RESCUE
Cooperative Navigation for Rescue Robots
Final Technical Report

A. Bernardino, L. Custódio, J. Frazão,
P. Lima, F. Melo, M. I. Ribeiro,
J. Santos-Victor, A. Vale

November 4, 2004

Contents

1 Introduction

The main goal of the "RESCUE – Cooperative Navigation for Rescue Robots" project was, as stated in its proposal, to provide integrated solutions for the design of teams of cooperative robots operating in outdoors environments, with special focus in the short and mid-terms on perception and representation issues, as well as cooperative navigation, and, in the mid to long-terms, on task modeling, planning and coordination.

One potential scenario for outdoors robot operation is the search and rescue for victims after a catastrophic event. This may be due to causes that range from natural occurrences, such as earthquakes or floods, to urban riots or terrorist attacks. Many regions of Europe, such as the Balkans, Italy, the South of Spain and Portugal are regions of high or moderate seismic activity.

Earthquakes are specially daunting phenomena. Even though their occurrence is fortunately sparse, the consequences are often tragic, leading to thousands of deaths and many more injured people, besides mass building destruction. Reports from the infamous 1995 Kobe earthquake in Japan show that many unexpected events hit the communications and civil protection infrastructure, disabling the execution of most of the available plans for disaster situations [21]. Especially the serious damage to the telecommunications network and the fact that the buildings of disaster mitigation organizations were hit caused serious delays in the arrival of local and external assistance to the victims. Structures in risk of collapsing are often inaccessible to humans because they are too dangerous. Survivors may be unintentionally injured during removal of debris due to the unawareness of their location or presence by the rescue teams.

Teams of heterogeneous robots can represent an invaluable help for future search and rescue operations. Robots can crawl over the collapsed structures, depositing small-sized robots, and feeding air, water, food and medication to trapped individuals through tubes snaked into the collapsed structure. Small-sized robots can sneak inside very confined spaces, taking with them tiny cameras and other sensors to detect survivors and map survivor locations. Aerial robots can provide a broad view of the search and rescue scenario and map high-destruction locations. Cooperation among human-operated stations, a distributed network of sensors located around the disaster area and teams of tele-operated and/or autonomous robots can increase the amount of available information to the rescue teams. Whenever communications with the networked sensors or the sensors themselves fail due to the earthquake impact, robots can be dispatched to cover the areas most unaccessible to humans and help finding victims at those locations. These mobile robot networks can also provide a dynamic view of the scene at relevant locations.

The major drives of the project were:

- To foster research advances in its enabling disciplines (Computer Vision, Robot Navigation, Control and Artificial Intelligence), extendable to other outdoors applications, such as environmental monitoring and surveillance, agriculture or planetary exploration.

- To develop an application of strong social impact.

During the project lifetime, other initiatives helped promoting these goals:

- The creation, at the Associate Laboratory Institute for Systems and Robotics, at IST (ISR/IST), of a research line on "Robot Monitoring and Surveillance" that, among others, is concerned with the development of outdoors cooperative rescue robots.

- The approval of the RAPOSA project proposal by the *Agência de Inovação*, where a tele-operated rescue robot has been built and will soon be start its tests at Lisbon Fire Department. The project promoted the collaboration between ISR/IST and the Lisbon Fire Department on this type of activities.

1.1 Reference Scenario

In the first year of the RESCUE project, thorough scientific discussions took place among the involved groups. The objective of these discussions was three-folded:

- to help the different involved groups sharing a common language;

- to provide an integrated view of the project scientific goals, instead of a simple concatenation of each group contributions;

- to specify a reference scenario that would illustrate the research goals, including the required hardware and software.

After these discussions, a scenario was approved as the *project reference scenario*. Its main goals are to provide a realistic step toward the long term goal of developing practical rescue robots and to encompass the scientific interests of the project members. The reference scenario, is based on the ATRV-Jr land and the MiniZepp blimp aerial robots, described in the next subsection, which perform cooperative navigation tasks, some of them requiring formation control.

The reference scenario consists of two robots: one aerial blimp/zeppelin robot; one land outdoors robot. The aerial robot will perform several tasks, such as making a visual topological map of the destroyed site. The map will include information on the relevance of each of the mapped locations concerning degree of destruction, presence of victims, etc, as well as on the difficulty of traversing regions between them, due to the presence of debris or obstructed paths. The map will be stored as a graph and will be used to choose the best path for the land robot to reach a goal location (e.g., one with a larger number of victims). It can also be used to help the aerial robot navigating. The land robot will use several sensors (GPS, inertial, vision, sonars) to navigate toward the goal, handling the details associated to the path (e.g., debris, trees, people on the way, etc). While the land robot moves toward the goal location, the aerial robot should follow it using a formation control algorithm, so as to keep a reliable communications link and to serve as a relay for informations that the land robot may need to send to distant stations.

An animation illustrating the scenario above was developed to better explain the reference scenario to project members and non-members and is included in the companion CD as a .mpeg file.

1.2 Robots

In the initial phases of the project, and following the reference scenario discussion, the specification, purchase and integration of the hardware required to accomplish the goals of the reference scenario were carried out, resulting in the purchase of an iRobot ATRV-Jr land mobile robot and a BlimpGuys commercial blimp, to be used as the aerial robot. The latter was purchased using ISR funding, not provided by this project.

The ATRV-Jr is a 4-wheel differential drive robot, with 2 high torque 24V DC servo motors, capable of achieving linear speeds up to 1 m/s and turning speeds up to 120 deg/s. With a 25 Kg payload, it weights 50 Kg and is capable of overcoming 45 deg slopes. Its 2 lead acid, 672 Watt/hr batteries provide an autonomy of 3 to 5 hours, depending on terrain. The onboard computer is a Pentium III running at 800 MHz. Communications with other robots and external computers are ensured by either a 100/10 Mbps Ethernet board or IEEE 802.11b Wireless radio Ethernet. The robot includes an odometry system and Linux-based management software and libraries.

The ATRV-Jr was endowed with several navigation sensors:

- one computerized navigation compass,
- one 12-channel GPS receiver,
- one Inertial Navigation System (INS), including rate-gyros, and
- seventeen sonars (5 forward facing, 10 side facing and 2 rear facing).

The land robot was initially tested on several runs at IST gardens, with simple navigation algorithms based on odometry and sonars. The GPS, compass and INS were also tested and their performance assessed under several different conditions (e.g., number of available GPS satellites, robot inclination). A report on this work, performed by the company IdMind, can be found in the companion CD, detailing the robot hardware and the results of experiments made with it.

The blimp is 4 m long, with a 2 m diameter and 5.5 m^3 volume polyurethane bag. Two of its speed-controlled servo-motors can control altitude, as well as forward and reverse speed, under less than 5 Km/h winds. The remaining servo-motor is attached to the bottom tail and helps controlling the blimp orientation. The payload is 2 Kg. The onboard sensor is a pan and tilt color CCD camera with HF transmitter. Originally, it had a seven channel 41Mhz FM radio transmitter from Graupner (mc-10), now modified to act as a 5-channel PC-Radio Controller transmitter interface using RS-232 communication. The 5 channels are used to control, from a ground computer, the speed, altitude and rotation servos, as well as the pan and tilt system. The ground computer receives the video signal sent through the HF video channel, processes the images and closes the navigation loop by sending commands to the servos and the pan and tilt systems. The modified hardware includes an emergency switch for manual control. The robot is powered by 2 NiMh 7.2v 1900mAh batteries for motors and control, and 8 NiMh 1.2v 600mAh batteries for camera and transmitter power, providing an autonomy of 30–120 minutes.

A report on the blimp, performed by the company IdMind, is also included in the companion CD, detailing the aerial robot hardware, the modifications made and the results of experiments made with it.

1.3 Organization

This final project report summarizes the main scientific and technological achievements of the RESCUE project and is organized as follows. Section 2 focus on the topological navigation methods developed for the land robot. Section 3 covers the different work done on vision-based indoors blimp navigation. In Section 4, some theoretical work on the navigation controllability of heterogeneous robot teams, based on Discrete Event Systems theory, is presented. Some methods for Formation Control, and the corresponding simulations, where the aerial blimp and up to 6 land robots follow paths while keeping a geometrical formation, are presented in Section 5. Section 6 introduces work on distributed planning for a multi-robot rescue team, based on distributed Artificial Intelligence and Multi-Agent Systems techniques, and tested on a simulator which emulates scenarios that could well be the subject of future applications using real robots, as an extension of the current project. Section 7 summarizes the concepts concerning the software and functional architectures that will support the project development, and its implementation. Finally, major conclusions of the work carried out within the RESCUE project are drawn in Section 8, where plans for future research are also briefly listed.

2 Topological Navigation of Land Robot

2.1 Introduction

In this section it is proposed a topological based methodology to solve the problem of mobile robot navigation in unstructured outdoor scenarios. The methodology was tested in realistic indoors and outdoors scenarios, supported on a real mobile land robot. Its main contributions include a new topological environment representation built from a set of nodes defined by sum of Gaussian pdfs and connected by orientation, a dynamic version of expectation and maximization algorithm to build the world representation, a probabilistic approach for localization and navigation using an optimized version of forward-backward algorithm and a procedure for feature extraction and selection.

The operation of mobile robots in outdoors environments has become widely used in the past years raising a set of challenging scientific problems that include environment representation, map building and localization.

The most common approaches for environment representation are divided in three groups: geometric, topological and hybrid. The choice of the type of world representation constraints the approaches taken for localization and navigation.

Most of the topological research to recognize places and to record them as references, is bundled on information retrieved by vision sensors, in particular with edges, [18], the main components of the image [28] and by colors [16]. There are several approaches that use vision and motion commands as a qualitative way [15] for mapping and navigation, based on bio-inspired techniques. Given the uncertainty associated to the environments, as justified by [27], it is important that topological maps also include a probabilistic approach. Geometric maps and topological maps can be combined as hybrid maps, described in [2], where most often the topological representation arises from the metric information [11].

In spite of individual problems, the localization and map building must be handled as a single issue. One promising research area lays on Simultaneous Localization and Mapping - SLAM, [23], [14], [3], also known as Concurrent Map Localization - CML, [1] and [12]. Most of the SLAM approaches are oriented to indoors, well structured and static environments (like domestic ones [32]) and give only metric information of the position of the mobile robot and the landmarks.

Rather than using a metric approach, this methodology solves the SLAM underlined problem, including localization, navigation and mapping, relying on a topological approach. A topological map is incrementally built along the robot operation. This map supports methodologies for topological localization and navigation.

2.2 Mobile Robot Navigation

The mobile robot navigation has to deal with uncertainty included in the world, the perception of the world (observations and/or consequently on features) and also on the motion.

The localization problem is associated to the world representation, not limited to a simple referential, where the metric information is related. Using the perception given by the sensors according to the selected environment representation, the localization has to provide the information to bound the uncertainty increasing.

Once an appropriate world representation is selected, no less important is the problem of building the map that endows the environment representation. It is necessary to understand how to start the localization on the initial map and to progressively update that map. The updating procedure, or mapping algorithm must be running every time that new observations are acquired, which can be retrieved in places not visited yet.

Features discriminate the different types of information, which result from processing the acquired observations. The map can be used as a referential again but, at this time, with more dimensions, where each dimension corresponds to a different feature. This map provides a high level of abstraction of the environment, a *topological* representation. The precision is accomplished by the type and the amount of features extracted.

2.2.1 The three main problems: Localization, Navigation and Mapping

The mobile robot navigation has three main problems: Localization, Navigation and Mapping. When the robot is moving, it is necessary to estimate the position (the localization problem) and compute a new path (the navigation problem). Moreover if the robot moves, it yields to a new position, where the new acquired observations could improve the current map, the mapping problem. From these three problems emerges the loop in Figure 1, that is executed while the robot performs its mission.

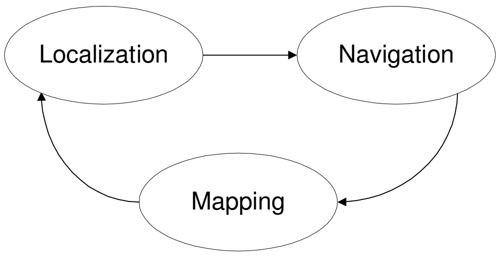


Figure 1: The main loop of mobile robot navigation

The loop does not require a specific sequence. Moreover, each problem can be accomplished at different frequencies, which could change dynamically, this meaning that the three issues have different priorities. Given that the robot is moving to reach a specific target or goal, it is important to recognizing its position at each time instant and to update the planned path if necessary. The localization and navigation are based on a map, which is build by the mapping algorithm. Since it is assumed that the robot does not travel long distance during short periods of time, it is expected that the scenario is still covered by the current map. Therefore, the map should be updated when the localization ambiguities occur and/or added/removed new/old type of features. Consequently, it is not necessary to run the mapping permanently.

The navigation is the next priority, to evaluate the best way to reach the main goal of the mission. Since all the procedures are based on the environment representation, it is also necessary to run the mapping procedure, even though it is not necessary to run it every time the robot acquires a new observation. Based on this concept, the procedure implemented with the highest rate is the localization, followed by the navigation and finally the mapping, as illustrated in Figure 2. If there is no map available at the beginning, the loop must start from the mapping procedure to provide a first map, or loading a previous estimation of the scenario.

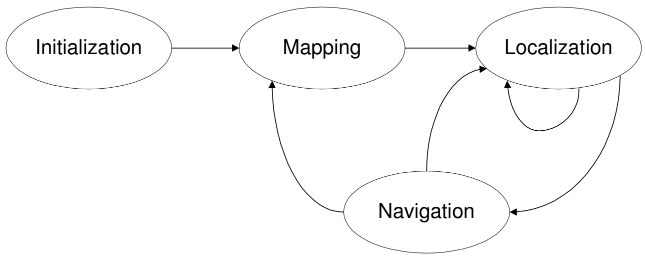


Figure 2: The 3 steps of mobile robot navigation at different rates

The localization starts with the initial robot position in the map. If not known, the initial position is assumed as equiprobable in all map. In the navigation, it is initialized or, imposed by the mission, a target goal. If there is a map available at the beginning, the target goal can be pointed in the map. After the initialization, the loop starts, following the priorities illustrated in Figure 2.

2.2.2 Environment Representation

The navigation is accomplished based on the available map and there are different types of maps, as illustrated in Figure 3.

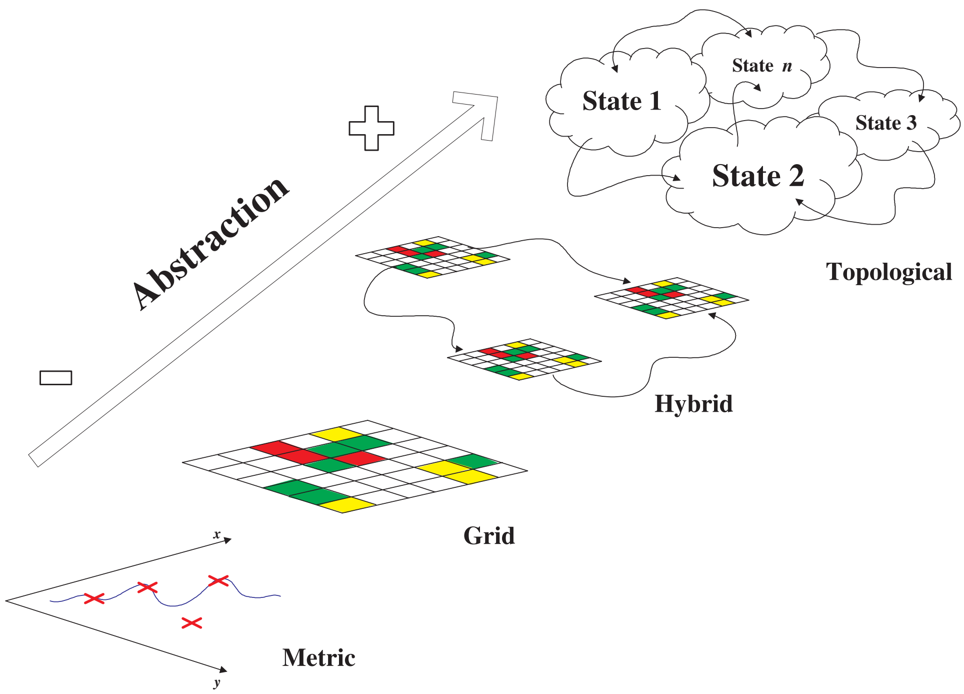


Figure 3: Environment representations at different levels of abstraction

Current global methods can be classified as topological (adjacency-graph based representation of the environment composed by nodes or states and links), geometric (metric representation of the environment landmarks position with respect to a referential; the metric representation also includes the common grid maps) or hybrid (topological maps containing sub-topological and metric maps in each state).

A topological map is a representation of an environment with no metric information available, showing physical (natural or artificial) features that characterize particular locations or places. The map expresses a functional relationship among relevant features with a resolution that is proportional to the complexity of the environment's representation.

The topological maps, are complex enough to travel long distances according to the appropriate way and simple to avoid the incumbency of recording ever information over the physical location covered by the map. This type of representation, topological maps, is the only one prepared for the diversity of scenarios, the large spectrum of information acquired by the available sensors, the unexpected events that occur during the operation and also robust to the scenario changes. The notation used to define a topological map is the following: s_i is the state $i, i = 1 \ldots, N$ of the map defined by a sum of Gaussians modeling the features that characterize each state, a_{ij} is the transition probability between state i and state j and θ_{ij} is the direction between these states.

2.2.3 Localization

The localization procedure proposed in this work states that, at each time instant, the robot location, q_t, is equal to the map's state in its closest vicinity using a probabilistic approach to decide on this proximity function. The robot estimated location is the map's state that is most likely to have produced the observations acquired by the robot sensors during a given time interval. When the proposed localization procedure yields a robot estimated location $\hat{q}_t = s_i$ this does not mean that the robot physical location (pose) coincides with that of the environment place that lead to the map state s_i.

As a result of the measurements uncertainty, the current robot state estimation can not be performed using a deterministic criteria. Consequently, the main issue of the localization problem is to find the state that minimizes the uncertainty, given the observations. The state estimation at each time instant t is evaluated using all the available observations during the interval T. According to a probabilistic approach, the current state estimation, $\hat{q}_t$, is given by

$$\hat{q}_t = \arg\max_{q_t} P(q_t = s_i \mid o_1, \ldots, o_T), \tag{1}$$

where $O_T = \{o_1, o_2, \ldots, o_t, \ldots, o_T\}$ is a sequence of observations. This problem is addressed in particularly in [29], by a changed version of Forward-Backward Algorithm.

2.2.4 Navigation

The navigation procedure proposed, is also developed using a topological approach, based on the robot location at each time instant and the topological map. The navigation procedure consists on finding the best way to reach a goal, a state in the topological map, given the current robot's state.

To reach the target state, the robot moves through other states endowed by uncertainty. Given the localization result at each time instant, the navigation algorithm provides the best sequence of states from the current state to the goal. However, if the robot fails the sequence, it means, the robot reaches a state not included in the sequence, the topological navigation has to compute a new sequence, starting at the new current robot's state.

Given the robot's state, $q_t = s_i$, retrieved by the localization algorithm, the navigation determines the sequence of states from the q_t to the goal state, s_j. The main goal consists on, after a period of time, $t + \Delta$, the robot to be placed in a given state, or, equivalently, $q_{t+\Delta} = s_j$. The key-question is what the robot should observe to reach the main state, or equivalently,

$$\max_{o_{t+1},\ldots,o_{t+\Delta}} P(q_{t+\Delta} = s_j \mid o_{t+1}, \ldots, o_{t+\Delta}, q_t = s_i) \tag{2}$$

Since the robot is a mobile vehicle, it is necessary to convert the sequence of states into motion commands, which is equivalent to change from topological navigation (high level of abstraction) to a metric navigation (low level of abstraction). To accomplish this step it is necessary to know the states which composed the map, the angles between them and the robot's orientation.

The direction between states could be retrieved as a particular feature of the states - orientation, if available, or by saving the orientation assumed by the robot between the transition of two consecutive states. For instance, the direction between s_i and s_j, θ_{ij} is NW. It could be repeated hundred times and the direction between states is refined. Even though the robot is trying to follow that direction, it may reach another state, not s_j, over few runs. If the reached state is not complemented by the sequence of states returned by the navigation algorithm, it will compute a new sequence given the new current state.

The next issue consists on finding a way to target the robot to the desired direction, which corresponds to the direction between the current state s_i (resulted by the localization estimation) and the next goal state s_j. The unexpected obstacles lead to a necessity of developing a process of taking control away from the undesirable situations. The selected approach is more oriented to attractive and repulsive behaviors [10], where the action leads the robot to the target, avoiding obstacles.

2.2.5 Mapping

The robot perception is condensed in observations, o_t, that represent the information obtained from the processing of the raw data acquired at each time instant t. With the map characterization, a set of nodes, defined by Gaussians, the mapping procedure estimates the mean vectors and the covariance matrices that maximize the probability of all observations given the environment model, i.e., that maximize the likelihood function,

$$L(S) \quad = \quad \sum_{i=1}^{t} \log \left(\sum_{k=1}^{N} c_k \cdot p(o_i \mid s_k) \right). \tag{3}$$

This problem is addressed in particularly in [30], by a Dynamic Expectation and Maximization algorithm.

Features have to support different scenarios but not every type of feature is essential to a particular scenario, this requiring a feature selection criteria. In this work it is also addressed the problem of feature extraction and proposes a feature selection criteria, which is described with more detail in [31].

2.3 Experimental Results

The experimental results were accomplished in real environments, using the mobile robot ATRV-Jr. The map building algorithm, given the high level of abstraction as the topological approach, is strictly dependent of the type of features. Concerning this dependence, the experimental results presented in this section also enable understanding how the features selected improve the topological approach.

2.3.1 Indoor Results

In this first experience, it was only used range sensors: the laser range scanner and the ring of ultra-sonic sensors installed at ATRV-Jr. The type of features were based on free-area measured by the laser and the sonars. The free-area could be represented by the mean, variance or other combinations of the free-area along different directions measured by the sensors.

The selected indoor scenario to test the mapping algorithm based on the type of features described above, is a laboratory environment, where the robot moved from a room to a corridor. The room is a common place containing chairs, tables and people walking, usually larger than a corridor.

The mapping algorithm was tested using the observed features (related with free-area) acquired during the travel from the middle of the room to the corridor. Since the selected features do not contain metric information of the robot's position, the raw-data recorded during the trajectory is displayed in Figure 4-a) using the odometry (sensor fusion), only for illustration and to simplify the data visualization.

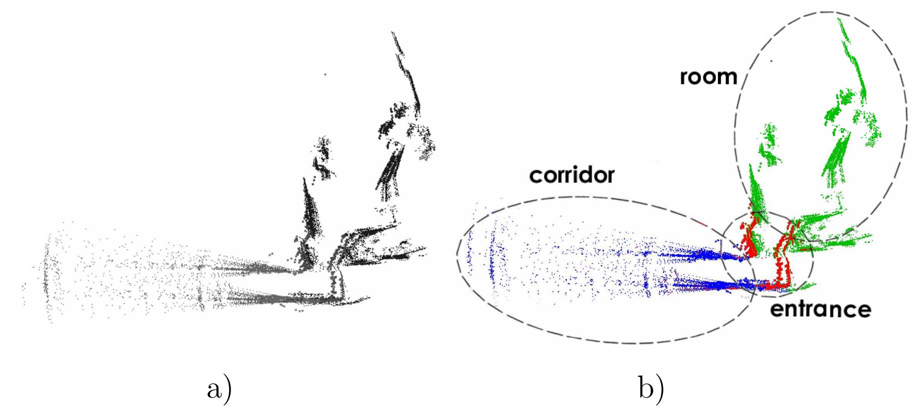
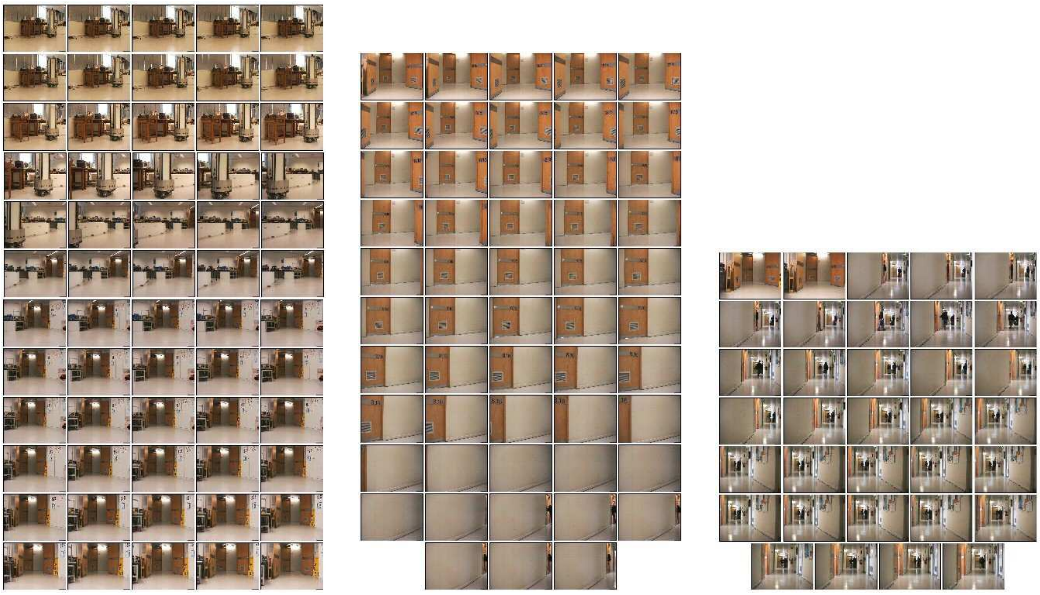


Figure 4: The resulted topological map: a) the laser and sonar measurements and b) the states that compose the map

Setting low accuracy to the mapping algorithm, the result is a topological map with three states, corresponding to the room and the corridor as expected and to another state, which corresponds to the transition between them, defined as an entrance. The Figure 4-b) represents the measurements of the laser and sonar with three different colors, corresponding to each state.

The resulted topological map, presented in Figure 4, does not contain neither metric information nor features extracted from a vision camera. However, during the experience, the images were grabbed at same time as the laser and sonar data. The camera is mounted on the top of the robot pointing in the front direction. After knowing the topological map, the images were divided according to each state, as depicted in Figure 5.

The images were clustered according to each state, based only on range sensor data and no vision information. Just by a vision inspection, the first state may be also decomposed



a) b) c)

Figure 5: The images associated to the a) state 1 (equivalent to the lab), b) state 2 (equivalent to the entrance) and c) state 3 (equivalent to the corridor)

in two possible new states. Therefore, image sensor retains more information than range finders.

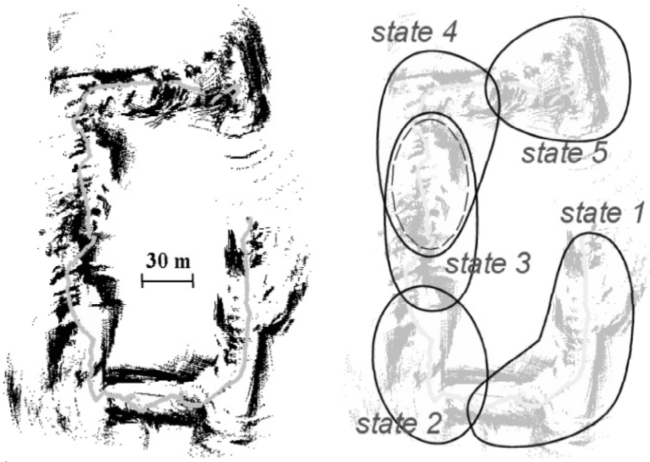
2.3.2 Outdoor Results

The next experience was targeted to test the topological representation using features extracted from the images. The proposed scenario is the Campus of IST (Instituto Superior Técnico) at Lisbon - Alameda. The robot acquired the raw-data, while it was navigating around the central building and computing the topological map. Since no map was available at the beginning, the navigation algorithm was proposed to guide the robot to follow some via-points defined by latitude and longitude coordinates, using the behavior approach. The robot moved along a distance of approximately 400 meters and acquired raw-data during 1586 iterations, later converted into features.

At this point, it is necessary to evaluate the resulted topological map and the quality of the information retained by the features extraction procedure. The central building of IST is located in the center of image in Figure 6-a), surrounded by trees, cars, people walking and other buildings. Only features extracted from the vision camera were used, given the range sensors' limitations. The resulted topological map using histograms (of colors) and PCA is presented in Figure 6-b).

The quality of information included in the features is not enough to create a middle state between state 3 and 4, leading to an ambiguity between these two states. Increasing the number of features does not imply an information improvement to represent what is observed, which requires to analyze the features correlation and choose the low correlated.

As expected, the correlation differs between experiences and features, as shown in [31], where vertical edges, Hue/Saturation-colors histograms parameterization, 2D histograms

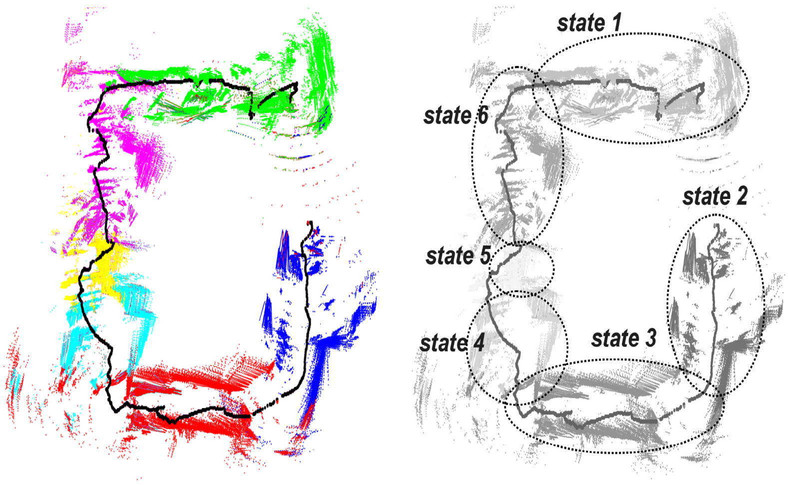


a) b)

Figure 6: The resulted topological map using color histograms and PCA: a) the laser measurements and b) the 5 states

(boundary-boxes), PCA and ICA were tested in different scenarios. ¿From the results presented in the previous paper, the edges and ICA or edges and histograms (or even the three type of features simultaneously) are the best features to build a map in current scenario. Based on the results obtained from the features selection, it was adopted the histograms and edges as one of the best features combinations. Moreover, the orientation was also included on the features: histograms, edges, and orientation.

The mapping algorithm computed a new topological map, as illustrated in Figure 7, containing six states. The ambiguity of the previous experiment was removed, since the topological map identified a new state (state 5) according to the selected features.



a) b)

Figure 7: The topological map using color histograms and edges a) the laser measurements and b) the states identified by circles

Using the topological map resulted from the last experience, in Figure 7, the localization algorithm was tested. To localize the robot in the topological map it is also necessary the same type of features used in the mapping algorithm.

The experimental results of localization are presented in Figure 8, with the probabilities of the robot to be localized in each state, given the observed features along time. The robot started with some doubt between state 2 and 3, quickly solved after few iterations, remaining in state 2 during more than 200 iterations. The following state is 3 and there is a period of uncertainty between state 3 and 4, lately solved to the last one. It is also verified in Figure 7, when the mapping algorithm shows a small noise between state 3 and 4, visible by the blue and red colors. This subject stresses not only the features selection issue, but also the importance of the topological map resolution.

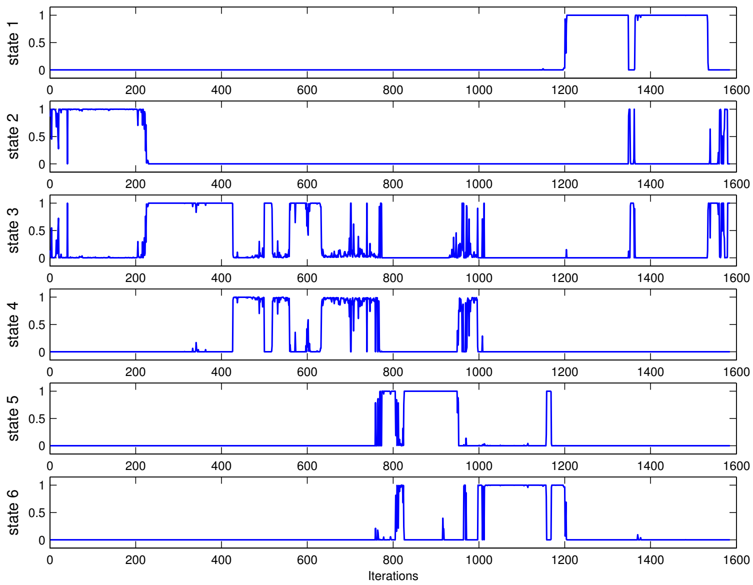


Figure 8: The localization evolution on the resulted topological map

Along the path and based on the localization results on the topological map, it is also possible to estimate the transition probability between states. The robot has a high probability to remain in the current state, for every state of the topological map. If the robot transits to another state, it is more probable to jump to the next or to the previous state. It is also possible to estimate the orientation between states.

3 Vision-Based Control of the Aerial Robot

The aerial blimp robot was first flown autonomously during the project second year. The first step consisted of developing a driver to provide access to the radio controller, whose hardware had been modified in the first year to allow remote control by a ground computer. Then, several measurements were made to obtain the characteristic functions that relate the voltage sent to the motors with the thrust generated by the propellers. Using that, a control library for position and velocity control was developed. The library is modular and consists of three control levels (see Figure 9), from position control in world coordinates down to velocity control of forward/backward, upward/downward and rotation around the vertical axis movements, in vehicle coordinates. This control library can be used like a black box. The user can control separately the local velocities over the global direction control up to the global position control.

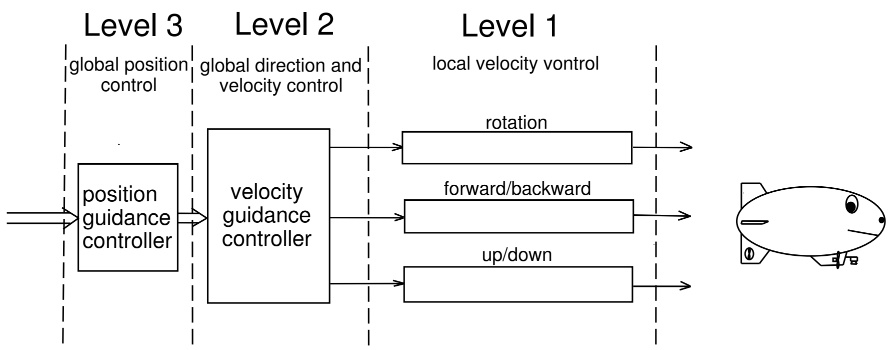


Figure 9: Aerial blimp control architecture.

During the RESCUE project, the only sensor used in the aerial robot was a vision system consisting of a micro-camera placed onboard the blimp, whose images are subject to real-time processing. From the homographies between consecutive images and assuming some prior information regarding the surrounding environment, it is possible to estimate velocities and displacements of the robot in 3D space.

Several indoors autonomous navigation tests were performed and their results are available in a technical report TKrause-TechRep.pdf included in the companion CD. Generally, the results were positive, and it is possible to control the robot velocity in 3D space, including the compensation of dynamical effects like simulated weak wind. Nevertheless, optical flow has some drawbacks, especially notable after integration, resulting in an often poor odometric estimate of the blimp position in world coordinates.

To perform positioning or trajectory following tasks with an aerial robot, the development of control algorithms that overcome the underlying limitations of the system dynamics and kinematics, as well as the external disturbances, is required. Control methodologies were developed for the aerial robot that enable the system to accomplish positioning or trajectory following tasks, surpassing some limitations imposed by the physical system and the sensor. Image processing algorithms that enable obtaining the vehicle pose (position + orientation) and velocity were studied. Several types of linear and non-linear controllers were used to control the vehicle velocity, as well as its heading in 3D space. Two strategies for the reference definition were proposed, one based in position and coordinates in Cartesian space and the other based in image measurements, thus avoiding the need for high precision camera calibration.

The work developed consisted of system modeling and parameter identification, as well as control and image processing tests in a special-purpose simulator developed within the project. Furthermore, experiments were made with the real setup in which the algorithms were implemented, running in real-time.

More detailed information on this work can be found in the FilipeLuis-TechRep.pdf file included in the companion CD.

Additional work has been done on localization and landmark selection for navigation on aerial images. It is assumed that the aerial robot has previously extracted an extended visual map of the working area. Once this is done, alternative techniques are compared to select salient regions in the scene that can be used as visual landmarks for navigation. The work is described in [19]. Salient regions are defined as regions very distinct from their neighborhood and this can also be useful in the detection of isolated destruction loci (fires, wrecks), though this was not fully exploited in this project.

More detailed information on this work can be found in the Lorenz-TechRep.pdf file included in the companion CD.

4 Navigation Controllability for Heterogeneous Robot Teams

In order to achieve coordination/cooperation between a set of robots, it is of great importance to understand how the ability of each individual robot to achieve some goal affects the team's overall ability to succeed in the mission.

In order to have a deeper insight regarding this problem, in this section an analysis tool is presented, which will allow better understanding of how the navigation of each robot in a robot population may or may not prevent the final objective from being attained.

4.1 The Environment Representation

Consider, then, a set of N mobile robots. The robots are moving in a discrete environment, i.e., an environment which can be divided in a discrete set of sites, $\mathcal{S} = \{1, ..., M\}$. A discrete environment can properly be described using a topological map $\mathcal{T} = (\mathcal{S}, \mathcal{E})$, where, as stated, $\mathcal{S}$ is a finite set of sites and $\mathcal{E} = \mathcal{S} \times \mathcal{S}$ is a set of edges, such that, given $s_1, s_2 \in \mathcal{S}$, $(s_1, s_2) \in \mathcal{E}$ if it is possible to move from site s_1 to site s_2. In figure 10 an example of an indoors environment and its topological map are depicted.

Although this kind of representation is particularly suited for describing indoors environments, since these environments are naturally divided in sites (rooms, hallways, etc), recent work has been developed regarding mapping of unstructured environments as topological maps (e.g. outdoors environments [30]).

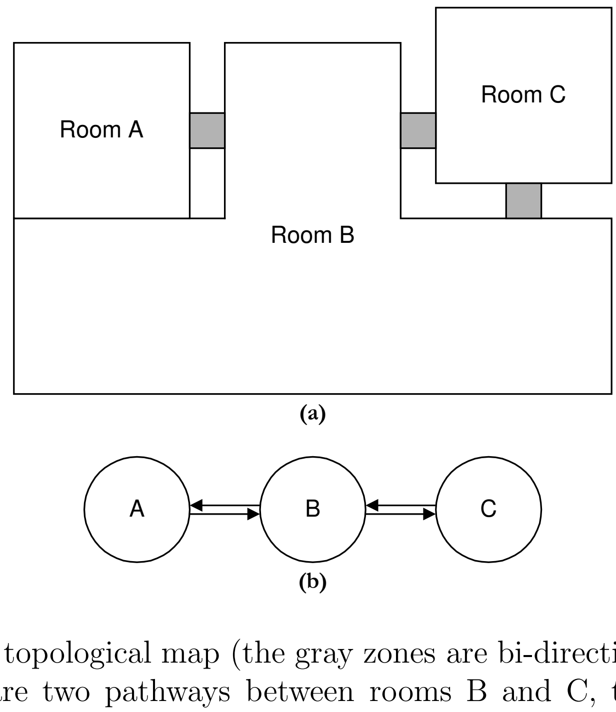


Figure 10: Example of topological map (the gray zones are bi-directional pathways). Notice that, although there are two pathways between rooms B and C, they aren't both represented in the topological map. (a) Example of an indoors environment. (b) Corresponding topological map.

However, the topology of the environment is not enough to model the movement of a mobile robot in that environment. For example, in Figure 10, suppose that the pathway between rooms A and B is a staircase. If the robot is not properly equipped to cope with such a terrain, it may be impossible for it to travel between rooms A and B, although there is a physical pathway between the two rooms.

This leads to an immediate conclusion: *the information in the topological map of the environment may not be enough to represent the movement of the robot in the environment.*

In the next sub-section, a model of the robot will then be introduced.

4.2 The Navigation Automata

¿From what was stated in the previous subsection, a robot moving in an environment described by a topological map $\mathcal{T} = (\mathcal{S}, \mathcal{E})$ may not be able to move across all edges of the topological map.

Thus, it is necessary to have a model for the mobile robot which properly describes its movement on the environment. A first approach would be to describe the workspace of the robot as a topological map $\mathcal{T}' = (\mathcal{S}, \mathcal{E}')$, where $\mathcal{E}' \subset \mathcal{E}$ is the subset of edges in $\mathcal{E}$ which the robot can traverse. However, in order to model the control actions and the state of the robot, a more complete representation of the robot is necessary.

Since the robot is moving in a discrete environment, it is possible to model it as a discrete-event system. Automata theory provides a sound framework when working with discrete-event systems (see [5] for details on automata theory). If a robot's workspace is described by a topological map $\mathcal{T}' = (\mathcal{S}, \mathcal{E}')$, it is now possible to model the mobile robot as an unmarked automaton $G = (X, E, f, \Gamma, x_0)$ [24], where

- X is the state-space and $X = \mathcal{S}$;
- E is the event set, a set of unique labels for the edges in $\mathcal{E}'$;
- $f : X \times E \longrightarrow X$ is the transition function; for $x \in X, e \in E, f(x, e) = y$, with $y \in X$, if and only if $(x, y) \in \mathcal{E}$ and the label of $(x, y) \in \mathcal{E}'$ is $e \in E$;
- $\Gamma : X \longrightarrow 2^E$ is the active event function, such that, for $x, y \in X$, $y \in \Gamma(x)$ if and only if $(x, y) \in \mathcal{E}'$;
- x_0 is the initial state.

Since each robot is meant to reach one of a set of target sites $\mathcal{S}_T$, define *Navigation Automata* $G(Y_k)$ as the marked automata $G(Y_k) = (X, E, f, \Gamma, x_0, Y_k)$, where X, E, f, Γ, x_0 are as defined previously and Y_k is the set of marked states $Y_k \subset \mathcal{S}_T$.

One may think of a navigation automaton $G(Y_k)$ as being obtained from the topological map $\mathcal{T}'$ by labeling the edges $\mathcal{E}'$ in $\mathcal{T}'$, setting an initial state x_0 and defining as marked states the set Y_k. For example, a navigation automaton for a robot moving in the environment of Figure 10 may be seen in Figure 11.

4.3 The Blocking Information Matrix

The Navigation Automata described in the previous subsection, although modeling the movement of a single robot, can be used to determine the properties of the system of N robots.

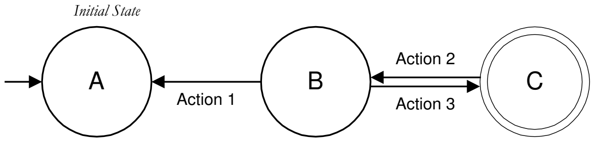


Figure 11: Example of a Navigation Automaton in environment of figure 10.

In fact, it is possible to relate the blocking and controllability properties of these small-dimension automata with the same properties of the large-dimension automaton describing the complete system (for a definition of automaton controllability and blocking, see [5]).

Consider then a system of N mobile robots, which, generally, are considered different. Let G be the automaton describing this system.

The Blocking Information Matrix (BIM) $\mathbf{B}_N$ is a $N \times N$ matrix such that

$$[\mathbf{B}_N]_{k,m} = \begin{cases} 0, & \text{if } G_k(y_m) \text{ is blocking and uncontrollable w.r.t. } \mathcal{K}; \\ -1, & \text{if } G_k(y_m) \text{ is blocking but controllable w.r.t. } \mathcal{K}; \\ 1, & \text{if } G_k(y_m) \text{ is non-blocking} \end{cases},$$

where $G_k(y_m)$ is a navigation automaton of robot k and $\mathcal{K} = \mathcal{L}_m(G_k(y_m))$(see [24, 25] for further details).

It is possible to infer from the BIM about the blocking and controllability properties of G (the automaton describing the system with N robots), from the properties of the navigation automata $G_k(y_m)$, as stated in Theorem 3 of [25].

Basically, if it is possible to permute the lines of $\mathbf{B}_N$ such that there are only non-zero elements in the main diagonal, the system is controllable. If it is possible to permute the lines of $\mathbf{B}_N$ such that there are only ones in the main diagonal, the system is non-blocking.

4.3.1 Example

After what was exposed, an example of the application of the BIM in a simple indoor rescue situation is presented. Consider a set of three robots in which:

The Crawler (Cr) has tracker wheels and is capable of climbing and descending stairs. It is able to open doors only by pushing;

The Puller (Pl) is a wheeled mobile manipulator, able to open doors either by pushing or pulling. However, it is not able to climb stairs;

The Pusher (Ps) is a wheeled robot, able to open doors only by pushing. It cannot climb stairs.

The rescue operation takes place in the indoor environment depicted in Figure 12 (e.g., a fire scenario). On the left is the physical map of the place, and on the right is the corresponding topological map.

Each of the robots is described by a different automaton, as represented in Figure 13.

The robots will leave Room 1 to assist three different victims, somewhere in the building. The doors open as shown in Figure 12 which limits the robots access to the different rooms.

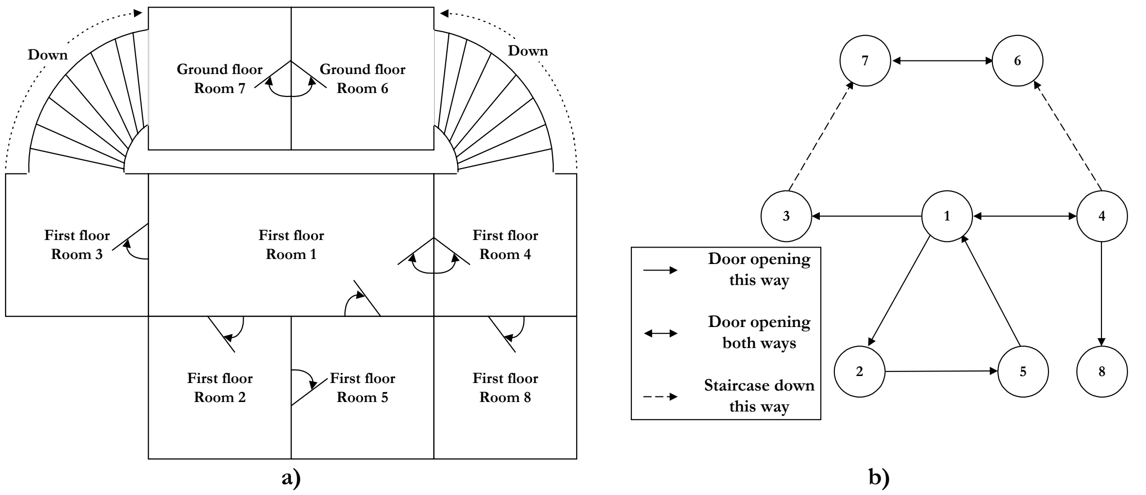


Figure 12: Map of the Environment.

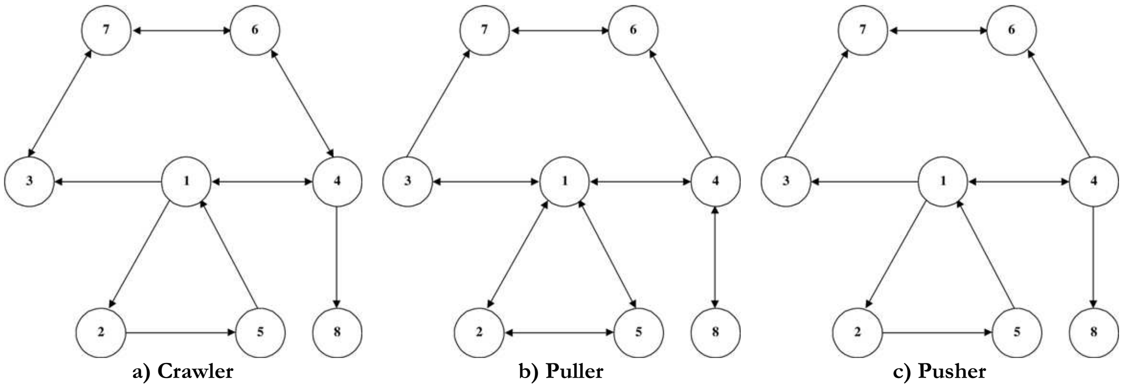


Figure 13: Automata for the robots.

Moreover, when in Rooms 6 or 7, only the Crawler can go upstairs. Finally, when in Rooms 3 and 4, all the robots may fall downstairs. The automata for the robots are in Figure 13.

The following example illustrates the practical use of the BIM in determining configurations that prevent the success of a given rescue operation. The victims will be in rooms 6, 7 and 8.

In this situation, the BIM for the system is:

$$\mathbf{B}_3 = \begin{bmatrix} -1 & -1 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}, \tag{4}$$

where the lines correspond to Pusher, Puller and Crawler and the columns correspond to target sites 6, 7 and 8, respectively.

Note that both Pusher and Crawler, once inside Room 8, are not able to leave. However, by preventing Pusher from going to room 8, this can be prevented. On the other hand, if Pusher or Puller get downstairs, they cannot go back upstairs. However, they cannot get to Room 8 without going through Room 4 and eventually falling to Room 6. Finally, there is no room from which Puller cannot reach Rooms 6 and 7, and from which Crawler cannot reach Room 8.

Looking at matrix $\mathbf{B}_3$ the system is blocking but controllable. In fact, for example in the configuration where Crawler and Pusher are in room 8, it is impossible to reach the target configuration. However, this can be avoided, by preventing, for example, Pusher from going to room 8.

5 Formation Control

The control of robots moving in formation is a relevant research topic these days, as it provides new results of utmost importance for several applications, namely for formation flying spacecraft. Within the RESCUE project, the focus of the preliminary research on this topic, carried out in the last year of the project, has been on formations of land robots moving on flat surfaces, borrowing algorithms from previous work [7], as well as on the extension of such algorithms to formations which include an aerial robot. The mid-term goal is to have these algorithms implemented in the robots so that, in a rescue operation, the blimp may follow a formation of land robots moving toward a destruction scenario, and act as a relay of information, or also making the blimp act as the leader that guides the land robots to the destruction area determined from the aerial observation.

The algorithms used are based on a leader-following principle, but several robots can be connected through a structural formation graph, representing the internal topology of the formation, where followers of one robot may act as leaders of other robot(s) [13].

Algorithms such as Separation-Bearing (where a follower robot attempts to keep given distance and orientation with respect to a leader robot) and Separation-Separation (where a follower robot attempts to keep given distances with respect to two leader robots) [7] were simulated for up to 6 land robots linked by a structural graph, where only the robot kinematics was considered.

Then, an extension of such algorithms that allows the inclusion of an aerial robot in the formation was developed. This extension consists of Separation-Bearing algorithms on the plane orthogonal to the (assumed flat) ground where the land robots move, with some peculiarities resulting from the problem geometry. Simulations of 6 land robots moving in formation with an aerial blimp were carried out, using a linearized version of a detailed dynamics model of the aerial blimp robot, including the identification of its parameters from actual measurements made in the real blimp.

All the algorithms and results are detailed in the JCostal-TechReport.pdf file included in the companion CD. They seem very promising for a future implementation in the actual robots, given the realistic models used.

6 Distributed Planning for a Multi-Robot Rescue Team

Typically, a rescue operation within a situation of catastrophe involves several and different rescue elements (individuals and/or teams), none of which can effectively handling the rescue situation by itself. Only the cooperative work among all those rescue elements may solve it. Considering that most of the rescue operations involve a certain level of risk for humans, depending on the type of catastrophe and its extension, it is understandable why robotics can play a major role in Search and Rescue situations (SaR), especially teams of multiple heterogeneous robots.

The overall goal of the RESCUE project is to develop a robotic team, constituted by more than one robot, capable of autonomously handle a rescue operation. This project can be seen at different levels of abstraction, such as a technological level (e.g., hardware development), a control level (e.g., motor control), a robot navigation level, and a task planning level, if an individual robot is considered. If we are to consider a team of robots, new levels must be added, for instance a level of robot cooperation and a level of mission management. At these levels, the objectives are making robots cooperate to fulfill their common goals, both through cooperative planning and cooperative execution. The RESCUE project aims at the development of an integrated approach to most of referred levels of abstraction, initially for a simplified rescue scenario and a team of two robots (an aerial one and a terrestrial one).

The work developed on multi-robot planning is mainly focused on the problem of distributed task planning for a team of heterogeneous robots. However, all considerations related with technology and utilization of real robots was not an issue in this work. So our rescue team is composed of agents, virtual entities interacting within a simulated environment and capable of some intelligent actions, both individual and cooperative.

The problems of task planning, task allocation and cooperative execution are dealt with mainly in the areas of Distributed Artificial Intelligence or Multi-Agent Systems. The main questions to be answered when solving these problems are:

Selection of goals and their allocation among agents Given a non-empty set of (rescue) objectives, agents must be capable of selecting the right sequence of goals to be fulfilled and distributing these goals among them.

Task planning restricted by the agents actions Given a particular goal, an agent must design a plan of actions that enable it to achieve the goal. Planning in this context means finding a sequence of actions that takes the agent from an initial state of beliefs to another state where a certain set of beliefs is included. This plan can include not only the actions the agent has but also all actions it knows other agents have. Therefore, plans tend to become non linear, i.e., there are actions to be performed in parallel by different agents.

Plan execution Besides ensuring that all pre-conditions for the whole plan and each one of its actions are met, some actions must be synchronized among agents.

Resources management One of the main resources in rescue operations is time, in the sense that timing is usually vital for the rescue success, not only the plan execution time but also the planning time. So a trade off is needed between the quality of plans and time to determine them.

Failures recovery The problem here is to decide what to do when premises for the plan being executed change. Agents must react promptly to changing conditions not only by deciding what to do next, either adapt the current plan or re-plan, but also in order to bring the team of robots, if that is the case, to a common and consistent state of beliefs.

Distributed planning One of the advantages of having several robots is also the possibility of dividing the computational needs among them. For instance, instead of performing task planning in only one robot or computer, one might divide it between two, three or even more robots. The problem of course is to decide who and what each one will plan.

Coherence and cooperation A known problem in multi-robot/agent systems is the possibility that one agents actions could invalidate other agents actions, due to, e.g., non-shared resources. So it is necessary to ensure at execution time the coherence of plan being executed.

Communication Obviously, in a multiple robot scenario, communication is always a relevant issue, both because it is limited and the agent must decide what to communicate.

Given all the problems described above, the project work has focused mainly on the topics of task planning and task allocation in a multi-robot rescue system, assuming that teamwork (i.e., cooperative tasks) plays an important role on the overall planning system.

An agent architecture has been developed, inspired on a Belief-Desire-Intention (BDI) architecture [4], considering that each agent interacts with others in the same rescue scenario, with the same interface and ontology. Moreover, the proposed architecture takes into account issues as agent heterogeneity, failures recover, cooperation, to name but a few. Besides that, agents equipped with this architecture are prepared to act in a non deterministic environment (where its state could change without any agent action), incomplete (meaning that only information agents have is acquired by their sensors which provided only incomplete data about the environment state), dynamic (meaning that planning decisions made for a certain environment state could be invalid when they are executed, claiming for some re-planning).

Since teamwork is a key aspect of this work, agents need to negotiate the execution of certain actions, either because an agent does not have the right skills to do it, or it evaluates that another agent could do it better (with a lower cost). To implement this a Contract-Net system [8] was developed and integrate in the agent architecture. This system allows agents to propose and negotiate contracts with other agents, and gives the necessary guarantees for maintaining signed contracts consistency (i.e., if an agent cannot fulfill a contract it must inform others involved in that contract).

The main decision process, the planner, was implemented based on a Hierarchical Decomposition Partial Order Planner (HDPOP) approach, with an important extension, the possibility to handle (plan) the resources needed for each of the tasks [6, 9]. The planner was developed using the STRIPS language and is supported on a variation of the well-known A^* search algorithm, the Iterative Deepening A^* (IDA^*).

To experiment and evaluate the proposed planning system, a simplified version of a rescue simulator was also developed. This simulator allows to create virtual rescue scenarios where rescue teams should face building and forest fires, civilians trapped in collapsed buildings, and roads blocked. The rescue teams are composed of aerial and land robots, with different skills. The former could perform a survey of the affected region (for instance, by defining a topological map and send it to the other robots, namely the land ones). They are also capable of transporting victims to rescue spots. On the other hand, the latter are endowed with first aid resources and have the autonomy to decide if the victim might be transport by air (in which case it contracts an aerial robot to take care of that transport).

Although this work did not cover all the problems mentioned earlier, the results obtained show that a distributed approach to a rescue problem is clearly an interesting solution when compared with a centralized one. One might lose some quality of the planning solutions, but gains more flexibility, redundancy and the possibility of parallelizing the planning process. One key word emerging from this work and its results was *delegation*, meaning that agents should delegate as much as possible given other agents skills, particularly whenever planning is concerned.

A detailed report on this work can be found in the DinisRodrigo-TechReport.pdf file included in the companion CD.

7 Software and Functional Architectures

The software architecture developed for the RESCUE project is supported on agent-oriented programming concepts that provide a systematic method for task design, task planning, task execution, task coordination and task analysis for a multi- robot system. An application program interface (API) was implemented [17].

The conceptual model of the agent-based software architecture includes different types of agents that can be combined both hierarchically and in a distributed manner. The architecture supports information fusion between several sensors and the sharing of information between the agents by a Blackboard (a distributed structure that gives support to the data exchange between the Agents), and is geared toward the cooperation between robots. Agents are generically organized hierarchically. At the top of the hierarchy, the algorithms associated with the agents are likely to be planners, whilst at the bottom they are interfaces to control and sensing hardware. The planner agents are able to control the execution of the lower level agents to service high-level goals. The latter can be distributed across several processors and/or robots. To offer platform independence, only the lowest level agents are specific to the hardware, and these have a consistent interface for communication with the planning agents that control their execution. The elements of the architecture are the **Agents**, the **Blackboard**, and the **Control/Communication Ports**. Agents communicate with each other through *control ports* and with the blackboard through *data ports*. The latter is effectively another means of sharing information among the agents.

In Robotics research and development, much time and resources are consumed in system design, system calibration and system analysis. A well-designed architecture targets the support and speed-up of these development phases. Usually, properties such as system distribution and concurrency are relevant during the mission execution, since they provide better resource allocation and robustness. Under this architecture, a different execution mode exists for each development phase of a multi-robot system. Five execution modes are available for each of the elements described in the previous section:

Control Mode that refers mostly to the run-time interactions between the elements.

Design Mode

Calibration Mode

Supervisory Control Mode

Logging and Data Mode .

A detailed report on this work can be found in the JFrazao-TechRep.pdf file included in the companion CD. This reference guide is targeted for researchers and students working on the RESCUE project, as well as to future users of the architecture, extendable to other projects.

8 Conclusions and Future Plans

During the RESCUE project lifetime, the research team at ISR/IST, composed of senior and young researchers from the Intelligent Systems, Mobile Robotics and Computer Vision Labs, was engaged in carrying out the implementation of a reference scenario defined in the 1st year, that drove, as intended, research in multiple complementary fields, namely on the navigation of multiple outdoors robots.

Major achievements of the project were:

- the topological navigation of a land robot, with tests in both indoors and outdoors scenarios - this work contributed to the accomplishment of a PhD thesis;
- vision-based navigation of the aerial (blimp) robot in indoors environment, including a full modeling of its dynamics, used in control simulations (a simplified version was used in real tests);
- theoretical results on the navigation controllability (using Discrete Event System models) of a Mobile Robot population, composed of heterogeneous robots, with examples in a Search and Rescue application;
- simulated results of formation control involving 1 aerial blimp robot and up to 6 land robots moving in formation according to a specified geometry, and changing that geometry at some point in the path;
- simulated results of distributed planning for a multi-robot rescue team, mainly focusing on task planning and task allocation, assuming that teamwork (i.e., cooperative tasks) plays an important role on the overall planning system.
- foundations of software and functional architectures which support the integration and development of all the multi-robot team sub-systems (e.g., navigation, formation control, planning).

Not all goals of the reference scenario were accomplished, but the research will proceed under the Theme B of the Associate Laboratory ISR-Lisboa, "Robot Monitoring and Surveillance".

Future plans include at least the following topics:

- use vision to track robots in formation control;
- obtain aerial topological maps of disaster scenes using the vision camera onboard the blimp robot;
- endow the aerial robot with GPS to decouple navigation and mapping/robot-tracking sensors;
- tackle representation problems resulting from the required matching between the topological maps created by the aerial and land robots;
- move on to research on task planning and coordination for multi-robot teams, applied to real outdoors robots.

References

[1] J. Andrade-Cetto, A. Sanfeliu (2002). Concurrent map building and localization on indoor dynamic environment. *International Journal of Pattern Recognition and Artificial Intelligence* **16**(3), 361 – 374.

[2] T. Bailey, E. Nebot (2001). Localization in large-scale environment. *Robotics and Autonomous Systems* **37**, 261 – 281.

[3] M. Bosse, P. Newman, J. Leonard S. Teller (2004). Slam in large-scale cyclic environments using the atlas framework. *International Journal of Robotics Research.*

[4] M.E. Bratman "Intentions, Plans and Pratical Reason" (2nd edition) Addison-Wesley, 1987.

[5] C. G. Cassandras, S. Lafortune, "Introduction to Discrete Event Systems", Kluwer Academic Publishers, 1999

[6] K. Decker and V. Lesser "Designing a Family of Coordination Algorithms" U. Massachussets Computer Science Technical Report 94-14, Aug 1995

[7] J. P. Desai, J. Ostrowski, V. Kumar, "Controlling formations of multiple mobile robots" *IEEE International Conference on Robotics and Automation*, Belgium, May 1998, vol. 4, 2864 -2869

[8] M. D'Inverno and M. Luck "Normalizing the Contract-Net as a Goal Directed System" *in MAAMAW'96, Springer-Verlag, 1996*

[9] E. Durfee and V. Lesser "Generalizing the Partial Global Planning Algorithm" U. Massachusetts, June 1993.

[10] M. Egerstedt, (2000). Behavior based robotics using hybrid automata. *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science. Springer-Verlag* pp. 103 – 116.

[11] E. Fabrizi, A. Saffioti (2000). Extracting topology-based maps from gridmaps. *Proceedings of the 2000 IEEE Conference on Robotics and Automation, San Francisco* pp. 2972 – 2978.

[12] J. Fenwick, P. Newman, J. Leonard (2002). Collaborative concurrent mapping and localization. *IEEE Conf on Robotics and Automation - Washington, DC.*

[13] R. Fierro, A. Das, J. Spletzer, R. Alur, J. Esposito, Y. Hur, G. Grudic, V. Kumar, I. Lee, J. P. Ostrowski, G. Pappas, J. Southall and C. J. Taylor, "A framework and architecture for multirobot coordination" *International Journal of Robotics Research*, vol 21, num 10-11, pp 977-995, Oct-Nov 2002.

[14] J. Folkesson, H. Christensen (2004). Graphical slam - a self-correcting map. *Proceedings of IEEE International Conference on Robotics and Automation* **1**, 383 – 390.

[15] M. Franz, H. Mallot (2000). Biomimetic robot navigation. *Robotics and Autonomous Systems 30.*

[16] M. Franz, B. Scholkopf, H. Mallot, H. Bulthoff (1998). Learning view graphs for robot navigation. *Autonomous Robots 5.*

[17] J. Frazao, P. Lima, (2004). Agent-Based Software Architecture for Multi-Robot Teams, *Proceedings of The 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal.*

[18] J. Gaspar, N. Winters, J. Santos-Victor (2000). Vision based navigation and environmental representations with an omni-directional camera. *IEEE TRA.*

[19] L. Gerstmayr, A. Bernardino, J. Santos-Victor (2004). Appearance based landmark selection and reliability evaluation for topological navigation. *Proceedings of The 4th IFAC/EURON Symposium on Autonomous Vehicles (IAV04), Lisbon, Portugal, July 2004.*

[20] S. Grigorescu, N. Petkov, P. Kruizinga. "Comparison of Texture Features Based on Gabor Filters", IEEE Transactions on Image Processing, **11(10)**, 2002.

[21] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, S. Shimada "RoboCup Rescue: Search and Rescue in Large-Scale Disasters as a Domain for Autonomous Agents Research" *Proceedings of IEEE International Conference on Man, Systems and Cybernetics, 1999*

[22] P. Lima, M. I. Ribeiro, L. Custodio, J. Santos-Victor "The RESCUE Project - Cooperative Navigation for Rescue Robots", Proc. of ASER'03 - 1st International Workshop on Advances in Service Robotics, March 13-15, 2003 - Bardolino, Italy.

[23] Y. Liu, S. Thrun (2003). Results for outdoor-SLAM using sparse extended information filters. *Proceedings of The IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan* pp. 1227 – 1233.

[24] F. A. Melo, M. I. Ribeiro, P. Lima, "Blocking Controllability of a Mobile Robot Population", Technical Report RT-601-04, Institute for Systems and Robotics, 2004

[25] F. A. Melo, M. I. Ribeiro, P. Lima, "Navigation Controllability of a Mobile Robot Population", RoboCup 2004 Symposium, Lisboa, Portugal, 2004

[26] F. A. Melo, P. Lima, M. I. Ribeiro, "Event-driven Modelling and Control of a Mobile Robot Population", Proceedings of the 8th Conference on Intelligent Autonomous Systems, Amsterdam, Netherlands, 2004

[27] R. Smith, P. Cheeseman (1987). On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research, Published by MIT Press* **5**(4), 56 – 68.

[28] I. Ulrich, I. Nourbakhsh (2000). Appearance-based place recognition for topological localization. *Proceedings IEEE International Conference on Robotics and Automation, San Francisco,* pp. 1023 – 1029.

[29] A. Vale, M. I. Ribeiro. "A Probabilistic Approach for the Localization of Mobile Robots in Topological Maps", Proc. of the 10th IEEE Mediterranean Conf. on Control and Automation, Lisboa, Portugal, 2002.

[30] A. Vale, M. I. Ribeiro. "Environment Mapping as a Topological Representation", Proc. of the 11th International Conference on Advanced Robotics, Coimbra, Portugal, vol. 1, pp. 29-34, 2003.

[31] A. Vale, J. M. Lucas and M. I. Ribeiro (2004). Feature extraction and selection for mobile robot navigation in unstructured environments. *Proceedings of The 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal.*

[32] G. Zunino, H. I. Christensen (2001). Simultaneous localisation and mapping in domestic environments. *Multisensor Fusion and Integration for Intelligent Systems* pp. 67 – 72.