are:

- *DTAM*. This module allows robots to autonomously negotiate task allocations in a distributed way by using a market-based approach. At this point, different auction algorithms have been implemented: SIT, SET [12], and S + T. This last algorithm creates services dynamically and handles their allocation when a robot cannot execute a task by itself (see Section 3).
- *Task Manager Module*. This module manages tasks and their states. It receives tasks from the MPS (manual allocation mode) or from the DTAM (autonomous allocation mode). Furthermore, it sends basic tasks to the lower software layers and reports the state of the tasks to the MPS.
- *Task Synchronization Module*. Tasks can be synchronized using preconditions, i.e., a task will not be executed until all its preconditions are satisfied. Once every task is allocated, the Task Manager Module communicates the preconditions of each task to this module. Before a task is going to be executed, the Task Manager Module asks this module if the task can start. This information is available because when a robot finishes a task, it will transmit a message to the rest of the team.
- *Environment Model Module*. This builds a local map of the environment and transmits it to other robots in order to combine them and generate a more complete and accurate model for the whole team [3, 17].

3. Services and Tasks: S + T Algorithm

As with any other market-based algorithm based on the Contract Net Protocol [18, 19], there are two roles (bidders and auctioneer) that are played dynamically by the robots. The auctioneer is the agent in charge of announcing the tasks and selecting the best bid from all the received bids. The algorithms associated with each role are detailed in Algorithms 1 and 2. In the bidding process, when a robot needs a service to execute a given task, it will bid initially with just the cost of the task (because

Algorithm 1.
S + T auctioneer algorithm

```
if there is any task to announce then
  announce task
  while timer is running do
    receive bids
  end while
  calculate best bid (lowest cost)
  if best bid is lower than the auctioneer bid then
    if best bid requires a service then
      allow robot to start a new auction in order to find a robot who can execute that service
    end if
    wait until the second auction is finished and the total cost of the task (including the service cost)
    is sent
    send task to best bidder taking into account the updated bids
  end if
  delete task from announcement list
  if task has an associated service then
    send a message to the robot that will execute the service in order to delete it from its local plan
  end if
end if
```

Algorithm 2.
S + T bidder algorithm

```
a new message is received
if new message is a task announcement then
  compute the optimal insertion point for the task in the local plan
  calculate bid (marginal cost)
  if the task requires a service then
    send initial bid to the auctioneer and indicate that a service is needed
  else
    send bid to the auctioneer
  end if
else if new message allows to ask for a service then
  start a new auction in order to find a robot that can execute the service
  receive all the bids for the service
  calculate the complete cost for the task including the cost for the service
  send the new cost to the auctioneer
else if new message is a task award then
  insert task in the local plan in the position calculated before
  add task in the announcement list
  if the task needs a service, allocate the service to the robot that won the auction
  if the cost of any allocated service (in case it exists) has changed because of the insertion of the
  new task in the local plan then
    send the new cost of the service to the robot with the task
  end if
end if
```

it still does not know the cost of the required services) labeling the message to the auctioneer as ‘provisional’. The auctioneer will evaluate all the bids and if the best bid requiring a service is better than the best bid without the need of a service, the robot requiring the service will start another auction in order to find which robots can perform that service. When this second auction is finished, the robot will send to the auctioneer the complete cost of the task, including the cost of the associated services. Afterwards, the auctioneer will decide which robot executes the task based on the updated costs. If a task is allocated to a robot requiring a service, that service will be allocated also at the same time.

It should be pointed out that both the protocol used to allocate the services and the algorithm to allocate the tasks are based on the SIT algorithm presented in Ref. [12]. The only differences are:

- Services cannot be reallocated dynamically.
- When a robot that will execute a service changes its local plan, it has to report the new cost of the service to the robot that required it (that can start another auction to check if a different robot has a lower cost now for that task).

A relevant feature of the protocol is that services can be allocated recursively, i.e., a robot that executes a service could also require another service to accomplish the first one and in this way to any number of recursive services. Therefore, the algorithm takes full advantage of the possibilities that a team of robots can offer (it is even possible to execute missions with a task involving the whole team).

In order to illustrate this characteristic, a surveillance mission will be considered. The mission consists in transmitting information from a certain area to a base station in real-time. The robot has to be within the communication range of the base or in the range of another robot acting as a communication relay. As can be seen in Fig. 3, the transmission to the base requires two robots acting as communication relays. The most relevant messages involved in the negotiation process are represented in

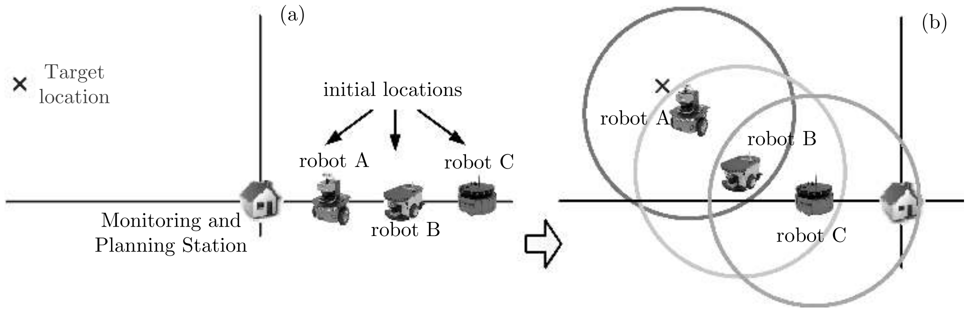


Figure 3. Example of multiple recursive services required to accomplish one task consisting of transmitting images from a target location: (a) initial positions of the robots and the monitoring station and (b) final assignment of tasks and services that allows robot A to transmit images to the monitoring station using robots B and C as communication relays.

the diagram depicted in Fig. 4 and a simulation of this mission using helicopters as robots can be downloaded from http://grvc.us.es/ADV_ROBOT_2010.

It should be pointed out that when a robot announces a service required for a certain task, the robot that will execute that task cannot take part in the auction process for the service.

The use of services increases the cooperation among robots and allows us to achieve missions that could be impossible using a regular task allocation algorithm, e.g., transmitting images in a surveillance mission from a position that does not have direct coverage with the base of operations. However, services can also increase the total time of the mission since more than one robot could be used to execute one task and, therefore, less tasks can be executed 'in parallel'. In this context, if a robot can execute a task by itself with a larger cost than another robot using services, it should be decided which option is better. From our point of view, the answer to this question depends on the specific application and two different approaches have been developed to tackle different scenarios:

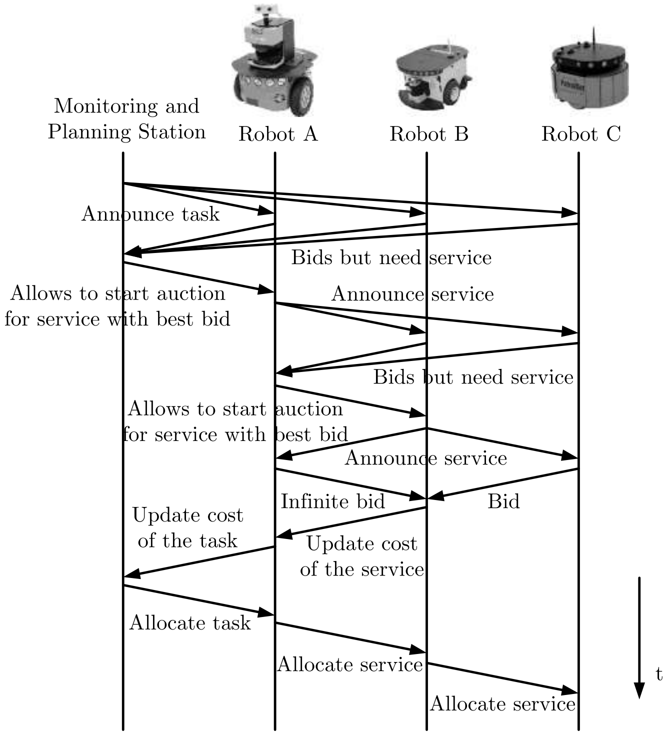


Figure 4. Messages interchanged in the negotiation process using the S + T algorithm for the example illustrated in Fig. 3 (one task requiring two services to be executed). When several robots ask for a service, only the robot with the lowest bid is allowed to start an auction for the service. For example, robot C asks for a service twice, but it is never allowed to start an auction, because the negotiation is over once robot A can execute the task using the communication relay services from robots B and C.

- In our first approach, tasks have a higher priority than services and, therefore, it should be applied to scenarios where the goal is to minimize the total execution time of the mission. Basically, when an auctioneer receives bids from robots and at least one of them does not require a service, the task will be directly allocated to it. This approach also needs less communication messages since services will be only considered when they are totally necessary for the success of the mission.
- In the second approach, the priority between the total time of the mission and the energy consumed by the team can be adjusted with a parameter α defined as:

  $$\alpha = \frac{P}{1 - P},$$

  where $P \in [0, 1]$ is the priority to minimize the total time of the mission. This parameter is used in the computation of the cost for the service:

  $$C_s = C_o \cdot (1 + \alpha \cdot L),$$

  where C_o is the original cost of the service, C_s is the new cost of the service and L is the level of the service, i.e., if it is the first service that depends on a task, L is equal to 1, if it is a service that depends on the first service, then L is equal to 2, and so on. This second parameter is used to penalize the use of more than one robot to execute one task. Moreover, when the use of services is unavoidable, L allows us to increase the priority of services that need less robots. The value of the parameter P should be selected depending on the type of mission. If it is more important to minimize the energy spent on the mission and the total time is not important, we should select $P = 0$, which means $\alpha = 0$. On the other hand, if we want to minimize the total time of the mission without considering a complete execution of all the tasks, we should select $P = 1$, which means $\alpha \to \infty$. In this case, services will not be considered and the algorithm will behave as the SIT market-based algorithm with local plans and reallocations.

3.1. Deadlock Situations

Until now, the allocation process of tasks and services has been presented, but not the synchronization issues related to the relation between tasks and services during the execution. From a general point of view, when the execution of tasks depends on others, the generation of deadlocks must be considered, and even more so when the process is distributed. It has been noticed in simulations that this problem appears frequently since each robot only has local information and there is no direct way to know if its particular local plan will generate a deadlock in the execution of all the tasks and services by the team of robots. For example, Fig. 5 shows how an execution loop can be generated using the S + T algorithm for

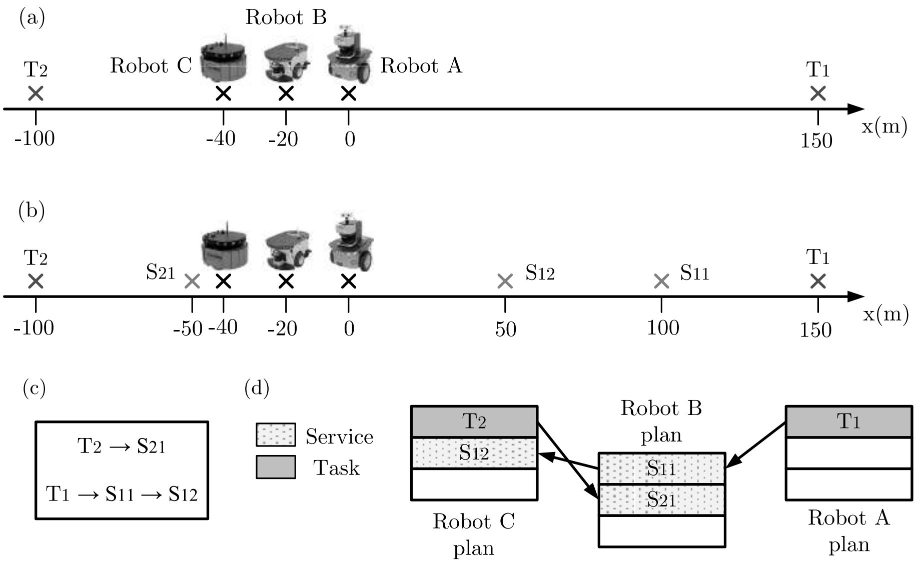


Figure 5. Example where a deadlock is generated since the execution of the tasks depends on the execution of services. (a) Initial position of the robots and the tasks to be allocated. Those tasks are represented by cross marks and consist of visiting target locations to transmit images to a monitoring station located in $x = 0$, assuming a radius of communication of 50 m. (b) Services (also represented by cross marks) needed to execute the tasks. (c) Relation of execution between the tasks and the services. (d) Relation in terms of execution in the final allocation using the S + T algorithm along with the plan of each robot (with the order of execution of its tasks and services). T_i represents the task with identifier i and S_{jk} means a service associated with the task j with level k.

a particular example with data transmission tasks and communication relay services.

This problem does not have an easy solution since robots only have knowledge of their own plans. It is also important to find an algorithm to solve this problem in a distributed way since the key idea is to have a whole functional robotic system that works without the presence of a centralized entity. Our solution is based on the use of ‘check loop’ messages, i.e., every time a robot wins a task, it will broadcast a message indicating the service associated with the new task (if it exists). The robot that has won that service will process the message and will send a message for every task or service that appears in its local plan before the mentioned service and has also a service associated to it. As is shown in Fig. 6, when a robot receives back a ‘check loop’ message with its ID, it will sell the task that provokes the loop and it will introduce it in a blacklist in order to avoid biding again for it. The use of a blacklist has the purpose to prevent the generation of allocation loops when the best two robots for a task are involved in an execution loop when they integrate the task in their local plans (i.e., they start to reallocate the task to each other and in both cases an execution loop is formed). Finally, the complete algorithm is shown in Algorithm 3.

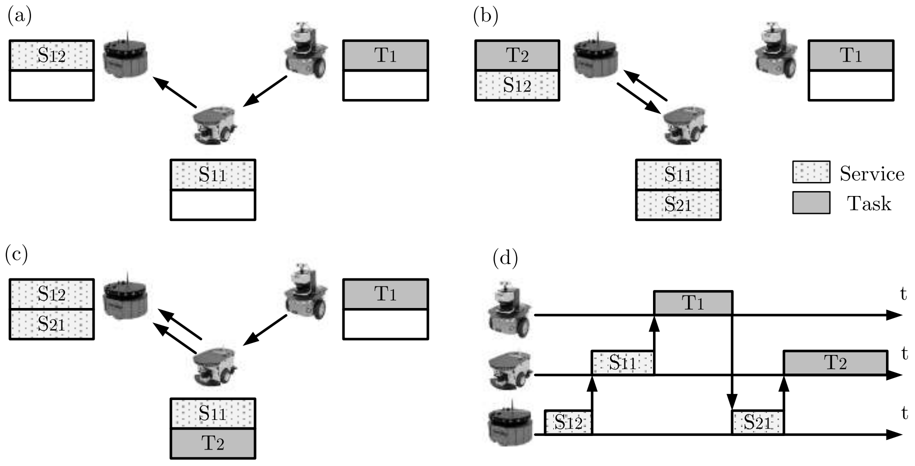


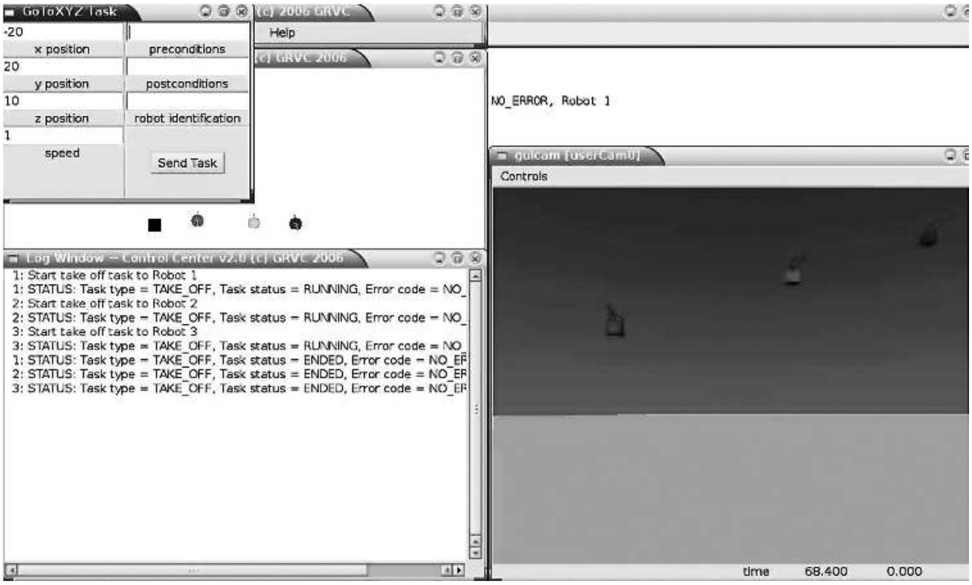
Figure 6. Considering the initial configuration presented in Fig. 5. (a) Allocation after the announcement of the first task and the path followed by the 'check loop' messages. (b) Allocation of the second task and the path of the 'check loop' messages that detects the execution loop. (c) Allocation after the reallocation of the task and how the execution loop has been removed. (d) Final execution sequence of the different tasks and services with one timeline per robot (the arrows represent the required synchronization during the distributed execution).

Algorithm 3.
Distributed loop detection algorithm

```
wait until receive a 'check loop' message with a task or service that the robot has in the local plan
if ID message == robot ID then
  if task has an associated service then
    send a cancel service message
  end if
  delete task from won-tasks list (loop detected)
  insert task in black-tasks list
  insert task in announcement-tasks list
else
  move to the initial position of the local plan
  repeat
    if task or service has a service associated to it then
      send 'check loop' message
    end if
    next task or service in the local plan
  until task or service != task received in the 'check loop' message
end if
```

4. Simulation Results

A multi-robot simulator based on the system architecture defined in Section 2 has been programmed. An important objective in the design of the simulator was the reusability of the code in the real robots. For this reason, the simulator, as well as the



-20	
x position	preconditions
20	
y position	postconditions
10	
z position	robot identification
1	
speed	

```
NO_ERROR, Robot 1
```

```
1: Start take off task to Robot 1
1: STATUS: Task type = TAKE_OFF, Task status = RUNNING, Error code = NO_
2: Start take off task to Robot 2
2: STATUS: Task type = TAKE_OFF, Task status = RUNNING, Error code = NO_
3: Start take off task to Robot 3
3: STATUS: Task type = TAKE_OFF, Task status = RUNNING, Error code = NO_
1: STATUS: Task type = TAKE_OFF, Task status = ENDED, Error code = NO_ER
2: STATUS: Task type = TAKE_OFF, Task status = ENDED, Error code = NO_ER
3: STATUS: Task type = TAKE_OFF, Task status = ENDED, Error code = NO_ER
```

```
time    68.400    0.000
```

Figure 7. Simulation environment and interface used to simulate the S + T algorithm with surveillance tasks where robots have to send back images from an area.

multi-robot architecture, is organized in three layers. The highest layer is independent of the type of robot and is the one aware of the existence of other robots. Thus, the task allocation algorithm is implemented in this layer. The other two layers are used to execute the different tasks allocated to the robot, and make the simulation of new algorithms easier by using a modular and component-based architecture. Therefore, each simulated robot runs the same software, which is onboard the real robot and uses a module in the lowest layer that emulates the hardware and its interfaces. Also, inter-processes communication has been implemented by using the BBCS [15]. This communication system allows us to run a multi-robot simulation in a single or multiple machines and is used also to communicate to the real robots as has been mentioned in Section 2.

In the simulations (see Fig. 7), surveillance missions were considered where robots had to send back images from a disaster area to the MPS. Therefore, a robot transmitting images had to be within the communication range of the MPS using its own communication device or using one or more robots as communication relays. For this particular scenario, the execution synchronization between tasks and services has been implemented using preconditions, i.e., a task cannot start until all the services associated to it have been executed. Moreover, the robot or robots that execute a service cannot start the next task or service in their local plan until the associated services have been completed.

Numerous simulations with different numbers of robots were performed for the surveillance missions mentioned above with several communication range values in a scenario of 1000 $\times$ 1000 m. In Fig. 8, it can be observed that the total distance

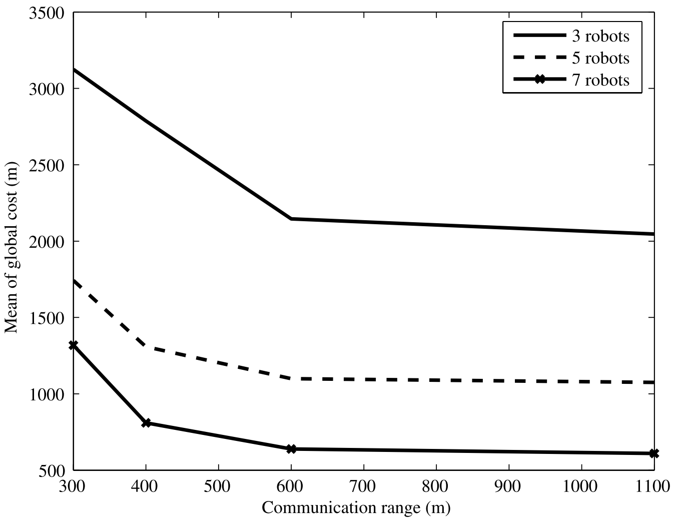


Figure 8. Mean of the total distance traveled by all the robots over 100 missions with different communication ranges, numbers of robots and five tasks.

traveled by all the robots decreases when the communication range increases as far as the probability to require a service decreases. The total distance traveled by all the robots is considered as a good measure of the energy spent during the mission. Moreover, the mean of the total distance traveled decreases when the number of robots increases due to the fact that a constant number of tasks is used in all the missions.

Table 1 shows the resulting mean values of some parameters in missions with five tasks, different numbers of robots and values for the communication range. The number of services executed increases when the communication range of the robots decreases and, as a logical consequence, the number of messages received by one robot and the total distance traveled by all of them also increases, as it was mentioned above. This means that the communication requirements and the energy needed to execute the mission will be higher when the number of services increases.

On the other hand, simulations have been run with different values of the α parameter that depends on $P \in [0, 1]$ (see Section 3). As it can be seen in Fig. 9, 100 random simulations have been executed for different values of P. $P = 0$ is an extreme value applied when the user wants to minimize the total distance traveled by all the robots in the mission in terms of energy and, therefore, the cost of the services is not modified. Also in Fig. 9, it can be observed how the maximum distance traveled by one robot decreases when P increases and, therefore, the time of the mission will be smaller (assuming that all the robots move at the same speed) because of the penalization of the costs associated with the services. However, if the execution time is critical, with $P = 1.0$ the S + T algorithm services are not considered and some tasks could be undone (mission partially accomplished). Figure 10

Table 1.
Results with five tasks, different numbers of unmanned aerial vehicles (UAVs) and values for the communication range

UAVs	Communication range (m)	Total distance (m)	Messages received	Services
3	600	2145.15	47.96	0.56
	400	2786.52	80.32	2.44
	300	3125.23	150.45	4.36
5	1100	1075.23	48.06	0.0
	600	1099.43	52.3	0.30
	400	1307.97	85.66	1.36
	300	1742.34	164.87	3.45
7	1100	609.14	45.06	0.0
	600	638.42	45.8	0.24
	400	810.23	79.76	1.24
	300	1318.31	142.96	2.76

The means of the values from 100 random missions are shown, where total distance is the distance travelled by all the UAVs, messages received is the number of messages received by one UAV due to the S + T algorithm and number of services is related to the ones executed by one UAV.

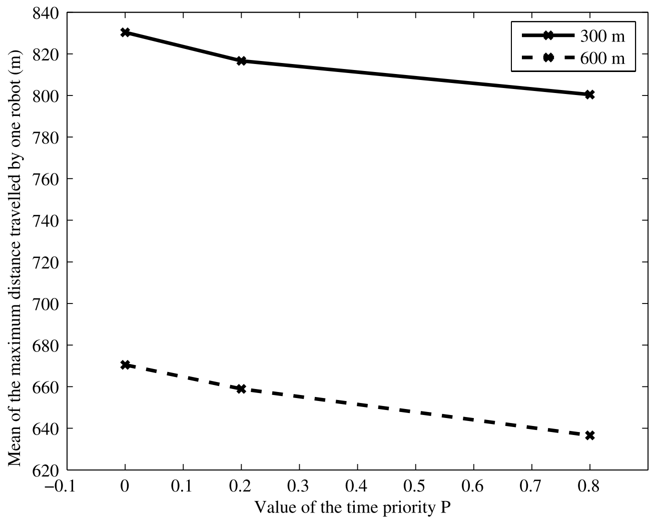


Figure 9. Mean of the maximum distance traveled by one robot over 100 missions with 300 and 600 m as the communication range, and five robots and tasks.

shows the mean of the number of tasks executed over 100 missions with different values for the communication range and with $P = 1.0$. Up to 600 m, it can be seen that a significant number of tasks cannot be accomplished for the group of robots if the use of services is not considered. Therefore, we have to be careful when the parameter P is equal to 1.0 and a given mission needs services to execute most of the

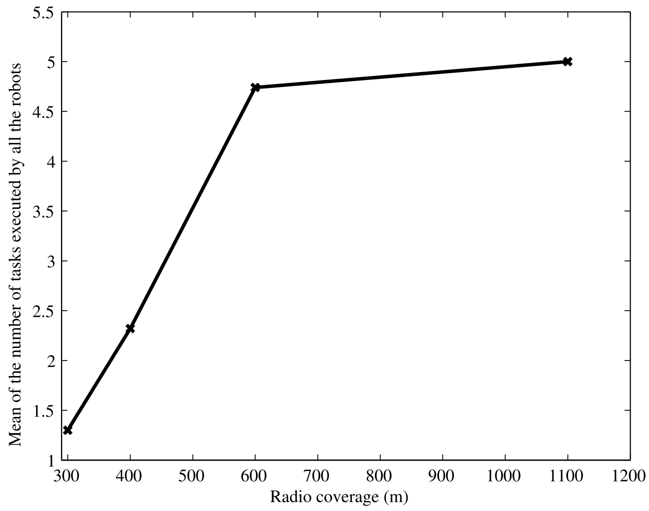


Figure 10. Mean of the number of tasks executed by all the robots over 100 missions with different values of the communication range, and five robots and tasks. The use of services is not considered in this simulations, i.e., $P = 1.0$ or $\alpha \to \infty$.

tasks. In that case, the time of the mission will be minimized, but many tasks will not be executed. Then, it is advisable to only use $P = 1.0$ when most of the tasks can be executed without services and the execution time of the mission is very critical.

5. Experimental Demonstration

A demonstration was conducted in the 'Alamillo' park in the city of Seville, in cooperation with the University of Malaga. Three different robots were involved: (i) the autonomous ground vehicle ROMEO-4R developed by the GRVC at the University of Seville, provided with a trailer for helicopter take-off and landing, (ii) the helicopter HERO2 also developed by GRVC, and (iii) a mobile fire extinguisher unit. This unit was the all-terrain tracked robot AURIGA developed by the University of Malaga, which was provided with a conventional fire extinguisher. The area considered for the demonstration was around 1 km^2.

The demonstration was performed so as to be significant for many disaster management activities, with a team of robots performing the following activities: detection, confirmation, localization, monitoring and actuation. In this demonstration, the S + T algorithm was used with $\alpha = 0$ since the energy of the robots is important in disaster scenarios where robots should be operative the maximum possible time.

5.1. Tasks and Services Considered in the Experiments

The following subset of tasks was selected for the field experiments:

- `Go-to(P)` tasks: to visit a point `P` given by its GPS coordinates.
- `Survey-area(A,object)` tasks: to cover an area `A` given by a convex polygon searching for objects of interest. The local planner of the robot com-

putes a sequence of way-points to cover the area of interest easily and efficiently by back and forth motion along rows perpendicular to the sweep direction sending images to the alarm monitoring station and performing autonomous detection. The task finishes when the object is detected.

- `Extinguish(P)` tasks: to locate, point and activate a fire extinguisher attached to the robot in order to extinguish a fire around GPS coordinates `P`.
- `Monitor(object,final_state)` tasks: to monitor an `object` until its state changes to `final_state`.

On the other hand, the services considered were:

- `Transport(P)` services: some robots (e.g., ROMEO-4R in the experiment described in Section 5) are equipped with platforms allowing aerial robots to be transported from an initial location to a point `P`.
- `Communication-relay (CRP)` services: the alarm monitoring station should receive images from the area during the execution of a `survey-area` task. Thus, if the communication range does not allow this link, another robot (or a chain of robots) should provide the `communication-relay` service moving to a certain point `CRP`.

5.2. *Description of the Demonstration in the 'Alamillo' Park*

The goal of the mission was to detect a fire and extinguish it with the collaboration between the three robots mentioned above (ROMEO-4R, HERO2 and AURIGA). Some photographs of the demonstration are shown in Fig. 11. The mission execu-

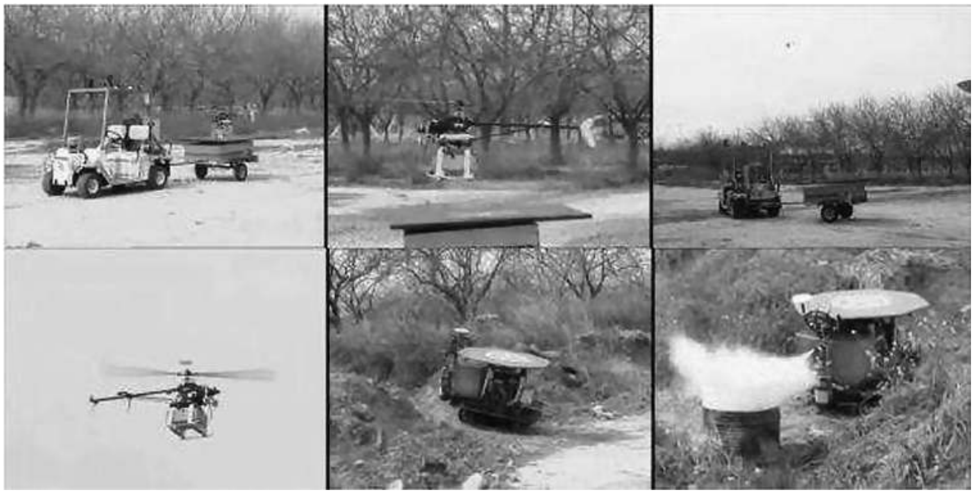


Figure 11. Some pictures of the demonstration in the 'Alamillo' park. The goal of the mission is to detect a fire and extinguish it with the collaboration of three robots: ROMEO-4R, HERO2 and AURIGA.

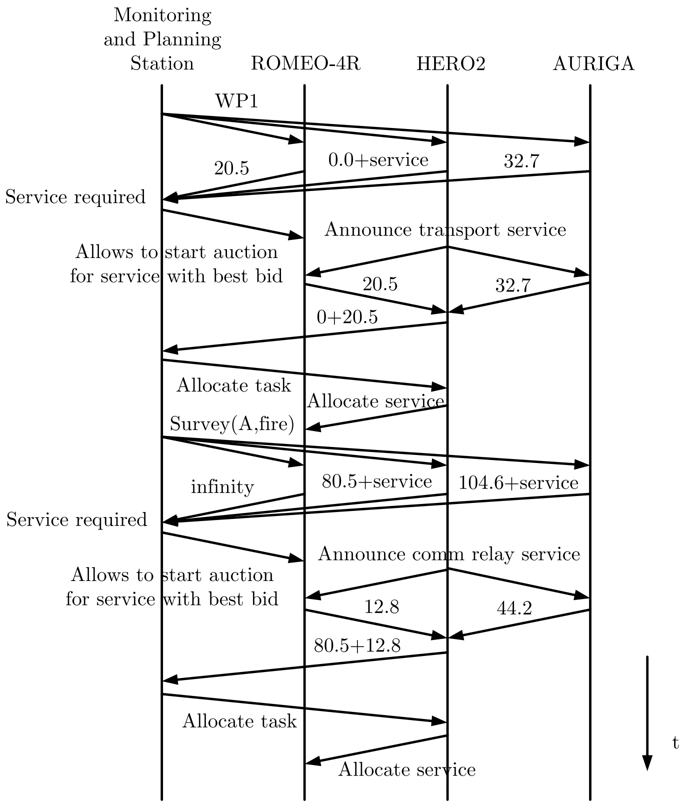


Figure 12. Messages interchanged in the initial negotiation process using the S + T algorithm for the demonstration in the 'Alamillo' park. HERO2 wins the `go-to(WP1)` task since it does not have any cost when a robot executes the transport service. This service is won by ROMEO-4R (it is closer than AURIGA to WP1. HERO2 also wins the `survey-area(A,fire)` task since after executing `go-to(WP1)` it needs to travel less distance to survey the area than AURIGA (due to the traversability of the terrain ROMEO-4R cannot execute this task and bids with infinity cost). Finally, ROMEO-4R wins the communication relay service since its cost is again smaller than the cost for AURIGA.

tion was as follows (Figs 12–14):

(i) At the beginning, the two ground vehicles were in their initial positions (marked as `H` in Fig. 13) and HERO2 (continuous line) was on the take-off/landing platform on the ROMEO-4R (dashed line) trailer. In the MPS, the human operator inserted a waypoint `WP1` to be visited as a starting exploration point and an area `A` (given by a polygon) to be surveyed.

(ii) Figure 12 shows the messages exchanged between the robots and the MPS. After the distributed negotiation process using the S + T algorithm (see Fig. 14), HERO2 won the `go-to(WP1)` task and the `survey-area(A,fire)` task. Each of these tasks has an associated service won by ROMEO-4R (the `transport(WP1)` service with the `go-to(WP1)`

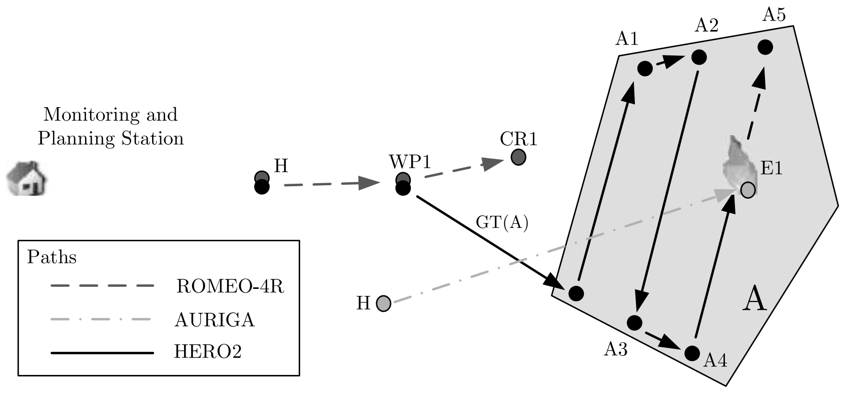


Figure 13. Diagram of the different elements involved in the mission execution. It represents the different tasks executed and an illustration of the paths traveled by each robot.

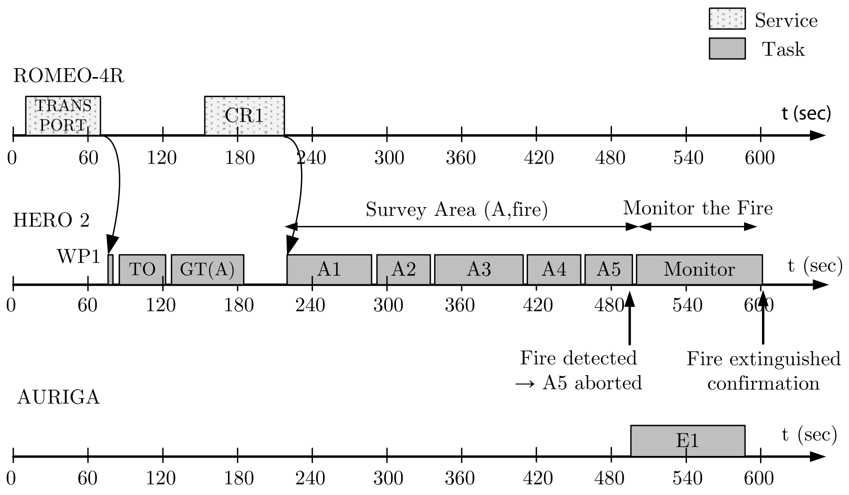


Figure 14. Tasks and services executed by each robot during the mission. This diagram shows the relation (preconditions) of the tasks and also a timeline with the duration of each task. All the acronyms are explained in the description of the demonstration, see Section 5.2.

task and the `communication-relay(CR1)` service with the `survey-area(A,fire)` task). Due to the limited flight autonomy of HERO2 (20 min), the transport service was expected to arise. The other service (`communication-relay`) was created since HERO2 has a limited communication range (virtually reduced for this demonstration) and it needs to send images back to the MPS during the `survey-area` task.

(iii) `Go-to(WP1)` was the first task to be executed since the cost of the HERO2 plan is minimized in this case. Then, ROMEO-4R moved to the `WP1` coordinates (see Fig. 13) with HERO2 on the take-off/landing platform. After reaching `WP1`, the first task and its associated service were completed. HERO2 started the `survey-area(A, fire)` task, which implied take-off (`TO`) and flying towards the `A` zone, `GT(A)`. At the same time, ROMEO-4R executed the service associated with this task and moved to the `CR1` point (Fig. 13) to act as a communication relay between HERO2 and the MPS.

(iv) When ROMEO-4R arrived to CR1, HERO2 started to survey `A` following a list of waypoints generated by its local planner (marked as `A1`, `A2`, `A3`, `A4` and `A5` in Fig. 13).

(v) HERO2 detected the fire [20] and geolocalized it [3, 17] computing the GPS coordinates of the fire. Then, HERO2 generated a new task (`Extinguish(E1)`) and inserted it in the multi-robot negotiation process. This task was allocated dynamically during the mission execution to the AURIGA (dash-dotted line) robot since it was the only robot with the required systems (the other two robots bid an infinite cost for this task). AURIGA started its execution, went close to the fire and activated the extinguisher using a teleoperation interface. Even though the execution of the AURIGA robot was teleoperated, we would like to clarify that the whole task allocation process was done autonomously using the $S + T$ algorithm. After the allocation, the task was transmitted to the teleoperation interface where the human operator executed the task. This fact also demonstrates that our architecture is flexible and enables us to combine robots with different degrees of autonomy.

(vi) At the same time, the fire detection alarm was sent by HERO2 to the alarm monitoring station that automatically created a `Monitor(fire, extinguished)` task and started an allocation process. This task was clearly allocated to HERO2 since it was the closer robot to the fire. HERO2 was executing this task that finishes when the fire is extinguished (mission completed).

A video of the demonstration can be downloaded from http://grvc.us.es/ADV_ROBOT_2010. This experiment has demonstrated the coordination among ground and aerial robots using a distributed task allocation system based on a new market approach. Moreover, it has been show that our system can handle tasks created dynamically during the mission execution. The use of tasks and services allows us to take advantage of the different characteristics available in an heterogeneous team of robots. The extension of this system to a larger number of robots can be accomplished easily due to the distributed nature of the system, as was demonstrated in the simulations.

6. Conclusions and Future Work

The cooperation of heterogeneous (aerial and ground) robots is a promising approach for fire detection, localization and extinguishing. This scenario requires the application of MRTA methods. In this paper, a novel algorithm ($S + T$) that considers services for the distributed solution of the MRTA problem has been presented. This protocol allows a team of robots to achieve tasks that could not be executed by a single robot. It does not just coordinate the robots, but rather introduces co-operation among them in order to increase the capabilities of the robot team. Also, a modification of this algorithm that allows us to give priority either to the execution time or to the global cost of the mission has been explained. The problem of execution loops that appears from the relation between tasks and services has been stated, and a distributed algorithm that solves this problem presented.

The S + T protocol has been implemented, simulated and tested in a demonstration with three robots. Simulations have shown that when the number of services increases, the number of messages and the global cost increases, which means that the communication requirements and the energy required to execute the mission will be larger. Also, the use of a larger number of services increases the maximum cost per robot and, therefore, the total time of the mission will be longer. Moreover, we have shown the effects of some values of the parameter α used to adapt our S + T algorithm to different types of objectives. Finally, regarding the demonstration, it should be pointed out that this is the first experiment of these characteristics and it involved not only exploration, detection and monitoring, but also actuation in order to extinguish the fire.

Future work includes evaluating the impact of partial or total communication and robot failures on the performance of the algorithms. Also, we propose to implement an algorithm that will change automatically the value of the parameter α, so the task allocation algorithm can be adapted autonomously to different kind of missions.

Acknowledgements

This work was partially funded by the European Union AWARE Project (IST-2006-33579), the CONET Network of Excellence (INFSO-ICT-224053), and the ROBAIR Project of the Spanish Research and Development Program (DPI2008-03847). We would like to thank the University of Malaga for their support and help in the demonstration, specially to Alfonso J. Garcia-Cerezo.

References

1. I. Maza, A. Viguria and A. Ollero, Networked aerial-ground robot system with distributed task allocation for disaster management, presented at: *IEEE Int. Workshop on Safety, Security and Rescue Robotics*, Gaithersburg, MD (2006).
2. A. Viguria, I. Maza and A. Ollero, S + T: an algorithm for distributed multirobot task allocation based on services for improving robot cooperation, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, pp. 3339–3344 (2008).
3. L. Merino, F. Caballero, J. R. Martinez-de Dios, J. Ferruz and A. Ollero, A cooperative perception system for multiple UAVs: application to automatic detection of forest fires, *J. Field Robotics* **23**, 165–184 (2006).

4. A. Gross, SALAMANDER: a firefighting robot, presented at: *IEEE Int. Workshop on Safety, Security and Rescue Robotics*, Gaithersburg, MD (2006).
5. B. Brumitt and A. Stenz, Grammps: a generalized mission planner for multiple mobile robots, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Leuven, vol. 2, pp. 1564–1571 (1998).
6. P. Caloud, W. Choi, J. Latombe, C. Le Pape and M. Yim, Indoor automation with many mobile robots, in: *Proc. IEEE Int. Workshop on Intelligent Robotics and Systems*, Ibaraki, vol. 1, pp. 67–72 (1990).
7. M. B. Dias and A. Stenz, Opportunistic optimization for market-based multirobot control, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Lausanne, pp. 2714–2720 (2002).
8. J. Ko, B. Stewart, D. Fox, K. Konolige and B. Limketkai, A practical decision-theoretic approach to multi-robot mapping and exploration, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas, NV, vol. 3, pp. 3232–3238 (2003).
9. B. P. Gerkey and M. J. Matarić, Murdoch: publish/subscribe task allocation for heterogeneous agents, in: *Proc. Int. Conf. on Autonomous Agents*, Barcelona, pp. 203–204 (2000).
10. B. B. Werger and M. J. Matarić, Broadcast of local eligibility for multi-target observation, *Distrib. Autonomous Robotic Syst.* **4**, 347–356 (2000).
11. M. B. Dias, R. Zlot, N. Karla and A. Stentz, Market-based multirobot coordination: a survey and analysis, *Proc. IEEE* **94**, 1257–1270 (2006).
12. A. Viguria, I. Maza and A. Ollero, SET: an algorithm for distributed multirobot task allocation with dynamic negotiation based on task subsets, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Rome, pp. 3339–3344 (2007).
13. T. Lemaire, R. Alami and S. Lacroix, A distributed tasks allocation scheme in multi-UAV context, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, vol. 4, pp. 3622–3627 (2004).
14. R. Zlot and A. Stentz, Market-based multirobot coordination for complex tasks, *Int. J. Robotics Res. (Special Issue on the 4th International Conference on Field and Service Robotics)* **25**, 73–101 (2006).
15. V. Remuss, M. Musial and U. W. Brandenburg, BBCS robust communication system for distributed system, presented at: *IEEE Int. Workshop on Safety, Security, and Rescue Robotics*, Bonn (2004).
16. A. Ollero, S. Lacroix, L. Merino, J. Gancet, J. Wiklund, V. Remuss, I. V. Perez, L. G. Gutierrez, D. X. Viegas, M. A. Gonzalez, R. Mallet, A. Alami, R. Chatila, G. Hommel, F. J. Colmenero, B. C. Arrue, J. Ferruz, J. Martinez de Dios and F. Caballero, Multiple eyes in the skies. architecture and perception issues in the COMETS unmanned air vehicles project, *IEEE Robotics Automat. Mag.* **12**, 46–57 (2005).
17. L. Merino, F. Caballero, J. Wiklund, A. Moe, J. R. Martinez-de Dios, P.-E. Forssen, K. Nordberg and A. Ollero, Vision-based multi-UAV position estimation, *IEEE Robotics Automat. Mag.* **13**, 53–62 (2006).
18. G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, *IEEE Trans. Comp.* **C-29**, 1104–1113 (1980).
19. T. Sandholm, An implementation of the contract net protocol based on marginal cost calculations, in: *Proc. Int. Workshop on Distributed Artificial Intelligence*, Hidden Valley, PA, pp. 295–308 (1993).
20. B. C. Arrue, J. R. Martinez-de Dios and A. Ollero, An intelligent system for false alarm reduction in infrared detection of forest fires, *IEEE Intell. Syst.* **15**, 64–73 (2000).

About the Authors

Antidio Viguria received the 'Engineering' degree in Telecommunication from the University of Seville, Spain, in 2004. From 2004 to 2006, he was working as a Researcher at the University of Seville. In 2006, he became a Fulbright Scholar and a Graduate Student at the Georgia Institute of Technology, where he received the MSECE Degree, in 2008. He is now a PhD candidate at the University of Seville and he recently joined the Advanced Center for Aerospace Technologies (CATEC). His main research interests are in multi-robot coordination and cooperation, and robot architectures.

Ivan Maza is currently assistant professor at the University of Seville, Spain. He received his degree in Telecommunication Engineering, and joined the Robotics, Vision and Control Group, in 2000. He has participated in several Spanish and European R&D projects in the field of aerial robotics and multi-robot cooperation, such as the COMETS and AWARE EU Projects. His main research interests are in the field of multi-robot decision architectures, and include distributed task allocation and collision avoidance for aerial robots.

Anibal Ollero is Professor at the University of Seville. He has been Professor at the Universities of Santiago and Málaga, Spain, and Researcher at the Robotics Institute CMU, Pittsburgh, PA, USA, and LAAS-CNRS, Toulouse, France. He is author or co-author of more than 350 publications including four books, and led or participated in more than 90 projects, including 14 projects funded by the European Commission. He is the recipient of eight national and international awards. He is currently the Coordinator of the European project AWARE on the integration of unmanned aerial vehicles with wireless sensor and actuator networks, and the coordinator of several Spanish projects on robotics, unmanned aerial vehicles and networked systems. He is Vice-Chair of the Technical Board of the International Federation of Automatic Control and Scientific Manager of the Advanced Center for Aerospace Technologies (CATEC).