ROMA TRE
UNIVERSITÀ DEGLI STUDI

Scuola Dottorale in Ingegneria - Sezione Informatica e Automazione

XXIII Ciclo

A multi-robot system: from design to intelligent cooperation

Maurizio Di Rocco

A.A. 2010/2011

Tutor: Prof. Giovanni Ulivi

Coordinatore: Prof. Stefano Panzieri

1. Reviewer: Achim J. Lilienthal

2. Reviewer: Paolo Robuffo Giordano

Day of the defense: 26 March 2011

A multi-robot system: from design to intelligent cooperation

Maurizio Di Rocco

Scuola Dottorale in Ingegneria - Sezione Informatica e Automazione
XXIII Ciclo

A thesis submitted for the degree of
PhilosophiæDoctor (PhD)

A.A. 2010/2011

Abstract

The proposed work aims at giving a contribution in the domain of multi robot systems.

In the last years these architectures, thanks to their flexibility and robustness with respect to monolithic structures, have been deeply investigated in various applications.

Looking at commercial platforms, one of the main limitations relies on the hardware configuration. In most of cases, the low-level control software is not open source, compromising a detailed customization of the robot. This drawback can be partially overcome by the usage of plug-in boards which ease the use of a particular sensor. However, whenever an application requires heterogeneous transducers, the management of several boards increases the computational burden of the CPU, the energy consumption and costs.

The first contribution of this thesis is the description of the prototyping of a multi robot platform. The robots, based on a dual layer architecture, have been designed to be low-cost and flexible in order to cope with several scenarios. The reduced dimensions allow the units to execute experiments in small environments, like the ones typically available in a research laboratory, even interacting with commercial platforms like sensor networks.

Afterward, several experimental research activities are covered.

In the field of motion control, two techniques for the encircling problem, i.e. the task in which a set of robots rotates around a target, are presented. This toy problem is very appealing to the scientific community because it deals with a series of technological issues which involve communications, motion estimation and control design.

Also the data fusion, which is one of the most treated problem in robotics, has been taken into account. One of the traditional approaches resides on the Bayesian framework. This approach allows a straightforward combination of data gathered by several sensors. In this dissertation, such technique has been used in environmental

monitoring. In particular extensive experiments were performed using the aforementioned platforms in the domain of gas source localization: the task consists in the search of a source releasing gas in an environment. Besides the Bayesian framework, other techniques can be used. In particular a focus on the Transferable Belief Model (TBM) is provided. Unlike the probabilistic formulation, the TBM is able to effectively represent the concept of ignorance and contradiction, concepts which, for example, belong to the human kind of thinking. In this work a technique for the data fusion in a multi agent context is proposed. The theoretical results have been validated by extensive experiments concerning the cooperative topological map building.

Finally a lookup view about the integration between mobile and static sensor networks (SN) is provided. One of the main issues for a static network is to know about the interdistances between nodes. In this work a localization algorithm, based on a modified Extended Kalman Filter, is showed. As for the previous topics, experimental results support the theoretical approach.

In the last chapter conclusions and possible improvements are discussed.

To every person who transparently and selflessly gave me the gift of his time,

Acknowledgements

This thesis represents the work of 3 years. During these years, I learned tons of things and I really would like to thank all of those people who helped me. First of all, I really would like to thank all of the students who worked in a passionate way: they gave me enthusiasm during the hours spent in the lab. A particular thanks goes to Giovanni and Enrico. Second, I really thank people who contributed to develop my skills during this period: the biggest thanks goes to Alessandro. In a very short period, he gave me many advices. By these advices, I truly realized that my PhD has been a valuable engineering work but, at the same time, a modest research contribution. Although this is a difficult matter to realize, I am happy to have understood many, probably too much, mistakes that have been present during those years: however, in my opinion, it is better to accept this situation than to ignore what happened. I truly thank him for the time he spent with me speaking about how to approach a problem and how to organize things.

I really would like to thank Achim: he contributed with many advices, too. Similarly to Alessandro, he gave me a new and orthogonal point of view about what the word ”research” means: whenever we spoke, he always stressed points which, in my opinion, a researcher should always keep in mind.

A great contribution was also given by Prof. Lorenzo Sciavicco: although he was not in contact with my research studies, he always demonstrated a great curiosity and interest whenever we spoke: in my opinion, he is a great example of how people should approach the research. Thank you very much.

I really would like to thank all the colleagues who actively contributed in those years: Antonio and Paolo, and all the guys, first of all friends, from Örebro. Thanks to Matteo, he was simply amazing when I arrived in Örebro: MITICO !. A special thank to Marco, Marcello, Federico, Tosho and Robert: it was a priceless experience (apart from TOFU :-D). You are a great group:TACK !

A special thanks to all other people who withstood me in those three years: unfortunately I think it has been a very useful experience although not happy, thus I apologize for the moments during which I was really stressed.

Last, but not least: I would like to thank people that, similarly to me, made mistakes. I learned a lot of things from these situations, too. I hope this experience taught to all of us something.

Contents

Chapter 1

Introduction

In the last years, Multi robot systems have gathered a lot of interest. This topic represents the $9-10\%$ of the titles in the IEEE International Conference on Robotics and Automation in the last decade.

The great advantage in using multi robot architectures resides in an enhanced robustness, flexibility and efficiency with respect to monolithic structures. The usage of multiple units intrinsically allows a greater resiliency to faults, a parallelization of the task execution among multiple agents and consequently a shortening in time execution. The development of these systems has gone hand in hand with the technological improvements in the micro-machining fields. Nowadays the miniaturization, i.e. the energy consumption reduction, allows the development of intelligent electro-mechanical devices keeping low the costs. Current embedded CPUs can run standard OS and consequently can mount conventional sensors/devices. De facto, these robots can be considered like small wireless units equipped with a considerable computation power.

The enhancement with respect to single robot architectures relies largely on the ability of exchanging information by wireless channels. In this manner, the team can instantaneously have a perception regarding different zones and consequently achieves a better understanding of what happens within the operating scenario.

On the other side, such advantages are paid with sophistication in the dynamics involving the overall system. This architecture can be represented by a set of simple, possibly different, systems linked each other by communication channels.

As a consequence, the stability of the algorithms running over a distributed system is much more difficult to analyze with respect to the past. Indeed, the communication topology, which in most of cases changes rapidly over the time, couples the dynamics of the interconnected systems.

In last years, a big effort has been given to the study of ”networked” systems, especially those concerning coordinated motion and the data fusion.

The first topic aims at moving a set of agents in a coordinated manner while the second one tries to achieve useful considerations taking into account observations coming from different units, possibly dislocated over the operating scenario.

1.1 Movement control

The motion ability is one of the distinctive features of a multi-robot system with respect to conventional sensor networks. While the control for a single agent has been deeply studied and can be considered an achieved goal by the scientific community, the extension to multiple units is still susceptible of improvements.

The proposed techniques can be currently divided into two main areas: the first one tries to steer the agents into precise geometrical shapes by means of analytical control laws, while the second one finds inspiration from the biological world.

The former approach results effective to mathematically characterize the behavior of the system in terms of stability. However these algorithms rely on assumptions which hardly fit real scenarios, although they are very precise and well posed.

The bio-inspired algorithms instead aim to create an emergent behavior. They are based on the principle that local interactions, based on very simple rules, can lead to a complex global behavior when a large swarm is considered. With respect to analytical approaches, the convergence properties are rather difficult to demonstrate even if results can be very impressive.

However, in both cases the interaction between agents plays the fundamental role. The interaction is mainly based on the sensorial and communication capabilities of the robots.

Sensors are important in order to gather relative information, e.g. distance and possibly angular displacement, among agents. They vary from simple noisy transducers to sophisticated and expensive range finders which are able to gather very precise measurements. The kind of measures and their quality directly influence both the control design and the quality of the movement.

The kind of communication is instead strictly related to the task time completion: power transmission and bandwidth tune the propagation speed of the information among agents. As for the sensors, these devices can be very simple, e.g. a light emitter, or very expensive, e.g. a large-bandwidth modem. Besides performance and costs analysis, a particular care about power consumption has also to be taken into account: especially when small devices are considered, the capacity of accumulators plays a fundamental role in the overall design.

1.2 Sensor fusion

Data fusion is a fundamental topic in robotics and in the overall scientific community. To have a full autonomy, a robot has to effectively process the data gathered by sensors in order to decide what to do.

Multi agent systems naturally enlarge the amount of information that a unit has to process: data coming from different places allow a better understanding of what is happening in the operating scenario.

This advantage is paid with sophistication in the model description: the estimation processes running on each robot are coupled by means of wireless communication channels.

Furthermore, this coupling, i.e. the topology of the communications, is dependent on the transmission capabilities and on the movements of the robots, i.e. it can rapidly change over the time.

In last years, a big effort has been made to study the behavior of these systems: a large amount of studies exploits the Bayesian framework. It allows a straightforward combination of data coming from different sources. In this context, a multi robot system can be modeled like a network whose nodes are the robots and the edges represent the communication links. Each node samples a given probability distribution and consequently fuses data coming from other sources by the probability theory. This approach is very effective when the probability distributions can be described by specific characteristics, i.e. the noise affecting the transducers can be described with a Gaussian function. In this case the approach is computationally efficient and very easy to implement. However, even in this case, the topology of the communication links can lead to complicate situations which have to be carefully taken into account.

If the Bayesian framework provides an effective and easy-to-implement method to process data gathered from transducers, on the other side it does not provide the same expressivity in high level reasoning. For example, its rigid structure does not provide a natural representation of contradiction between sources. Furthermore, it difficulty handles the concept of ignorance: when an ambiguous measure is considered, typically the amount of belief is equally split among the most prominent hypotheses; this approach, for example, differs from the human kind of thinking which takes into account the union of prominent hypotheses and not its specifications.

Related to these situations, alternative frameworks have been proposed in literature. These techniques describe an effective paradigm in cases where complex, i.e. contradictory and ambiguous, situations raise. However, this flexibility is paid with a huge increase of the computational burden. This is the main reason why they are often relegated to monolithic and static structures. On the other side, the advent of multi agent systems may naturally enhance the execution of these techniques. Currently, the application of alternative paradigms over distributed systems is still an open field susceptible of improvements.

1.3 Contributions

The main contributions of this thesis can be stated in three points.

The first one concerns about the prototyping of a mobile platform. This robot, replicated in 12 units, has been designed in order to execute multi robot algorithms. Differently from commercial platforms, this architecture is fully accessible in terms of control software and can be easily adapted to several scenarios without the insertion of adjunct boards. These peculiarities are demonstrated by several experiments which are described in this dissertation. At the moment, related publication about this topic has been published in (1). Currently, (2) has been submitted.

The second contribution is related to the formation control: in particular the encirclement problem is treated. This topic has been deeply studied and many techniques have been proposed over the years. However, the majority of the algorithms require the knowledge of certain parameters which are difficult to estimate: to overcome this drawback, an external and centralized monitoring system is often used. In this dissertation, a fully autonomous multi robot system has been considered. Furthermore, control laws have been designed in order to avoid the use of parameters which are particularly sensitive to error estimate. At the moment, the related publication about this topic is (3).

The last contribution is related to data fusion. In this context, a big effort has been given by the scientific community to the enhancement of the Bayesian framework for multi robot systems. In this thesis, the Bayesian approach has been used in the domain of environmental monitoring in combination with a stigmergic approach in a smart environment. In particular, extensive experiments concerning the gas source localization have been conducted. Another contribution relies on the study of non-conventional paradigms in multi agent systems: in particular, an algorithm concerning the data fusion upon the Transferable Belief Model framework is presented. In this case as well the algorithm is supported by real experiments concerning the cooperative topological map building. At the moment, the related publications about this topic are (4), (5), (6), (7). Currently, (8) and (9) have been submitted.

In the final part of the dissertation, some notes are devoted to the integration between static and mobile systems. The related chapter does not pretend to go in deep about this topic: it just presents some preliminary technique about the static sensor network localization. In particular, this task can be viewed like a fundamental prerequisite in order to achieve the aforementioned integration. Before starting any task execution, in fact, a static sensor network has to know the location of the nodes. Here we propose a discussion about a distributed technique based on the Extended Kalman Filter. The algorithms are supported by experiments conducted using real devices. At the moment, (10) has been published about this topic.

Chapter 2

An experimental platform for multi robot algorithms

In this Chapter a mobile platform built from scratch is described. The main peculiarity relies in its flexibility: it mounts several sensors and others can be added in order to cope with several scenarios. A particular care regarding mechanical and electrical design has been taken in order to keep low the costs. Furthermore the communication capabilities allow the interaction with standard sensor networks. All these components are managed by dedicated software architecture

2.1 Introduction

A rather new and very interesting research field in mobile robotics investigates the cooperation of several units and in particular the possibility of getting a collaborative behavior by decentralized algorithms. Indeed, decentralized algorithms show definite advantages over centralized ones. They are naturally resilient to one point failure (the loss of one unit), can easily reconfigure themselves and don't require a leader in the team. Moreover, they can take benefit from the increase of computing power that can be achieved parallelizing the activities of all the computers of the team. At the same time, the number and the length of inter-robot communications can be kept smaller than those necessary for centralized activities.

The field of decentralized algorithms can be roughly divided in two areas. The first and older one is that of swarm robotics where the control is performed by rule-based de-

signed behaviors (e.g. (11), (12), (13)). Tuned by a trial and error approach, they can provide a very good performance and can be adapted to different robots and environments. Research works in the other area try to decompose well posed (in an analytical sense) algorithms in fragments that can be run onboard each robot using its small computing and communication capabilities (e.g. (14), (15)). Under these regards, the latter area shows several similarities with sensor networks and can even be useful to consider each robot as a moving node of such a network. The analytical demonstration of their convergence to the correct results is often carried on under milder assumptions than their centralized relatives and often assuming far from reality characteristics as Gaussian, zero mean errors, lossless communications, strict inter robot synchronization. In both cases, the results are strongly influenced by the available onboard sensors, the reliability and the range of the communication system and obviously the working environment.
A serious assessment about the effectiveness of these algorithms requires therefore a good deal of experimental work and a team of robots. Thus, our research group decided to develop a prototype of a mobile robot that is low priced, in order to be replicated in many copies, and at the same time is equipped with many sensors, able to communicate over different channels, endowed with a sufficient computing power and easy to program.

SAETTA has been designed to fit these requirements. To have a low cost, we made some mechanical choices that positively influenced also the electronic part. The first is to build a standard differential-drive wheeled robot for indoor use and give up interesting, exotic locomotion approaches that are useful in more challenging environments. Low cost derives also from the recent improvements of computer boards and even more from the advance of micromachined sensor technology. Several kinds of kinematic transducers are indeed included in the design, the utility of which will be discussed in the sequel. Moreover, we gave up any aesthetic concern, at least till now. The size of the robot was chosen in order to conduct experiments in a limited space with, say, ten units but at the same time sufficiently large to require no complex (and costly) miniaturization and to allow a few limited expansions.

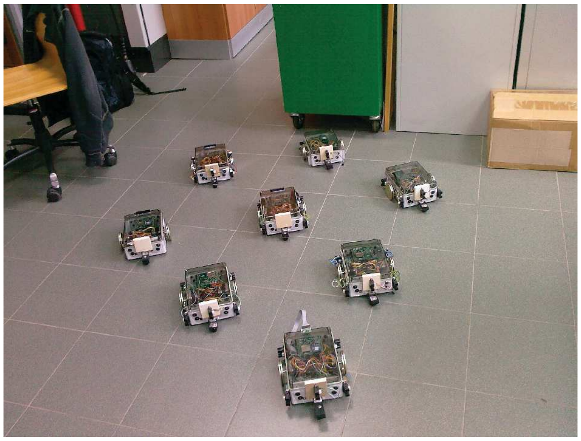


Figure 2.1: Saetta robots: a part of the 12 Saetta robots developed at DIA.

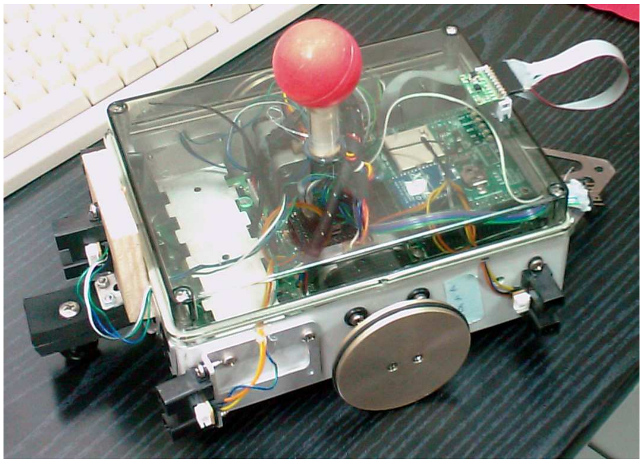


Figure 2.2: The Saetta robot: along the body IR sensors are visible, while in the rear part, above the cap, the compass is located. On the top, a beacon allows to acquire the ground truth for performance comparison by an external camera.

2.2 Mechanical and traction design

The traction system of the robot and its mechanical structure play an important role in keeping simple the overall project and low the cost of each unit. One important choice has been the use of stepper motors instead of the more common d.c. ones. They offer several advantages. First of all, a stepper motor needs neither a tachometer nor an encoder nor, as a consequence, the circuitry associated with the transducer. As for the controller, the situation is not so clear, as a stepper motor requires no kinematic control loop, but it implies a rather complex timing, in particular when micro-stepping is used.

Thus, for this specific point, the difference is made by the hardware resources of the control processor that, in our case, are sufficient to perform all the needed operations. Moreover, a stepper motor requires lower supply voltages as it has a very low back EMF; this means less cells in the accumulator and a simpler circuitry. Finally, the provided torque is sufficient to move the robot without a gearbox even using large diameter wheels.

On the other hand, a stepper motor leads to longitudinal vibrations whose level has been kept low by a proper choice of the motor and of the driving technique (see Section 2.4.3).

The case of the robots has been kept as simple as possible without paying attention to the aesthetical results. We started from a standard PVC wiring box that is cheap and offers a sufficient mechanical robustness. Its rigidity, already good when the cover is inserted, has been enhanced by an internal aluminium bar that also carries the accumulators while the motors are screwed on the long sides of the box. The motors can carry a rather heavy transversal load that is several times the weight of the robot. Therefore, they can be placed so that the center of gravity of the finished unit is about aligned with the motor axes that support the wheels. In this way, a simple plastic sphere can be used for the third rest avoiding the use of a castor. The castor has been coupled with a small part of neoprene which acts like a damped spring. This choice was evaluated in order to reduce the amount of longitudinal vibrations induced by the stepper motors which resulted in a discontinuous floor/sphere contact. In this way the vibrations do not disturb the motion of the robot, that indeed shows a very good odometry. The relevant data of the robot are referred in Table 2.1.

Table 2.1: Robot specifications

Size	14×29cm
Battery voltage	9,6 V
Current consumption	$600 \div 1000$ mA
Wheel radius	3.6cm
Max speed	30 cm/s

2.3 System architecture

The electronics of the robot is organized as a two tiers architecture. The first tier takes care of motors and sensors while the other one executes higher level control and communication tasks. This conceptual division has an immediate correspondence in the hardware realization: each tier is realized on a separate electronic board and has its own CPU.

The first board is custom designed around a Programmable Interface Controller (PIC) while the other is a commercial ARM9 board running embedded Linux (NETUS). The two boards communicate via a high speed serial interface. Interboard connectors provide, besides the electrical connections, the mechanical support to the ARM9 board.

At the moment, a 2 axis gyroscope, a 3 axis magnetometer (only two used), a 3 axis accelerometer and 5 infrared (IR) sensors are connected to the lower level board.

2.4 Low level tier

This tier is implemented on a 18F87J50 PIC from Microchip. It is hosted by a board whose size is approximately 13×27cm that, besides the CPU, accommodates the sensors (that are actually mounted on mini piggy-backs), the motor drivers, the power supply and the connectors carrying the other board.

The power source is constituted by the series of 8 AA cells providing a nominal voltage of 9.6V and a total capacity of 2.3Ah. Two supply voltages are needed: 5V and 3.3V. The conversion from the battery pack voltage is obtained by the cascade of two high efficiency converters. The overall autonomy depends on the characteristics of the experiment, i.e. the traversed space and the used peripheral and sensors. On average, the robot has a 90-minutes autonomy.

The board can work in a stand-alone way when the ARM9 board is not installed, connecting the PIC by jumpers to a dedicated connector for a wireless ZigBee module, thus obtaining a simpler but cheaper configuration.

2.4.1 Central processing unit tasks

The PIC is an 8 bit controller. The main features which are interesting for the project are detailed below:

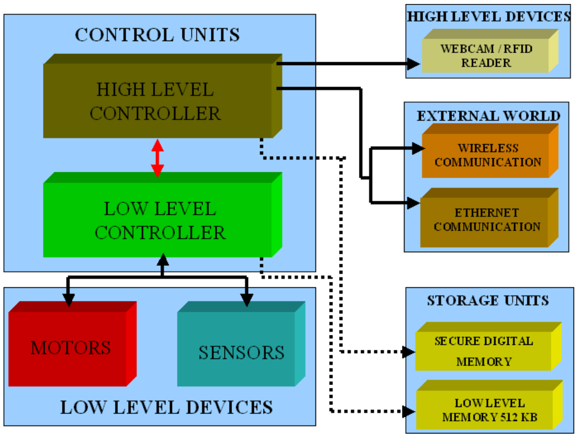


Figure 2.3: System architecture: each peripheral is connected to a specific controller. The two controllers are linked (red arrow) by an RS232 communication.

- Clock frequency 48MHz
- Supply voltage 3.3V
- Flash program memory 128KB
- A serial RS232 communication port
- 12 analog inputs with 10 bit converter
- 4 timers/counters
- several digital I/O lines

A timer has been used to trigger the control cycle T_{PIC} of the processor, in our case 25ms. The control period is responsible for managing the synchronous operations like channel sampling, data filtering and the coordination with the high level tier.

The main activities the PIC has to perform are: transducers sampling, control of the actuators by means of digital lines and communication with the high level tier and eventually with a ZigBee module if no upper level tier is used. In the latter case, the PIC is also used to run all the subtasks of the mission.

Asynchronous operations, like actuation and communication management, have to be taken into account, too. To this aim, some interrupts originated by the internal PIC hardware are used: the related description is detailed in Section 2.4.2.

2.4.2 Software

The software architecture is similar to the *interrupt design pattern* described in (16).

The program is constituted by two main blocks: an infinite loop that executes low speed operations, and a very frequent and short routine. The latter is triggered by a timer which expires every 50μs: it checks for events related to some peripherals (serial port, motors, analog sampling, etc) and, if needed, it performs some fast actions (setting a flag, buffering of a received character, step movement, storage of acquired data, etc). It performs only very simple operations and is not preemptive.

Flags enable the execution of slower operations in the infinite loop: if a flag is set, the launch of the related long term action (like message parsing, notification of served positions, data filtering, etc) is performed.

2.4.3 Motor Control

Each motor is controlled by a dual full-bridge pulse width modulation (PWM) driver. The driver is a 24 pins dual inline package (DIP). It provides both electrical and thermal protections. Stepper motors provide a 200 steps resolution with a nominal torque of 12 Ncm. The motor windings are independently controlled: four current levels can be imposed, ranging from 0% to 100% of the max current value. The control is done acting on a couple of pins controlled by the PIC. Furthermore, another pin is used to impose the current direction in the winding. Globally, six digital lines per motor are used by the PIC to perform the control which results in a PWM of fixed period. The electrical period is at least constituted by four intervals. When using micro-stepping, the intervals are multiple of four and in our case eight or twelve intervals have been tested. As a consequence a resolution respectively of 400 and 600 steps can be achieved. Figure 2.4 shows the vibration analyses on the mechanical structure for these two kinds of control when the wheel rotates at 2.5Hz. Data were collected by a *Type 4371* accelerometer from *Bruel & Kjaer*. The twelve intervals solution clearly shows lower vibration levels and it is used for the whole range of operation even if it is slightly more demanding in terms of CPU operations. As a result, the robot odometry on a typical floor is very good. After estimating the wheel diameter by calibration, the error along a straight line is less than 1%.

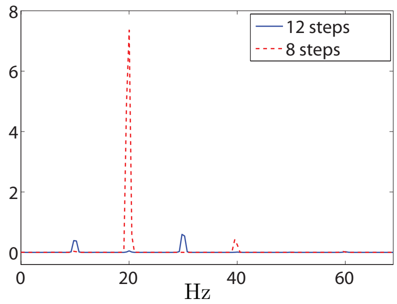


Figure 2.4: Vibrations gathered with twelve (continuous blue) and eight intervals control (dotted red). On the x-axis Hertz units are represented while on the y-axis the magnitude of the signal gathered from the accelerometer is showed. The eight intervals control shows a peak centered at about 20Hz.

2.5 Sensor characteristics and calibration

The robot is provided with a rather complete set of kinematic sensors. In particular, the proprioceptive ones are a 3-axes accelerometer, a 2-axes gyroscope and a 3-axes magnetometer (to be used as a compass); odometry sensors have not been included as stepping motors are used. The range of exteroceptive sensors connected to the PIC comprehends just five IR range finders as more sophisticated units are driven by the upper tier processor. In this tier, the transducer furnishes analog sources which are sampled by the analogical to digital converter provided by the PIC. When needed, a small circuitry constituted by an amplifier and low pass filter is put between the transducer and the PIC.

All the channels are sampled with a burst acquisition: in each control cycle T_{PIC}, the PIC collects N (in our case $N = 8$) samples for every channel at maximum sampling rate ($60 - 100\mu s$ per channel). The acquisitions are then averaged to reduce random fluctuations. It is worth noting that the number of samples is a power of 2 to improve the computation time using shift operators.

In the following the aforementioned sensors are described, together with their performances and calibration, when needed.

2.5.1 Infrared sensors

Each infrared sensor, a GP2Y0A02YK from Sharp, is composed by an emitter and a separate detector. The analog range measurement is obtained by a triangulation of the spot projected on the obstacle. Intensity is discarded, thus the sensor is insensitive to the obstacle reflectance. However, differently from the other sensors mounted on the robot, the relation between the output voltage of the sensor and the distance is not monotonic and shows a maximum near 20cm. Therefore the same reading can refer to two different distances (see Figure 2.5-a) and this ambiguity is located in a range that can be very dangerous with respect to the safety of the robot (e.g. collision avoidance). Thus, a *dangerous* and *safe* zone must be considered in the IR curve. One way to overcome this problem is considering subsequent readings to disambiguate from the two zones. For each unit, the calibration was effected using a standard polynomial fitting considering a separate subset of data for each zone. As shown in Figure 2.5-a, a small difference results between the tested units, however it can be reduced to acceptable values by a simple vertical translation of the calibration curve: the result of this operation can be viewed in Figure 2.5-b. The operative range can be considered to be from 7cm to 120cm: out of this range, measures are useless due to the noise and the discretization of the converters. Errors are particularly important for distances greater than 80 cm: the standard deviation of the measure, even averaging 64 samples, is near 10cm and affects in particular localization and mapping tasks. Instead, for distances between 25cm and 50cm, the repeatability is about 1cm. Along this range, the mean error is under 2cm.

Each IR sinks about 50mA, thus the PIC can turn ON or OFF their supply line by a transistor.

2.5.2 Accelerometers

The accelerometer device is a 3 axes unit (MMA7260QT from Freescale semiconductor). Its range can be chosen among four values from 1.5g up to 6g by 2 pins. This setting is typically applied at power up, but online changes are possible. Actually, considering the noise superimposed to the measures and the difficulty to get rid of the offsets, these sensors cannot be reliably used for velocity or position estimation and they find the main application to state the position with respect to the horizontal plane, an information that can be very useful in connection with the sensors connected to the upper tier, e.g. the camera. In particular, supposing that initially the robot starts the mission on a horizontal plane, a different posture is recognized if the values gathered on the axes differ from the initial ones. To this purpose, the maximum sensitivity is

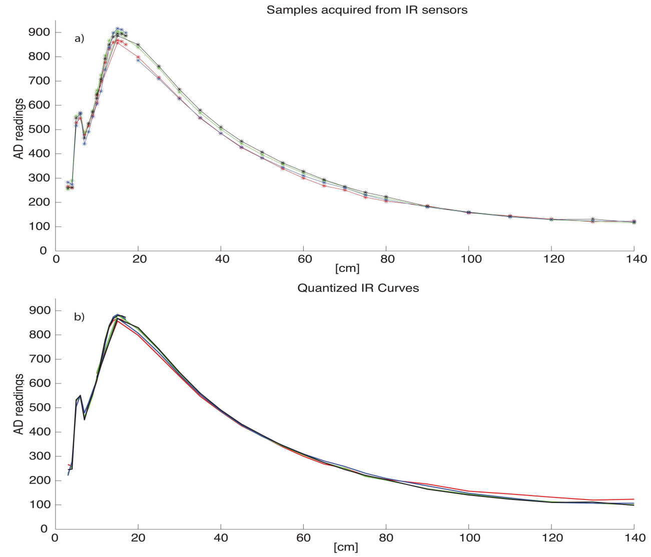


Figure 2.5: Calibration curves obtained from different sensors: (up) original data; (down) curves after the unbiasing. Y axis are AD units, on the x axis distance in cm is reported.

used. As in this case two axes are sufficient, it is possible to leave out one axis by a jumper (the one oriented along the non-holonomic constraint of the vehicle) to sense the battery voltage.

2.5.3 Rotation sensors

This section briefly describes the behavior both of gyro and magnetometer sensors. The gyro is a 2-axes ADXRS150 sensor from Analog Devices, oriented along the roll and yaw axes of the robot. At the moment, the most used part is the one oriented along the yaw axis, which measures the rotational speed of the robot.

The magnetometer is a three-axes HMC1043 sensor from Honeywell. To spare an analog input, only two axes were considered, being them sufficient to determine the position of the Earth magnetic field on the horizontal plane. To adapt the chip output levels to those of the analog/digital converter, two operational amplifiers are needed. The two components can be affected by a bias related to the residual magnetization of the sensitive core. Each of these biases cannot be estimated as they change randomly at every power-on. To overcome this drawback, a calibration, consisting in a full self-rotation of the robot, has to be performed at the beginning of each experiment. Alternatively, the magnetometer provides a procedure consisting in a reset driven by a train of current pulses injected into a digital line of the chip. Although effective, in our case the amount of current needed by this operation is too much onerous for the battery capabilities.

Also in the case of gyro, a calibration at each power-on is performed by self-rotations. The gyro calibration curve revealed a linear behavior with respect to the rotational velocity of the robot (see Figure 2.6).

The magnetometer outputs instead showed the expected profile: each axis returned a sinusoidal curve representing the projection of the magnetic field on that direction. Two remarks are in order: first, the currents flowing in the robot's circuits has a negligible effect on the readings of the magnetometer; second, the gyro is very sensitive to temperature conditions. In our experiments, we noted that these variations were not negligible. In order to reduce this phenomenon, we tried to estimate the bias whenever the robot stopped: tough that some improvements was obtained, the gyro did not result reliable over long periods to achieve velocity and nevertheless position measures, especially over long periods.

Consequently the velocity integration is unreliable on long time intervals.

During the navigation experiments, however, the two sensors showed rather different error characteristics. While the gyro maintained the expected behavior, the compass resulted unreliable in some areas of the arena where experiments were performed: these situations were observed when passing over big iron beams installed under the floor. One typical behavior is depicted in Figure 2.8. In this test, SAETTA traveled two straight paths interleaved by a 90 degrees rotation. During the self rotation, the field direction was constant and the angle was correctly measured (the sensor returned 88.8 degrees); on the contrary, during the straight paths, the reading exhibited large errors because they passed over a beam that modified the magnetic field. Obviously, any large ferromagnetic mass induces the same problems. When no disturbing mass is present, the compass is highly reliable as can be seen by Figure 2.14. In this graph the magnetometer measures are compared with the odometry ones that can be considered reliable due to the very slow velocity imposed to actuators. The maximum error in this experiment was 7^o.

Thus, we can have some indications on the use of the two kinds of sensor. When self rotating, compass is reliable for rotational measures but cannot provide an indication of the absolute position of the robot wrt the North direction. Absolute angle can be

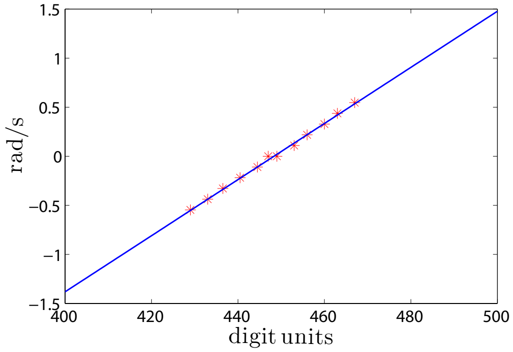


Figure 2.6: Calibration curve for the gyroscope sensor: red asterisks represent the measured data, the blue line is the calibration curve achieved by the least square estimation. Robot rotation speed is on the y axis (radiant/sec), digit units on the x one.

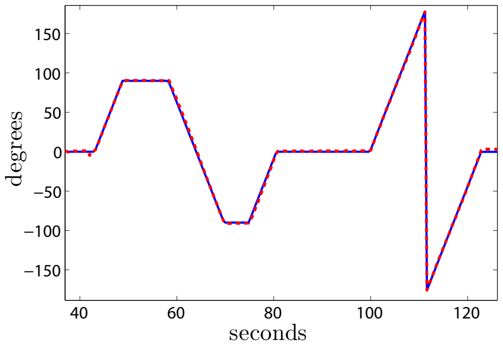


Figure 2.7: Magnetometer measures compared with the bearing achieved by odometry: continuous line (blue) is the odometry of the robot, dotted line(red) is the magnetometer reading. The signals are expressed in degrees units over a temporal scale (seconds). The maximum displacement is about 7 degrees.

relied upon only if there is an a priori knowledge about the presence and the position of disturbing masses, or after a long straight path during which the compass readings are constant. Moreover, the compass can be used to go back on a path already followed, as in each point the error is the same in the two ways. For paths with a sufficiently fast velocity and a large radius of curvature, it is safer to rely on the integration of the gyro readings if the bias can be estimated in some way (for example during stop phases).

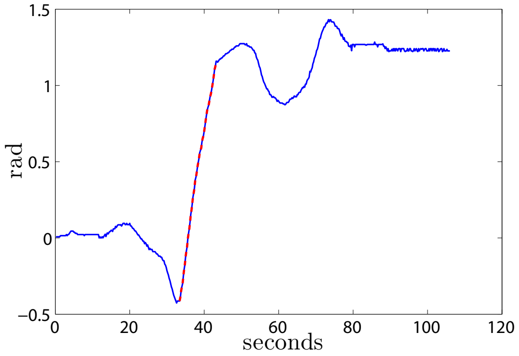


Figure 2.8: Curve acquired by the magnetometer: continuous line (blue) is relative to straight motion affected by iron beams, the dotted (red) one is relative to a self rotation of $\pi/2$. Signals are represented in degrees units over a temporal scale (seconds).

2.6 High-level tier

The high level tier is implemented on a Linux embedded board, the NETUS board from ACME SYSTEMS srl. It has an AT91SAM9G20 Atmel processor at 400MHz. The board comes with 64MB RAM and 8MB FLASH. It sinks 60mA from a $5V$ source and offers 5 serial, 2 USB and 1 ethernet ports plus several digital input/output lines. It carries an expansion board hosting a ZigBee module for communications in an unstructured environment. Linux offers the advantages of a solid open source Operating System that can be tailored on the user needs and gives the possibility to easily access the low level resources. The availability of many widespread interfaces as the USB and the RS232 in a standard OS opens the system to the use of an almost unlimited number of peripherals. At now we have connected webcams for vision and inter-robot distance estimation, standard Wi-Fi keys, an RFID tag reader for navigation in partially structured environments and a small laser scanner. With a careful programming, even rather complex vision algorithms can be implemented and, at the moment, the actual bottleneck seems to be in the batteries, since all these expansions sink a lot of current.

2.6.1 ZigBee communications

In a multirobot scenario, the ZigBee wireless communication channel is very important for the system architecture. In particular, the ZigBee module from Maxstream has been used. This module can communicate with a baud rate up to $250000bps$ within a range up to $20m$ in an indoor environment and up to 100m in outdoors without an infrastructure [1]. The protocol enables the robots to interact also with ZigBee compliant sensor networks. The use of this chip makes it possible to embed SAETTA in etherogeneous systems including both sophisticated devices as well as tiny platforms (see for example (17)) .

Differently from Wi-Fi devices, ZigBee ones have low power consumption: supplied at 3.3V, our module sinks 55mA and 270mA respectively in RX and TX modes and less than 1mA when idle. On the contrary, they show a smaller bandwidth that is anyway more than sufficient for most cooperation purposes.

Furthermore, several options using the ZigBee API are available. It is worth to underline the main features related to the typical multirobot scenarios. The module is able to work in different modes, which can be switched at run time: sleep, receive, transmit, idle and command mode. The last one allows the master device to set configuration parameters of the chip. Furthermore, it is possible to establish either a connection with implicit management of the acknowledgement (similar to TCP/IP) or without it (similar to UDP) depending on the specific application. Due to the cost of transmitting a packet, it is possible to send data in a broadcast mode or to a specific subset of other nodes (specifying the Personal Area Network ID to which the nodes of interest belong). These options (that are completely managed by the module without further computational load for the processor) find a direct correspondence in multi robot applications where the communication graph plays more ((18)) or less ((19)) importance, requiring a strictly connected net or a sporadic exchange of information without acknowledgements.

When an interaction both with sophisticated hardware and with a sensor network is required in the same experiment, it is possible to use at the same time both the ZigBee (802.15.4 standard) and the standard Wi-Fi channels (802.11 standard).

2.6.2 Software

Relying on a Linux OS, the software architecture has been developed in a more sophisticated manner than the PIC code.

[1] www.sparkfun.com/datasheets/Wireless/Zigbee/ZigBee-Manual.pdf

In literature several frameworks have been proposed. In (20), the Player project is introduced. It is a multi robot framework where each robot can be seen like a node of the network. Even in the internal architecture of each agent, devices and algorithms can be implemented as nodes connected each other: this approach is very effective in terms of abstraction but has the inconvenient in the overhead of communications. In (21) an interesting approach to domestic robotic networks is described: in this framework, every piece of data, referred as ”a tuple”, is ”published” in a distributed and connected network. Also in this case the overhead introduced into the communication was not appropriate for tiny devices. To overcome this issue, an optimized version running even on very limited platforms is described in (22). Another interesting approach is given in the Multi-Robot Integrated Platform(MIP) [1]: this framework is an object oriented environment which divides algorithms, tasks and devices into categories. Within each category, a component can be easily substituted by another one, resulting in short time-to-develop new experiments. Furthermore the algorithms can be executed both on real platforms (Khepera III) and in a simulated environment based on Player: the swapping between the two environments is easy as the change of one flag. Although very flexible, the use of OOP programming introduces an overhead which highs the computational burden, especially for embedded platforms like the one used on SAETTA.

In our case, we have chosen to develop a framework which encompasses both the overhead about sophisticated communications and OOP programming. A particular care about the timing performances was taken at the price of losing the abstraction level previously described.

The developing system is based on the C language. Our board mounts a standard Ubuntu 9.10 version, thus we could take all the advantages of the standard programming techniques. Currently, we chose a multi-threading approach to separate the kind of activities to be executed.

The critical ones are put in the main thread. This thread has a cyclic structure whose period is 200ms. It contains the routines to communicate with the PIC, motion commands and all the subtasks which concern about the safety of the robot. The control period is triggered by an interrupt deriving from the interface with the PIC (see Section 2.7) which starts the communication exchange between the two CPUs.

[1] http://www.dis.uniroma1.it/ labrob/software/MIP/

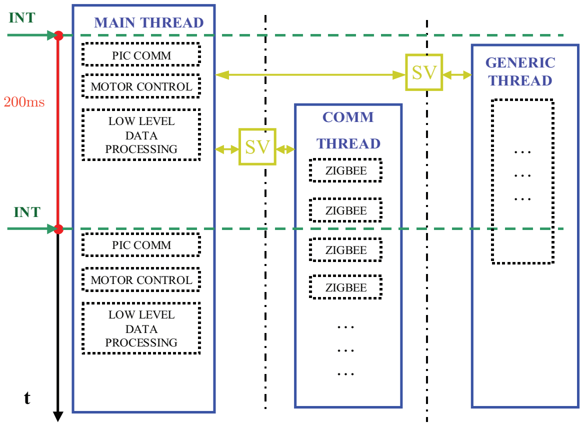


Figure 2.9: High level program structure: on the left the vertical time line is depicted. The first block, the main thread, is constituted by the periodic execution of subtasks: the communication with the PIC (PIC COMM block), motor control, and low level data processing. It can be seen that the periodicity of this thread is triggered by the interrupt INT which raises every 200ms. Conceptually, other parallel periodic threads can be run at the same time: they can be executed by subtasks whose execution can be shorter than the control cycle (e.g. COMM THREAD) or longer (GENERIC THREAD). The communication between threads is performed by shared variables SV.

This thread can be flanked by other ones which contain auxiliary activities: currently one is used for wireless communication (ZigBee or Wi-Fi), another for an RFID reader and so on. Typically each thread has a cycle which depends upon the activity it has to do: if a precise synchronization is needed, semaphores can be adopted. Communication between threads is implemented by the use of shared variables (see Fig. 2.9).

It is worth to note that whenever a thread doesn't have to perform any operation, it is put in idle mode, i.e. it is not time consuming, by the OS (23).

2.7 Communication between the two tiers

The interaction between the two tiers, i.e. between the NETUS and the PIC boards, is performed under the $RS232$ communication protocol. Denote with T_{NETUS} and T_{PIC} respectively the NETUS and the PIC control period cycle. The exchange of information is made at the beginning of T_{NETUS} (see Figure 2.10). The *starting message*, sent by the PIC, raises an interrupt on the NETUS processor enabling the main thread to

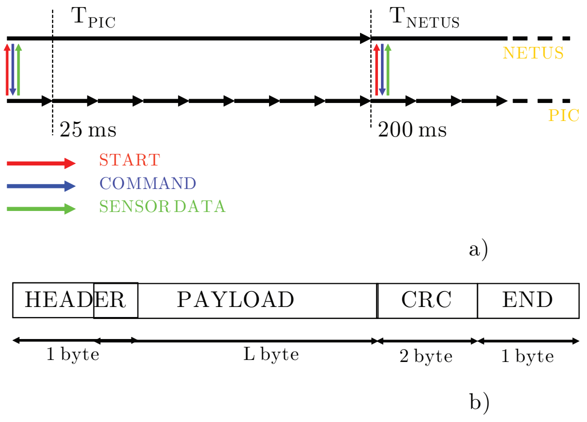


Figure 2.10: System timing. a) Communication process between the two controllers. The PIC sends a starting message (red arrow), the NETUS replies with a command (blue) and finally the PIC send processed data (green). b) Packet structures: header (1 byte ID), payload (varying length), a crc code (2 bytes) and an end character (END). The end character is unique as it cannot compare in the other fields: a masking technique is used to avoid this condition.

run. The NETUS board answers with a packet containing commands computed by the previous cycle. Finally, after receiving the command packet, the PIC board sends the data collected and processed over the 8 previous sub-periods. Each message, in both directions, is composed by the following fields:

- header: it is an ID that identifies the kind of packet
- payload: it is the information owned by the packet
- crc: it is a verification code
- end: it identifies the end of the packet

The commands that the upper tier can send are: system state conditions, reference velocities or displacements and stop commands. The lower tier sends data about odometry, sensor readings and acknowledgments about correct command execution.

2.8 Basic Testing

In this section two simple experiments are described in order to show the performances of the platform. In each test the robot was asked to execute some well known task in robotics. The first experiment is devoted to characterize the interaction between multiple units: robots have to meet in a common point using the *consensus algorithm* (18). In this scenario both communication and odometry performances were tested. The second experiment is related to infrared sensors: one robot explores a small area in order to build a map of the surrounding environment. In Section 2.9 a more detailed experiment about localization will be provided.

2.8.1 Consensus algorithm

In this scenario a set of 3 robots was used. The test was made to evaluate the performances of the system about motion precision in presence of heavy inter-robot communications.

Each agent, having a prior knowledge of its own state with respect to a common frame, broadcasted repeatedly its position to the others. Applying the consensus filter, the robots would meet to the mean of the initial positions if the dynamics of the robots were linear. Due to the nonholonomic constraints, the input generated by this control law had to be filtered by a Cartesian controller (24). As depicted in Fig. 2.11, this controller overcomes the drawback: the trajectories of the robots converge to a common point, which obviously is not the mean of the initial states. In detail, in this test, each robot traveled meanly for $84.3cm$: the maximum linear speed was set to 8 cm/s while the rotational one was 0.98 rad/s. Each robot broadcasted its state with a frequency of 10 Hz (double with respect to the control cycle to prevent data loss): on average each robot sent 137 packets. In tab. 2.2 it can be seen that the number of lost packets is negligible as well as the computational load: the algorithm took about 3ms per iteration and could be furthermore reduced avoiding the use of trigonometric functions with lookup tables and integer implementations.

Similarly, the odometry furnishes precise data. Robots arrived to a common point without having feedback about their positions. As mentioned before, several tests revealed errors smaller than one percent over long straight paths (about 12m) and contained angular displacements as well.

2.8.2 Mapping

In this test a robot explored a small area to build a map by the IR sensors. In particular a closed tunnel with a length of 130cm (Figure 2.12) has been considered. The

Table 2.2: Consensus experiment data

	Robot 1	Robot 2	Robot 3
Path (cm)	59	98	96
max speed (cm/s)	7.54	7.36	7.25
mean speed (cm/s)	3.15	3.85	3.99
Packets sent*	129	152	130
Total data loss	3.5%	2.9%	2.1 %
Max data loss with respect to one target	3.7%	3.1%	2.4 %

* different numbers of packets sent are due to different power-on time of the robots: the broadcasting is related to the power-on time while the consensus algorithm starts in a synchronous manner with an external starting command.

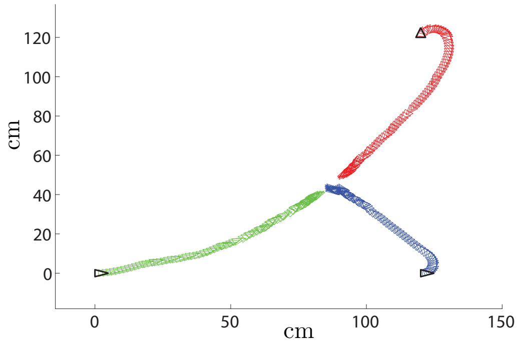


Figure 2.11: Consensus algorithm for 3 agents: black triangles represent the robot initial positions. The axes units are centimeters.

dimensions were chosen in order to effectively exploit the IR sensors.

In the test the robot made a smooth trajectory: along a straight path it stopped for two times to span the surrounding environment. The results of this experiment can be seen in Figure 2.13. The maximum linear speed was likely 6cm/s with a mean speed, without taking into account the resting, of 2.79cm/s while the maximum rotational speed was 0.3 rad/s.

The mapping results are good (see Fig.2.13): the sides of the open square are estimated with an error less than 5cm without using any sophisticated data processing. When more demanding mapping is needed, more performing sensors, e.g. laser range finders, can be mounted on SAETTA.

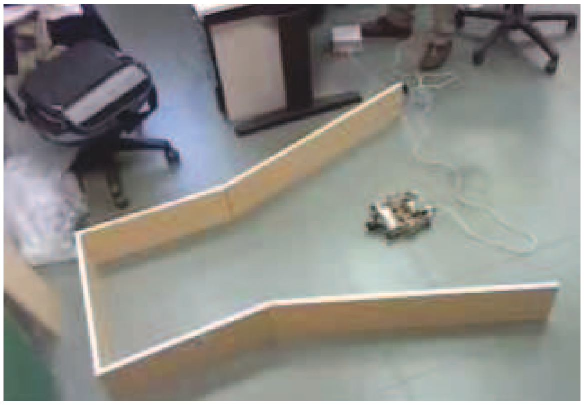


Figure 2.12: Test bed for the map building: the sides of the square are 60cm long; the two slanted sides are 1m long.

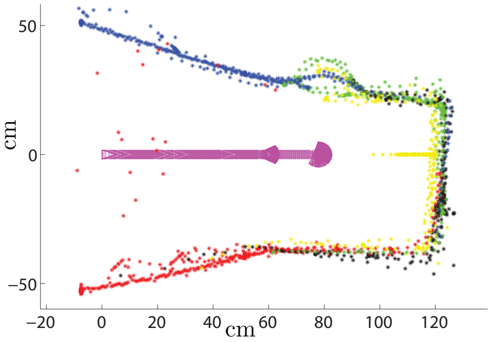


Figure 2.13: Map built by IR sensors. the robot trajectory is represented in violet, asterisks show the sensor acquisitions. Each color identifies an IR sensor: the front side ones are blue and red dots, front central one is yellow and the rear lateral ones are green and black.

2.9 A modified particle filter for low performance robotic platforms

In this section a localization algorithm, for small platforms like SAETTA, is presented. These experiments were conducted with a previous version of the NETUS board, the FOX one mounting a chip of 100MHz, making the scenario more demanding. Due to the limited processing capabilities, some ad hoc solutions have been used: the lack of processing resources has been compensated by an efficient implementation of the estimator and by the use of compass measures which ease the computational load. The results show how a careful design allows the implementation of sophisticated algorithms even on small platforms.

2.9.1 Problem setting

The localization problem can be approached using a probabilistic framework. It basically consists in the state estimation of a system whose temporal evolution can be modeled as:

$$q_k = f(q_{k-1}, u_{k-1}, \nu_k) \tag{2.1}$$

where $k \in \mathbb{N}$, $f(\cdot, \cdot)$ is a generic nonlinear function, $q_k \in \Omega$ represents the state of the system at the $k - th$ time step, u_k represents the control input, $\nu_k \in \mathcal{N}(0, \sigma_\nu^2)$ is a zero mean Gaussian noise with variance σ_ν^2 and Ω is the set of all the states x. To improve the estimation process, an observation system is used to gather measures that are function of the state x. It can be modeled as:

$$z_k = h(q_k, \lambda_k) \tag{2.2}$$

where $h(\cdot, \cdot)$ is a nonlinear function and $\lambda_k \in \mathcal{N}(0, \sigma_\lambda)$ is a zero mean Gaussian noise with variance σ_λ. It is assumed that the sequence of states q_k can be modeled as a first order Markov process, e.g.

$$p(q_k|Q_{k-1}, U_{k-1}) = p(q_k|q_{k-1}, u_{k-1}) \tag{2.3}$$

and also that the observations z_k are conditionally independent, hence:

$$p(z_k|Q_k) = p(z_k|q_k) \tag{2.4}$$

where Q_k represents the sequence of states up to the instant k. Similarly, denote with Z_k and U_k respectively the set of measurements and control inputs up to the generic instant k.

Under the previous assumptions, a prior and posterior belief about the state estimation can be introduced. The first represents the degree of belief about a certain state based on the knowledge of the previous estimates, the control input, that is assumed to be perfectly known, and the observations, all of them up to instant $k - 1$:

$$bel^-(q_k) = p(q_k|q_{k-1}, u_{k-1}, Z_{k-1}) \tag{2.5}$$

the latter instead represents the degree of belief after acquiring a new measure z_k; due to the assumptions described by eq. 2.3 and eq. 2.4 and using the Bayes rule, the posterior belief can be defined as:

$$bel^+(q_k) = \frac{p(z_k|q_k)bel^-(q_k)}{p(z_k|U_{k-1},Z_{k-1})} \tag{2.6}$$

where $p(z_k|q_k)$ is the perceptual model. In this way, the recursive estimation can be performed by defining the prior belief as:

$$bel^-(q_k) = \int_\Omega p(q_k|q_{k-1},u_{k-1}) \times bel^+(q_{k-1})dq_{k-1} \tag{2.7}$$

Particle filter adopts the aforementioned recursive model by a sampling approach. The posterior belief function is approximated by a set of N samples extracted by a normalized importance sampling distribution $\pi(q_k|u_{k-1},z_k)$:

$$bel^+(q_k) \approx \frac{1}{N}\sum_{i=1}^N w_i \cdot \delta(x - q_i) \tag{2.8}$$

where $\delta(\cdot)$ represent the Dirac pulse, and $q_i, i \in 1,\ldots,N$ is a sample extracted by the distribution π which, in the robotics field, is commonly represented by the prior distribution $bel^-(q_k)$ ((25)). Under this assumption, using the Bayes rule, the weight assigned to each particle is:

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot \frac{p(z_k|q_k^{(i)})bel^-(q_k)}{\pi(q_k|u_{k-1},z_k)} = w_{k-1}^{(i)}p(z_k|q_k^{(i)}) \tag{2.9}$$

Furthermore the prior belief can be achieved in a recursive fashion as:

$$bel^-(q_k) = p(q_k|q_{k-1},u_{k-1}) \cdot bel^+(q_{k-1}) \tag{2.10}$$

This recursive procedure has to be supported by a re-sampling technique. As new data become available, most of the particles have a negligible weight. These ones, with a re-sampling procedure, can be substituted with particles which fit better the new measures. To determine when a re-sampling iteration has to be done, the following performance index is commonly used:

$$T_k = \frac{1}{\sum_{i=1}^N (w_k^{*i})^2} \tag{2.11}$$

where w_k^{*i} is the normalized weight:

$$w_k^{*i} = \frac{w_k^i}{\sum_{i=1}^N w_k^i} \tag{2.12}$$

Whenever this value is less than a constant T, a re-sampling process is started. This consists in choosing a new set of particles according to the distribution $p(q_k|Z_k)$:

$$p(q_k|Z_k) \approx \sum_{i=1}^{N} w_k^i \delta(q_k - q_k^i) \tag{2.13}$$

in other words, the new particles are sampled with a major probability in the neighborhood of the most promising particles of the previous iteration.

2.9.2 Algorithm implementation

In this section the implementation of the algorithm on the SAETTA robot is discussed. In its complete formulation, for each particle the algorithm computes the intersections between the five beams originating by the infrared sensors and all the walls and then selects the feasible (shortest distance) intersection to compute $p(z_k|q_k)$. These computations involve multiple rototranslations, each of which makes use of trigonometric function. This kind of operation represent the main bottleneck in the algorithm execution, thus the attention will be focused just on this part.

Denote with $\mathcal{F}_0$ the frame origin and omit the temporal index k. Let the generic pose of the particle with respect to $\mathcal{F}_0$ be $q_i = \{p_i, \theta_i\}$, with $p_i = (p_i^x \; p_i^y)^T \in \mathbb{R}^2$ and $\theta_i \in \mathcal{S}$. Furthermore denote with $\mathcal{F}_b$ a frame related to a generic sensor $b = (p_b \; \theta_b)^T \in \mathbb{R}^2 \times \mathcal{S}$, centered at p_b and oriented like θ_b with respect to $\mathcal{F}_0$. Similarly, let ${}^b z = ({}^b p_z \; {}^b \theta_z)^T$ the representation of the pose $z \in (\mathbb{R}^2 \times \mathcal{S})$ with respect to $\mathcal{F}_b$. It is assumed that each particle i is equipped with M sensors (in our case $M = 5$), each of which having a pose $s_{ji} = (q_{ji} \; \beta_{ji})^T$,$j = 1, \ldots, M$, with respect to $\mathcal{F}_0$.

Consider a generic particle q_i and define a fitness function $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, for s_{ji}, as the Gaussian:

$$g(d_j, \hat{d}_{ji}) = \frac{1}{\sqrt{2\pi\sigma_s^2}} e^{\frac{(d_j - \hat{d}_{ji})^2}{2\sigma_s^2}} \tag{2.14}$$

where d_j is the measure actually acquired by the j-th sensor, $\hat{d}_{ji}$ is the measure the sensor would acquire if the robot were in the particle pose q_i, and σ_s is a fitness function parameter.

The most of computation relies on the determination of $\hat{d}_{ji}$. Consider a linear obstacle $\mathcal{O}$ delimited by the endpoints $\xi_1 = (\xi_{1x} \; \xi_{1y})^T$ and $\xi_2 = (\xi_{2x} \; \xi_{2y})^T$, with $\xi_1, \xi_2 \in \mathbb{R}^2$. A way to calculate the range $\hat{d}_{ji}$ between the sensor and the obstacle is to represent, by means of a rototranslation, the endpoints with respect to $\mathcal{F}_{s_{ji}}$:

$$^{s_{ji}}\xi_o = R(-(\beta_{ji} + \theta_{ji}))(\xi_o - q_{ji}),\ o = 1, 2 \tag{2.15}$$

where $R(-(\beta_{ji} + \theta_{ji}))$ is the rotation matrix along the axis orthogonal to the $\mathbb{R}^2$ plane. The computation of the distance is performed whenever at least one of the x coordinate is positive and the y coordinate of the two points are opposite each other. If these two conditions are satisfied, the distance between the sensor ji and the obstacle $\mathcal{O}$ is:

$$\hat{d}_{ji}|_{\mathcal{O}} = -^{s_{ji}}\xi_{1y}\tfrac{^{s_{ji}}\xi_{1x} - ^{s_{ji}}\xi_{2x}}{^{s_{ji}}\xi_{1y} - ^{s_{ji}}\xi_{2y}} +^{s_{ji}} \xi_{1x} \tag{2.16}$$

Consider the environment $\mathcal{E} = \bigcup_{i=1}^{P} \mathcal{O}_i$ like the union of the all P linear obstacles present into the environment, $P \in \mathbb{N}$. To determine the expected reading of a generic sensor, its minimum distance from the obstacles in $\mathcal{E}$ must be considered;

$$\hat{d}_{ji} = \min_p \quad \hat{d}_{ji}|_{\mathcal{O}=\mathcal{O}_p} \tag{2.17}$$

where the term $\hat{d}_{ji}|_{\mathcal{O}=\mathcal{O}_p}$ denote the distance between sensor pose s_{ji} and the obstacle $\mathcal{O}_p$. The complexity for the determination of the distances for the whole population of samples is $O(N \cdot M \cdot P)$. This requires a lot of computations and constitutes the worst bottleneck of the implementation. Each iteration of the aforementioned algorithm, using floating point numbers, takes about 2.6s considering a population of 20 particles. In the optimized implementation, shown in the next section, the same iteration takes about 40ms.

2.9.2.1 Optimized algorithm

The efficiency improvements are based on three actions. The first one consists in using the compass readings to impose the particle bearings. The feasibility of this action is related to the precision shown by the compass, at least in our experiments. In this case the computing time for $p(z_k|q_k)$ is only 1.5 seconds, which is about one half of the original one. Note that the particle number (equal to 20) has been kept the same. Therefore it is used to span a 2-dimension space instead of a 3-dimensional one thus improving the particle "density" which in turn provides a smaller error and a better convergence of the algorithm as shown by the reduced number of effected re-samplings. Actually, to get the same properties of the original algorithm, this number could be reduced with a consequent improvement of the iteration speed.

Furthermore, because the execution time is still unsuitable for real time use, an integer representation has been adopted (26) for the numeric variables and a lookup table T_{cos} has been used for the cosine operator. These actions make much faster the rototranslation of walls. In fact, every wall is rotated by the same angle given by the sum of the compass reading and the sensor relative angular position. It is worth of note that the same approach is applied to shorten the computation of s_{ji} by rotating the vector position of the sensor wrt the robot center of the angle.

In particular, being our scenario confined in few meters, the 32 bit length which is native of our processor for integer variables makes it negligible the loss of precision due to the quantization (27). As for the angles, the range from 0 to $\pi/2$ has been discretized into 128 levels. The correspondent cosine range, $[1, \ldots, 0]$, has been represented by the interval $[1023, 0]$. In this way a generic angle α is represented by an index l s.t. $T_{cos}[l]/1023 \approx cos(\alpha)$.

This representation (26) is sufficient to represent the cosine of every angle into the range $[-\pi, \pi]$. Similarly, the sine representation reduces by a proper shifting in the table. The same approach, obtaining the table $T_{fitness}$, has been used for the computation of the fitness function in Eq. 2.14. Also the range of values which the fitness function can assume has a maximum which is a power of two, thus reducing the normalization operator to a simple shift. It has to be stressed that this approach, making use of analytical representation for the environment, can be easily adapted to gridmaps. The schematization of the algorithm can be viewed in Alg. 1.

2.9.3 Experimental results

In this section some experiments are presented to show the performance achieved by SAETTA. The test bed is constituted by a rectangular arena which presents a recess (see fig. 2.15). Several tests have been conducted: here we describe one of the most significant. The ground truth about the Cartesian position of the robot has been logged by an external webcam calibrated by the Matlab Vision Toolbox. In this experiment the robot has traveled for about five minutes in the arena. Due to the stochastic nature of the particle filter, beyond the onboard real time implementation described before, the data collected by the robot were stored in a PC to perform also a statistical analysis of the behavior wrt different initial conditions randomly chosen to start the population of the particle filter. The comparison was performed between a standard particle filter

Algorithm 1: Fitness value for the particles population

```
Data: $d_j, q_i, s_{ji}$ , $j = 1, \ldots M, i = 1, \ldots, N$
let $\hat{\theta}$ be the compass measure
Result: $fitness\ values\ \omega_i$, i=1, ..., N

for each sensor j=1, ..., M do
    get $\cos(-(\hat{\theta} + \beta_{ji}))$ and $\sin(-(\hat{\theta} + \beta_{ji}))$ from $T_{cos}[l]$
    for each particle $i = 1, \ldots, N$ do
        $\hat{d}_{ji} \leftarrow \infty$
        for each $\mathcal{O}_k \in \mathcal{E}, k = 1, \ldots, P$ do
            ${}^{s_{ji}}\xi_o = R(-(\hat{\theta} + \beta_{ji}))(\xi_o^k - p_{ji}), o = 1, 2$, making use of $T_{\cos}[l]$
            if ( (${}^{s_{ji}}\xi_{1x} > 0$ or ${}^{s_{ji}}\xi_{2x} > 0$) and ($sign({}^{s_{ji}}\xi_{1y}) \neq sign({}^{s_{ji}}\xi_{2y})$) ) then
                compute $d_{aux}$ according to eq. 2.16
                if $d_{aux} < \hat{d}_{ji}$ then
                    $\hat{d}_{ji} \leftarrow d_{aux}$
                end
            end
        end
        $\omega_{ji} = g(d_j, \hat{d}_{ji})$ making use of $T_{fitness}(\lfloor d_j - \hat{d}_{ji} \rfloor)$
    end
end
for each particle $i = 1, \ldots, N$ do
    $w_i = \frac{1}{M} \sum_{j=1}^{M} w_{ji}$
end
```

(PF) and the fixed bearing algorithm (FBA). Due to the reliability of the compass in our test-bed and the reduced dimensionality of the problem, the convergence of the FBA is faster and the errors are smaller than the ones achieved by the PF. Table 2.3 has been obtained running the algorithm on the same data set for 500 iterations (approximately 100 seconds) and starting from 50 different initial conditions. The error is relative to the weighted mean of the best five particles, in terms of importance weight, over a total population of 20. The FBA shows better performance than PF, both with respect to the error statistics and to the number of re-samplings. Running the standard algorithm with a bigger population, e.g. 50 particles, the results are, as expected, slightly better

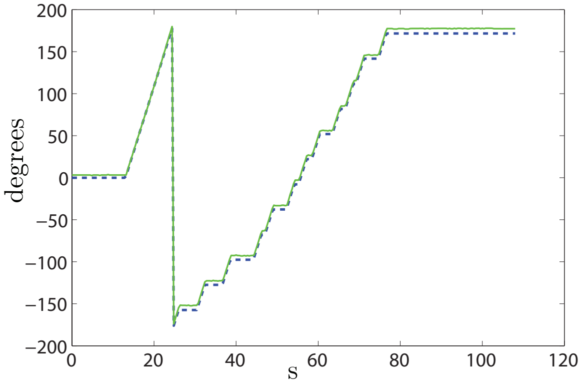


Figure 2.14: Magnetometer data compared with bearing obtained by odometry : solid line is the odometry of the robot, dashed line represents magnetometer readings. The units for y axis are degrees. The maximum displacement is about 10 degrees.). On x axis the time, in seconds, is reported.

but with a computation time almost twice as high. Some data concerning about time execution of the FBA algorithm with respect to the particle population are reported in Table 2.6).

In Table 2.4, the results of the whole navigation are showed: also in this case the FBA performs better with respect to the PF. The error trajectory for one trial is described in Fig. 2.16: the curves represent the error obtained considering respectively the weighted mean of the 5 best particles at each iteration and the mean of the total population. The correspondent representation of the first curve is represented in fig. 2.15. The red dots represent the mentioned weighted average; the isolated ones represent initial estimates and correspond to the first part of the curve. After 400 iterations, when the error starts to decrease, the best hypotheses track the robot trajectory. This situation is represented by the dense set of hypotheses lying on the neighborhood of the robot trajectory. Finally, a kidnapping test is showed. In this case, the data set has been dropped from iterations 200 to 900. As detailed in Table 2.5, results show how the FBA algorithm performs after about 500 iterations. In this section, we did not consider the Extended Kalman Filter because, although effective, this estimator needs to know the initial location of the robot within a certain precision. In our case, instead, we run our experiments without having a prior knowledge about the initial state of the robot.

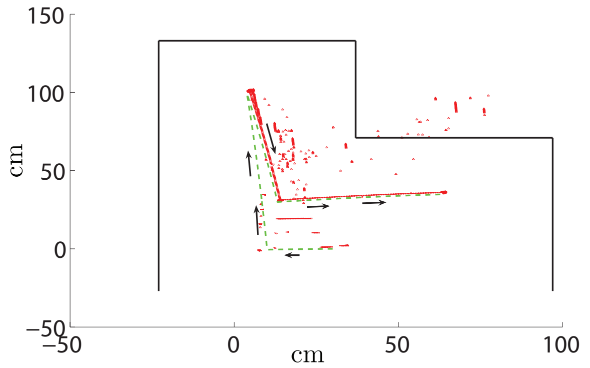


Figure 2.15: Algorithm evolution: the real robot trajectory is the dashed line; the hypotheses related to the estimate of the FBA are red dots. Arrows indicate the moving direction of the robot along the trajectory. After some iterations (clustered isolated hypothesis) during which the real pose is wrongly estimated, the algorithm correctly tracks the robot. Both axes coordinates are expressed in cm

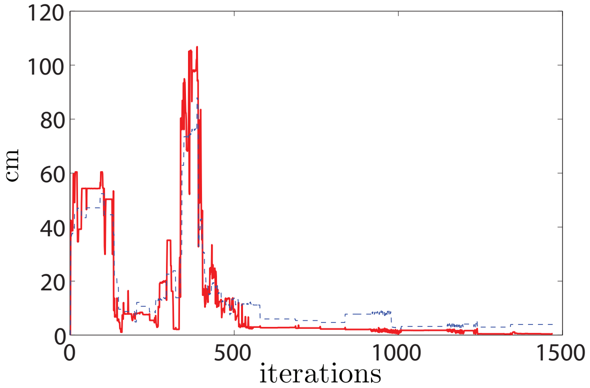


Figure 2.16: Error convergence: dashed line represents the hypotheses obtained averaging the whole population; solid line represents the hypotheses obtained averaging the best 5 particles. Units on the y axis are in cm, number of iterations are reported on x axis.

Table 2.3: Tests for 500 iterations

Alg.	e_{mean} (cm)	σ^2_{mean} (cm^2)	# of resamplings
PF	48.25	1478	90
FBA	5.89	8.24	35

2.10 Conclusions

In this work, a new small mobile platform has been presented. The main characteristics of the robot are the contained costs and the flexibility. The latter one has been showed both through hardware and software architecture which adapt easily to different contexts, as stated by several examples.

Currently the core feature, hence the base software and the management of important navigation sensors, has been developed. Future works will focus on the implementation of a software framework based on a very structured scheme: it is useful to structure the base features of the robot upon which the various high level tasks will be executed.

Other works will be done in the research algorithms direction: many open problems related to the multi robot fields could be faced by this architecture. Furthermore another very interesting aspect is the interaction between these platforms and other systems, e.g. static sensor networks or human beings.

Table 2.4: Tests for 1400 iterations.

Alg.	e_{mean} (cm)	σ^2_{mean} (cm^2)	# of resamplings
PF	9.86	111.0903	137
FBA	5.7831	13.8332	46

Table 2.5: Results obtained in an experiment where kidnapping conditions are used.

e_{min} (cm)	σ^2_{min} (cm^2)	e_{mean}(cm)	σ^2_{mean} (cm^2)
6.1356	11.7727	6.84	11.83

Table 2.6: Computation time for one single iteration of the FBA with mean error algorithm.

# of particles	time (ms)	e_{mean} (cm)
5	13	21.956
10	25	13.75
15	34	9.69
20	45	5.78
30	57	5.23
50	83	4.98
100	139	4.34

Chapter 3

Formation control: the encirclement case

In this Chapter, the encirclement problem is analyzed. This problem consists in the design of control laws able to steer a set of robots around a target, which can be static or dynamic, at constant speed. Furthermore the agents have to be equally spaced over the circle they travel. In the further two techniques are proposed: both of them are experimentally validated through extensive tests[1].

3.1 Introduction

In this chapter a particular motion technique, the encirclement, is taken into account.

This task has gathered a big interest in the scientific community, mainly for two reasons: first it deals with a series of technological issues related to communications, motion estimation and control design; second, the encircling is a fundamental part in missions like entrapment, escorting or patrolling.

The design of control laws has to carefully take into account both the kind of communication and the quality of the available sensorial system.

The former is fundamental in order to characterize the communication graph, i.e. the flux of the information spreading among robots. In fact the swarm can be conceptually represented by a graph, whose nodes are the robots and the edges represent the communication channels.

[1] This is a joint work with the Robotics Lab, La Sapienza. The experiments were conducted with the Khepera III robots of the Robotics Lab, La Sapienza

The latter is important in order to determine the quality and the performances that can be achieved by the control laws.

Furthermore, another sophistication is represented by the specific kinematic of the vehicles.

In literature a large variety of works have been proposed. In the formation control, one of the first approaches was based on the potential fields. In (28) a study about a swarm of robot which has to traverse a space in which multiple obstacles raise is showed. In particular a potential-based technique is exploited and considerations about the ability of the swarm to keep cohesion are reported. In (29) a fleet is guided through the interaction between agents and virtual points which deform the shape of the formation to achieve some tasks and guarantee collision avoidance. In both previous works the communication between agents was considered unlimited. The main drawback in the use of potential fields is the coarse effectiveness in presence of local minima: when multi robot systems are considered, the mixture of potential fields can lead to stall situations.

While these works take into account holonomic vehicles, in (30) a nonholonomic constraint is introduced. A group of unicycles is steered at constant linear speed around a common point. In (31) an extension with communication limitations is considered: in this case the global information is propagated through local interactions. In both previous works a Lyapunov control approach is used. The uniform spacing along the circle needed by the encircling task is achieved using the so called (M, N) *patterns*, for which a controller based on phase potentials is proposed. This controller requires relative bearings that in real experiments are rather difficult to achieve. Furthermore the controller is nested in the sense that if the circle has to be divided in M sectors, moments of order $M - 1$ have to be calculated. Another issue is the constant linear speed in the control law: this constrain does not fits with situations, e.g. proximity to obstacles, where the agents can not keep constant the linear velocity.

In (32) a work strictly related to the previous one is presented. Although very effective, the relative bearing and the distance estimation is done by a centralized vision system mounted on the roof.

A fully centralized approach has been proposed in (33). A vision system provides the configuration of the robots to a centralized controller that computes the trajectories. This algorithm, derived from the control theory related to manipulators, computes the trajectories considering a set of required tasks. Each of them has a different priority and is projected on the null space of the previous one. This controller requires a good estimate of the pose of every robot with respect to a fixed frame in order to compute the Jacobian of the formation.

In (34) a control for nonholonomic agents with directional sensors using a Lyapunov approach based on the cartesian displacement between robots is presented . However the control law requires the previous knowledge of the number of agents, therefore possible faults of robots are not taken into account.

Some algorithms present control laws whose convergence is based on the topology of the communication graph which describes the information flow between the agents (35). In (36) a fixed topology algorithm, based on the concept of the α *stability* property, is discussed to drive holonomic robots into a regular formation. Under the assumption that the information graph is strongly connected, it can be demonstrated that agents reach a common estimate (consensus) of such global quantities (37). Extensions to consensus problem have been proposed in (38) and (39) where more robust second order filters allow propagation of time varying quantities.

In (40) an artificial intelligence approach to the *evader-pursuer* problem is used. Both pursuers and evaders use networks whose evolution is influenced by environmental measures. During the task execution, each agent has a particular role, possibly with different characteristics, with respect to the rest of the formation. In (41) a real experiment with an heterogeneous formation is performed. Using both aerial and ground vehicles, probabilistic algorithms to entrap an evader are performed. During the task execution, a grid map of the environment is used.

In (42) the effectiveness of a four-phase bio-inspired cooperative strategy for multiple pursuers to confine an evader in a bounded region is demonstrated. The authors assume that all the players have unlimited sensing capabilities and that both instantaneous position and velocity of the evader are available to all pursuers. These strict assumptions are removed in (43), in which an algorithm to solve the visibility-based pursuit-evasion problem is presented for a single agent and suggested for a multi robot case.

In all the previous works where an experimental framework is presented, the quantities required in order to execute the encircling task are gathered by a centralized system which results to be more precise with respect to a distributed one: this simplification hinders an effective analysis about the performances achieved by a specific technique.

Furthermore, a centralized architecture is usually not robust to faults and a structured environment is typically required. In a real context it is very important to design a *decentralized estimation system* that uses the measures directly gathered by the agents to estimate the quantities involved in the control law.

For the best of our knowledge, the work described in the following is innovative because it takes into account a more realistic and minimal model of the measured quantities, and achieve the encircling task in an more unstructured environment. Furthermore the controller of each unit is independent from the orientation of the other robots: relative bearings are very critic parameters to estimate and nonholonomic constraints are taken into account. Therefore, each robot has only to performs measures within its own current local frame: this also avoids the use of compass and/or gyros during the task execution.

The high degree of decentralization of the multi-robot system also guarantees a high robustness to single agent faults. The proposed control laws are experimentally validated by a set of real robots: each unit executes the encirclement after performing an estimate of the formation configuration by the Mutual Localization Algorithm (44). The cascade of the two tasks results to be effective also in case of faults.

In the following two approaches are proposed: the first relies on a planner/controller scheme, while the second tries to avoid the trajectory generation using directly a feedback law.

3.2 Problem setting

Consider a multi robot system constituted by a set of N robots $\mathcal{R} = \{\mathcal{A}_1, \ldots, \mathcal{A}_N\}$. Denote with $p_i = (p_i^x \, p_i^y)^T \in \mathbb{R}^2$ and $\theta \in S^1$ respectively the unknown position of a representative point and the unknown orientation of $\mathcal{A}_i$, both expressed in an fixed frame $\mathcal{F}$. Denote with $p_t = (p_t^x \, p_t^y)^T$ the unknown target position in $\mathcal{F}$. Each $\mathcal{A}_i$ is modeled as a unicycle, i.e.:

$$\dot{q}_i = (\dot{p}_i^T \, \dot{\theta}_i)^T = \begin{pmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_i \\ w_i \end{pmatrix} \tag{3.1}$$

where v_i and w_i are respectively the speed and the steering controls. Each $\mathcal{A}_i$ has an attached moving frame $\mathcal{F}_i$ with origin p_i and orientation θ_i. Denote with ${}^iq_j = ({}^ip_j^T \, {}^i\theta_j)^T$ the unknown configuration of $\mathcal{A}_j$ expressed in $\mathcal{F}_i$.

Furthermore suppose that each unit is equipped with a sensorial system, the *Robot detector*, able to extract some informations about other robots, and a communication system.

The *robot detector* is a sensor device that measures the relative position (not the orientation) of other robots and of the target w.r.t. the detector, provided that they fall in a perception set D_p that is rigidly attached to it. The shape of D_p is arbitrary, and in particular it may contain blind zones. The relative position measures provided by the robot detector are *anonymous*, i.e., they do not convey the specific identity of the detected robot (hence, the target is detected as a robot).

The communication module can send/receive data to/from any other robot contained in a communication set D_c such that $D_p \subseteq D_c$. For simplicity suppose that D_c is a circle of radius R_c.

Associated to $\mathcal{R}$, denote with $G = (V, E)$ the communication graph whose nodes $v_i \in V$ represent the robots, i.e. $|V| = |\mathcal{R}|$, and E is the set of edges representing a communication link between two robots. In the latter case a communication link e_{ij} exists if and only if the distance between two robots $\mathcal{A}_i$ and $\mathcal{A}_j$, represented by two nodes v_i and v_j, is less than R_c. This supposition implies a bidirectional communication, i.e. if $\mathcal{A}_i$ is able to communicate with $\mathcal{A}_j$ then also $\mathcal{A}_j$ is able to communicate with $\mathcal{A}_i$. The graph can be represented by the associated Laplacian matrix (37),

$$L = \begin{cases} -\sum_{k=1, k\neq i}^{N} e_{ik} & if \quad i = j \\ 1 & if \quad i \neq j \\ 0 & otherwise \end{cases} \tag{3.2}$$

with $L \in \mathbb{R}^{N \times N}$. This matrix has the following properties: it has at least one null eigenvalue, which is the greatest one, whose associated eigenvector is $1 = \{1, \ldots, 1\}^T \in \mathbb{R}^N$, i.e. it spans the subspace constituted by vectors having identical components.

These properties can be used effectively when a set of agents has to agree about some information, e.g. the inter-distance between the agents. In fact, considering a system having the following dynamics:

$$\dot{\eta}(t) = \frac{1}{N} L \eta(t) \tag{3.3}$$

with $\eta \in \mathbb{R}^{\mathbb{N}}$ and t denoting the time, the state η converges to a vector whose components are all equal to the mean of initial state components $\eta(0)$. This system will be useful in the description of the algorithms.

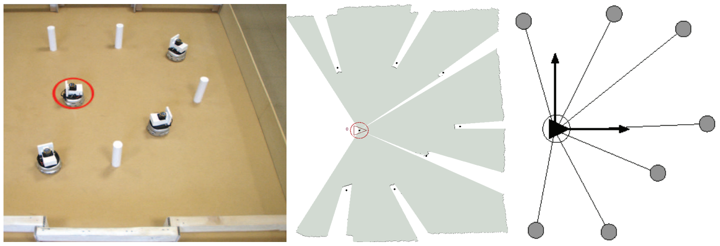


Figure 3.1: Sequence of the feature extraction process, from left to right: the robot with red circle acquires scan measures, the feature extraction extracts features which are compliant with the robot shape, the output consisting of the set of extracted features is furnished.

The encirclement problem is the problem of finding a control law $u_i(t), \forall i \in 1, \ldots, N$ such that every agent is able to rotate with constant velocity v_i onto a circle centered at a point p_t and having a radius R. Furthermore, the agents have to distribute themselves over the circle with a regular distribution, i.e. each agent has to be equally distant from the preceding and consequent neighbor.

3.3 Mutual Localization

Mutual Localization algorithm is a technique which allows to estimate in a distributed fashion the configuration, in metrical terms, of a team of mobile robots. In this chapter a particular algorithm has been used in order to validate the effectiveness of the controllers: the mutual localization algorithm with anonymous measures (45). This algorithm was chosen because it easily adapts to a real context. In fact, differently from previous works, it does not implies that measures gathered by one unit are related to the identity of the other agents. Initially each robot is only able to extract features which resemble to a robot shape (Figure3.1).

Upon this information and the one provided by the other robots (called observations), the algorithm onboard the robot furnishes an estimate of the formation configuration, i.e. the configuration of the team mates. This result is gathered by a two step process: the first consists of a multi registration algorithm, called *Multireg*, which takes in input the observations and furnishes a set of possible configurations. The output of this phase is then injected into an estimator (a multi Extended Kalman Filter or a particle one) which provides the best hypothesis. Because the detailed description is out of the purpose of this thesis, the reader is referred to (46), (47)

3.4 Orbital Controller

In this section a controller based on dynamic feedback linearization (48) is proposed. In view of the encirclement objective, it is convenient to express the configuration of the generic i-th robot in polar coordinates with respect to a reference frame centered at the target, as in Fig. 3.2. In particular, these coordinates are the distance ρ_i of the unicycle wheel center from the origin, the angle γ_i that the sagittal (forward) axis of the robot forms with the line joining the unicycle wheel center to the origin, and the angle ϕ_i between the same line and the x axis. In the following, ϕ_i is also called *phase* of the robot). The kinematic model of the unicycle is then written as (49)

$$\begin{aligned}\dot{\rho}_i &= -v_i \cos\gamma_i\\ \dot{\gamma}_i &= (\sin\gamma_i)/\rho_i - \omega_i\\ \dot{\phi}_i &= v_i(\sin\gamma_i)/\rho_i,\end{aligned}$$

where v_i and ω_i are respectively the driving and steering velocity inputs.

It is assumed that the robot index i refers to the cyclic counterclockwise ordering of the robots defined by their increasing phase angles (see Fig. 3.2). For the i-th robot, denote by $\bar{\phi}_i$ the mean between the phases of the successor (robot $i+1$) and the predecessor (robot $i-1$) of the robot. Correct execution of the encirclement task requires that

$$\lim_{t\to\infty}\rho_i(t) = R \qquad \lim_{t\to\infty}\phi_i(t) = \bar{\phi}_i(t) \qquad \lim_{t\to\infty}\dot{\phi}_i(t) = \Omega \quad \forall i, \tag{3.4}$$

where R and Ω are respectively the encirclement radius and angular speed, which must be the same for all robots.

The architecture of the system used for the algorithm execution is depicted in Fig. 3.5. The initial configuration $(\rho_i^0, \gamma_i^0, \phi_i^0)$, provided by the mutual localization module[1] is used to plan a reference trajectory for the robot. In particular, such trajectory is specified by an exosystem that assigns reference evolutions ρ_i^r, ϕ_i^r to the

[1] This assumes that the configuration estimate is immediately reliable. In practice, it may be necessary to perform a preliminary motion of the multi-robot system aimed at improving the accuracy of the estimate. To this end, the anti-symmetry control law proposed in (45) may be used.

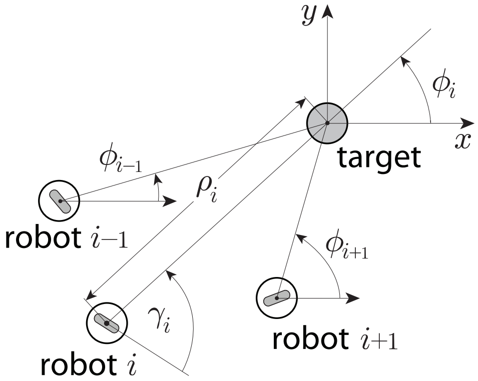


Figure 3.2: Polar coordinates for the i-th robot and the cyclic ordering defined by phases.

coordinates ρ_i, ϕ_i. In fact, it can be demonstrated that these two are *flat* outputs (50) for the unicycle in polar coordinates, i.e., once an evolution is assigned to them it is possible to compute algebraically the corresponding evolution γ_i^r of the remaining variable γ_i as well as the reference inputs v_i^r, ω_i^r. The reference outputs ρ_i^r, ϕ_i^r are fed to a feedback controller based on Dynamic Feedback Linearization (DFL), that generates the control inputs v_i, ω_i so as to guarantee global exponential tracking of the reference trajectory (48). It should be noted that ρ_i^r, ϕ_i^r are initialized at ρ_i^0, γ_i^0, so that the transient is extremely fast. During its operation, the DFL tracker uses the current estimate of the target-frame robot configuration $(\rho_i, \gamma_i, \phi_i)$ computed by the mutual localization module.

In the following, we consider three slightly different versions of the basic encirclement task entailed by (3.4), and give the appropriate form of the trajectory planner (exosystem). In all versions, the encirclement radius R is assigned in advance. The reference radius $\rho_i^r(t)$ is therefore always generated by

$$\dot{\rho}_i^r = K_\rho(R - \rho_i^r) \qquad \rho_i^r(0) = \rho_i^0, \tag{3.5}$$

where K_ρ is a positive gain. As a consequence, $\rho_i^r(t)$ exponentially converges to R for any initial condition. Note that $\rho_i^r(t)$ does not depend on the reference radius of any other robot.

The three versions of the encirclement task differ on the procedure used by the robots to agree on the common value of the angular speed Ω in (3.4). They are analyzed in detail below.

3.4.0.1 Encirclement – Version 1.

In the first version, the angular speed Ω is also specified in advance. The reference phase $\phi_i^r(t)$ for the i-th robot is generated by

$$\dot{\phi}_i^r = \Omega + K_\phi(\bar{\phi}_i^r - \phi_i^r) \qquad \phi_i^r(0) = \phi_i^0, \tag{3.6}$$

where K_ϕ is a positive gain and $\bar{\phi}_i^r$ is the mean between the reference phases of the predecessor and the successor (in accordance with the counterclockwise cyclical ordering of the reference phases). We have the following result (the proofs of all the propositions can be found in (51)).

Proposition 1 *The flow of (3.5),(3.6) yields exponential convergence of ρ_i^r to R, of ϕ_i^r to $\bar{\phi}_i^r$, and of $\dot{\phi}_i^r$ to Ω, for any assigned R, Ω and any initial ρ_i^0, ϕ_i^0.*

An example of reference robot trajectories corresponding to the flow of (3.5),(3.6) is shown in Fig. 3.3. The robots approach the circle in such a way that the 'insertion points' are almost uniformly spaced, and actually achieve the required formation very quickly.

3.4.0.2 Encirclement – Version 2.

In the second version, the robots are assigned an *escape window* s, i.e., the time interval in which a point on the circle remains unvisited at the steady state corresponding to the asymptotic conditions (3.4). Being $s = 2\pi/n\Omega$, where N is the number of robots, the robots can in principle easily compute the required value of Ω as $\Omega = 2\pi/ns$; however, since N is not known a priori, an estimate $\hat{n}$ of this number is required.

Assume that each robot instantaneously computes its own estimate as $\hat{n}_i = 2\pi/\Delta_i^r$, where $2\Delta_i^r$ is the reference phase difference between the successor and the predecessor. The required angular speed for the robot is then computed as $\Omega_i = 2\pi/\hat{n}_i s = \Delta_i^r/s$.

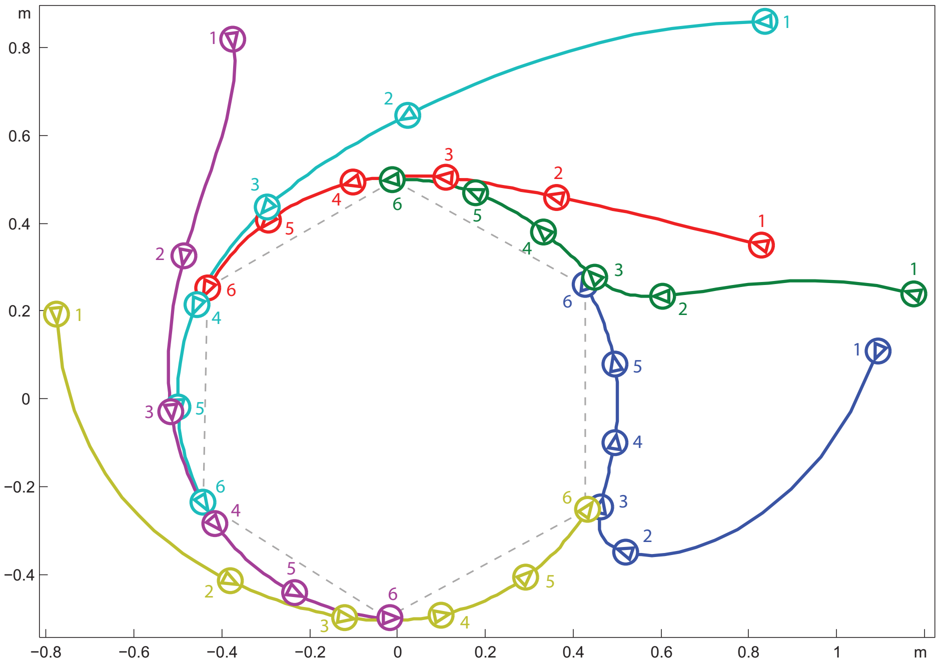


Figure 3.3: Reference trajectories corresponding to the flow of (3.5),(3.6) for a generic initial configuration of a 6-robot system, with a target located at the origin. The configuration of each reference robot along its trajectory is explicitly shown at six equispaced time instants, identified by $1, \ldots, 6$.

Using this expression for Ω in (3.6) we obtain the following exosystem for the reference phase:

$$\dot{\phi}_i^r = \Delta_i^r / s + K_\phi(\bar{\phi}_i^r - \phi_i^r) \qquad \phi_i^r(0) = \phi_i^0. \tag{3.7}$$

Proposition 2 *The flow of (3.5),(3.7) yields exponential convergence of ρ_i^r to R, of ϕ_i^r to $\bar{\phi}_i^r$, and of $\dot{\phi}_i^r$ to $2\pi/ns$, for any assigned R, s and any initial ρ_i^0, ϕ_i^0.*

3.4.0.3 Encirclement – Version 3.

In the third version, only the radius R is assigned, and the robots must autonomously agree on a common value of the angular speed Ω. The reference phase exosystem for the i-th robot is

$$\dot{\Omega}_i^r = K_\Omega(\bar{\phi}_i^r - \phi_i^r) \qquad \Omega_i^r(0) = 0 \tag{3.8}$$

$$\dot{\phi}_i^r = \Omega_i^r + K_\phi(\bar{\phi}_i^r - \phi_i^r) + \xi_i \qquad \phi_i^r(0) = \phi_i^0 \tag{3.9}$$

where ξ_i is a constant forcing term. Denote by $\bar{\xi}$ the average of the forcing terms ξ_i over the multi-robot system.

Proposition 3 *The flow of (3.5), (3.8–3.9) yields exponential convergence of ρ_i^r to R, of ϕ_i^r to $\bar{\phi}_i^r$, and of $\dot{\phi}_i^r$ to $\bar{\xi}$, for any assigned R and any initial ρ_i^0, ϕ_i^0.*

An interesting feature of this third scheme is that the common frequency of the phase reference trajectories can be regulated by acting on a single robot; to this end, it is sufficient to let $\xi_i = 0$ for all the robots but one.

To allow the implementation of (3.6), (3.7), or (3.8–3.9) all the robots must broadcast their current reference phase through the communication system. However, each robot computes its reference trajectory and control inputs autonomously on the basis of local information, i.e., its own configuration and data coming from the neighbors.

3.5 Tangential Controller

In this section the tangential controller, based on weaker assumptions about the information gathered by each agent with respect to the orbital one, is presented. In particular each robot is not asked to achieve the phase of other robots. Only inter-distances between units and polar coordinates of the target with respect to the frame of each robot are needed. This control has been demonstrated to asymptotically solve the encirclement problem only with agents starting outside the circle, while in the other case only practical effectiveness by experimental results is showed.

The proposed control law can be conceptually split into two orthogonal phases. The former steers the robot over the circle rounding at constant velocity v_d, while the latter allows a uniform distribution by the knowledge of local information.

For the moment, the attention will be focused only on the case where agents start outside the circle (see Fig. 3.4). Suppose, without loss of generality, that the target is centered at the origin. Thus it is possible to define the error vector as the following:

$$e_i = d_i^{ref}\ (\cos\theta_i^{\text{ref}}\ \sin\theta_i^{\text{ref}})^T \tag{3.10}$$

where:

$$d_i^{ref} = \sqrt{\rho_i^2 - R^2} \tag{3.11}$$

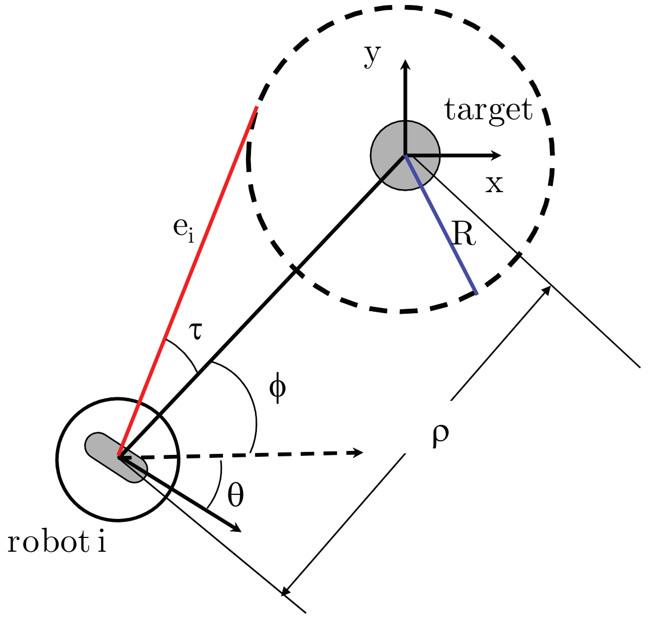


Figure 3.4: Representation of the problem setting related to the tangential controller.

$$\theta_i^{\mathrm{ref}} = \phi_i + \tau_i \tag{3.12}$$

with $\phi_i = -\arctan 2(y_i, x_i)$ and $\tau_i = \arctan\left(\frac{R}{d_i^{ref}}\right)$

The control that steers each agent rounding at constant linear speed v_d around p_t, i.e. asymptotically both $e_i \to 0$ and $\theta_i \to \theta_i^{ref}$, is:

$$v_i = K_v \cdot |e_i| + v_f \tag{3.13}$$

$$\omega_i = K_\omega(\theta_i^{\mathrm{ref}} - \theta_i) + \dot{\theta}_i^{\mathrm{ref}} \tag{3.14}$$

where $v_f > 0$ is a feedforward term and $\dot{\theta}_i^{ref}$ is a nonlinear compensation taking into account the nonlinearities due to the variation of θ_i^{ref}.

The meaning of v_f will be clarified into a while. For the moment consider the rotational dynamics. The rotational error e_θ is described by the following dynamics.

$$\dot{e}_\theta = \dot{\theta}_i^{ref} - \dot{\theta} = -Ke_\theta \tag{3.15}$$

Consider the dynamic of θ_i^{ref}:

$$\dot{\theta}_i^{ref} = \dot{\phi}_i - \dot{\tau}_i = \frac{v_i}{\rho}(\sin(\phi_i - \theta_i) + \tan\tau\cos(\phi_i - \theta_i)) \tag{3.16}$$

with:

$$\dot{\phi}_i = -\frac{d}{dt}\arctan 2(p_i^y, p_i^x)$$

$$= -\frac{1}{1+(\frac{p_i^y}{p_i^x})^2}\frac{\dot{p}_i^y p_i^x - \dot{p}_i^x p_i^y}{(p_i^x)^2}$$

$$= -\frac{1}{\rho^2}\,[\ \rho\cos(\phi_i)\ v_i\ \ \sin(\theta_i) -\ \ \rho\sin(\phi_i)\ v_i\ \ \cos(\theta_i)]$$

$$= -\frac{1}{\rho^2}\,[\rho v_i \sin(\phi_i - \theta_i)]$$

$$= -\frac{1}{\rho}\,[v_i \sin(\theta_i - \phi_i)]$$

$$= \frac{1}{\rho}\,[v_i \sin(\phi_i - \theta_i)]$$

and:

$$\dot{\tau} = \frac{d}{dt}\arctan 2(R, d_i^{ref}) =$$

$$= \frac{1}{1+\frac{R^2}{|e_i|^2}}\left(-\frac{R}{|e_i|^2}\right)\left(\frac{d}{dt}d_i^{ref}\right)$$

$$= -\frac{R}{\rho^2}\frac{p^T\ \dot{p}}{d_i^{ref}}$$

$$= -\frac{R}{\rho^2}\frac{\rho(\cos(\phi-\pi)\ \ \sin(\phi-\pi))v_i(\cos(\theta_i)\sin(\theta_i))^T}{d_i^{ref}}$$

$$= -\frac{R}{\rho^2}\frac{\rho(-\cos(\phi)\ \ -\sin(\phi))v_i(\cos(\theta_i)\sin(\theta_i))^T}{d_i^{ref}}$$

$$= \frac{v_i}{\rho}\frac{R}{d_i^{ref}}\cos(\phi_i - \theta_i)$$

$$= \frac{v_i}{\rho}\tan\tau\ \ \cos(\phi_i - \theta_i)$$

Summarizing the dynamic is:

$$\dot{\theta}_i^{ref} = \frac{v_i}{\rho}(\sin(\phi_i - \theta_i) + \tan\tau\cos(\phi_i - \theta_i))$$

this term reveals a problematic situation when the robot is in the proximity of the circle but with a wrong bearing: in this case $\tau \to \infty$ but $\cos(\phi - \theta) \neq 0$. This drawback can be skipped by acting a modulation over v_f. For hypothesis v_f is a signal greater or equal than zero. In order to achieve the task, it is possible to module v_f in a manner such that input controls remain bounded. In the case where the robot is next to the circle but with a wrong bearing, then $\tau \to \frac{\pi}{2}, |e| \to 0$ and $\rho \to R$; consequently the feedforward $\dot{\theta}_i^{ref}$ becomes:

$$\begin{aligned}
\dot{\theta}_i^{ref} &= \frac{v_i}{\rho}(\sin(\phi - \theta) + \tan\tau\cos(\phi - \theta)) = \\
&= \frac{K_v|e_i| + v_f}{\rho}(\sin(\phi - \theta) + \tan\tau\cos(\phi - \theta)) = \\
&= \frac{0 + v_f}{\rho}(\sin(\phi - \theta) + \tan\tau\cos(\phi - \theta)) = \\
&= \frac{v_f}{R}(\sin(\phi - \theta) + \tan\tau\cos(\phi - \theta)) = \\
&= \frac{v_f}{R}\sin(\phi - \theta) + \frac{v_f}{R}\tan\tau\cos(\phi - \theta)) = \\
&= \epsilon + \frac{v_f}{R}\tan\tau\cos(\phi - \theta))
\end{aligned}$$

where ϵ can be neglected because it is bounded. In order to keep the control ω bounded, it is possible to apply the following saturation:

$$\begin{aligned}
&\frac{v_f}{R}\tan\tau\cos(\phi - \theta)) < M \Rightarrow \\
&\Rightarrow v_f < \frac{M\ R}{\tan\tau\cos(\phi - \theta)}
\end{aligned}$$

with $M > 0$ constant. Applying this rule, the rotational control is bounded and consequently the system respects the BIBO (bounded input - bounded output) property. This allows to state that the rotational error goes to zero at exponential rate.

This saturation can also be interpreted like an initial limitation on the input in order to steer the bearing of the robot in a correct direction. Formally, the term v_f has the following form:

$$v_f = \min(v_d, \frac{M\ R}{\tan\tau\cos(\phi - \theta)})$$

Whenever a correct task execution is asymptotically achieved, the variation of the angle ϕ tends to be constant, $\dot{\phi} \rightarrow \frac{v_d}{R}$ while $\dot{\gamma} \rightarrow 0$. Therefore asymptotically $|e| \rightarrow 0$ and $\theta_i \rightarrow \phi_i + \tau_i$. This can be demonstrated by the following Lyapunov candidate:

$$V = \frac{1}{2}(e_i^2 + e_\theta^2)$$

this is a radially unbounded function with:

$$\begin{aligned}
\dot{V} \quad & = e_i^T \dot{e}_i + e_\theta \dot{e}_\theta = \\
& d_i^{ref}\ (\cos(\theta_i^{\text{ref}})\ \sin(\theta_i^{\text{ref}})) \\
& \Big(\tfrac{d}{dt}\ |e_i|(\ \cos(\theta_i^{\text{ref}})\ \sin(\theta_i^{\text{ref}})\ )^T + \\
& d_i^{ref} \dot{\theta}_i^{\text{ref}}(\ -\sin(\theta_i^{\text{ref}})\ \cos(\theta_i^{\text{ref}})^T\ )\Big) - K_\theta e_\theta^2 \\
& = d_i^{ref} \cdot (p^T \dot{p})/|e_i| + 0 - K_\theta e_\theta^2 \\
& = \rho(\ \cos(\phi - \pi) \sin(\phi - \pi)) \cdot v_i(\cos\theta_i \quad \sin\theta_i)^T - K_\theta e_\theta^2 \\
& = \rho(\ -\cos(\phi) - \sin(\phi)) \cdot v_i(\cos\theta_i \quad \sin\theta_i)^T - K_\theta e_\theta^2 \\
& = -\rho(\ \cos(\phi) \sin(\phi)) \cdot v_i(\cos\theta_i \quad \sin\theta_i)^T - K_\theta e_\theta^2 \\
& = -\rho\ v_i\ \cos(\phi_i - \theta_i) - K_\theta e_\theta^2 \\
& = -\rho\ (K_v|e_i| + v_f)\ \cos(\phi_i - \theta_i) - K_\theta e_\theta^2
\end{aligned} \tag{3.17}$$

After a certain instant the rotational error will be such that the $\dot{V}$ will be permanently negative definite. The only cases where $\dot{V} = 0$ is when $\cos(\phi - \theta) = 0$. This implies that $\tau = \frac{\pi}{2} \Rightarrow |e| \rightarrow 0$, $\theta \rightarrow \theta_i^{ref}$ and $v_f = v_d$. The invariant set will be achieved permanently and the dynamic of the robot will be:

$$v_i = v_d \quad w_i = v_d/R$$

as desired. The regular distribution is obtained by a further modification over the v_f signal. In particular a consensus technique acting on the neighborhood of $\mathcal{A}_i$ can be used. The consensus component v_c is defined as

$$v_c = \max(0, K_c(d_{ij} - d_{ik})) \tag{3.18}$$

where:

$$d_{ij} = |{}^i p_{i+1}|, \tag{3.19}$$

$$d_{ik} = |{}^i p_{i-1}|, \tag{3.20}$$

where d_{ij} and d_{ik} are respectively the distance of the robot $\mathcal{A}i$ from the subsequent and previous agent with respect to the aforementioned notation.

This perturbation is a feedback control that acts on the invariant set of the previous controller. Asymptotically it will lead to $d_{ij} = d_{ik}, \forall i, j, k \in 1, \ldots, N$. In fact, because robots asymptotically arrive to the circumference, the final neighborhood of every robot is constituted by a front and a rear neighbor, i.e. the multi agent topology, in metrical terms, can be represented by a ring. The consensus interaction, described by Equation 3.2, will guarantee the regular distribution of the formation:

$$v'_f = min(v_f + v_c, \frac{M\ R}{\tan\tau\cos(\phi - \theta)})$$

therefore, asymptotically the robots converge to a circle with a linear velocity v_f (v_c asymptotically vanishes) and with a rotational speed $w'_f = \frac{v'_f}{R}$.
In the case robots start inside the circle, a saturation law is proposed. In particular the controller takes the following form:

$$\begin{aligned} v_i &= v_f && (3.21) \\ \omega_i &= K_\omega(\phi_i + \frac{\pi}{2} - \theta_i) + \dot{\phi}_i + K'(R - \rho_i) && (3.22) \end{aligned}$$

it corresponds to a control that steers the robot along the circle having a radius ρ_i (smaller than R). Asymptotically, the perturbation proportional to $(R - \rho_i)$ steers the robot on the circle of radius R. It is interesting that this control law keeps continuity with respect to the transition towards the other side of the circle. As a consequence, the switching Lyapunov theorem could be used in order to determine stability properties (52). Currently only experimental effectiveness is showed.

A particular consideration about this formulation is needed: the coupling of the term τ with the cosine of the angular displacement has to be carefully managed in order to avoid numerical instability issues. In practical case, a saturation over the term τ can be used. During experiments, this device has been demonstrated to be effective although it can lead to a transient in which a switching between the external and internal controller is present due to the approximation induced on τ term

3.6 Conditions for Task Achievement

The proposed methods will achieve the encirclement task provided that the robots can localize the target and each other. In this section, we briefly discuss the conditions under which these two requirements are actually satisfied. Recall that the mutual localization module used in our encirclement scheme is effective within weakly connected components (simply called *subnets* in the following) of the robot detection graph, provided that $D_p \subseteq D_c$ and multi-hop communication is used (44).

The first condition may be derived from the analysis of the desired steady state of the system, in which the N robots are uniformly distributed along a circle of radius R. In this formation, the whole detection graph must be weakly connected, i.e., a single subnet must exist. In view of the circle topology, this property is guaranteed if each robot can detect the target and the successor robot with respect to the cyclic phase ordering (that is actually the predecessor if Ω is positive). For example, this is true if D_p is a frontal circular sector with central angle at least $\pi + \epsilon$ wide, with ϵ any positive number, and radius at least $\max\{R, 2R \sin \pi/n\}$.

The second condition is instead obtained considering the beginning of the encirclement task. To localize the target at $t = 0$, each subnet of the detection graph must contain at least one robot that detects the target. From that moment on, all the robots will get closer to the target in view of the reference evolution (3.5) for ρ, and therefore target detection is guaranteed throughout the task (this is easy to show if D_p has the shape discussed above). In particular, all the subnets will merge into a single connected component that includes the whole graph.

Note that the first condition (on D_p) concerns the robot detector, whereas the second (on the detection graph at $t = 0$) restricts the admissible initial arrangements of the robots with respect to the target. Taken together, they are a *sufficient* condition for task achievement — less demanding requirements may be enforced (in particular, on the shape of D_p) but their efficacy would be more difficult to prove.

3.7 Experiments

We have experimentally validated our encirclement scheme with a system of five Khepera III robots. Each robot is equipped with a wi-fi card and a Hukuyo URG-04LX laser sensor with an angular range of 240° and a linear range artificially limited to 2 m. The robot detector is a simple feature extraction algorithm that inspects the laser scan searching for the indentations made by the vertical cardboard squares mounted atop each robot (in the blind zone of the range finder). Since each square can give

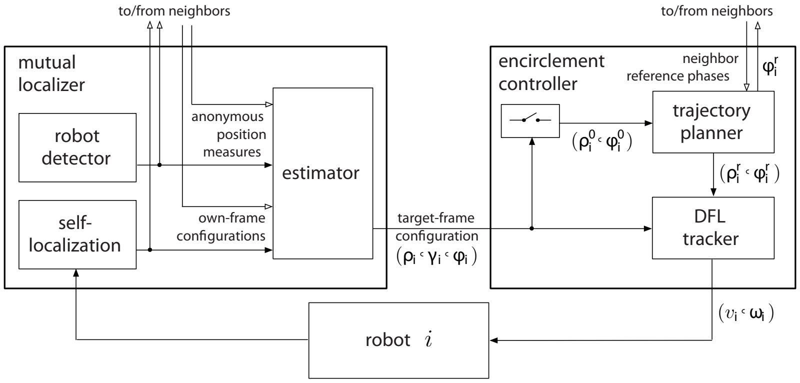


Figure 3.5: The structure of the encirclement system that runs on the$i$-th robot.

indentations of the laser scan that range from 1 to $12cm$, depending on the relative orientation between the measuring and the measured robot, the detector cannot distinguish among robots, the target and obstacles whose size is in the same range. The encirclement scheme has been implemented using the MIP architecture[1] which provides a multi-tasking estimation/control framework, a realistic simulation environment, and allows direct porting for execution on real robots.

The mutual localization module implements the method proposed by (44), to which the reader is referred for a detailed description. The inputs to this component are (1) the anonymous relative position measures (which include the target, if this is contained in D_p) coming from the robot detector (2) an estimate of the configuration of the robot in its own frame, computed by any self-localization (position tracking) algorithm (3) the same information (i.e., anonymous relative position measures and robot configuration in its own fixed frame) obtained via communication from each neighbor. A multiple registration algorithm followed by a multi-EKF are used to process these data and compute an accurate estimate of the configuration of each robot in a common fixed frame. While a cooperating target is directly identified and localized with this procedure, it is interesting to note that a non-cooperating target can still be singled out by the mutual localization module as the only 'robot-like' object that does not communicate its data. From the mutual localization results, each robot can directly derive an estimate of

[1] http://www.dis.uniroma1.it/∼labrob/software/MIP/index.html.

its configuration (position and orientation) with respect to a *common* target-centered frame.

The encirclement control module generates the control inputs to the robot using the target-centered configuration of the robot computed by the mutual localizer as well as information coming from the neighbor robots.

3.7.1 Experimental results - Orbital controller

A typical experimental result is summarized in Fig. 3.6. Here we are considering version 1 of the encirclement task, and hence (3.5–3.6) as a trajectory planner, with $R = 0.5$ m and $\Omega = 0.06$ rad/s. One of the robot is used as a stationary (cooperating) target. At the start (snapshot 1), only three robots are active. At $t_1 = 200$ s, with the three robots rotating around the target in a regular formation, another robot is added (snapshot 2); the four robots then achieve a regular formation (snapshot 3). At $t_2 = 310$ s one of the robot is kidnapped and powered off (snapshot 4). The three remaining robots rearrange themselves in a regular formation (snapshot 5). Finally, at $t_4 = 600$ s another robot is kidnapped and the formation becomes a 1 m wide dipole. The evolution of the experiment is also illustrated by Fig. 3.7, that shows the plots of the distances between consecutive robots, the distances between each robot and the target, and the robot angular speeds. The fact that during each phase the correct regular formation is promptly reached shows the reactivity of the proposed encirclement scheme.

We have also run experiments with moving targets, obtaining satisfactory results as long as the speed of the target remains at least one order of magnitude smaller than that of the robots. One such experiment is shown in Fig. 3.8. Video clips of the experiments are available at http://www.dis.uniroma1.it/labrob/research/encirclement.html.

3.7.2 Experimental results - Tangential controller

In this section an experiment considering the tangential controller is reported. In particular the situation where one or more units are suddenly moved away, i.e. kidnapped, from their position is analyzed[1]. Also some considerations about the experimental

[1] Other videos and data are available at http://www.dis.uniroma1.it/~labrob/research/entrapment.html.

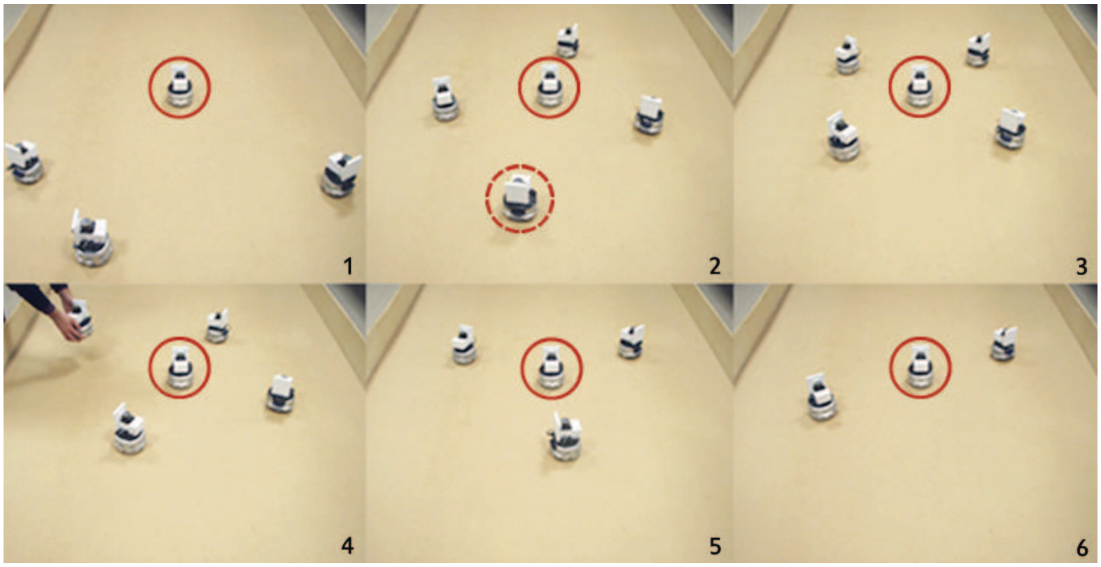


Figure 3.6: Encirclement of a stationary target (solid red circle).

setup responsiveness is detailed. The circle radius is set to $R = 0.4$ m and the estimated distances between each agent and the target are showed in Fig. 3.9. The 3 estimated distances between $\mathcal{A}_1$ and each other robot are showed in Fig. 3.10, where the 3 horizontal dotted lines represent the 3 correct inter-robot distances for a formation of 4, 3 and 2 robots, in fact d_1 corresponds to $\sqrt{2}R$, d_2 is $\sqrt{3}R$ while d_3 is equal to $2R$. Relevant phases of the experiment are reported in Fig. 3.11. At the beginning the 4 robots start 2 m away from the target and move to encircle it. At instant t_0 the desired 4-robot encircling is reached, at instant t_1 $\mathcal{A}_4$ is manually kidnapped and kept away, at instant t_2 the 3 remaining robots rearrange and reach a new equilibrium; at instant t_3 also $\mathcal{A}_3$ is manually kidnapped but immediately released 2 m away; in the meanwhile the two remaining robots reach the equilibrium on a 2-robot formation; at t_4, after $\mathcal{A}_3$ has reached the other two robots, a new equilibrium for 3-robot team is reached and $\mathcal{A}_4$ is reinserted 2 m away; finally at t_5 a new equilibrium on a 4-robot formation is reached again. All these kidnappings, that can also be considered as robots failures, show fault tolerant capabilities of the mutual localization. Note that the estimator of each robot also provides a steady relative measure related to a robot temporarily or definitely hidden by means of occlusions as well as the relative measure of a robot on the back which is invisible to the object detector due to the limited cone of the laser. Subsequently, the tangential controller provides an effective execution of the task.

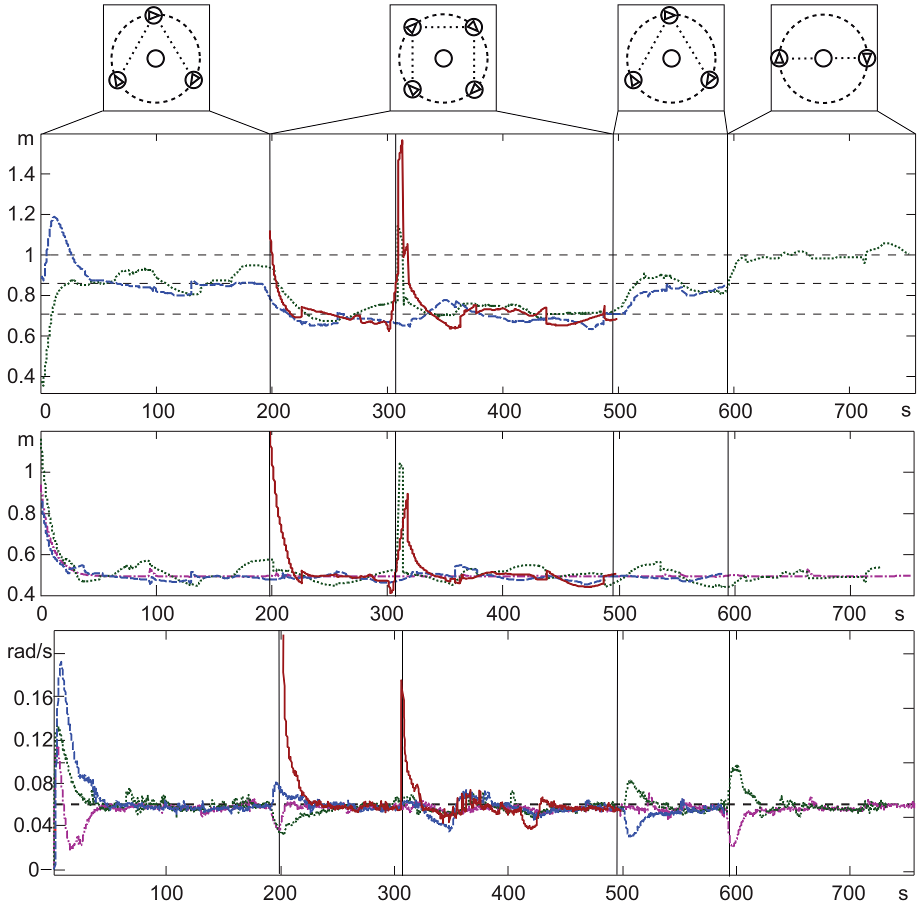


Figure 3.7: Orbital controller experiment: distances between consecutive robots (top), distances between each robot and the target (center), angular speeds of the robots (bottom).

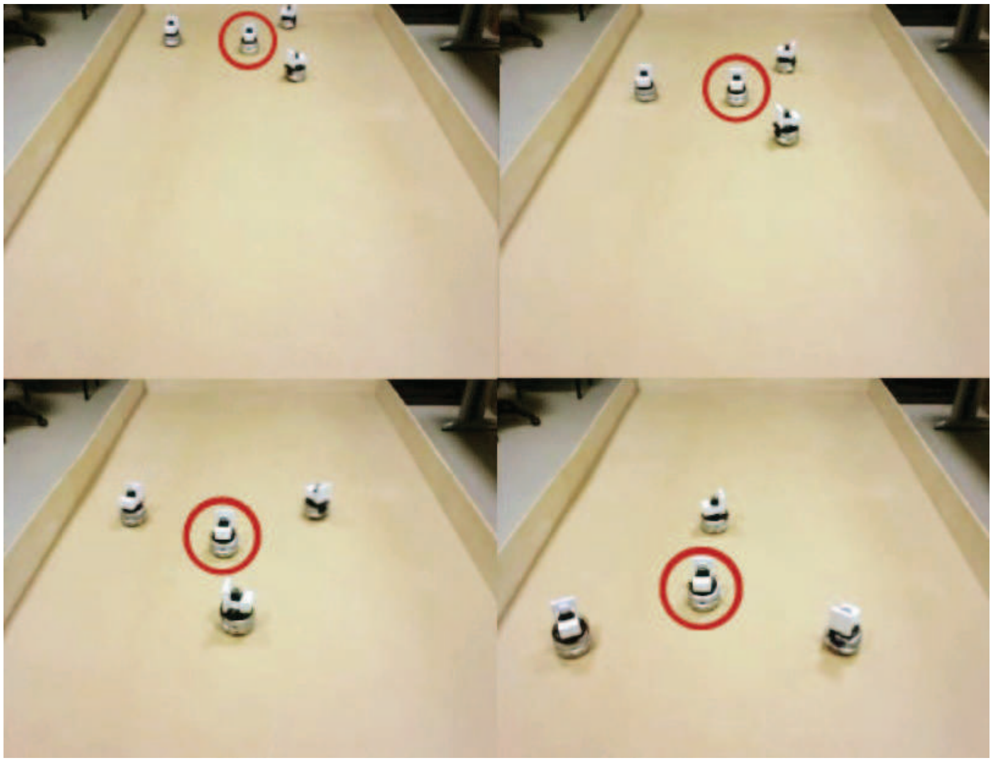


Figure 3.8: Encirclement of a moving target (solid red circle).

Since we do not use a global localization system we do not have a ground truth localization to compare with our mutual localization in order to understand how the performances of the mutual localization influence the performances of the controller. However, since the implementation of the mutual localization algorithm and the communication protocol are exactly the same in simulation and in a real experiment, a simulation reproducing the events sequence is perfectly suitable for this aim. The results of this simulation are showed in Fig. 3.12, where the real distances are compared with the estimated ones, and in Fig.3.13, where the estimation error is plotted. The unique relevant aspect to note is the small delay of the estimated quantity, due in most part to the communication delay between the agents.

3.8 Conclusions

In this chapter two techniques for the encirclement problem have been presented. The proposed schemes integrate a mutual localization module based on the developments

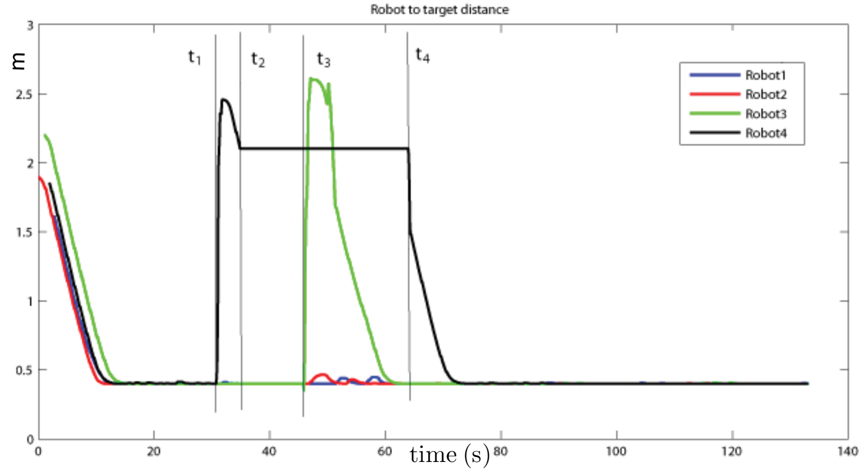


Figure 3.9: Robots to target distances: before t_1, robots reach the circle. Then at t_1 the robot $\mathcal{A}_4$ is kidnapped: it is physically blocked from t_2 to t_4. Also $\mathcal{A}_3$ is moved away at t_3. At the end, the right configuration is obtained.

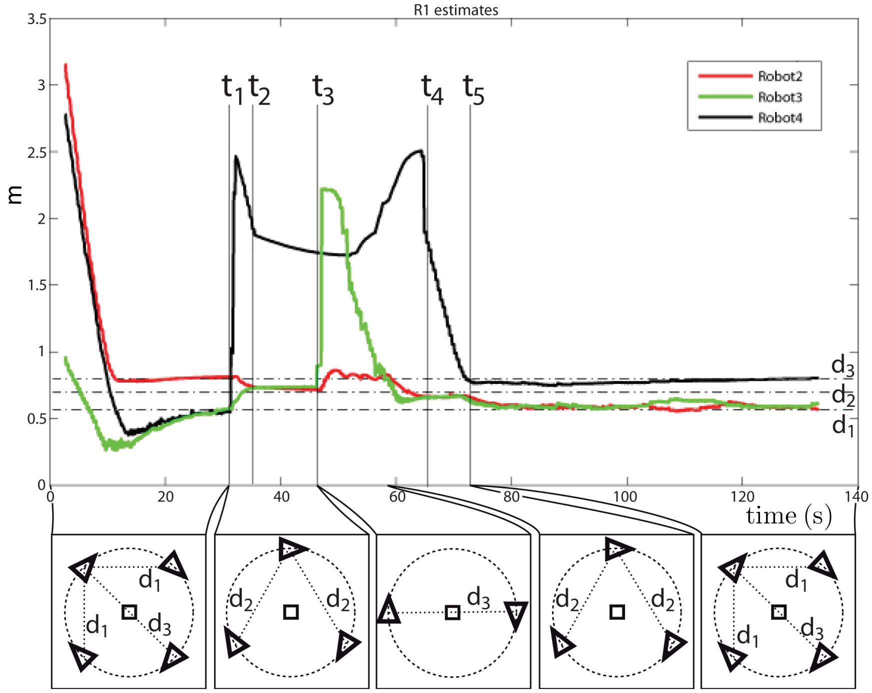


Figure 3.10: Robot1's estimate: before t_1 the four agents reach the correct 4-formation. At t_1 $\mathcal{A}_4$ is kidnapped and the remaining agent are $\sqrt{3}R$ far to each other. At t_2 also $\mathcal{A}_3$is kidnapped and consequently a 2-formation is established. Finally, reinserting both $\mathcal{A}_4$ at t_4 and $\mathcal{A}_3$ at t_5, the 4-formation is newly achieved.

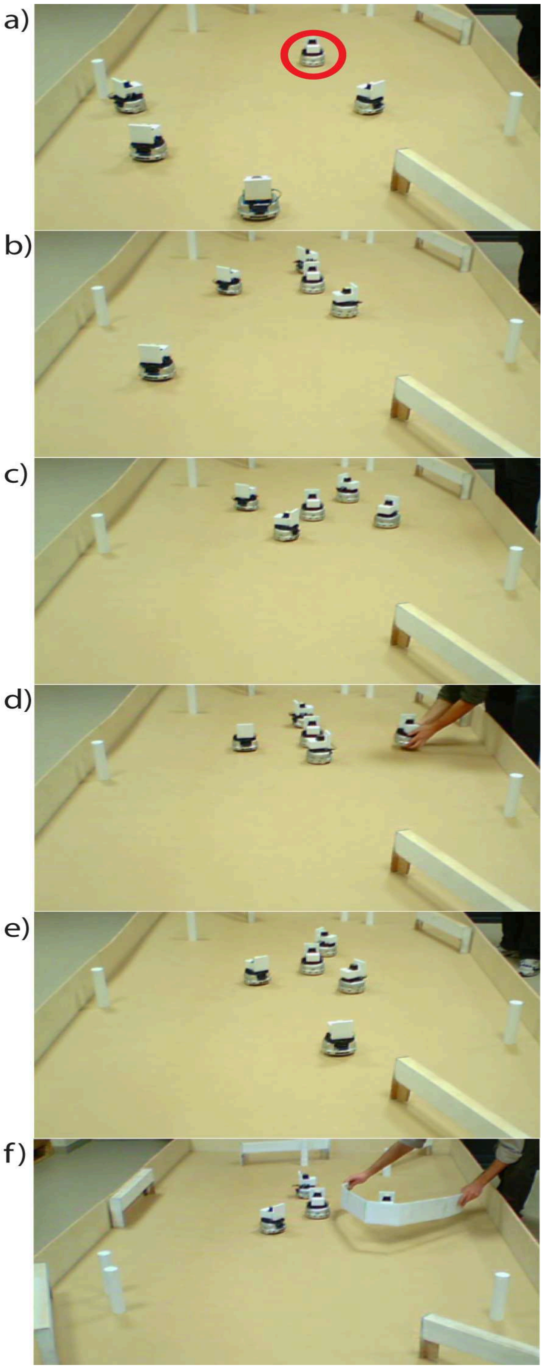


Figure 3.11: Snapshot of an experiment: a) starting formation: in the top of the figure can be seen the target b) three robots in the encircling's transient ; c) four robots encircling; d) kidnapping of a robot; e) three robots formation restored after kidnapping; f) hiding of a robot.

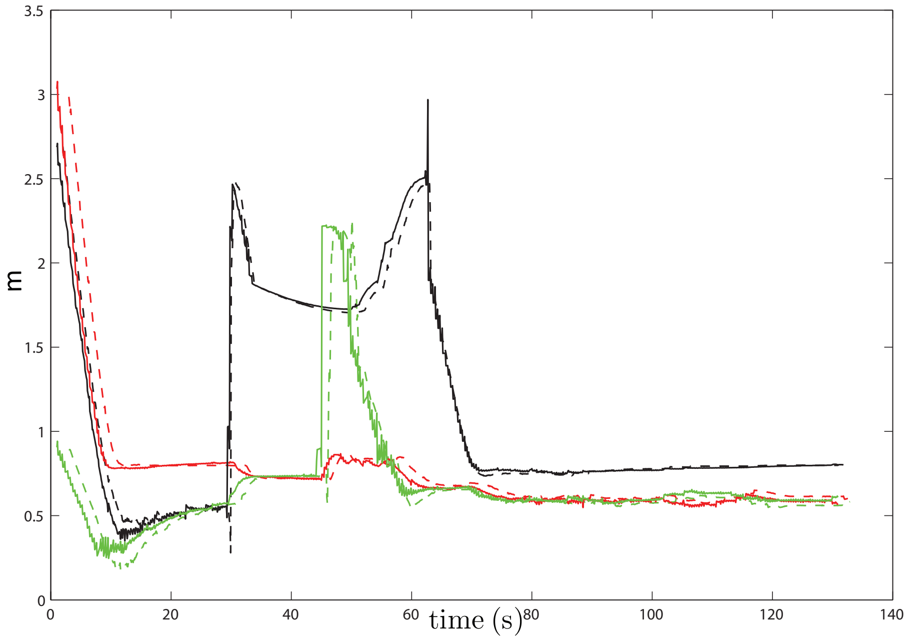


Figure 3.12: Real (continuous lines) and estimated (dashed lines) distances between $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$. The estimates follow the real values with a small delay.

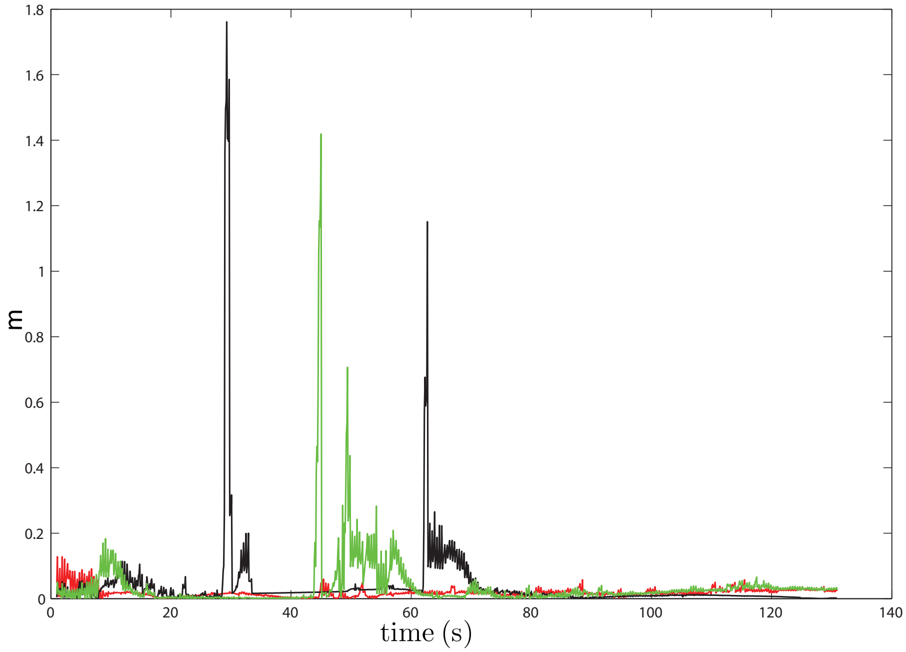


Figure 3.13: Error in the estimate of the distances between $\mathcal{A}_1$ and $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$. The errors are always sufficiently limited except when a robot is kidnapped, due to the fast displacement and the estimate delay.

in (44).

Concerning about the Orbital controller the theoretical proof of its effectiveness is supported by extensive experimental results.

Future work will be aimed at:

- proving that the reference trajectories never meet, so as to provide grounds for identifying a collision avoidance condition for robots of finite size;
- performing a theoretical analysys of a trajectory generation scheme based on *continuous replanning*, in which the actual robot coordinates (estimated through the mutual localization module) are used in place of their reference value (we already implemented such a variant with encouraging results);
- integrating a *consensus* mechanism among the robots on the results of the mutual localization, and especially the configuration of the target.

Concerning the tangential controller, only the part for robots coming from the external part of the circle has been proven. Improvements regarding the inner part will be considered as well as the effect of saturation blocks. The latter topic would be useful in order to use the linear input to avoid collisions.

Chapter 4

Data fusion with the Transferable Belief Model approach

In this Chapter a multi agent approach to the Transferable Belief Model is proposed. This framework shows an effective representation of the ignorance and contradiction among sources about phenomena which have to be estimated. The technique has been validated through extensive experiments concerning topological map building

4.1 Introduction

Data fusion is a research area that is growing rapidly due to the fact that it provides means for combining pieces of information coming from different sources/sensors. As a result, an enhanced overall system performance, i.e., improved decision making, increased detection capabilities, diminished number of false alarms, improved reliability, with respect to separate sensors/sources can be achieved (53).

Indeed, data fusion techniques play an important role in the context of multi-agent systems where information coming from different sources must be aggregated in order to provide a meaningful description of the surrounding environment. The majority of works available in the literature is based on the Bayesian framework, where the aggregation is achieved by applying the Bayes rule. The most representative example is the Kalman Filter, where noisy data are assumed to be described by means of a gaussian probability distribution (54). Several works have been proposed to deal with the multi-agent data fusion in a Bayesian framework (55, 56). In this context, the network topology of the multi-agent system is considered as a Bayesian network over which a message passing algorithm to perform inference on a graphical model is devised.

The Theory of Evidence (DS) introduced by Arthur P. Dempster and Glenn Shafer represents a valid alternative to the Bayesian framework (57). The main difference concerns the way in which the ignorance is handled: in the probabilistic framework the uncertainty is treated by splitting the amount of credibility among plausible events, in the DS framework a belief is assigned to the set describing all the plausible hypotheses without supporting any in particular. Depending on the specific application, one framework can be more adequate than the other (58). This framework was further extended by the Transferable Belief Model (TBM) introduced by Philippe Smets (59). In particular, TBM introduces the idea of open world assumption in the DS framework. This implies the set of hypotheses not to be exhaustive, therefore data can take to contradiction. Indeed, the concept of contradiction is a powerful tool to detect cases where information fusion has to be considered unreliable, case that is not considered in the Bayesian technique. The main limitation of this framework is the computational complexity, which grows exponentially with respect to the number of events, as their supersets have to be taken into account. To overcome this drawback, several approximation technique have been proposed (60), (61), (62). However, in case a minimal number of events is enough to model the problem, the TBM approach has been effectively used, e.g, in diagnostic applications (63) and target identification (64). Other methods, basing on the aforementioned approximation techniques, concern vehicle localization (65), and water treatment (66).

In this chapter the data aggregation problem for a multi-agent system is investigated. In the following dissertation, agents are assumed to be independent reliable sources which collect data collaborating to reach a common knowledge. A protocol for distributed data aggregation which is proved to converge to the basic belief assignment (BBA) given by a centralized aggregation based on the Transferable Belief Model (TBM) conjunctive rule is provided. The approach of the TBM to multi robot system is finally used in the topological map building: the cooperation between SAETTA robots allows the estimation of patterns which could not be identified by the single agent. The proposed technique fits properly the capabilities of the SAETTA robots: within the proposed framework, i.e. a limited number of hypotheses, our techniques is suitable for small and cheap robots like SAETTA.

4.2 Transferable belief model

The Theory of Evidence is a formalism which can be used for modeling uncertainty instead of classical probability. Theory of Evidence embraces the familiar idea of using a number between zero and one to indicate the degree of belief for a proposition on the basis of the available evidence.

Let $\Omega = \{\omega_1, \ldots, \omega_n\}$ be a finite set of possible values of a variable ω, where the elements ω_i are assumed to be mutually exclusive and exhaustive. Let $\Gamma(\Omega) \triangleq 2^\Omega = \{\gamma_1, \ldots, \gamma_{|\Gamma|}\}$ be the power set associated to it. In this framework, the interest is focused in quantifying the belief of propositions of the form: "the true value of ω is in γ", with $\gamma \in \Gamma$. The propositions of interest are therefore in one-to-one correspondence with the subset Ω, and the set of all propositions of interest corresponds to the elements of Γ. The set Ω so defined, is referred as *frame of discernments*.

Definition 1 (BBA) *A function* $m : 2^\Omega \to [0, 1]$ *is called a basic belief assignment if*

$$\begin{aligned} m(\emptyset) &= 0 \qquad (4.1) \\ \sum_{\gamma_a \in \Gamma} m(\gamma_a) &= 1 \qquad (4.2) \end{aligned}$$

Thus for $\gamma_a \in \Gamma$, $m(\gamma_a)$ is the part of belief that supports exactly γ_a, i.e. the fact that the true value of ω is in γ_a, but due to the lack of further information, does not support any strict subset of γ_a. The first condition reflects the fact that no belief should be committed to $\emptyset$ and the second condition reflects that the total belief has measure one.

Notice that $m(\gamma_a)$ and $m(\gamma_b)$ can be both equal to zero even if $m(\gamma_a \cup \gamma_b) \neq 0$. Further, $m(\cdot)$ is not monotone under inclusion, i.e. $\gamma_a \subset \gamma_b$ does not imply $m(\gamma_a) < m(\gamma_b)$. Notice that the BBA represents the atomic information in the theory of evidence.

Definition 2 *A function* $Bel : 2^\Omega \to [0, 1]$ *is called belief function over* Ω *if it satisfies the following relationship:*

$$Bel(\gamma_a) = \sum_{\gamma_b \subseteq \gamma_a} m(\gamma_b) \qquad (4.3)$$

this function quantifies the total specific amount of belief supporting the event, and it is often taken into account in the decision making process after data aggregation has been performed (59).

The main criticism to Shafer formulation concerns the application of the Dempster-Shafer (DS) combination rule. In fact, whenever there is a strong conflict between sources to be combined, the straightforward application of DS combination rule can produce a result in which certainty is assigned to the minority opinion (67).

A more refined approach is based on the Transferable Belief Model (TBM) proposed by Philips Smets in (68). The TBM theory, like the Shafer formulation, relies on the concept of basic belief assignment function, but removes the assumption of $m(\emptyset) = 0$. This allows to omit the normalization constant in the Dempster's rule of combination and conditioning.

Definition 3 (Smets - operator $\bigotimes$)
In the TBM, the combination rule is, therefore, defined in this way:

$$s_{ij} \triangleq s_i \bigotimes s_j = \Big\{ (m_i \otimes m_j)(\gamma_a);\ \gamma_a \in\ \Gamma \Big\} \tag{4.4}$$

where:

$$m_{ij} \triangleq (m_i \otimes m_j)(\gamma_a) = \sum_{\substack{\gamma_b,\, \gamma_c \\ \gamma_b \cap \gamma_c = \gamma_a}} m_i(\gamma_b) m_j(\gamma_c). \tag{4.5}$$

Note that, for sake of clarity the following notation $m_i(\gamma_a) \otimes m_j(\gamma_a) \triangleq (m_i \otimes m_j)(\gamma_a)$ will be used indiscriminately in the further.

The fact that $m(\emptyset) > 0$ can be explained in two ways: the open world assumption and the quantified conflict. The open world assumption reflects the idea that Ω might not be exhaustive, i.e. it might not contain all the possibilities. Under this interpretation, being $\emptyset$ the complement of Ω, the mass $m(\emptyset) > 0$ represent the modeling errors, that is the fact that the truth might not be contained in Ω. The second interpretation of $m(\emptyset) > 0$ is that there is some underlying conflict between the sources that are combined in order to produce the BBA m. Hence, the mass assigned to $m(\emptyset)$ represents the degree of conflict.

In particular, it can be computed as follows:

$$m_{ij}(\emptyset) = 1 - \sum_{\substack{\gamma_a \neq \emptyset \\ \gamma_a \in \Gamma}} m_{ij}(\gamma_a) \tag{4.6}$$

4.3 A Transferable Belief model applied on formation pattern selection

In this section an algorithm to select a geometric shape into which a set of robots have to arrange is proposed. This technique uses the classical TBM: although being distributed, the application shows several limitations (see Section 4.3) which will be removed in the following Sections.

4.3.1 Problem setting

Consider a set of N holonomic robots $\mathcal{R} = \{\mathcal{A}_i, i = 1 \dots N\}$, each of which is able to communicate within a *communication disk* whose radius is C_{max} and to sense within a *sensing disk* whose radius is D_{max}, where typically $C_{max} \geq D_{max}$. Suppose that each robot is equipped with a sensorial system s_i that is composed by an array of sonar range finders. Denote with *formation pattern* a set of virtual robots $\mathcal{V} = \{v_i, i = 1, \dots, N\}$. Furthermore, some patterns are supplied: they can be grouped in a single set $\mathcal{P}$ defined as $\mathcal{P} = \{p_k, k = 1, \dots, S\}, S \in \mathbb{N}$. Furthermore, suppose that the set $\mathcal{R}$ explores an environment which can be approximated by the union of atomic environmental patterns. These patterns can be detailed in corridor, T-junction, open-space and corner. Call the set of these patterns Ω. The aim of this algorithm is to describe a technique to arrange the set $\mathcal{R}$ into the best formation pattern with respect to proper performance indices, after classifying the surrounding environment.

4.3.2 Environmental categorization

The fusion process is performed after that each robot creates its own basic mass assignment using the map building technique described in (69). In each iteration of the algorithm, every robot provides a BBA $M_i = \{m_i(p), \forall p \in 2^\Omega\}$. The fusion information

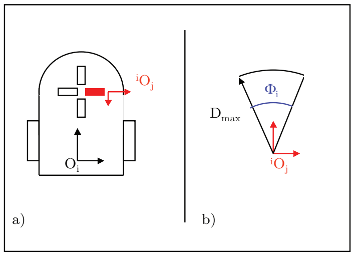


Figure 4.1: a)Frames on the robot: in black O_i is the robot frame while ^iO_j is the orientation of the j-th scan b) Sonar parameters with respect to the frame attached to the j-th scan.

is done using the Eq. 4.5:

$$m_{ij}(z) = \sum_{t,w:z=t\cap w} \mathrm{m}_i(t)\mathrm{m}_j(w), \forall z \in \Gamma. \tag{4.7}$$

This approach show several limitations related to the information exchange among the agents. The framework requires to adopt a rigid information passing scheme: the flux of information can not be modified and furthermore synchronization is required. Synchronization is required to overcome situations where an an agent, during the process, elaborates a new set of masses from a new observation: this new set of data can not be instantaneously inserted in the estimate. This work has been briefly explained in the thesis in order to describe which are the possible limitations in using a standard

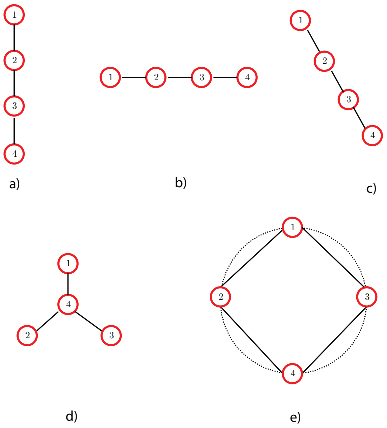


Figure 4.2: Formation patterns composed by four elements: a) Column b) Row c) Slash d) Delta e) Circle.

formulation related to the possibility theory. The details about the implementation can be found in (4).In the further sections, these limitations will be removed.

4.3.3 Simulations

In this section, simulations about the proposed *environment recognition* and *formation selection* algorithms are presented. Algorithms were run only in simulated environment under the *Player/Stage* framework (20).

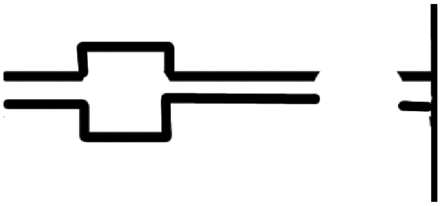


Figure 4.3: Environment map used in simulations.

In this scenario robots are asked to pass through the environment depicted in Figure 4.3. During the execution, the robots optimize some performance indices: for example they have to optimize the coverage, i.e. they have to cover as space as they can, or the crossing speed, i.e. the ability of the formation to pass as quick as possible through certain environment.

In the first simulation, whose salient moments are depicted in Figures 4.4a-d, the robots pass through the environment optimizing the coverage and the quickness of passing. It can be seen that the crossing of corridors brings the set of robots to arrange in a column formation because it is the only feasible solution (Fig. 4.4-a and Fig. 4.4-c). When the platoon is instead in a larger section, robots move towards a row formation (Figure 4.4-b,d). The estimation of the environment over the time is depicted in Figure 4.5: on the x-axis there is the number of iterations while the y-axis shows the mass values over the time. It is worth to mention that here only atomic elements are figured. In particular the green line represent the mass associated to the corridor, the blue dotted line is the one associated with the open space and finally the red one is related to the T-junction. In accordance with the results, the categorization changing is evident.

As mentioned before, the fusion process constrains the communication protocol: each robot collects the masses of its neighbors taking care that no cycles are formed.

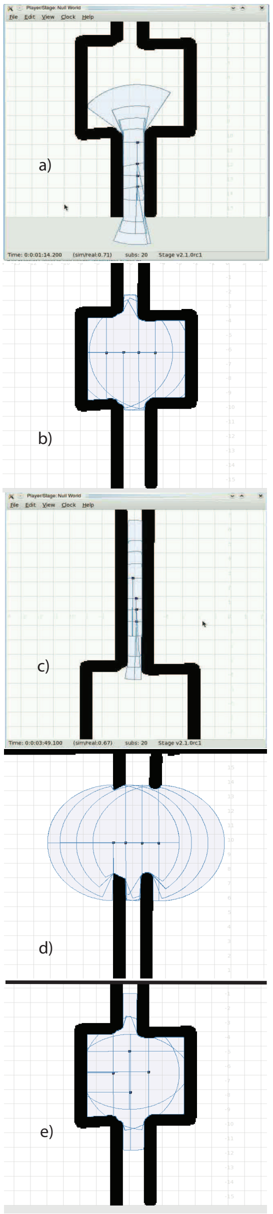

Player/Stage: Null World
File Edit View Clock Help
a)
Time: 0:0:01:14.200 (sim/real:0.71) subs: 20 Stage v2.1.0rc1
b)
Player/Stage: Null World
File Edit View Clock Help
c)
Time: 0:0:03:49.100 (sim/real:0.67) subs: 20 Stage v2.1.0rc1
d)
e)

Figure 4.4: a) - d) Arranged formation with $\alpha = [0.4, 0.1, 0.4, 0.1]^T$ e) formation with $\alpha = [0.25, 0.4, 0.25, 0.1]^T$.

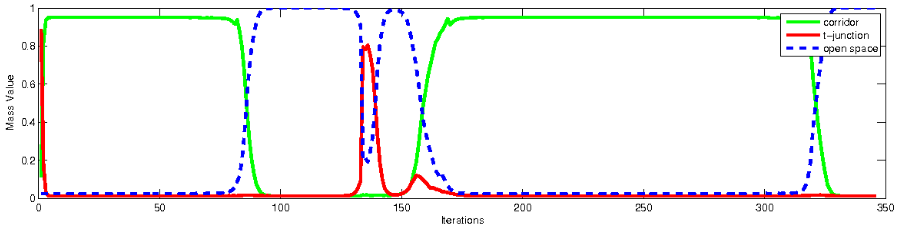


Figure 4.5: Mass distribution of atomic elements over the time

4.4 Distributed TBM for multi agent systems

Let us consider a simple scenario, where three agents, whose network topology is depicted in Figure 4.6, observe the same event. Furthermore, let us assume that the objective of these agents is to perform a data aggregation in order to reach a common knowledge about such an event. To this end, let us suppose that agent 1 first collaborates with agent 2 and successively sets up a collaboration with agent 3. At this point, a question arises: what happens if agent 1 collaborates again with agent 2? Let us further investigate this situation.

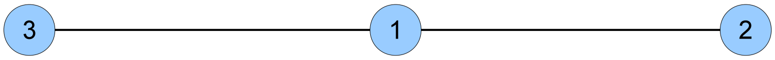


Figure 4.6: Network topology.

Every agent can apply the Smets combination rule to interact with its neighbors. When agent 1 collaborates with agent 2 they share the owned information to reach a new common knowledge (s_{12}). After that, when agent 1 sets up a collaboration with agent 3, they reach a new knowledge (s_{123}). Now if agent 1 and agent 2 collaborate again, by using the Smets combination rule, they would reach a new "wrong" knowledge (s_{12123}), where the information due to the first communication would be considered twice.

Therefore, in a distributed scenario a different combination strategy must be designed in order to overcome this issue. To this end, let us assume the current knowledge of an agent can be divided with respect to any of its neighbors in two parts, i.e, common knowledge and novelty. In particular, the common knowledge represents the portion of information shared by the agents, while the novelty is the novel portion of information brought by an agent. At this point, this issue can be simply overcome by restricting the aggregation among agents to the novelty, and then combining the obtained result with the common knowledge. In this way, when agent 1 and agent 2 collaborate again, they can reach a new knowledge avoiding to consider twice the result of their previous aggregation.

In section 4.4.1 the problem setting related to this scenario is summarized. In Section 4.4.2 a technique to deal with this static scenario is proposed, while in 4.4.3 an additional assumption is made: agents can dynamically collect new observations over time. As a consequence, the following question holds: what happens if some agents update their observations while performing the data aggregation? Indeed, the protocol must take into account the fact a node might update its direct knowledge. To this end, a proper extension of the proposed distributed data aggregation protocol is devised in order to overcome this limitation. In particular conditions related to the rate of the new incoming observations will be provided to guarantee the convergence of the system to a common BBA.

4.4.1 Problem Setting

Let the network of agents be described by an undirect graph $\mathcal{G} = \{V, E\}$, where $V = \{v_i : i = 1, \dots, n\}$ is the set of nodes (agents) and $E = \{e_{ij} = (v_i, v_j)\}$ is the set of edges (connectivity) representing the point-to-point communication channel availability. A position $p_i \in \mathbb{R}^d$ in the d^{th} dimensional space is associated to each node $v_i \in V$, with $i = 1, \dots, n$.

In particular, an edge representing a connection between two agents exists if and only if the distance between these agents is less then or equal to their communication radius $r \in \mathbb{R}$, namely

$$E = \{e_{ij} : \ \|p_i - p_j\|_d \leq r, \ i \neq j\},$$

where $\|\cdot\|_d$ is the Euclidean norm in $\mathbb{R}^d$. Since the graph is undirect the existence of the edge e_{ij} (from node i to node j) implies the existence of the edge e_{ji} (from node j to node i). Therefore, in the following they will be used without distinction to indicate a connection between node i and j. Moreover, we will refer to $\mathcal{N}(i)$ as the neighborhood of agent i, namely the set of indices of the agents directly connected through an edge with agent i.

In the proposed framework a gossip algorithm (70) is defined as a triplet $\{\mathcal{S}, \mathcal{R}, \mathfrak{e}\}$ where:

- $\mathcal{S} = \{s_1, \ldots, s_n\}$ is the set containing the local state $s_i \in \mathbb{R}^q$ of each agent i in the network.
- $\mathcal{R}$ is the local interaction rule ($\oplus$ binary operator) that, for any couple of agents (i, j) with $e_{ij} \in E$, gives:

$$\mathcal{R} : \quad \mathbb{R}^q \times \mathbb{R}^q \longrightarrow \mathbb{R}^q.$$

- $\mathfrak{e}$ is the edge selection process that specifies which edge $e_{ij} \in E(t)$ is selected at time t.

```
Algorithm 2: Gossip Algorithm.
  Data: $t = 0$, $s_i(0) \quad \forall\, i = 1, \ldots, n$.
  Result: $s_i(t_{stop}) \quad \forall\, i = 1, \ldots, n$.

  while stop_condition do
        • Select an edge $e_{ij} \in E(t)$ according to $\mathfrak{e}$.
        • Update the states of the selected agents applying $\mathcal{R}$:
              $s_i(t+1) = s_i(t) \oplus s_j(t)$
              $s_j(t+1) = s_j(t) \oplus s_i(t)$
        • Let $t = t + 1$.
  end
```

From an algorithmic point of view, a possible implementation of the gossip algorithm is given in Algorithm 2.

The following assumptions on the network of agents are made:

Assumptions 1

- *The network can be described by a connected undirected graph $\mathcal{G}$.*
- *The communication range is limited by a maximum communication radius r.*
- *The communication among agents is asynchronous,* gossip *like (70).*
- *A distributed algorithm to build a spanning tree $\mathcal{T}$ is available to the agents, for example by using (71, 72).*

- *Agents are capable to handle the storage required for the algorithm execution.*

■

4.4.2 Static version

In this section, a local interaction rule $\mathcal{R}$ to perform the distributed TBM data aggregation over a network topology for a multi-agent system, under assumptions 1, is described. Furthermore, every agent is assumed to provide only a single observation over the time.

Let us assume $\mathcal{T} = \{V, \hat{E}\}$ to be the spanning-tree over the graph G computed by the agents in a distributed fashion, where $V = \{v_i : i = 1, \ldots, n\}$ is the set of nodes (agents) and $\hat{E} = \{e_{ij} = (v_i, v_j)\},\ \hat{E} \subseteq E$ is a subset of the edges of E required to build a spanning-tree.

Definition 4 ($\mathcal{S}$)
Let $\mathcal{S}(t) = \{s_1(t), \ldots, s_n(t)\}$ be the set of the agents states defined with respect to a finite frame of discernment $\Omega = \{\omega_1, \ldots, \omega_m\}$, where $s_i(t) = \{m_i(t, \gamma_a), \gamma_a \in \Gamma\},\ s_i(t) \in \mathbb{R}^{|\Gamma|}$ is the set of basic belief assignment (BBA) of agent i over the power set $\Gamma(\Omega)$ at a given time $t \in \mathbb{N}$. Note that, in the following the time dependence will be omitted for sake of clarity if not strictly required.

Let us now introduce the binary operator $\odot$, which is useful to break up any set of basic belief assignment with respect to any other one.

Lemma 4.1 *Let us consider two sets of basic belief assignment (BBA) $s_k = \big\{m_k(\gamma_a); \quad \forall \gamma_a \in \Gamma\big\}$, and $s_i = \big\{m_i(\gamma_a); \quad \forall \gamma_a \in \Gamma\big\}$. It can be defined an operator $\odot$:*

$$\tilde{s}_k^i \triangleq s_k \odot s_i = s_j \tag{4.8}$$

such that:

$$s_k = s_i \otimes s_j. \tag{4.9}$$

In particular, each element of $s_j = \Big\{m_j(\gamma_a); \quad \forall \gamma_a \in \Gamma\Big\}$*, can be computed recursively as follows:*

$$m_j(\gamma_a) = \frac{m_k(\gamma_a) - \sum_{\substack{\gamma_b \,\cap\, \gamma_c \,=\, \gamma_a \\ \gamma_b \,\supset\, \gamma_a}} m_j(\gamma_b) m_i(\gamma_c)}{\sum_{\gamma_a \subseteq \gamma_b} m_i(\gamma_b)}, \tag{4.10}$$

by starting from the element of the power-set with highest cardinality, $\gamma_{|\Gamma|} = \{\Omega\}$*, and moving down to the elements with cardinality equal to one, i.e.,* $\{\gamma_{i+1} = \{\omega_i\},\ i = 1, \ldots, n\}$*, with* $\gamma_1 = \emptyset$*.*

Proof: *The proof is a simple consequence of the application of Smets operator* $\otimes$*. Let us assume* s_k *can be written as the Smets aggregation of* s_i *and* s_j*:*

$$s_k \triangleq s_i \otimes s_j = \Big\{(m_i \otimes m_j)(\gamma_i);\ \gamma_i \in\ \Gamma\Big\}$$

where:

$$\begin{aligned} m_k(\gamma_a) &\triangleq (m_i \otimes m_j)(\gamma_a) \\ &= \sum_{\gamma_b \cap \gamma_c = \gamma_a} m_i(\gamma_b) \cdot m_j(\gamma_c) \\ &= m_j(\gamma_a) \sum_{\gamma_b \subseteq \gamma_a} m_i(\gamma_b) + \sum_{\substack{\gamma_b \,\cap\, \gamma_c \,=\, \gamma_a \\ \gamma_b \,\supset\, \gamma_a}} m_j(\gamma_b) m_i(\gamma_c). \end{aligned}$$

At this point, by collecting with respect to $m_j(\gamma_a)$ *the following expression is obtained:*

$$m_j(\gamma_a) = \frac{m_k(\gamma_a) - \sum_{\substack{\gamma_b \,\cap\, \gamma_c \,=\, \gamma_a \\ \gamma_b \,\supset\, \gamma_a}} m_j(\gamma_b) m_i(\gamma_c)}{\sum_{\gamma_a \subseteq \gamma_b} m_i(\gamma_b)}.$$

Therefore, $s_j = \Big\{m_j(\gamma_a); \quad \forall \gamma_a \in \Gamma\Big\}$ *is obtained.* □

Definition 5 ($\mathcal{R}$ **- operator** $\oplus$ **)**
Let $\mathcal{R}$ *be a rule to combine the basic belief assignments for two agents* (i, j) *such that* $e_{ij} \in \hat{E}$ *as follows:*

$$\begin{aligned} s_i(t+1) &= s_i(t) \oplus s_j(t) \\ &= \Big\{ \big(\tilde{m}_i^j(t,\gamma_a) \otimes \tilde{m}_j^i(t,\gamma_a)\big) \otimes \bar{m}_{i,j}(t,\gamma_a); \\ &\quad \forall\, \gamma_a \in \Gamma \Big\}, \end{aligned} \tag{4.11}$$

with $\otimes$ *the Smets operator. Let us denote with* $\tilde{s}_i^j(t) = \Big\{\tilde{m}_i^j(t,\gamma_a); \quad \forall \gamma_a \in \Gamma\Big\}$ *the novelty of the agent* i *with respect to the agent* j*, which can be computed recursively as follows:*

$$\tilde{m}_i^j(t,\gamma_a) = \frac{m_i(t,\gamma_a) - \sum_{\substack{\gamma_b \cap \gamma_c = \gamma_a \\ \gamma_b \supset \gamma_a}} \tilde{m}_i^j(t,\gamma_b)\bar{m}_{i,j}(t,\gamma_c)}{\sum_{\gamma_a \subseteq \gamma_b} \bar{m}_{i,j}(t,\gamma_b)} \tag{4.12}$$

and $\bar{s}_{i,j}(t) = \{\bar{m}_{i,j}(t,\gamma_a); \quad \forall \gamma_a \in \Gamma\}$ *(or equivalently* $\bar{s}_{j,i}(t)$*) is the common knowledge, i.e., the knowledge stored by both agents after their last aggregation, set to the neutral element* $\mathfrak{n} = \{0,\, 0\ \ldots,\, 0,\, 1\}$ *of the TBM conjunctive rule before their first aggregation.*

Note that, as a consequence of Lemma 4.1, for a given agent i the following relation holds between the novelty and the common knowledge with any other agent j:

$$\begin{aligned} s_i(t) &= \Big\{m_i(t,\gamma_a); \quad \forall \gamma_a \in \Gamma\Big\} = \tilde{s}_i^j(t) \otimes \bar{s}_{ij}(t) \\ &= \Big\{\tilde{m}_i^j(t,\gamma_a) \otimes \bar{m}_{i,j}(t,\gamma_a); \quad \forall \gamma_a \in \Gamma\Big\} \end{aligned} \tag{4.13}$$

Furthermore, for any couple of agents (i,j), the related states s_i and s_j are equal if and only if they are completely described by their common knowledge, i.e. $s_i = s_j = \bar{s}_{i,j}$.

Remark 1 *A few important remarks are now in order:*

- *In order to apply the local interaction rule* $\mathcal{R}$*, an agent must have stored all the most recent collaborations with its neighbors, that is* $\Big\{s_i \oplus s_j; \quad j \in \mathcal{N}(i)\Big\}$.
- *As only information concerning collaborations among (1-hop) neighbors are required, the algorithm is fully distributed and scalable in terms of memory requirements with respect to the size of the network.*

At this point, in order to prove the convergence of the proposed algorithm, some properties concerning the local interaction rule $\mathcal{R}$ must be introduced.

Lemma 4.2 ($\mathcal{R}$ properties) *The local interaction rule $\mathcal{R}$ defined according to eq. (4.19) has the following properties:*

$$\begin{array}{rcll} s_i \oplus s_j & = & s_j \oplus s_i & (\textit{commutativity}) \\ s_i \oplus s_i & = & s_i & (\textit{idempotence}) \\ (s_i \oplus s_j) \oplus s_k & = & s_i \oplus (s_j \oplus s_k) & (\textit{associativity}) \end{array} \tag{4.14}$$

for each triple $(i,\, j,\, k) : e_{ij}, e_{jk} \in \hat{E}$.

Proof: The properties can be proven by applying the definition given in eq. (4.19).

- Commutativity:

  Let us consider two agents (i, j), then from Definition 5 we have:

  $$\begin{aligned} s_i \oplus s_j &= \left(\tilde{s}_i^j \otimes \tilde{s}_j^i\right) \otimes \bar{s}_{i,j} \\ &= \left(\tilde{s}_j^i \otimes \tilde{s}_i^j\right) \otimes \bar{s}_{j,i} \\ &= s_j \oplus s_i \end{aligned}$$

  where $\left(\tilde{s}_i \otimes \tilde{s}_j\right) = \left(\tilde{s}_j \otimes \tilde{s}_i\right)$ comes from the commutativity property of the Smets operator $\otimes$ and $\bar{s}_{i,j} = \bar{s}_{j,i}$ by definition.

- Idempotence:

  Let us supposed two agents (i, j) at a given time t have their BBA equal to their common knowledge (acquired at certain time previous t), that is $s_i = s_j = \bar{s}_{i,j}$, then we have:

  $$\begin{aligned} s_i \oplus s_j &= \left(\tilde{s}_i^j \otimes \tilde{s}_j^i\right) \otimes \bar{s}_{i,j} \\ &= \left(\mathfrak{n} \otimes \mathfrak{n}\right) \otimes \bar{s}_{i,j} \\ &= \bar{s}_{i,j} \\ &= s_i \end{aligned}$$

- Associativity:

  Let us consider a triplet of agents $(i,\ j,\ k)$ such that $e_{ij}, e_{jk} \in \hat{E}$, then we have:

$$
\begin{aligned}
\Big(s_i \oplus s_j\Big) \oplus s_k &= s_{ij} \oplus s_k \\
&= \tilde{s}_{ij}^{k} \otimes \tilde{s}_{k}^{ij} \otimes \bar{s}_{ij,k} \\
&= s_{ij} \otimes \tilde{s}_{k}^{ij} \\
&= \tilde{s}_{i}^{j} \otimes s_j \otimes \tilde{s}_{k}^{ij} \\
&= \tilde{s}_{i}^{j} \otimes s_j \otimes \tilde{s}_{k}^{j} \\
&= \tilde{s}_{i}^{j} \otimes s_{jk} \\
&= \tilde{s}_{i}^{jk} \otimes s_{jk} \\
&= \tilde{s}_{i}^{jk} \otimes \tilde{s}_{jk}^{i} \otimes \bar{s}_{ij,k} \\
&= s_i \oplus s_{jk} \\
&= s_i \oplus \Big(s_j \oplus s_k\Big)
\end{aligned}
$$

  where the equivalent relations $\tilde{s}_{k}^{ij} = \tilde{s}_{k}^{j}$ and $\tilde{s}_{i}^{j} = \tilde{s}_{i}^{jk}$ come from the independence of knowledge between nodes i and k with respect to j, due to the properties of the topology structure of the communication graph, i.e., a spanning-tree $\mathcal{T}$.

□

So far, we have introduced a local interaction rule $\mathcal{R}$ and we have described its properties. In the following, it will be shown that if the agents apply the gossip algorithm given in Algorithm 2 with such a local interaction rule $\mathcal{R}$ over the spanning-tree $\mathcal{T}$, they converge toward a common BBA. In particular, it will be shown that such a BBA is the same as in the case of a centralized aggregation based on TBM conjunctive rule (68).

Definition 6 (Centralized TBM) *Let us consider a system of n agents (sources) where each agent i provides an independent set of observations at time $t = 0$ described* *by the basic belief assignment $s_i(0) = \{m_i(0, \gamma_a);\ \gamma_a \in \Gamma\}$. A centralized aggregation schema would provide the following aggregated BBA:*

$$s_{1\,2\,\dots\,n} = s_1 \otimes s_2 \otimes \dots \otimes s_n \tag{4.15}$$

Let us now introduce the concept of a time-dependent forest $\mathcal{F}(t, t+\Delta)$ with respect to a given tree $\mathcal{T}$ over time as follows:

Definition 7 *Let us define $\mathcal{F}(t, t+\Delta t) = \{V, \hat{E}(t, t+\Delta t)\}$, with $\hat{E}(t, t+\Delta t) = \bigcup_{z=t}^{t+\Delta t} \mathfrak{e}(z)$ and $\mathfrak{e}(z) \in \hat{E}$, as the forest resulting from the union of all the edges given by the edge selection process over the set $\hat{E}$ from time t to time $t+\Delta t$. Obviously, if the edge process $\mathfrak{e}$ is such that in the time interval $(t, t+\Delta t)$ the forest $\mathcal{F}(t, t+\Delta t)$ is connected, then the spanning tree $\mathcal{T}$ is obtained.*

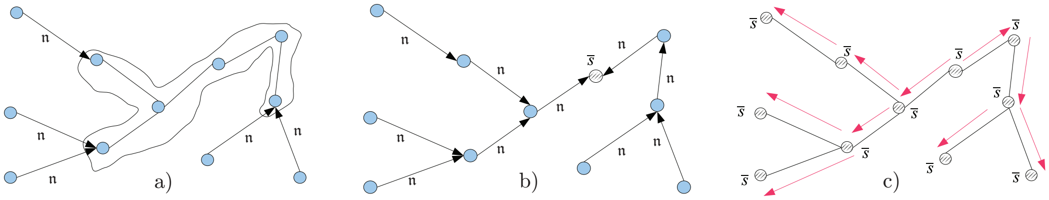


Figure 4.7: Steady-state convergence over the spanning tree $\mathcal{T}$ at different time-interval: a) after the first interval of time during which the tree $\mathcal{T}$ is obtained, the related leaves will send only the neutral element b) after a certain amount of time the root of $\mathcal{T}$ achieve the steady state $\bar{s}$ c) Finally the steady state is spread over the whole net.

In order to prove the main result of the paper, a useful relationship between the Smets operator $\otimes$ and the proposed interaction rule $\oplus$ is now introduced.

Lemma 4.3 *Let us consider three agents (i, j, k) such that $e_{ij}, e_{jk} \in \hat{E}$ with their observations $s_i(0), s_j(0), s_k(0)$ at time $t = 0$. The following holds:*

$$s_i(0) \otimes s_j(0) \otimes s_k(0) = s_i(0) \oplus s_j(0) \oplus s_k(0) \tag{4.16}$$

Proof: *The lemma can be proven by applying the definition given in eq. (4.19) along with the properties given in eq. (4.13):*

$$\begin{aligned} s_i(0) \oplus s_j(0) \oplus s_k(0) &= \Big(s_i(0) \oplus s_j(0) \Big) \oplus s_k(0) \\ &= \Big(\tilde{s}_i^j(0) \otimes \tilde{s}_j^i(0) \otimes \bar{s}_{i,j}(0) \Big) \oplus s_k(0) \\ &= \Big(s_i(0) \otimes s_j(0) \otimes \mathfrak{n} \Big) \oplus s_k(0) \end{aligned}$$

where

$$\tilde{s}_i^j(0) \otimes \tilde{s}_j^i(0) \otimes \bar{s}_{i,j}(0) = s_i(0) \otimes s_j(0) \otimes \mathfrak{n}$$

is due to the independence of the agents observation. Now, by defining $s_z(0) = s_i(0) \otimes s_j(0)$, it follows that:

$$\begin{aligned} s_z(0) \oplus s_k(0) &= \tilde{s}_z^k(0) \otimes \tilde{s}_k^z(0) \otimes \bar{s}_{z,k}(0) \\ &= s_z(0) \otimes s_k(0) \otimes \mathfrak{n} \\ &= s_i(0) \otimes s_j(0) \otimes s_k(0) \end{aligned}$$

□

Let us now introduce the main result of the paper, that is the convergence of the proposed gossip algorithm towards the basic belief assignment (BBA) as in the centralized aggregation schema given in Definition 6.

Theorem 4.1 (Distributed TBM) *Let us consider a gossip algorithm $\{\mathcal{S}, \mathcal{R}, \mathfrak{e}\}$ over a spanning-tree $\mathcal{T} = \{V, \hat{E}\}$ with $\mathcal{S}$ and $\mathcal{R}$ defined respectively as in Definition 4 and Definition 5. Let us assume each agent i at time t= 0 provides an independent set of observations described by the basic belief assignment $s_i(0) = \{m_i(0, \gamma_a);\ \gamma_a \in \Gamma\}$. If $\mathfrak{e}$ is such that $\forall\ t \quad \exists\, \Delta t \in \mathbb{N}$ so that the time-variant forest $\mathcal{F}(t, t + \Delta t)$ is connected, then there will exist a time $t = \bar{t}$ so that:*

$$s_i(t') = s_1(0) \otimes s_2(0) \otimes \ldots \otimes s_n(0) \quad \forall\, t' > \bar{t}, \tag{4.17}$$

that is, each agent i converges toward the same BBA as in the centralized aggregation schema given in Definition 6.

Proof: The proof of the theorem consists of three steps. First, it will be proven that a steady-state exists for the proposed gossip algorithm. Successively, it will be proven that such an algorithm always converges toward a steady-state. Finally, it will be proven that the steady-state is unique and it is the same as the result of the centralized aggregation schema given in eq. (4.15).

- *Steady-State Existence*

  In order to prove the existence of a steady-state for the proposed gossip algorithm, it will be shown that a sufficient and necessary condition is that all the agents share the same state $\bar{s}$. In fact, if all the agents have the same state $\bar{s}$, according to the interaction rule given in eq. (4.19), they will always send the neutral element $\mathfrak{n}$ for any further aggregation. Therefore, the state $\bar{s}$ is itself a steady state for the multi-agent system. Furthermore, let us prove by contradiction this condition to be necessary as well. To this end, let us consider a spanning-tree $\mathcal{T}$ computed by the agents in a distributed fashion. Now, let us suppose two agents i and j have reached two different steady states over the network, that is $s_i(t) = s'$ and $s_j(t) = s''$. Therefore, according to the definition of a spanning-tree, there will always exist a (unique) path connecting the two nodes i and j. Let us know consider for such a spanning-tree $\mathcal{T}$ the path $p_{ij} = \{v_i,\, v_{k \in \mathcal{N}_i},\, \ldots,\, v_{h \in \mathcal{N}_j},\, v_j\}$ connecting these two agents i and j. In particular, as agent i has reached the state s', its neighbor k will always send to it the neutral element $\mathfrak{n}$ as novelty for any further aggregation. This implies that, the agent k must have reached itself the same steady state s' and be receiving the neutral element $\mathfrak{n}$ by its neighbors. The same argument can be applied to the agent j and its neighbor h with respect to the steady state s''. Now, by iterating this reasoning from both ends of the

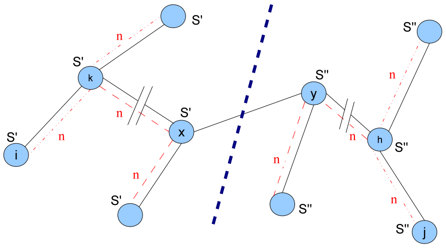


Figure 4.8: Novelty contraction over the spanning tree $\mathcal{T}$ at different time-interval.

path there will be a cut where all the nodes on a side will have reached the same steady-state s' as agent i, while on the other side all the agents will have reached the same steady-state s'' as agent j, as shown in Figure 4.8. Let us call x and y the two agents on the boundaries of the cut. Since, x and y have both reached a steady state, s' and s'' respectively, they must be sending the neutral element $\mathfrak{n}$ as novelty to each other. However, the two steady states s' and s'' have been supposed to be different, therefore the two agents x and y *cannot* be sending the neutral element $\mathfrak{n}$ to each other. Indeed, this would be possible only if the two steady states s' and s'' were the same steady state $\bar{s}$, which gives the absurd. Therefore a steady state holds if and only if $s_i = s_j \quad \forall i, j \in N$.

- *Steady-State Convergence*

In order to prove the convergence of the proposed algorithm towards a steady-state, let us consider a spanning tree $\mathcal{T}$ computed by the agents in a distributed fashion over the network topology $\mathcal{G}$, as shown in Figure 4.7. Now, let us consider an interval of time $[t_0, t_0 + \Delta t_0]$ for which the forest $\mathcal{F}(t_0, t_0 + \Delta t_0)$ is connected. This implies that some agents play the role of leaves for the resulting spanning-tree $\mathcal{T}$. According to the definition of the local interaction rule given in eq. (4.19), (at least) these agents will always send the neutral element $\mathfrak{n}$ to their fathers for any further aggregation (Figure 4.7-a). Now, let us consider a new interval of time $[t_1,\, t_1 + \Delta t_1]$ with $t_1 = t_0 + \Delta t_0 + 1$. We can use the same argument with respect to a new spanning-tree $\mathcal{T}'$ obtained by removing the leaves from the original spanning-tree $\mathcal{T}$. In fact, there are some other agents which play the role of leaves for the new spanning-tree $\mathcal{T}'$ in the time interval $[t_1,\, t_1 + \Delta t_1]$. This implies again that (at least) these agents will always send the neutral element $\mathfrak{n}$ for any further aggregation to their fathers. At this point, since the number of agents is finite, by repeating this reasoning it will exist an interval $[t_h,\, t_h + \Delta t_h]$ after which the residual spanning tree $\mathcal{T}^h$ will be composed of only one agent i, whose state $\bar{s}$ is the aggregation of all the observations available over the network (Figure 4.7-b). Let us now consider, a new spanning tree $\mathcal{T}^{h+1}$ composed of such an agent i and all of its one-hop neighbors. There will exist an interval $[t_{h+1},\, t_{h+1} + \Delta t_{h+1}]$ after which the forest $\mathcal{F}(t_{h+1}, t_{h+1} + \Delta t_{h+1})$ is connected. As a result, all the agents belonging to this spanning tree will have reached the same knowledge as the agent i. This is due to the fact, that agent i will be the only one to send a novelty different from $\mathfrak{n}$, and therefore any aggregation will let the other agents reach its state $\bar{s}$. By iterating the same reasoning, there will be an interval of time $[t_{2h},\, t_{2h} + \Delta t_{2h}]$ for which the related spanning tree $\mathcal{T}^{2h}$ will coincide with the original spanning tree $\mathcal{T}$. At this point, all the agents will have reached the same state $\bar{s}$ as the agent i (Figure 4.7-c). Therefore, according to the proof of existence, $\bar{s}$ is a steady state for the multi-agent system.

- *Steady-State Uniqueness* In order to prove the uniqueness of the steady state, it will be shown that any sequence of aggregations over the network, where each agent is considered at least once, is always the combination of the initial set of observations, that is:

  $$\bar{s}(t) = s_1(0) \oplus s_2(0) \oplus \ldots \oplus s_n(0).$$

  This can be proven by recalling the properties given in Lemma 4.5. In fact, the particular sequence of aggregations does not affect the result due to the commutativity and associativity properties, while the presence of several occurrences of the same state can be neglected due to the idempotence property.

As a result, the combination of the initial set of observations is achieved. At this point, by exploiting the result given in Lemma 4.3, the following holds:

$$\begin{aligned} \bar{s}(t) &= s_1(0) \oplus s_2(0) \oplus \ldots \oplus s_n(0) \\ &= s_1(0) \otimes s_2(0) \otimes \ldots \otimes s_n(0). \end{aligned}$$

Thus proving the theorem. Details concerning the algorithm execution are provided in Appendix 4.7. □

Let us now provide a characterization of the convergence time with respect to a given edge selection process $\mathfrak{e}$.

Lemma 4.4 (Convergence Time) *Let us consider an edge selection process $\mathfrak{e}$ such that $\forall\, t\ \exists\, \Delta t \in \mathbb{N}$, so that the forest $\mathcal{F}(t, t+\Delta t)$ is connected. If $\exists\, M \in \mathbb{N}\ :\ \Delta t < M\ \forall\, t$, then the convergence is reached by any agent at most at time $\bar{t} = d \cdot M$, where d is the diameter of the spanning tree $\mathcal{T}$.*

Proof: *The proof of the lemma follows the same arguments of the steady-state convergence proof (4.4.3.2) by assuming that an upper bound is available to the time required for the forest to be connected. In particular, for sake of simplicity and with no lack of generality let us assume to start at time $t = 0$. Under this assumption, the information contraction process towards a single agent i described in 4.4.3.2 takes in the worst case, i.e., the leaves are the last agents to perform an aggregation, time $t_1 = (d/2) \cdot M$. In the same way, the information propagation process from such an agent i to all the other agents over the network described in 4.4.3.2 takes in the worst case, i.e., one of the leaves of the previous spanning-tree is the last agent to perform an aggregation, time $t_2 = (d/2) \cdot M$. Therefore, the overall time required to the algorithm to converge in the worst case scenario is $t_{tot} = t_1 + t_2 = d \cdot M$.* □

4.4.3 Dynamic version

In this section, a local interaction rule $\mathcal{R}'$ to perform the distributed TBM data aggregation over a network topology for a multi-agent system, under assumptions 1 and time varying observations, is described.

As opposite to the static scenario, each agent collects new observations over the time, hence the following terms are introduced in the formulation:

- $p_i(t)$ BBA related to the previous observation;
- $l_i(t)$ BBA related to the latest observation;

For sake of clarity, the following notation will be used indiscriminately in the further:

$$\begin{aligned} s_c(t) &= \bar{s}_{i,j}(t) \\ s_s(t) &= s_i(t) \otimes s_j(t) \end{aligned}$$

Definition 8 (Combination rules) *In a dynamic scenario each agent can perform two type of data aggregation:*

- *Local Aggregation performed by a single agent.*
- *Dynamic Aggregation performed by a couple of agents.*

4.4.3.1 Local Aggregation

An agent requires a local aggregation any time a new observation is available. To this end, let us suppose that the initial observation is made at time $t = 0$ and a new one is collected at time $\hat{t}_k,\ k \in \mathbb{N}$. In this context, an agent must perform a *local aggregation* in order to remove the past observation from its current knowledge and add the new one. This can be achieved by first computing the novelty between the current knowledge s_i and the past observation p_i and then aggregating this novelty $\tilde{s}_i^p$ with the latest observation l_i. In this way, the updated current knowledge of the agent will take into account only the information coming from the previous collaboration with its neighbors and its last observation, that is:

$$s_i(\hat{t}_k) = \tilde{s}_i^p(\hat{t}_k) \otimes l_i(\hat{t}_k) \tag{4.18}$$

4.4.3.2 Dynamic Aggregation Between Agents

In order to perform the dynamic aggregation between two agents the operator $\vec{\oplus}$ must be introduced.

Definition 9 ($\mathcal{R}'$ - operator $\vec{\oplus}$)
Let $\mathcal{R}'$ be a rule to combine the basic belief assignments for two agents (i, j) such that $e_{ij} \in \hat{E}$ as follows:

$$\begin{aligned} s_i(t+1) &= s_i(t) \,\vec{\oplus}\, s_j(t) \\ &= \left(s_i\widetilde{(t) \otimes s_j}(t)\right)^c \\ &= \Big\{\tilde{m}_s^c(t, \gamma_a)\Big\}, \quad \forall \gamma_a \in \Gamma \end{aligned} \tag{4.19}$$

where $\tilde{m}_s^c(t, \gamma_a)$ is defined as in eq. (4.10) with respect to $s_s(t)$ and $s_c(t)$.

Remark 2 *A few remarks are now in order:*

- *A simple consequence of Lemma 4.1 with respect to the operator $\vec{\oplus}$ is that the aggregation of two state s_i and s_j such that $s_j = \bar{s}_{i,j}$ is:*

$$\begin{aligned} s_j(t+1) &= \widetilde{(s_j(t) \otimes s_i(t))}^{c} \\ &= \widetilde{(s_j(t) \otimes s_i(t))}^{j} \\ &= s_i(t) \end{aligned}$$

- *According to the operator $\vec{\oplus}$, two states $s_i(t)$ and $s_j(t)$ are equal if and only if they are completely described by their common knowledge, i.e. $s_i(t) = s_j(t) = \bar{s}_{i,j}(t) = s_c(t)$.*

Therefore, the aggregation rule used for the dynamic scenario turns out to be the same as in the static scenario if the agents do not collect new observations over time.

At this point, in order to prove the convergence of the proposed algorithm some properties concerning the local interaction rule $\mathcal{R}$ must be introduced.

Lemma 4.5 ($\vec{\oplus}$ properties) *The local interaction rule $\vec{\oplus}$ defined according to eq. (4.19) has the following properties:*

$$\begin{array}{rcll} s_i \vec{\oplus} s_j & = & s_j \vec{\oplus} s_i & (\textit{commutativity}) \\ s_i \vec{\oplus} s_i & = & s_i & (\textit{idempotence}) \\ (s_i \vec{\oplus} s_j) \vec{\oplus} s_k & = & s_i \vec{\oplus} (s_j \vec{\oplus} s_k) & (\textit{associativity}) \end{array} \tag{4.20}$$

for each triple $(i,\, j,\, k)$ such that $e_{ij}, e_{jk} \in \hat{E}$.

Proof: The properties can be proven by exploiting the Lemma 4.1 and the definition given in eq. (4.19). Furthermore, let us consider the more general case for which agents might have already performed an aggregation with each other. In particular, according to Lemma 4.1, the state s_i of an agent i with respect to any of its neighbors j can be always written as:

$$\begin{aligned} s_i(t_m) &= l_i(t_m) \otimes s_{\mathcal{N}(i)\setminus j}(t_m) \otimes l_j(t_q) \otimes s_{\mathcal{N}(j)\setminus i}(t_q) \\ &= v_i^j(t_m) \otimes v_j^i(t_q) \end{aligned}$$

where $s_{\mathcal{N}(i)\backslash j}(t_m), s_{\mathcal{N}(j)\backslash i}(t_q)$ describe respectively all the aggregated data coming from the neighborhood of agent i (at time t_m) and j (at time t_q) excluding each other, and $v_h^p(t_k) = l_h(t_k) \otimes s_{\mathcal{N}(h)\backslash p}(t_k)$.

- Commutativity:
  Let us consider two agents (i, j), then from Definition 5 we have:

$$\begin{aligned} s_i \vec{\bigoplus} s_j &= \tilde{s}_s^c \\ &= \left(\widetilde{s_i \otimes s_j}\right)^c \\ &= \left(\widetilde{s_j \otimes s_i}\right)^c = s_j \vec{\bigoplus} s_i \end{aligned}$$

  where $\left(s_i \otimes s_j\right) = \left(s_j \otimes s_i\right)$ comes from the commutativity property of the Smets operator $\otimes$.

- Idempotence:
  Let us supposed two agents (i, j) at a given time t have their BBA equal to their common knowledge (acquired at certain time previous t), that is $s_i = s_j = \bar{s}_{i,j}$, then we have:

$$\begin{aligned} s_i \vec{\bigoplus} s_j &= \left(\widetilde{s_i \otimes s_j}\right)^c \\ &= \left(\widetilde{s_i \otimes s_j}\right)^j = s_i \end{aligned}$$

- Associativity:
  Let us consider a triplet of agents $(i,\, j,\, k)$ such that $e_{ij}, e_{jk} \in \hat{E}$. Furthermore, according to Lemma 4.1, let us assume the current state of the three agents at time t_m to be written as follows:

$$\begin{aligned} s_i &= v_i^j(t_m) \otimes v_j^i(t_q) \\ s_j &= v_i^j(t_q) \otimes l_j(t_m) \otimes s_{\mathcal{N}(j)\backslash i,k}(t_m) \otimes v_k^j(t_{q'}) \\ &= v_i^j(t_q) \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_{q'}) \\ s_k &= v_k^j(t_m) \otimes v_j^k(t_{q'}) \end{aligned}$$

  and the common knowledge describing previous aggregation among these agents to be written as:

$$\begin{aligned} \bar{s}_{i,j}(t_q) &= s_{c'} = & v_i^j(t_q) \otimes v_j^i(t_q) \\ \bar{s}_{j,k}(t_{q'}) &= s_{c''} = & v_j^k(t_{q'}) \otimes v_k^j(t_{q'}). \end{aligned}$$

Then we have:

$$
\begin{aligned}
&\Big(s_i\,(t_m)\,\vec{\oplus}\,s_j(t_m)\Big)\,\vec{\oplus}\,s_k(t_m) = \\
&= \widetilde{\Big(v_i^j(t_m)\otimes v_j^i(t_q)\otimes v_i^j(t_q)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_{q'})\Big)}^{c'}\;\vec{\oplus}\,s_k(t_m) \\
&= \Big(v_i^j(t_m)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_{q'})\Big)\,\vec{\oplus}\,s_k(t_m) \\
&= s_{ij}(t_m)\,\vec{\oplus}\,s_k(t_m) \\
&= \widetilde{\Big(v_i^j(t_m)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_{q'})\otimes\; s_k(t_m)\Big)}^{c''} \\
&= \widetilde{\Big(v_i^j(t_m)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_{q'})\otimes v_k^j(t_m)\otimes v_j^k(t_{q'})\Big)}^{c''} \\
&= v_i^j(t_m)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_m)
\end{aligned}
$$

where the equivalence $\bar{s}_{ij,k} = \bar{s}_{j,k}$ comes from the independence of the knowledge between node i and k with respect to j, due to the properties of the topology structure of the communication graph, i.e., a spanning tree $\mathcal{T}$. And:

$$
\begin{aligned}
&s_i\,(t_m)\,\vec{\oplus}\,\Big(s_j(t_m)\,\vec{\oplus}\,s_k(t_m)\Big) = \\
&= s_i(t_m)\,\vec{\oplus}\widetilde{\Big(v_i^j(t_q)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_{q'})\otimes v_k^j(t_m)\otimes v_j^k(t_{q'})\Big)}^{c''} \\
&= s_i(t_m)\vec{\oplus}\Big(v_i^j(t_q)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_m)\Big) \\
&= s_i(t_m)\vec{\oplus}s_{jk}(t_m) \\
&= \widetilde{\Big(s_i(t_m)\otimes v_i^j(t_q)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_m)\Big)}^{c'} \\
&= \widetilde{\Big(v_i^j(t_m)\otimes v_j^i(t_q)\otimes v_i^j(t_q)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_m)\Big)}^{c'} \\
&= v_i^j(t_m)\otimes v_j^{i,k}(t_m)\otimes v_k^j(t_m)
\end{aligned}
$$

where the equivalence $\bar{s}_{i,jk} = \bar{s}_{i,j}$ comes again from the independence of the knowledge between node i and k with respect to j, due to the properties of the topology structure of the communication graph, i.e., a spanning tree $\mathcal{T}$.

□

In a dynamic scenario, referring to Def. (6), let us assume that every so often ($t = \hat{t}_k,\ k \in \mathbb{N}$), one or more agents perform an update of their observation. As a consequence, the aggregated BBA can be updated accordingly as follows:

$$s_{1\,2\,\ldots\,n}(\hat{t}_k) = l_1(\hat{t}_k) \otimes l_2(\hat{t}_k) \otimes \ldots \otimes l_n(\hat{t}_k) \tag{4.21}$$

where $l_i(\hat{t}_k)$ describes the BBA related to the most recent observation available to the agent i.

Under the dynamic scenario assumption, the convergence of the proposed gossip algorithm towards the basic belief assignment as in the centralized aggregation schema is guaranteed by the following theorem:

Theorem 4.2 (Distributed Dynamic TBM) *Let us consider a gossip algorithm* $\{\mathcal{S}, \mathcal{R}', \mathfrak{e}\}$ *over a spanning-tree* $\mathcal{T} = \{V, \hat{E}\}$ *with* $\mathcal{S}$ *and* $\mathcal{R}'$ *defined respectively as in Definition 4 and Definition 9. Let us assume each agent i at time t= 0 provides an independent observation described by the basic belief assignment* $s_i(0) = \{m_i(0, \gamma_a\},\ \gamma_a \in \Gamma\}$. *Furthermore, let us assume that every so often* ($t = \hat{t}_k,\ k \in \mathbb{N}$)*, one or more agents perform an update of their observation. If* $\mathfrak{e}$ *is such that* $\forall\ t \quad \exists\, \Delta t \in \mathbb{N}$ *so that the time-variant forest* $\mathcal{F}(t, t + \Delta t)$ *is connected, then for some k there will exist a time* $t = \bar{t}_k$ *so that:*

$$\begin{array}{rcl} s_i(t') & = & l_1(\hat{t}_k) \otimes l_2(\hat{t}_k) \otimes \ldots \otimes l_n(\hat{t}_k) \\ & & \forall\, t' \in [\,\bar{t}_k,\ \hat{t}_{k+1}\,) \end{array} \tag{4.22}$$

where $l_i(\hat{t}_k)$ *describes the most recent observation available to the agent i.*

Proof: The proof of the theorem consists of four steps. First, it will be proven that a steady-state exists for the proposed gossip algorithm. Successively, it will be shown that each agent by applying the aggregation operator $\vec{\oplus}$ can inject the updated observations into the network while transparently removing the previous ones. Then, by exploiting this property, it will be proven the convergence and finally, it will be shown that the steady-state is unique and equal to the result of the centralized aggregation schema given in eq (4.21).

Note that, for the existence, convergence and uniqueness analysis, the interval of time $[\hat{t}_k,\, \hat{t}_{k+1})$ between two consecutive observations update is supposed to be long enough with respect to the nature of the edge selection process $\mathfrak{e}$. This allows to guarantee that each agent can perform all the aggregations required to reach the steady-state. Necessary and sufficient conditions concerning the length of time interval with respect to the edge selection process $\mathfrak{e}$ are given in Lemma 4.6.

- *Steady-State Existence*:

  In order to prove the existence of a steady-state for the proposed gossip algorithm, it will be shown that a sufficient and necessary condition is that all the agents share the same state $\bar{s}$. According to the interaction rule given in eq. (4.19), this imply that all the agents will have the same common knowledge s_c. Therefore, the state $\bar{s}$ is itself a steady state for the multi-agent system. Furthermore, let us prove by contradiction this condition to be necessary as well. To this end, let us suppose two agents i and j have reached two different steady states over the network, that is $s_i(t) = s'$ and $s_j(t) = s''$. Therefore, according to the definition of a spanning-tree, there will always exist a (unique) path connecting the two nodes i and j. Let us now consider for such a spanning-tree $\mathcal{T}$ the path $p_{ij} = \{v_i, v_{k\in\mathcal{N}_i}, \ldots, v_{h\in\mathcal{N}_j}, v_j\}$ connecting these two agents i and j. In particular, as agent i has reached the state s', this implies that its neighbor k must be sending a state which is equal to their common knowledge. Furthermore, since agent k itself has reached a steady-state, agent i must be sending a state which is equal to their common knowledge. However, according to the Remark 2 this implies that both agents have the same state, that is s'. The same argument can be applied to the agent j and its neighbor h with respect to the steady state s''.

  Now, by iterating this reasoning from both ends of the path there will be a cut where all the nodes on a side will have reached the same steady-state s' as agent i, while on the other side all the agents will have reached the same steady-state s'' as agent j. Let us call x and y the two agents on the boundaries of the cut. Since, x and y have boot reached a steady state, s' and s'' respectively, they must be sending a state which is equal to their common knowledge to each other. However, the two steady states s' and s'' have been supposed to be different, so the two agents x and y *cannot* be sending a state equal to their common knowledge s_c to each other. Indeed, this would be possible only if the two steady states s' and s'' were the same steady state $\bar{s}$, which gives the absurd. Therefore a steady state holds if and only if $s_i = s_j \quad \forall i, j \in N$.

- *Observations Propagation*:

  Let us consider two agents (i, j) such that $e_{ij} \in \hat{E}$ and let us assume a collaboration was performed at time $t = t_q$. The updated states can be written as:

$$\begin{aligned} s_i(t_q) &= l_i(t_q) \otimes s_{\mathcal{N}(i)\backslash j}(t_q) \otimes l_j(t_q) \otimes s_{\mathcal{N}(j)\backslash i}(t_q) \\ &= v_i^j(t_q) \otimes v_j^i(t_q) \\ s_j(t_q) &= l_j(t_q) \otimes s_{\mathcal{N}(j)\backslash i}(t_q) \otimes l_i(t_q) \otimes s_{\mathcal{N}(i)\backslash q}(t_q) \\ &= v_j^i(t_q) \otimes v_i^j(t_q), \end{aligned}$$

  where $s_{\mathcal{N}(i)\backslash j}(t_q)$, $s_{\mathcal{N}(j)\backslash i}(t_q)$ describe respectively all the aggregated data coming from the neighborhood of agent i and j excluding each other, and $v_h^p(t_k) = l_h(t_k) \otimes s_{\mathcal{N}(h)\backslash p}(t_k)$. Furthermore the common knowledge at time $t = t_q$ between the two agents can be written as $s_c(t_q) = v_j^i(t_q) \otimes v_i^j(t_q)$, Now, let us consider a time $t = t_m,\ t_m$ such that the two agents have performed further aggregations (but not with each other) and an update of their observation. Their current state at time $t = t_m$ can be written as follows:

$$\begin{aligned} s_i(t_m) &= l_i(t_m) \otimes s_{\mathcal{N}(i)\backslash j}(t_m) \otimes l_j(t_q) \otimes s_{\mathcal{N}(j)\backslash i}(t_q) \\ &= v_i^j(t_m) \otimes v_j^i(t_q) \\ s_j(t_m) &= l_j(t_m) \otimes s_{\mathcal{N}(j)\backslash i}(t_m) \otimes l_i(t_q) \otimes s_{\mathcal{N}(i)\backslash q}(t_q) \\ &= v_j^i(t_m) \otimes v_i^j(t_q). \end{aligned}$$

  Now, let us assume that the two agents perform an aggregation at time $t = t_m + 1$. Their current state can be updated accordingly as follows:

$$\begin{aligned} s_i(t_m + 1) &= s_i(t_m) \vec{\otimes} s_j(t_m) \\ &= \Big(v_i^j(t_m) \otimes v_j^i(t_q) \widetilde{\otimes} v_j^i(t_m) \otimes v_i^j(t_q)\Big)^{c} \\ &= v_i^j(t_m) \otimes v_j^i(t_m) \\ s_j(t_m + 1) &= s_j(t_m) \vec{\otimes} s_i(t_m) \\ &= \Big(v_j^i(t_m) \otimes v_i^j(t_q) \widetilde{\otimes} v_i^j(t_m) \otimes v_j^i(t_q)\Big)^{c} \\ &= v_j^i(t_m) \otimes v_i^j(t_m) \end{aligned}$$

  where the common knowledge $s_c(t_q) = v_j^i(t_q) \otimes v_i^j(t_q)$, which represents both their previous observations and their neighbors previous observations, is removed. Note that, after the aggregation the common knowledge is set to the current state of the two agents $s_c(t_m + 1) = v_i^j(t_m) \otimes v_j^i(t_m)$. Therefore, anytime two agents perform an aggregation only the most recent observation of any agent is propagated over the network.

- *Steady-State Convergence*:

  In order to prove the convergence of the proposed algorithm towards a steady-state, let us consider an interval of time $[t_0, t_0 + \Delta t_0]$, with $t_0 = \hat{t}_k$, for which the forest $\mathcal{F}(t_0, t_0 + \Delta t_0)$ is connected. This implies that some agents play the role of leaves for the resulting spanning-tree $\mathcal{T}$. Indeed, any further aggregation of these agents with their fathers will not change the state of the fathers. This is due to the fact that all the knowledge brought by the leaves is already available to the fathers in their common knowledge (see Remark 2). Furthermore, according to the Observation Propagation proof (4.4.3.2) only the most recent observations will be sent by the leaves to the fathers. Now, let us consider a new interval of time $[t_1,\, t_1 + \Delta t_1]$ with $t_1 = t_0 + \Delta t_0 + 1$. We can use the same argument with respect to a new spanning-tree $\mathcal{T}'$ obtained by removing the leaves from the original spanning-tree $\mathcal{T}$. In fact, there are some other agents which play the role of leaves for the new spanning-tree $\mathcal{T}'$ in the time interval $[t_1,\, t_1 + \Delta t_1]$. This implies again that (at least) these agents will always send the common knowledge s'_c for any further aggregation to their fathers. At this point, since the number of agents is finite, by repeating this reasoning it will exist an interval $[t_h,\, t_h + \Delta t_h]$ after which the residual spanning tree $\mathcal{T}^h$ will be composed of only one agent i, whose state $\bar{s}$ is the aggregation of all the *most recent observations* available (at time $\hat{t}_k$) over the network. Let us now consider, a new spanning tree $\mathcal{T}^{h+1}$ composed of such an agent i and all of its one-hop neighbors. There will exist an interval $[t_{h+1},\, t_{h+1} + \Delta t_{h+1}]$ after which the forest $\mathcal{F}(t_{h+1}, t_{h+1} + \Delta t_{h+1})$ is connected. As a result, all the agents belonging to this spanning tree will have reached the same knowledge as the agent i. This is due to the fact that for any aggregation, agent i will be the only to have its state different from the common knowledge. Therefore according to the Remark (2), the other agents will reach its state $\bar{s}$. By iterating the same reasoning, there will be an interval of time $[t_{2h},\, t_{2h} + \Delta t_{2h}]$ for which the related spanning tree $\mathcal{T}^{2h}$ will coincide with the original spanning tree $\mathcal{T}$. This implies that all the agents will be reached the same state $\bar{s}$ as the agent i. Therefore, according to the proof of existence, $\bar{s}$ is a steady state for the multi-agent system.

- *Steady-State Uniqueness* In order to prove the uniqueness of the steady state, it will be shown that any sequence of aggregations over the network, where each agent is considered at least once, is always the $\vec{\oplus}$ combination of the observations set at time $t = \hat{t}_k$, that is:

  $$\bar{s}(t') = s_1(\hat{t}_k) \vec{\oplus} s_2(\hat{t}_k) \vec{\oplus} \ldots \vec{\oplus} s_n(\hat{t}_k), \quad \forall t' \in [\bar{t}_k, \hat{t}_{k+1}).$$

  This can be proven by recalling the properties given in Lemma 4.5 and the result concerning the proof of the Observation Propagation (4.4.3.2) along with the proof of Steady-State Convergence (4.4.3.2). In fact, the observation propagation result guarantees that, anytime two agents perform an aggregation, only the most recent observation of any agent is propagated over the network.

Furthermore, according to the convergence proof, when a steady-state is reached over the network, it embodies all the most recent observations available up to time t_k. Finally, due to Lemma 4.5, the particular sequence of aggregations does not affect the result due to the commutativity and associativity properties, while the presence of several occurrences of the same state can be neglected due to the idempotence property. As a result, the combination of the observations set at time $t = \hat{t}_k$ is achieved, that is:

$$\bar{s}(t) = l_1(\hat{t}_k) \vec{\oplus} l_2(\hat{t}_k) \vec{\oplus} \ldots \vec{\oplus} l_n(\hat{t}_k),$$

that is the same result as in the centralized aggregation schema given in eq. (4.21). □

Remark 3 *A few important remarks are now in order:*

- *The gossip algorithm described in the dynamic case allows the agents to "track" the steady-state (given by eq (4.21)). In fact, by applying the local aggregation rule given in eq (4.18) each agent can replace the previous observation with the latest one on its own state, and by applying the dynamic aggregation rule given in eq (4.19) if two agents perform an aggregation only the most recent observation of any agent is propagated over the network.*

- *The convergence capability of the proposed algorithm depends on whether the time interval between two consecutive observations update is sufficiently long with respect to the nature of the edge selection process $\mathfrak{e}$. However, even if some steady-states are missed, the agents still keep tracking the most recent one.*

In the following, an analysis to derive an upper-bound of the convergence time for the worst-case scenario is proposed.

Lemma 4.6 *Let us consider an edge selection process $\mathfrak{e}$ such that $\forall\, t \ \exists\, \Delta t \in \mathbb{N}$ so that the forest $\mathcal{F}(t, t+\Delta t)$ is connected. If $\exists\, M \in \mathbb{N} \ : \ \Delta t < M \ \forall\, t$, then the multi-agent system can always reach the convergence towards a steady-state if the following condition holds between two consecutive observations update:*

$$\hat{t}_{k+1} \geq \hat{t}_k + d \cdot M, \quad \forall\, k \in \mathbb{N} \tag{4.23}$$

where d is the diameter of the spanning tree $\mathcal{T}$.

***P**roof: The proof follows the same argument of the steady-state convergence proof (4.4.3.2) by assuming that an upper bound is available to the time required for the forest to be connected. Furthermore, let us assume that at time $t = \hat{t}_k$, one or more agents have performed an observation update over the network. Under this assumption, the information contraction process towards a single agent i described in 4.4.3.2 takes in the worst case, i.e., the leaves are the last agents to perform an aggregation, time $t_1 = (d/2) \cdot M$. In particular, the state of such an agent i represents the aggregation of the latest set of observation available over the network up to time t_k. In the same way, the information propagation process from such an agent i to all the other agents over the network described in 4.4.3.2 takes in the worst case, i.e., one of the leaves of the previous spanning-tree is the last agent to perform an aggregation, time $t_2 = (d/2) \cdot M$. Therefore, the overall time required to the algorithm to converge in the worst case scenario is $t_{tot} = t_1 + t_2 = d \cdot M$. Note that, in the case an update is performed by any agent before the contraction process ends, i.e., $\hat{t}_{k+1} < \hat{t}_k + d \cdot M$, the state spread by agent i will no longer represent the aggregation of the most recent set of observations available over the network, and therefore at the end of the propagation process, no steady-state will be reached for the interval $[\hat{t}_k, t_{k+1})$.* □

Note that Lemma 4.6 provides only a theoretical characterization of the convergence time for the proposed gossip algorithm. However, in a real scenario agents perform the update of their observations independently and asynchronously, therefore no control can be provided for the convergence of the algorithm apart from the design of a “smart” edge selection process able to keep the upper-bound M as small as possible.

4.5 An experimental case: the topological map building

Map-building addresses the problem of acquiring spatial models of physical environments by mobile robots (73). The map-building problem is generally considered as one of the most important problems in the pursuit of building truly autonomous mobile robots. Two different approaches for modeling an indoor environment have been proposed: the metric approach and the topological approach (74). Metric maps capture the geometric properties of the environment, whereas topological maps describe the connectivity of different places. These approaches exhibit orthogonal strengths and weaknesses (75). On the one hand, topological maps are computationally efficient, easy to maintain even in large scale environments while metric maps suffer from their enormous space and time complexity. On the other hand, metric maps provide a very detailed description of the environment while topological maps offer a limited representation of the surrounding world.

The majority of the approaches available in the literature deals with the simultaneous localization and mapping problem (SLAM) consisting in both building the map of the environment and localizing the robot that is moving within it (76). In (77) an approach to build a topological map based on the concept of Voronoi random fields is introduced. The idea is to extract a Voronoi graph from an occupancy grid map generated with a laser range-finder, and then represent each point on the Voronoi graph as a node of a conditional random field. The resulting Voronoi random field estimates the label of each node, integrating features from both the map and the Voronoi topology. Several works have been proposed for the SLAM problem in a multi-robot scenario as well. In (78) a platoon of four robots performs on-line the map building with a particle filter algorithm. In (69) an alternative approach for less effective sensors, e.g., sonar range-finders, is proposed. The idea is to build a grid-map representation of the environment modeling uncertainty by means of the fuzzy theory. In (79), an improvement of this work is presented. In particular, the grid-maps are exploited to extract a knowledge of the surrounding environment along which the robot travels. The uncertainties are managed through the Possibility Theory (80).

In this section a collaborative topological map-building approach for a team of robots moving in an indoor office-like environment is proposed. Each robot, after building a local map by infrared range-finders, builds a set of hypotheses about the topological nature of the surrounding environment by comparing the features extracted using the Hough transform with a set of predefined environmental patterns. The local view of each robot, which is significantly constrained by its limited sensing capabilities, is then strengthened by a collaborative aggregation schema based on the Transferable Belief Model. In this way, a better representation of the environment is achieved by each robot by means of a minimal exchange of information.

4.5.1 Problem Setting

In the proposed framework, a team of robots which explores an unknown office-like environment is considered. Robots are equipped with a sensorial system composed of an array of infrared range-finders along with an analog compass which allows the team to share a common heading direction. Therefore, a wireless channel is available for communication purposes.

The team of robots is assumed to move in a rigid formation. This assumption has been made in order to require a module for mutual localization: in fact it should be sufficient to know the inter robot configuration in order to perform this algorithm. For details about mutual localization techniques, the reader is referred to (46). Indeed, this can be achieved by exploiting one of the several control laws available in literature, e.g., (81, 82). In addition, robots are assumed to be aware of the sensing occlusions due to the other robots. Note that, this is not a strong limitation as robots are assumed to move in a rigid formation. In particular, for each couple of robots an angular section with respect to their line of sight is considered as occluded.This information can be taken into account when building the set of hypotheses to describe the surrounding environment. Figure 4.9 depicts the adopted geometrical model of occlusion for a team of three robots.

The office-like environment in which the team of robots moves is approximated by the union of a set of environmental patterns detailed in Subsection 4.5.2. In particular, the following patterns are taken into account in this framework: L-turn, corridor, dead-end, T-junction, and crossing.

The objective of this work is to develop a collaborative technique to let the team of robots achieve a proper topological description of the surrounding environment. The key idea is to provide an effective collaborative framework to make up for the limited sensorial capabilities of each single robot.

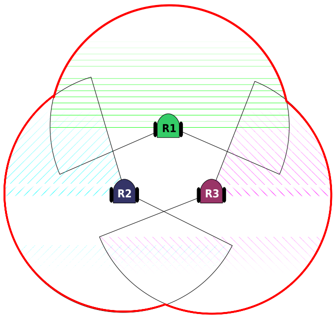


Figure 4.9: Geometric model of occlusion for the multi-robot system.

In the following the collaborative topological multi-robot map-building technique is described. First, the patterns adopted to represent the environment are described in Subsection 4.5.2. Then, the feature extraction process to build an actual view of the surrounding environment is explained in Subsection 4.5.3. Successively, the topological map building is explained in Subsection 4.5.4. Finally, the collaborative approach based on the TBM conjunctive rule to improve the local view of each robot is described in Subsection 4.5.4.1.

4.5.2 Environmental Patterns

In order to have a meaningful representation of an office-like area, environmental patterns are introduced. Indeed, these models constitute, in our context, a valid approximation into which classify more complex environments. In the proposed scenario, maps are composed of a combination of the following elements: *L-turn* (L), *corridor* (O), *dead-end* (D), *T-junction* (T), *crossing* (X).

In order to derive a mathematical description of these patterns some preliminary concepts must be introduced.

Let us first introduce the set $A = \{a_1, \dots, a_k\}$ of *atomic elements* as the set of basic features that a robot can detect. In this work, walls (W) and corners (C) are considered. An atomic element a_i is described by means of a simple parametrization with respect to the reference frame of the detecting robot. In detail, a wall is described by the angular coefficient θ of the detected segment, while a corner is represented by the angular coefficients θ_1 and θ_2 of the oriented segments connecting the two end-points to the vertex. Note that, two different kinds of corners are considered: convex and concave. According to the situation shown in Figure 4.10, a corner is said to be convex

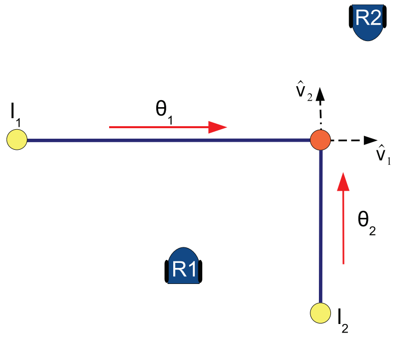


Figure 4.10: Corner detection. Robot R1 detects a convex corner while robot R2 detects a concave one.

if the robot r belongs to the third quadrant of the reference frame $\hat{O}(\hat{v}_1, \hat{v}_2)$ attached to the vertex v, concave otherwise.

Let us now introduce the concept of *relational features*. A relational feature f_i is a couple of atomic elements for which a set of geometric relationships hold. From a mathematical perspective, a relational feature f_i can be defined as follows:

$$f_i = \{a_j, a_k\} \; : \; \mathcal{M}(a_j, a_k) = \text{true}$$

where $\mathcal{M}(\cdot) : \mathcal{F} \times \mathcal{F} \rightarrow \{\text{false}, \text{true}\}$ is a boolean map describing a given set of geometric relationships. Note that, among all the possible relationships also the identity map, i.e., $\mathcal{M}(a_i, a_i) = \text{true}$, is considered. Indeed, this allows to define a relational feature even if only a single atomic element is recognized. The idea of relational feature turns out to be very useful as it allows to provide a graph-like representation of an environmental pattern and therefore a simple decomposition of it.

It is now possible to provide a formal characterization of the set $\mathcal{P} = \{p_1, \dots p_n\}$ of environmental patterns. In particular, an environmental pattern p_q can be described by a set $F_q = \{f_1, \dots, f_h\}$ of relational features. Note that, in order to be a valid representation, the set F_q must satisfy the following property:

$$F_q \nsubseteq F_k \wedge F_q \nsupseteq F_k, \; \forall p_k \in \mathcal{P} \backslash \{p_q\}, \tag{4.24}$$

which guarantees the pattern p_q to be *fully* described.

This formalization leads, as previously mentioned, to an intuitive graph-like representation of an environmental pattern where links represent the geometric relationships

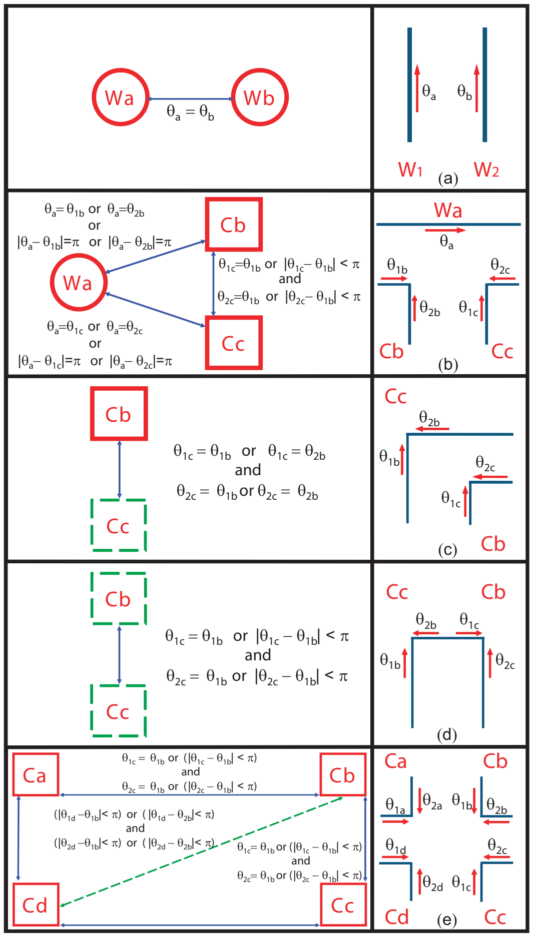

Wa
Wb
$\theta_a = \theta_b$
θ_a
θ_b
W1
W2
(a)
$\theta_a = \theta_{1b}$ or $\theta_a = \theta_{2b}$
or
$|\theta_a - \theta_{1b}| = \pi$ or $|\theta_a - \theta_{2b}| = \pi$
Cb
Wa
$\theta_{1c} = \theta_{1b}$ or $|\theta_{1c} - \theta_{1b}| < \pi$
and
$\theta_{2c} = \theta_{1b}$ or $|\theta_{2c} - \theta_{1b}| < \pi$
$\theta_a = \theta_{1c}$ or $\theta_a = \theta_{2c}$
or
$|\theta_a - \theta_{1c}| = \pi$ or $|\theta_a - \theta_{2c}| = \pi$
Cc
Wa
θ_a
θ_{1b}
θ_{2c}
θ_{2b}
θ_{1c}
Cb
Cc
(b)
Cb
$\theta_{1c} = \theta_{1b}$ or $\theta_{1c} = \theta_{2b}$
and
$\theta_{2c} = \theta_{1b}$ or $\theta_{2c} = \theta_{2b}$
Cc
Cc
θ_{2b}
θ_{1b}
θ_{2c}
θ_{1c}
(c)
Cb
Cb
$\theta_{1c} = \theta_{1b}$ or $|\theta_{1c} - \theta_{1b}| < \pi$
and
$\theta_{2c} = \theta_{1b}$ or $|\theta_{2c} - \theta_{1b}| < \pi$
Cc
Cc
Cb
θ_{2b}
θ_{1c}
θ_{1b}
θ_{2c}
(d)
Ca
$\theta_{1c} = \theta_{1b}$ or $(|\theta_{1c} - \theta_{1b}| < \pi)$
and
$\theta_{2c} = \theta_{1b}$ or $(|\theta_{2c} - \theta_{1b}| < \pi)$
Cb
$(|\theta_{1d} - \theta_{1b}| < \pi)$ or $(|\theta_{1d} - \theta_{2b}| < \pi)$
and
$(|\theta_{2d} - \theta_{1b}| < \pi)$ or $(|\theta_{2d} - \theta_{2b}| < \pi)$
$\theta_{1c} = \theta_{1b}$ or $(|\theta_{1c} - \theta_{1b}| < \pi)$
and
$\theta_{2c} = \theta_{1b}$ or $(|\theta_{2c} - \theta_{1b}| < \pi)$
Cd
Cc
Ca
Cb
θ_{1a}
θ_{2a}
θ_{1b}
θ_{2b}
θ_{1d}
θ_{2c}
θ_{2d}
θ_{1c}
Cd
(e)
Cc

Figure 4.11: Graph-based representation of an environmental pattern. Nodes represent atomic elements linked to each other by *relational edges*. Dashed squares are convex corners, solid squares are concave corners and solid circles are walls. a) corridor b) T-junction c) L-turn d) dead-end e) crossing.

existing among each pairs of atomic elements. Therefore, according to this alternative representation, an environmental pattern can be also thought as a fully connected graph of atomic features. Figure 4.11 gives a graphical overview of such a representation for the set of environmental patterns adopted in this work. In particular, a corridor can be viewed like a two *wall-nodes* connected by an edge for which the relationship of equal orientation holds. A T-junction graph is instead represented by two *concave-corner-nodes* and a *wall-node*. The *corner-nodes* share a common bearing for a couple of segments while the other one has a phase displacement of π. In addition, each corner shares a common bearing with *wall-node*. Note that, in order to avoid a wrong association between atomic elements, the relationship of parallelism is constrained by a minimal distance between the considered segments. A similar description can be easily derived for the remaining patterns.

4.5.3 Local Topological Features Extraction

Each robot while moving builds a local map of the surrounding environment by exploiting infrared range-finders. The expressiveness of the obtained map is highly constrained by the limited amount of data which can be collected at each time, i.e., only an array of 5 infrared sensors arranged over the 180° with respect to the heading direction of the robot is available. Figure 4.12 depicts an example of local map built exploiting the raw data collected by the robot while rotating over 360°. In particular, it can be noticed the effect of the noise affecting the measurements due to the intrinsic characteristics of the sensor. The local map is used to extract some features of interest by means of a Hough transform (83).

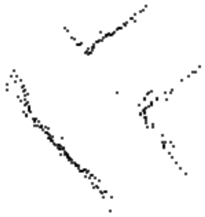


Figure 4.12: Example of local map built by letting the robot rotate over 360°.

4.5.4 Local Topological Map Building

In order to achieve the topological description of the environment, the framework of the "Theory of Evidence" introduced by Shafer in (57) is exploited. It gives an effective mathematical model for the representation of uncertainty. Hence, it turns out to be very suitable in a robotics context when dealing with noisy measurements coming from low-cost sensors. In this framework, an environmental pattern represents the proposition of a set Ω called *frame of discernment*, while the set of all propositions of interest corresponds to the elements of the power-set Γ. In addition, it can be defined a function $m : \Gamma \rightarrow 1$ called Basic Belief Assignment (BBA) which associates to each element

$\gamma \in \Gamma$ a belief mass. This mass $m(\gamma)$ describes the proportion of all relevant and available evidence that supports the claim that the actual "state" belongs to γ but to no particular subset of it. This framework suits very well the multi-robot topological mapping problem. In fact, elements of the power-set can be used to model the subset of patterns which fits the limited set of features that can be extracted from the partial view of a single robot. For example, if a corner is detected by a robot during the feature extraction process the set given by the union of all the environmental patterns which contain a corner is considered. Note that, due to the limited sensing capability, the whole surrounding environment will be hardly recognized by a single robot. For this reason an aggregation among the knowledge acquired by the team must be introduced. In this ways, ambiguities can be reduced and the set of plausible patterns can be restricted. Conditions under which the correct pattern can be detected will be discussed in Theorem 4.3.

As far as the construction of the set of masses $M = \{m(\gamma_1), \ldots, m(\gamma_{|\Gamma|})\}$ is concerned, let us assume each robot $\mathcal{A}_i$ can extract from its local map a set of relational features $s_i = \{f_1, \ldots, f_d\}$ every T seconds. In addition, let us define $\alpha_i = p_i \cap s_i$, with $\alpha \in [0,\, 1]$, as the "similarity" between the set of features s and an environmental pattern p_i. At this point, the set of masses, which must sum to one by definition, is computed as follows:

- a fraction η, is assigned to the union A of all patterns compatible with the features acquired that is:

$$m(A) = \eta, \quad A = \bigcup_{p_i \cap s \neq \emptyset} p_i \tag{4.25}$$

- the remaining $1-\eta$ of the mass is assigned proportionally (w.r.t. to the similarity) to each subset B_j having cardinality minus one with respect to A, except for the atomic patterns:

$$\begin{array}{rcl} m(B_j) & = & (1-\eta)\cdot\chi_j, \\ & \forall & B = \bigcup\limits_{p_i \cap s \neq \emptyset} p_i \quad s.t. \quad |B| = |A| - 1 > 1 \end{array} \tag{4.26}$$

with

$$\chi_j = \frac{\sum_{p_k \in B_j} \alpha_k}{\sum_{\substack{B_j \in A \\ |B_j| = |A|-1}} \sum_{p_k \in B_j} \alpha_k}.$$

Note that, the condition $|A| \geq 2$ is required in order to have an assignment to be self-consistent, i.e., a combination of a set M with itself should not introduce contradiction.

4.5.4.1 Topological Maps Aggregation

The Transferable Belief Model (TBM) introduced by Smets (68) is exploited for the aggregation of the topological description of the environment built by each robot. TBM allows to combine evidence from different sources and arrive at a degree of belief that takes into account all the available evidences.

For a couple or robots (i, j), the TBM conjunction rule can be defined as follows:

$$(m_i \otimes m_j)(\gamma_a) = \sum_{\substack{\gamma_b,\, \gamma_c \\ \gamma_b \cap \gamma_c = \gamma_a}} m_i(\gamma_b) m_j(\gamma_c). \tag{4.27}$$

From a computational perspective, the aggregation can be performed in two different ways. A simple approach is to let robots broadcast the acquired topological description and then each robot performs locally the aggregation. A more refined approach is to devise a local interaction rule which provides the same result as the previous one avoiding the overhead of the broadcast (5). The first approach is simpler and reasonable for a team of few robots, while the second approach is more suitable for a large team of robots.

Note that the proposed framework turns out to be exact in the ideal case of measurements without noise. The following theorem provides a mathematical characterization of this correctness.

Theorem 4.3 *Let us consider an environmental pattern $p_q \in \mathcal{P}$ described by a set $F_q = \{f_1, \ldots, f_h\}$ of relational features. Let us assume each robot to build a set of masses according to the rules given in eq. (4.25) and eq. (4.26). Finally, let us assume the set of masses to be aggregated according to the combination rule given in eq. (4.27). A sufficient condition for the recognition of the environmental pattern p_q is that :*

$$\bigcup_{i}^{N} s_i = F_q,$$

where N is the number of robots and s_i is the set of relational features computed by the i-th robot.

Proof: In order to prove the theorem, let us consider without any lack of generality a partition over F_q such that every subset of F_q is correctly identified by one robot. Let A_i be the union of all the patterns compatible with the observed subset of features related to the i-th robot. Now, let us consider two robots j and k performing an aggregation of their masses M_j and M_k according to eq. (4.27). The elements $m(\gamma)$ of the resulting set $M_{j,k}$ are obtained as follows:

$$m(\gamma) = \begin{cases} m_j(A_j) \cdot m_k(A_k) & \text{if} \quad \gamma = A_j \cap A_k \\ \geq 0 & \text{if} \quad \gamma \cap (A_j \cap A_k) \neq \emptyset \\ 0 & \text{if} \quad \text{otherwise} \end{cases}$$

Therefore, only the elements $\{\gamma_l\}$ of the power-set Γ representing the subset of environmental patterns supported by the intersection of the two sets of relational features A_j and A_k are assigned with a mass greater than 0, while the remaining content of information accumulates in the mass of the element $\gamma = \emptyset$ emphasizing the contradiction between sources. At this point, by iterating this aggregation process, the obtained set of masses $M_{j,k}$ will be necessarily aggregated with the set M_r provided by one of the other robots. Now, by recalling from eq. (4.24) that an environmental pattern is fully described by the union of its own relational features and by assuming measurements to be perfect, the following holds:

$$\bigcap_{i}^{N} A_i = p_q.$$

Therefore, after all the set of masses are aggregated, only the element $\gamma = p_q$ will have a mass greater than 0. Hence, the correct environmental pattern can be identified. Furthermore, the mass associated to $\gamma = \emptyset$ can be thought as the amount of contradiction due to the initial occlusions experienced by each robot. □

Although a proper mathematical characterization has been derived under the assumption of ideal measurements availability, in the real world data coming from sensors is always affected by noise. For this reason a simple but effective workaround to deal with the eventuality that a robot might perform an incorrect feature extraction is proposed. The idea is to assign a fraction ζ to the *universe set*, e.g. $m(\mathcal{P}) = \zeta$. This trick allows to partially recover from a bad feature extraction process. In this way also non common hypotheses can survive. A viable solution, making the assumption that a sort of index measuring the quality of the feature extraction is available, is to assign ζ proportionally to the goodness of the feature extraction process. Finally, the remaining part after assigning η and ζ could be assigned as explained before.

4.5.4.2 Computational Complexity and Implementation Details

From a computational perspective the proposed collaborative technique for topological map building cannot be implemented as it is. Indeed, a few tricks are required to let the technique be computationally affordable for a team of low-cost robots.

Strictly speaking, the major bottleneck is related to the construction of the powerset Γ. Indeed, given a frame of discernment Ω with cardinality $|\Omega|$ the related powerset will have a cardinality equals to $2^{|\Omega|}$, letting the formalization become intractable very quickly. In order to overcome this limitation, the problem formulation has been equivalently split in two parts. In detail, while the original formalization considers a frame of discernment where environmental patterns own an orientation with respect to the robot moving direction, in the algorithmic solution environmental patterns are not oriented and the discrimination is made in two steps: first the ones which are plausible according to the set of information coming from the team are discriminated, successively the correct one is identified by means of an agreement over the orientation. Indeed, this two-steps procedure allows to significantly reduce the cardinality of the set Ω and therefore to keep the overall complexity of the proposed technique affordable. The drawback is presented, if a non-rigid formation is assumed, by the fact that some problems could arise when an eventual mutual localization module would introduce error estimates. In this case the first phase could be prone to misunderstanding.

4.5.5 Experimental Results

In this section, an experimental validation of the proposed topological map-building technique is proposed. Experiments have been carried out by exploiting two robots

SAETTA having a complementary 180° field of view (see Figure 4.13). The experiments encompass all the environmental patterns described in Section 4.5.2. In particular, three different scenarios have been considered.

The motion of a robot is regulated by a very simple rule: if an obstacle is sensed on the heading direction, or if a large discontinuity is detected by the lateral sensors, a 360° rotation is performed by each robot. Note that, the rotation maneuver is required only to make up for the limited sensing resolution of the array of infrared range-finders. Indeed, this could be avoided if a sensor with a more refined resolution were available, e.g., a laser range-finder.

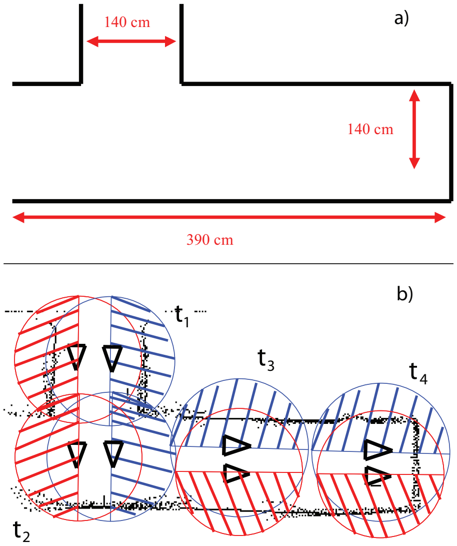


Figure 4.13: First scenario. a) scenario of the first experiment: the environment can be partitioned into a sequence constituted by a corridor (upper part), a T-junction, another corridor and a dead end b) the environment reconstructed by IR sensors is visible in the background. Triangles represents robot poses over time, while semi circles show the parts to be monitored by each robot.

The first scenario, with the related robots paths, is depicted in Figure 4.13. The sequence of mass aggregation performed by the two robots is shown in Figure 4.14. In particular, the solid (green) line represents the ground-truth, while the cross-dashed (red) line represents the output of the coarse aggregation involving a set of non oriented environmental patterns, and the dashed (blu) line describes the resulting (oriented) patterns obtained after the two-step aggregation procedure is performed.

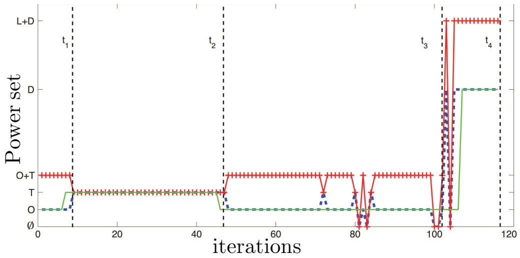


Figure 4.14: Estimation process over the time for the first experiment: solid (green) line represents the ground-truth, cross-dashed (red) line represents the output of the sensor fusion using the TBM conjunctive rule, dashed (blu) line represents the output of the algorithm using feature orientation information.

From the beginning until time t_1, each robot detects a wall and, therefore the element of the power-set supporting both a corridor and a T-junction will be set with a mass greater than 0. Now, since the algorithmic solution described in Subsection 4.5.4.2 is considered, the aggregation (first-step) cannot solve the ambiguous situation, due to the lack of knowledge about the orientation. However, by performing the second step a conflict regarding the orientation of the T-junction arises and therefore the corridor is taken as the correct pattern. Successively, at time t_2 the formation moves into a T-junction. In this case both robots detect a corner along with a wall, and therefore no doubt concerning the correct pattern remains after the masses aggregation (first-step) is performed. Furthermore, as a large discontinuity is detected, the two robots perform a full rotation as explained above. After that, till time t_3 the two robots remain within a corridor performing the same aggregation as discussed for the first time interval. Note that, due to the noise affecting the measurements, a couple of times a convex corner is erroneously detected by one of the two robots, making the two observations contradictory. As a result, a high value for the $m(\emptyset)$ is obtained by the two robots. After time t_3, the two robots approach a dead-end and also in this case a full rotation is performed. Apart from a few erroneous features extraction both robots detect a convex corner which supports both the L-turn and the dead-end. As for the corridor, the lack of information about the orientation does not allow to disambiguate the proper profile after the first step. Nevertheless, the dead-end is properly recognized after the second step is performed.

The second scenario, which considers two intersecting corridors, is shown in Figure 4.15. After a correct initial estimation of the corridor, the couple of robots repeat-

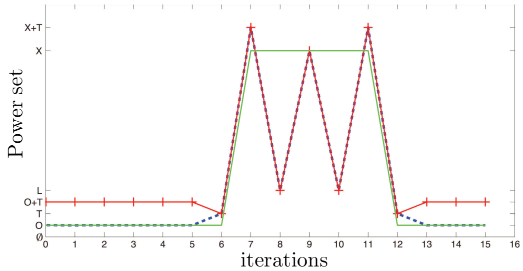


Figure 4.15: Estimation process over the time for the second experiment: solid (green) line represents the ground-truth, cross-dashed (red) line represents the output of the sensor fusion using the TBM, dashed (blu) line represents the output of the algorithm using feature orientation information.

edly fails to estimate the crossing. Indeed, this is due to the wrong feature extraction performed by one of the two robots. In particular, three erroneous detections are performed. On the first case, one of the two robots correctly identifies two corners while the other one simply recognizes a segment. This leads to the wrong (but plausible according to the data) detection of a T-junction. On the second case as shown in Figure 4.16, one of the two robots correctly identifies two corners while the other one recognizes two roughly parallel segments which do not respect the minimum distance constraint discussed in Subsection 4.5.2. Now, since no feature is available for the faulty robot the entire mass is assigned to the union of all the hypotheses, i.e., $m(\mathcal{P}) = 1$. As a result, the two profiles crossing and T-junction, supported by the detection of the other robot, cannot be disambiguated. Finally, on the third case an L-turn is detected. This can be explained by the fact that one of the two robots erroneously recognizes a convex corner instead of a concave one. As a consequence, by combining the patterns supported by the correct detection of a concave corner (by one robot), with the patterns supported by the wrong detection of a convex corner (by the other robot), the only plausible pattern turns out to be the L-turn (Figure 4.18). Differently, the crossing is properly recognized when both robots recognize a couple of concave corners each. Note that, this situation could be partially recovered by assigning a mass $m(\mathcal{P}) > 0$ to the element representing the whole set of environmental patterns. In addition, if an index of quality about the feature extraction process were available, the value of $m(\mathcal{P})$ could even be accurately tuned. Obviously, the higher the observation reliability is the lower the value of the mass would be. For sake of clarity, let us now consider a numerical

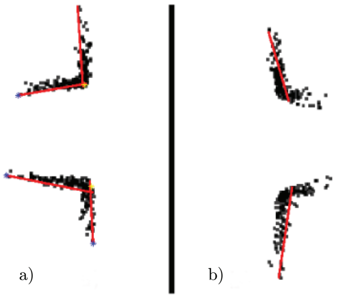


Figure 4.16: Feature extraction. a) a robot correctly detects a pair of angles b) a robot fails to recognize a pair of angles estimating a meaningless dashed line.

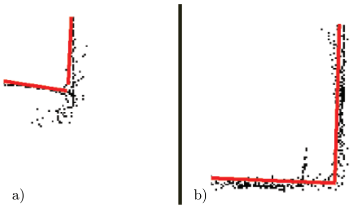


Figure 4.17: Corner identification. a) the robot correctly identify a concave corner b) the robot correctly identify a convex corner.

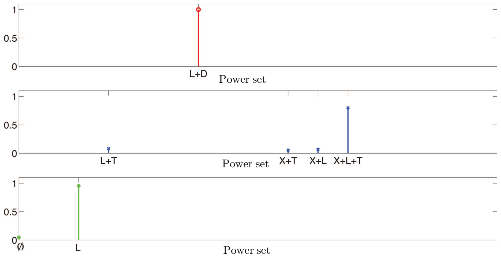


Figure 4.18: Mass aggregation for a couple of robots under the assumption of perfect measures. a) mass assignment for the first robot b) mass assignment for the second robot c) result.

example describing this situation. Figure 4.18 shows the result of the aggregation in the case of ideal measurements, while Figure 4.19 shows the result of the aggregation if the suggested workaround is taken into account. In the first case, the result of the aggregation does not allow to detect the correct pattern even if further aggregations are considered, while in the second case this would be possible as the correct pattern is still considered plausible.

The last experiment involves the detection of the L-turn. Also in this case the orien-

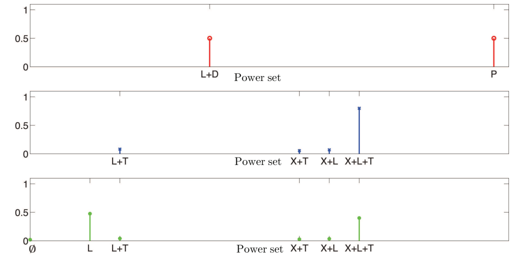


Figure 4.19: Mass aggregation for a couple of robots under the assumption of noisy measurements. a) mass assignment for the first robot b) mass assignment for the second robot c) result.

tation of the relational feature allows to properly estimate the surrounding environment as can be seen in Figure 4.17 with no contradiction.

4.6 Conclusions

In this chapter the data aggregation problem for a multi-agent system has been investigated. Agents are assumed to be independent reliable sources which collect data and collaborate to reach a common belief. A distributed protocol for data aggregation, which was proved to converge to the basic belief assignment (BBA) given by a centralized aggregation based on the Transferable Belief Model (TBM) conjunctive rule, has been provided either for static and dynamic scenarios. Furthermore a practical case for multi robot systems, related to the topological map building, has been described.

This framework finds its effectiveness in categorization problems, where the set of hypotheses taken into account is limited: it is worth of note that such technique is computationally expensive with respect to other approaches, e.g. Bayesian one, and it can be used only in proper contexts. Future work will be mainly focused on the possible extension of the proposed technique on more demanding topologies, e.g. graphs, thus on more demanding scenarios. While the trees can easily represent interaction between static sensors, the use of cyclic structures better describes the interaction between mobile units without taking into account the problem of a distributed spanning tree. Another issue is related to the possibility of this system to store information over the time: at now, only memoryless data fusion has been taken into account. Particularly related to the TBM, this formulation shows another drawback: the mass related to contradiction tends to increase over the time because it substantially integrates all possible ambiguities, i.e. noise, over the time. This peculiarity even more hinders the introduction of a recursive data fusion system.

4.7 Example of Distributed Data Aggregation

In the following, an example of distributed data aggregation involving a system of 5 agents (sources) observing an event is given. Two facts are assumed to be possible for the observed event, hence the following frame of discernment is defined $\Omega = \{a, b\}$ and the following power-set is obtained. $\Gamma = \{\emptyset, a, b, \{a, b\}\}$. Observations collected by the agents are summarized in Table 4.1.

Set #	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
$\emptyset$	0	0	0	0	0
$\{a\}$	0.1	0.2	0.1	0.2	0.3
$\{b\}$	0.8	0.7	0.8	0.7	0.4
$\{a, b\}$	0.1	0.1	0.1	0.1	0.3

Table 4.1: Observations collected by the system of 5 agents concerning the event.

In the case the centralized TBM aggregation schema is considered, according to Definition 6 the result of the data aggregation is shown in Table 4.2.

Set #	Agent 12	Agent 123	Agent 1234	Agent 12345	C-TBM
$\emptyset$	0.23	0.341	0.4781	0.63499	0.63499
$\{a\}$	0.05	0.011	0.0035	0.00213	0.00213
$\{b\}$	0.71	0.647	0.5183	0.36285	0.36285
$\{a, b\}$	0.01	0.001	0.0001	0.00003	0.00003

Table 4.2: Centralized TBM: final result and progressive aggregation.

Time	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9
Edge	e_{12}	e_{13}	e_{24}	e_{35}	e_{12}	e_{13}	e_{12}	e_{35}	e_{24}

Table 4.3: Edge selection process.

Let us now consider the distributed TBM aggregation schema proposed in this work. In particular, Fig 4.20 depicts the multi-agent system where the solid (black) lines describes the network topology $\mathcal{G}$, while the dashed (red) lines represents the spanning tree $\mathcal{T}$ computed in a distributed fashion by the agents. The edge selection process for

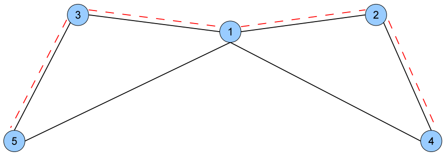


Figure 4.20: Multi-Agent system: solid (black) lines represent the network topology $\mathcal{G}$, dashed (red) lines describe the spanning tree $\mathcal{T}$.

the proposed example is described in Table 4.3. For sake of simplicity, only collaborations among agents which augment their common knowledge, i.e., the novelty is not the neutral element $\mathfrak{n}$ at least for one of the two agents, have been considered. Each agents has a table where the common knowledge with its neighbors is stored. In particular, by assuming each mass can be represent with a `double` (`4 bytes` in our representation), the memory occupancy for each agent is equal to `4 x 2 x 8 = 64 bytes`, where 2 is the maximum number of neighbors in the spanning-tree $\mathcal{T}$ and 8 is the cardinality of the power-set Γ. In the proposed example, the system of agents reaches a converge toward the same BBA after 9 steps. In particular, it can be noticed that each single agents has the same BBA (Tables 4.10,4.11, 4.12) as in the case of the centralized aggregation schema based on the TBM conjunctive rule (Table 4.2).

Set #	Agent 1	$\bar{s}_{1,2}$	$\widetilde{s}_{1,2}$	Agent 2	$s_1(t) \oplus s_2(t)\Big\vert_{t=1}$
$\emptyset$	0	0	0	0	0.23
$\{a\}$	0.1	0	0.1	0.2	0.05
$\{b\}$	0.8	0	0.8	0.7	0.71
$\{a,\, b\}$	0.1	1	0.1	0.1	0.01

Table 4.4: Distributed TBM: t=1, $s_1 \oplus s_2$.

Set #	Agent 1	$\bar{s}_{1,3}$	$\widetilde{s}_{1,3}$	Agent 3	$s_1(t) \oplus s_3(t)\Big\vert_{t=2}$
$\emptyset$	0.23	0	0.23	0	0.341
$\{a\}$	0.05	0	0.05	0.1	0.011
$\{b\}$	0.71	0	0.71	0.8	0.647
$\{a,\, b\}$	0.01	1	0.01	0.1	0.001

Table 4.5: Distributed TBM: t=2, $s_1 \oplus s_3$.

Set #	Agent 2	$\bar{s}_{2,4}$	$\widetilde{s}_{2,4}$	Agent 4	$s_2(t) \oplus s_4(t)\big\|_{t=3}$
$\emptyset$	0.23	0	0.23	0	0.407
$\{a\}$	0.05	0	0.05	0.2	0.017
$\{b\}$	0.71	0	0.71	0.7	0.575
$\{a,\, b\}$	0.01	1	0.01	0.1	0.001

Table 4.6: Distributed TBM: t=3, $s_2 \oplus s_4$.

Set #	Agent 3	$\bar{s}_{3,5}$	$\widetilde{s}_{3,5}$	Agent 5	$s_3(t) \oplus s_5(t)\big\|_{t=4}$
$\emptyset$	0.341	0	0.341	0	0.5395
$\{a\}$	0.011	0	0.011	0.3	0.0069
$\{b\}$	0.647	0	0.647	0.4	0.4533
$\{a,\, b\}$	0.001	1	0.001	0.3	0.0003

Table 4.7: Distributed TBM: t=4, $s_3 \oplus s_5$.

Set #	Agent 1	$\bar{s}_{1,2}$	$\widetilde{s}_{1,2}$	Agent 2	$s_1(t) \oplus s_2(t)\big\|_{t=5}$
$\emptyset$	0.341	0.23	0	0.407	0.4781
$\{a\}$	0.011	0.05	0.1	0.017	0.0035
$\{b\}$	0.647	0.71	0.8	0.575	0.5183
$\{a,\, b\}$	0.001	0.01	0.1	0.001	0.0001

Table 4.8: Distributed TBM: t=5, $s_1 \oplus s_2$.

Set #	Agent 1	$\bar{s}_{1,3}$	$\widetilde{s}_{1,3}$	Agent 3	$s_1(t) \oplus s_3(t)\big\|_{t=6}$
$\emptyset$	0.4781	0.341	0	0.5395	0.63499
$\{a\}$	0.0035	0.011	0.2	0.0069	0.00213
$\{b\}$	0.5183	0.647	0.7	0.4533	0.36285
$\{a,\, b\}$	0.0001	0.001	0.1	0.0003	0.00003

Table 4.9: Distributed TBM: t=6, $s_1 \oplus s_3$.

	Agent 1	$\bar{s}_{1,2}$	$\widetilde{s}_{1,2}$	Agent 2	$s_1(t) \oplus s_2(t)\big\|_{t=7}$
$\emptyset$	0.63499	0.4781	0	0.4781	0.63499
$\{a\}$	0.00213	0.0035	0.3	0.0035	0.00213
$\{b\}$	0.36285	0.5183	0.4	0.5183	0.36285
$\{a,\, b\}$	0.00003	0.0001	0.3	0.0001	0.00003

Table 4.10: Distributed TBM: t=7, $s_1 \oplus s_2$.

	Agent 3	$\bar{s}_{3,5}$	$\widetilde{s}_{3,5}$	Agent 5	$s_3(t) \oplus s_5(t)\big\|_{t=8}$
$\emptyset$	0.63499	0.5395	0	0.5395	0.63499
$\{a\}$	0.00213	0.0069	0.2	0.0069	0.00213
$\{b\}$	0.36285	0.4533	0.7	0.4533	0.36285
$\{a,\, b\}$	0.00003	0.0003	0.1	0.0003	0.00003

Table 4.11: Distributed TBM: t=8, $s_3 \oplus s_5$.

	Agent 2	$\bar{s}_{2,4}$	$\widetilde{s}_{2,4}$	Agent 4	$s_2(t) \oplus s_4(t)\big\|_{t=9}$
$\emptyset$	0.63499	0.407	0.28	0.407	0.63499
$\{a\}$	0.00213	0.017	0.09	0.017	0.00213
$\{b\}$	0.36285	0.575	0.6	0.575	0.36285
$\{a,\, b\}$	0.00003	0.001	0.03	0.001	0.00003

Table 4.12: Distributed TBM: t=9, $s_2 \oplus s_4$.

Chapter 5

A stigmergic approach for environmental monitoring

In this Chapter the gas source localization problem is treated. In particular a stigmergic approach, i.e. the process by which agents indirectly coordinate/cooperate, is used: inspired by a well known algorithm in literature, the Kernel DM+V algorithm, an environmental monitoring technique using low cost devices and smart environments, i.e. places which provide some services to the robots, is proposed[1]

5.1 Introduction

The interest in robotic olfaction for air quality monitoring is growing steadily, mostly driven by the need to protect the humans and the environment from the presence of toxic contaminants and pathogens in the air. A mobile robot equipped with an electronic-nose (e-nose) can act as an autonomous wireless node for monitoring purposes. The use of a mobile platform offers a number of important advantages compared to stationary sensors, including: higher and adaptive monitoring resolution; compensation for inactive sensors; adaption to dynamic changes in the environment; the possibility to do source localization and tracking; and the possibility to perform first aid and cleanup of hazardous sites.

Robotic olfaction is a young research field, and a number of research challenges remain open. The two key topics in this field are gas source localization and gas

[1] This is a joint work with the AASS Research Center, Örebro University, Sweden

distribution modelling. Gas source localization with mobile robots deal with the goal of finding a source that releases some chemical substance. Gas distribution modelling is the task of deriving a truthful representation of the observed gas distribution from a set of spatially and temporally distributed measurements of gas concentration. Both topics are very challenging tasks. One main reason is that in many realistic scenarios gas is dispersed by turbulent advection. Turbulent flow creates packets of gas that follow chaotic trajectories (84). This results in a concentration field that consists of fluctuating, intermittent patches of high concentration.

The two topics above are closely connected, and can be seen as dual aspects of the same problem. On one hand, knowing the gas source may allow us to form predictions about the gas distribution; on the other hand, gas distribution modelling can provide useful information to locate a gas source. There are, however, several difficulties that limit the practical applicability of current techniques. Gas distribution modelling requires that the robot is able to self-localize (with respect to a fixed frame or respect to other robots) and possibly to build a map of the environment. This in turn typically requires the use of expensive sensors and/or computationally heavy algorithms, which makes these techniques unsuitable for practical, inexpensive service or consumer robots. Gas source localization, by contrast, can be performed in a reactive or quasi-reactive way with minimal sensing and computation. Unfortunately, the effectiveness of reactive approaches is seriously hindered by the turbulent nature of gas distributions, which prevents the formation of a smooth concentration gradient that the robot can follow.

In this chapter, we propose to overcome these limitations by allowing the robot to use the environment to store information. For this, we use an RFID floor, that is, a floor with a grid of RFID tags buried underneath it. RFID floors are an emerging technology (85), and several authors have used them to improve localization accuracy. For instance, in (86) the robot gathers the absolute positions stored into the floor tags to approximatively compute its own location. In (87) a Monte Carlo approach is used to estimate robot position, where the probabilistic model related to the antenna range is used to instantiate a particle filter.

In our work, we use an RFID floor to store information about gas distribution. In a nutshell, our approach works in two phases. In the first phase, the robot stores information about the local gas concentration, derived from sensor measurements. In the second phase, the robot stores information about the global distance to the areas of highest concentration. The two phases build upon a combination of an algorithm for gas distribution mapping (88) and an algorithm for stigmergic path planning (19), respectively. The result is the creation of a map, stored in the RFID floor, that approximates the distance to a gas source. This map provides a smooth (metric) gradient toward the gas source, which can be used as a sort of "virtual" gradient of the gas concentration. This enables a robot to navigate to the source by cheap gradient descent on the values read from the RFID tags.

Our algorithm only relies on the information coming from the gas sensor and read from the floor tags. The robot does not need to know where it is or to store any global information onboard. Therefore, our approach is suitable for low-cost robots with very limited sensing and computational capabilities. As for the RFID floor, the very limited storage capacities of standard RFID tags proved to be sufficient for our purposes, even under long lasting experiments.

The use of the environment to store information is a well known mechanism in nature, called stigmergy (89). Stigmergic approaches are used both in natural and artificial agents to enable the cooperative performance of tasks (90). Accordingly, we show in this chapter how our approach naturally extends to multiple robots, enabling them to perform the mapping task in a cooperative way by simply storing and reading information from the same RFID floor. Interestingly, cooperation emerges even if the robots are not aware of each-other.

5.2 Background

In this section we recall some concepts about the two basic approaches upon which our algorithm is based. In Subsection 5.2.1 the gas map building is described while in the Subsection 5.2.2 distance mapping process over an RFID floor is summarized.

5.2.1 Olfaction maps

As mentioned above, the two key topics in robot olfaction are gas source localization and gas distribution modelling. Gas source localization involves three subtasks: gas finding, gas source tracking and gas source declaration (91). Gas finding is the task of detecting an increased concentration of a target gas; gas source tracking is the task of following the cues determined from the sensed gas distribution towards the source; and gas source localization is the decision process finding out whether the source has been reached, at which location it is, or which object it is. Most of literature copes with gas source localization using bio-inspired approaches: for instance, in (92) a genetic algorithm is applied to a swarm of robots in order to locate the leakage of gas into an environment presenting a contained airflow. The algorithm is split in two phases: in the first each robot perform a spiral movement to explore the surrounding environment while in the second it moves according to the gathered information and data coming from the other robots. A similar bio-inspired spiral motion is proposed in (93), where robots execute different strategies depending on the concentration sensed and the wind direction measured by an anemometer. Experiments are executed under the assumption of laminar flow. In (94), three bio-inspired odor source localization algorithms (casting, surge-spiral and surge-cast) are tested both in simulation and with real robots in a wind tunnel, and their expected performance is derived through a theoretical model.

Current approaches to gas distribution mapping can be divided into two groups. Model-based approaches, such as the one proposed by Ishida et al. (95), assume a particular model of the time-averaged gas distribution and estimate the corresponding parameters. Model-free approaches deal with the fluctuating nature of the gas distribution without assuming a particular form of the model. In (96) and (97) individual concentration samples were recorded over a prolonged time (several minutes) at grid locations, and in (91, 98) gas sensor readings were statistically integrated into a spatial grid. Lilienthal et al. (88) propose a statistical approach (the Kernel DM+V algorithm) of the observed gas distribution, treating gas sensor measurements as random variables. The learned model is represented as a pair of grid maps, one representing the distribution mean and the other one the corresponding predictive variance per grid cell. Reggente et al. (99) show that the variance distribution map typically provides more accurate informations about the source location than the mean distribution map.

In this work, we use the Kernel DM+V algorithm (88) because it allows a systematic and meaningful representation of the maps, which capture important aspects related with the gas source location thanks to the computation of the variance per grid cell, The general gas distribution mapping problem addressed is to learn a predictive two dimensional model $p(s|\mathbf{x}, \mathbf{x}_{1:n}, s_{1:n})$ for the gas reading s at location $\mathbf{x}$, given the robot trajectory $\mathbf{x}_{1:n}$ and the corresponding measurements $s_{1:n}$. To study how gas is dispersed in the environment we consider measurements from metal oxide sensors. The central idea of kernel extrapolation methods is to understand gas distribution mapping as a density estimation problem addressed using convolution with a Gaussian kernel $\mathcal{N}$, which governs the amount of extrapolation. The map building process, at each time instant t_i, computes for every cell center x_l the following quantities, basing on the current position x_i and gas concentration measure s_i of the robot:

$$\Omega_l = \sum_{i=1:n} \mathcal{N}(|x_i - x_l|, \sigma) \tag{5.1}$$

$$M_l = \sum_{i=1:n} \mathcal{N}(|x_i - x_l|, \sigma) \cdot s_i \tag{5.2}$$

$$V_l = \sum_{i=1:n} \mathcal{N}(|x_i - x_l|, \sigma) \cdot (s_i - s_{l(i)})^2 \tag{5.3}$$

where Ω_l is the weight map which intuitively represent the information content of a sensor measurement i at grid cell l. M_l, V_l are respectively the mean distribution map and the variance distribution map. The term $s_{(l(i))}$ is the mean prediction of the cell l closest to the measurement point x_i and, consequently, the quantity $(s_i - s_{(l(i))})^2$ is the variance contribution of reading i, and σ is the standard deviation of the Gaussian $\mathcal{N}$. The quantity Ω_l is used as a normalization factor in order to build the estimate taking into account the particular trajectory of the robot which both may privilege certain locations respect to others or to not cover some locations:

$$m_l = \alpha \frac{M_l}{\Omega_l} + (1 - \alpha) m_0 \tag{5.4}$$

$$v_l = \alpha \frac{V_l}{\Omega_l} + (1 - \alpha) v_0 \tag{5.5}$$

where α is a confidence factor which indicates high confidence for cells for which a large number of readings close to the center of the respective grid cell is available. m_0 and v_0 represent the best estimates, which are used for cells with a low confidence, i.e. for cells for which we do not have sufficient information from nearby readings. As the best guess of the mean concentration m_0 we use the average over all sensor readings and v_0 is the average over all variance contributions.

5.2.2 Building the artificial gradient

Gradient maps over an RFID floor have been presented in (19). The environment can be represented by a grid of cells into which one or more robots write some data.

In particular, in this approach, an estimate of the distance to a given goal is stored into every cell. To this purpose, the robot starts from a known goal writing a 0 value both into this cell and into an internal counter. Then robot executes random trajectories updating the value of the cells when needed.

In particular, during the execution of the algorithm, the robot compares the estimate stored into the cell over which it is, with the actual estimate obtained from its motion. In particular the latter one is obtained incrementing the internal counter whenever a transition from a cell to another is detected.

It has been analitically proven that, under a full coverage of the environment, a distance map reporting for each cell the minimum distance to the goal is asymptotically obtained. This process does not need data storage on the robot except for the actual estimated distance. Furthermore the algorithm can be run concurrently by multiple units and it is robust with respect to the deployment of obstacles on the floor. The presence of multiple goals is managed building several maps at the same time using the multiple fields available on an RFID floor.

5.3 Building the map

In our scenario one or more robots travel over an hexagonal grid of cells. In particular each cell is described by an Id and a readable/writeble memory f. It is worth to note that the position of the cells is unknown to the robots, and that the Id does not furnish any metrical information as it is only used to detect the passage of the robot between two cells. The grid can be represented by a graph $\mathcal{G} = (\mathcal{R}, \mathcal{E})$, where the set of nodes $\mathcal{R}$ represents the cells and the set of edges $\mathcal{E}$ represents the links between adjacent cells. The centers of two adjacent cells, p_i and p_j, are separated by an interdistance d, i.e. $\mathcal{E} = \{e_{ij} : ||p_i - p_j|| = d\}$. The gas dispersal is supposed to be induced by a stationary source, i.e. the gas dispersal is described by a stationary aleatory process (88).

In the further, the description of the algorithm will be detailed taking into account only a robot. This description does not loose of generality with respect to the multi robot case thanks to the stygmergic approach that is used in our algorithm: in fact the data stored in each cell are retrieved and manipulated without taking care of the identity of the robot that performs these operations. As it will be detailed in the further, this scheme will lead to an implicit collaboration even if the agents are not aware of each other.

The time execution is discretized into equal slots $t_k, k \in \mathbb{N}$. For each t_k, the robot is supposed to interact with the cell that it occupies. In particular our algorithm can be split in two phases. In the first the agent acquires and elaborates gas-concentration measures. Once this process stops, the robot builds a distance map towards interesting regions.

5.3.1 Phase I : map seeding

During this phase the robot gathers concentration measures. In particular at each instant t_k the agent collects a sample $s_i(t_k)$ on a generic cell $r_i \in \mathcal{R}$. The value $s_i(t_k)$ is represented by an integer variable: this choice is supported by the fact that these values are gathered sampling the transducers by an analog to digital (AD) converter.

These data are used to iteratively build both concentration ($\mathcal{M}$) and variance ($\mathcal{V}$) maps.

In particular the informative content of a cell r_i is defined as:

$$f_i(t_k) = (m_i(t_k), v_i(t_k), c_i(t_k)) \tag{5.6}$$

where $m_i(t_k)$ and $v_i(t_k)$ represent respectively the actual mean and variance concentration related to the cell r_i and $c_i(t_k)$ is a counter representing the number of time slots during which the robot was on the cell c_i.

This representation avoids the storage of any information on-board the robot: in this phase all the needed information is stored in the cells.

In detail, both the values of concentration and variance are supposed to be discretized into M levels $l_j \in 0, \ldots, L_M, j = 0, \ldots, M$ with $l_{j+1} = l_j + \Delta_l, \Delta_l \in \mathbb{N}^+$.

Whenever a robot acquires a measure over the cell r_i, the memory f_i is updated as the following:

$$m_i(t_k) = k_m \left(\frac{c_i(t_{k-1}) m_i(t_{k-1}) + s(t_k)}{c_i(t_{k-1}) + 1} \right) \tag{5.7}$$

$$v_i(t_k) = k_v \left(\frac{c_i(t_{k-1}) v_i(t_{k-1}) + c_i(t_{k-1}) m_i^2(t_{k-1}) + s^2(t_k)}{c_i(t_{k-1}) + 1} - m_i^2(t_k) \right) \tag{5.8}$$

$$c_i(t_k) = c_i(t_{k-1}) + 1 \tag{5.9}$$

where k_m and k_v are constants to normalize the values into the aforementioned levels.

In this manner, at each location, an estimate of the current concentration and variance is iteratively built.

As the robot does not have any metrical information, the related motion is governed by a random behavior: the agent goes straight until an obstacle forbids the motion. When this happens, random rotations are performed until a free path is available.

This scheme, despite the initial deployment of the RFID infrastructure, allows to drastically reduce the amount of resources to be installed on the mobile platform: in this phase the robot does not store anything on its own memory and the computational burden is reduced only to integer operations.

After a certain amount of time this phase is declared completed: at the end of this procedure the RFID floor is seeded by the triplets reported in Eq.(5.6).

Some considerations about the end of this phase are needed: being the navigation governed by a stochastic behavior, it is not possible to compute in a closed form the completion time. However, it is possible to empirically establish a good estimate basing on the dimensions of the environment, the number of the robots and the related cruise speed. In order to clarify this consideration, a couple of examples is reported: to effectively cover the PEIS Home, the first phase took about 50 minutes with a single robot while the same task took about two hours in the PEIS Home 2 with a couple of SAETTAs. It is worth to note that only very few objects were present on the floor.

In our experiment we chose a much longer period in order to study the behavior of the algorithm with respect to saturation phenomena. In particular, we wanted to emphasize the robustness of our algorithm with respect to modification of the gas distribution. Over the time in fact the environment tends to be saturated and, as it will be shown in Section 5.4, some areas show a growing concentration. In order to consider this particular phenomenon, we assume that the source which releases the gas can be described as a stationary stochastic source.

Secondly, similarly to (88), a full coverage is not needed: the filtering effect of the second phase will update the empty cells by the values obtained on the adjacent ones. An empty cell is characterized by a triplet $f = (0, 0, 0)$. Obviously, the less the number of empty cells, the better the overall estimate.

Finally, a consideration about the multi robot execution is needed. Before the task execution, we used a simple calibration which resulted to be effective to our purposes: we calibrated the sensors using the values gathered by the e-nose respectively when the gas-compound is not present and when it is next to the transducer. In this manner each robot can furnish normalized measures which are compliant with the ones provided by the other ones: as it will be detailed in the further even this simple calibration is sufficient to effectively perform a multi robot seeding.

5.3.2 Phase II : gradient map

In this phase the robot travels on the grid in order to build potentials centered at interesting zones. While in (19) the case of a single goal was treated, in this scenario we try to effectively represent multi modal profiles in which local maxima represent more interesting locations. During the building process, the robot stores in memory a variable, the potential counter, for each map to be built.

This variable is initially set to zero; whenever a transition is detected by a robot, i.e. it passes from a cell to another, the potential counter is decremented by a factor ρ. This value is then compared with the one retrieved from the tag where the robot has arrived: the minimum between these two values is written in the memory of the cell and the navigation continues. At the end of the algorithm, each tag, for each map, will contain a value which is proportional to the vicinity to the more influencing maximum.

As described before, also in this case the navigation has a random nature.

It is worth to note the role of ρ: as it will be detailed in the further, this parameter plays an essential role in order to obtain an effective representation of a multi modal profile.

A description of the gradient mapping is reported in Algorithm 3.

Algorithm 3:

```
Notation: $Y \in \mathcal{R}$ subset of cells, $w(y)$ generic value (wether mean or variance) on the cell y
Require: all the tags have an initial value $w$, a step increment value $\delta = \Delta_l/k, k \in \mathbb{N}$
Ensure: each cell will contain a value proportional to its proximity with respect to the more influencing maximum
$Y_{curr} \leftarrow \emptyset$
$potential\_counter = 0$
while $Explore$ do
    $Y_{prev} \leftarrow Y_{curr}$
    $Y_{curr} \leftarrow tags\_in\_range()$
    $Y_{new} \leftarrow Y_{curr} - Y_{prev}$
    choose random $y \in Y_{new}$
    $potential\_counter \leftarrow potential\_counter - \delta$
    $w(y) \leftarrow \max(w(y), potential\_counter)$
    $potential\_counter \leftarrow w(y)$
end
```

5.3.3 Simultaneous building: the Olfaction gradient map

At the end of the first phase, it could be straightforward to apply the algorithm in (19). However, this approach presents limitations when a multi modal profile is considered. In fact the turbulent nature of the gas dispersal could take to unuseful representations. In order to cope with that, the parameter ρ is used to filter out the effect of the turbulence.

In practice this parameter is used to isolate interesting locations within a certain zone. In particular it is possible to relate ρ with the previous discretization step Δ_l as the following:

$$D = \frac{\Delta_l}{\delta} \tag{5.10}$$

with $D \in \mathbb{N}$. As a result, whenever D is greater than one, a finer smoothing is performed. In particular some asymptotic properties about the distance maps can be inferred under the following assumptions:

Assumptions 2

(i) *the reader never skips over a tag*

(ii) $\forall t_k$ *only a tag is in the range of the robot*

(iii) *the exploration strategy is complete*

these conditions can be assumed achievable being supported by experimental results. A particular consideration about the completeness of the exploration is due: this is an asymptotic property that is respected taking into account our navigation strategy and the fact that the environment is of finite dimensions.

Furthermore, it is useful to formalize the definition of local maximum in our context:

Definition 5.1 *Under Assumptions 2, a cell r_i is referred as a maximum for the map $\mathcal{X}$ if it contains a value $\bar{x}_i(t)$ s.t. $\exists T' \in \mathbb{N}_0$ s.t. $\bar{x}_i(t_k) > x_j(t_k)$, $\forall t_k \geq T'$ and $\forall e_{ij} \in E$*

where x_i and $\mathcal{X}$ can be referred whether to the mean (m_i and $\mathcal{M}$) or variance (v_i and $\mathcal{V}$) map representations.

As a consequence, it is possible to infer the following property about the maps which are built by the algorithm:

Theorem 5.1 *Under assumptions 2, asymptotically in the map $\mathcal{X}$ each area having at least radius $D-1$, centered at the maximum r_{max}, will contain values which are smaller or equal to $\bar{x}_{max}$.*

As it can be argued, the choice of D corresponds to filter out higher frequencies on the gradient profile: for example, choosing $D = 1$ will correspond to have no smoothing while choosing $D \geq 2$ will take to a filtering effect. Denote with $\bar{p}$ the longest path between all over the shortest paths connecting two cells of the map. If $\delta < \frac{\Delta_l}{\bar{p}}$, then only the highest value collected during the first phase will not be updated. As a consequence, all the other cells will contain a value whose magnitude depends only on the distance to this maximum. For further details, the related demonstration is reported in Appendix.

5.4 Experimental results

In this section some experiments are analyzed with a focus on the properties previously discussed about the algorithm. In particular two data sets are considered: the first was gathered in an empty space free of obstacles, while in the second walls and one table in the middle of the room were added. The experiments lasted 9 and 18 hours respectively. To the best of our knowledge, no experiment lasted so long in a similar scenario. Although a shorter time for this space is typically needed (2 hours with two robots), this time was taken into account because we wanted to test the robustness of our algorithm with respect to the first phase: one of the main issues related to the gas dispersal is that it can lead to saturation phenomena. For this reason, we considered a stationary source (this does not imply a stationary gas distribution but conversely it only characterizes the gas releasing). This assumption fits many scenarios. In a domestic environment, it can be associated with a gas leakage in a pipe or when a certain mass, e.g. a basket plenty of organic compound, releases chemical substances in the air. The gas distribution changes over the time, and the source localization problem becomes more difficult especially when the concentration increases in several locations.

As stated in (19), the second phase requires several hours. This drawback is compensated by the minimal hardware which is installed on the mobile platform and by the fact that the results of this phase depend only by the end of the seeding process.

In Subsection 5.4.1 the experimental setup is described and considerations about the data sets are reported in Subsection 5.4.2 . In Subsection 5.4.3 results about an empty space are presented, while in Subsection 5.4.4 map building with obstacles is contemplated. Finally Subsection 5.4.5 discusses about the results gathered using a multi robot system for the seeding phase.

5.4.1 Experimental setup

Experiments have been carried out into the PEIS Home 2. Compared to its predecessor, the PEIS Home (17), this environment furnishes a wider area, about 60m^2, which allowed us to conduct extensive experiments. One of the distinctive features of these structures is that their floors contain an hexagonal grid of RFID tags, which can be individually accessed by the robot while it navigates, and thus act as a physical read/write memory . In particular, each tag can store up to 8Bytes and the interdistance between two adjacent units is 20cm. The tests were conducted using a couple of SAETTA mobile platforms. Thanks to their small dimensions, these vehicles can be easily used into indoor environments and they can carry on several sensors. In this work robots mount IR sensors, a webcam, one SkyeModule M1 RFID reader and a

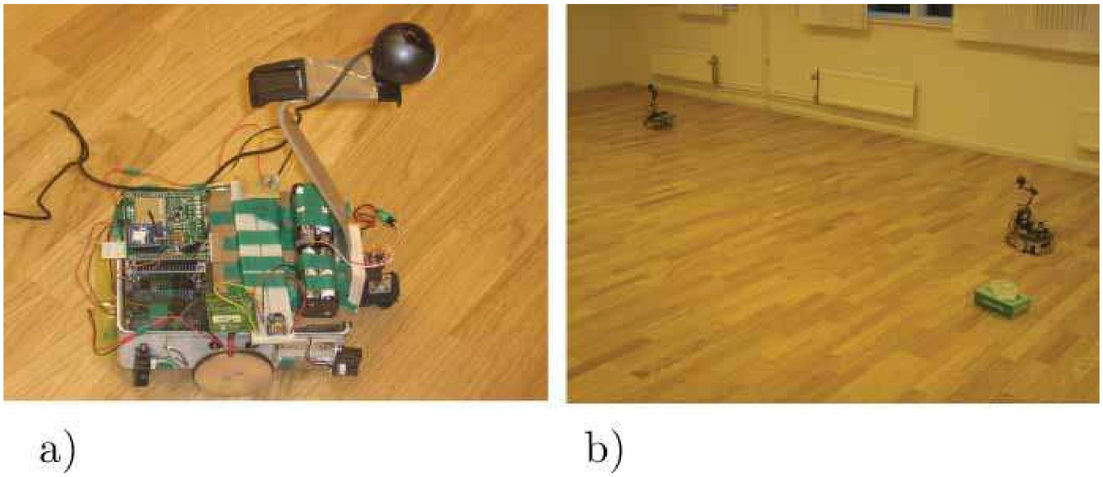


Figure 5.1: a) Robot SAETTA: on the robot sides an RFID reader and a gas sensor are mounted b) SAETTAs explore the PEIS Home 2.

MiCS-5521 metal oxide sensor as depicted in Fig. 5.1. The first two are used for obstacle and collision avoidance, while the latter ones are used to execute our algorithm (see Fig.5.1). The RFID reader checks for tag presence with a frequency of 2.5Hz; the gas sensors are collected at the same frequency after performing an average of samples collected at 32Hz. The webcam grabs images at 5Hz while the IR sensors are checked at 40Hz. The maximum speed of the robot is 7cm/s. The exploration is performed by a simple behavior which consists in moving straight until an obstacle is detected. When this happens, a random rotation is performed. The collision avoidance, being the landscape uniform, was implemented using the Polly algorithm (101) coupled with IR sensors. The latter ones were used in order to cope with obstacles suddenly appeared in front of the robot after performing a rotation. Due to the long lasting experiments, a voltage supplier was used: electric cables were passed over the lamps on the roof. For the task execution a person was in charge to avoid that cables would tangle each other.

This limitation is overcome in typical scenarios: as mentioned before, for the first phase, a shorter time is sufficient in order to effectively seed the map. Another solution, when multiple units are available, consists in using some robots when other are recharging the batteries. As it will be detailed in the further, the seeding performed with multiple robots is demonstrated to be effective.

The second phase instead does not suffer of time interruption as it depends only by the end of the first phase.

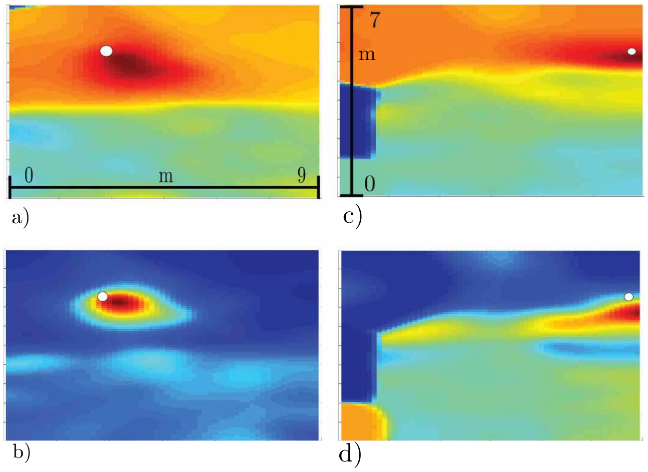


Figure 5.2: Kernel DM+V algorithm results. White circles represent gas source location a) mean map related to Experiment 1 b) variance map related to Experiment 1 c) mean map related to Experiment 2 d) variance map related to Experiment 2.

5.4.2 Data set validation

In order to estimate the goodness of the data, the Kernel DM+V algorithm (88), summarized in Subsection 5.2.1 , has been executed for both datasets. The results, depicted in Fig.5.2, show that the source is estimated with good accuracy both in mean and variance maps. The error estimate is always within 50cm of the exact location and, as expected, the variance representation shows more effectively the source location: in fact,regarding the mean representation, the peak is in the middle of an high concentration area, while the variance representation is less susceptible to saturation phenomena.

5.4.3 Mapping with no obstacles

In this experiment Saetta robots collected data over a time of 9 hours in an environment free of obstacles. In this test, data were used only for the first phase while the second one was only simulated. The results are depicted in Figure 5.3. In this case both the mean (Fig.5.3-b) and the variance (Fig.5.3-d) maps correctly estimated the source location with an error of one cell. The distance map process was executed after that the seeding phase showed a particularly relevant maximum in the vicinity of the source location: in Fig.5.3-a and Fig.5.3-c the mean and variance ones are respectively depicted. It is worth to note that, in this experiment, data were not filtered.

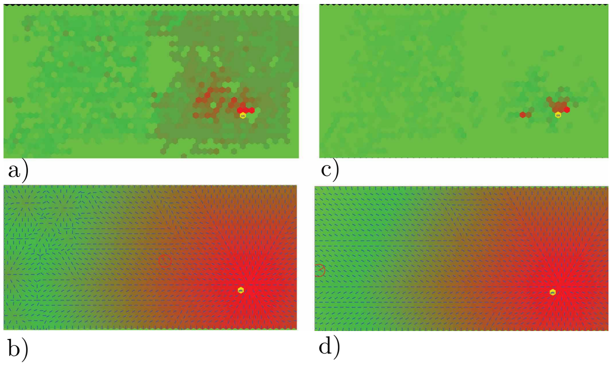


Figure 5.3: Experiment 1 performed in the PEIS Home 2 with no obstacles. Gas level was discretized into $L_M = 50$levels, while $D = 5$. Yellow cells represent gas source location: a) end of phase 1: seeding of the mean map, b) mean map. As showed, the error is contained (1 cell). In the upper left part some local maxima are represented c) end of phase 1:seeding of the variance map, d) variance map: thanks to a better seeding respect to the computation of the mean, no local maxima are present.

5.4.4 Mapping with obstacles

The second experiment was executed under more demanding conditions due to the presence of obstacles and a longer execution period. In the second phase, we considered the paths traversed by the two robots to validate the concurrency of this phase. In this case a thresholding of the values was needed. This was necessary especially for the variance maps: being this environment prone to saturation phenomena due to the long lasting time, some areas, which were initially clean, showed an increase of variance as the gas concentration increased over the time into the whole environment. This phenomenon could be partially handled by filtering data, for example with a forgetting factor: this kind of solutions, although effective, would introduce an unwanted sophistication of the algorithm. The threshold instead can be easily set by a sensor calibration prior of the task execution. This operation consists in acquiring the measures when the sensor is far away from the source and when it is very close to ethanol. In this manner, the range of the transducer can be determined. In our scenario we chose the threshold equal either to the 70% of the maximum value or the 130% of the minimum: in both cases we achieved good results. However, without applying a threshold, some interesting observations about the smoothing effect of the parameter δ are possible. In Fig.5.4-a a distance map considering $D = 1$ is showed, while in Fig.5.4-c the same data set has

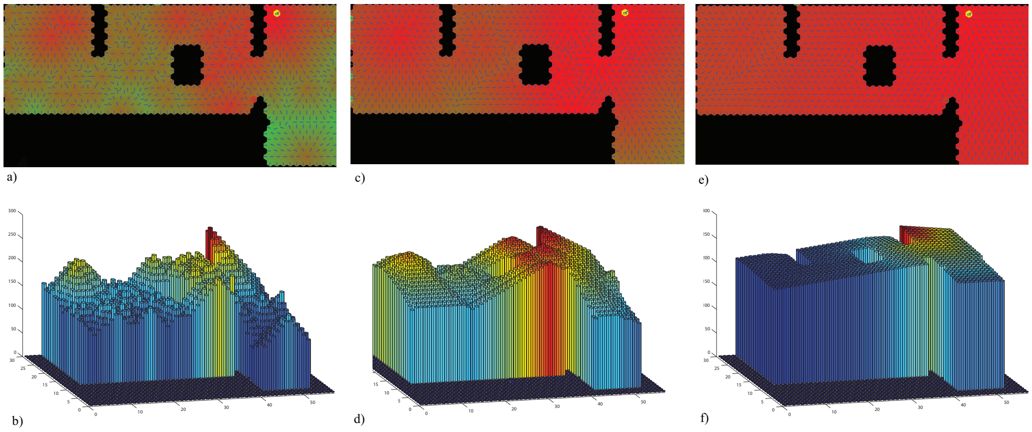


Figure 5.4: Distance maps obtained with different values of D, considering variance measures in Experiment 2 without threshold. In Fig. a,c,e Yellow blobs represent gas source location while the black ones are inaccessible tags a) distance map obtained with $D = 1$, b) three dimensional representation of the distance map with $D = 1$, c) distance map obtainded with $D = 5$, d) three dimensional representation of the distance map with $D = 5$, e) distance map obtainded with $D = 20$, f) three dimensional representation of the distance map with $D = 20$.

been processed with $D = 20$. As it can be noticed, the first map is very indented, while the latter one shows only one maximum located close to the source. Intermedium solutions can be obtained with a proper tuning of D: for example, Fig.5.4-b shows a map were two peaks arise setting $D = 5$. In this case it is also possible to see how the maxima are distant at least 4 cells each other.

In Fig.5.5-a the variance map obtained considering thresholded values is depicted: it was built considering the real paths traversed by the two robots. It can be seen that this representation resembles the asymptotic one obtained by simulation (Fig.5.5-b). Furthermore, it can be seen that the presence of obstacles distorts the gradient flux: in this manner the real distance for a cell to the source, greater than the line of sight range, is correctly represented. It is worth noting the difference between the two snapshots regarding the center of the image: this is due to a table put in the room; in the simulated map it is represented by a regular hole, while in the real map it has a different shape which depends also on the presence of power supply cables deployed on the floor.

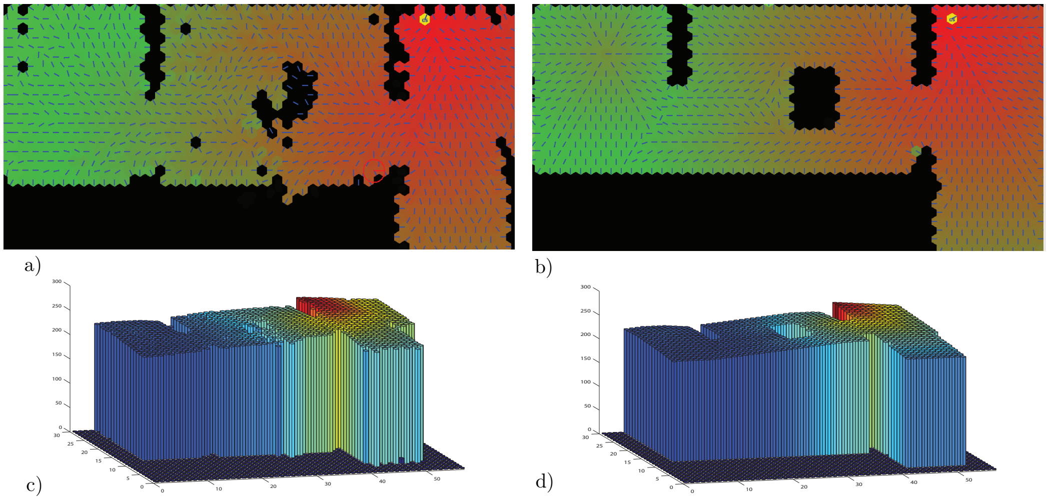


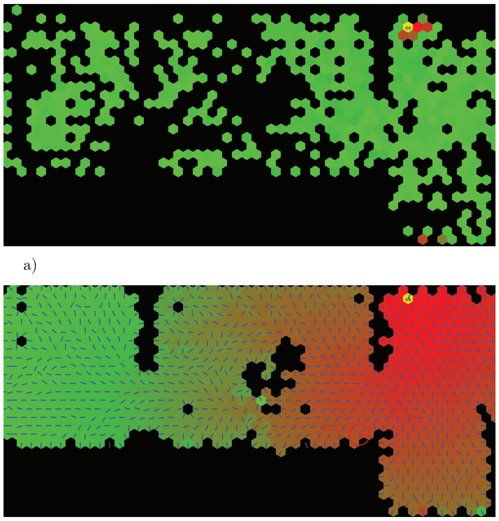
Figure 5.5: Variance map obtained from Experiment 2: $L_M = 25$ and $D = 2$. Data were thresholded considering only values greater than the 70% of the maximum value achievable by the sensor. Black cells represent obstacles or broken tags, while yellow ones represent gas source location: a) map obtained considering the real paths performed by the robots b) asymptotic map obtained integrating real paths with simulation c-d) 3D representation of the aforementioned maps.

5.4.5 Concurrent seeding

In this experiment also the first phase is parallelized: the measurements performed by each robot, due to the calibration procedure described in Subsection 5.3.1, are normalized with respect to a known level. As depicted in Fig.5.6-b, the gap between the estimate and true position is one cell. It is interesting to analyze the seeding phase depicted in Fig.5.6-a: the normalization procedure allows the measures gathered by the robots to be compatible each other. Also in this case the map was evaluated by the real paths of the robots: although the asymptotic map is not still gathered, it can be seen how more interesting cells are in the neighborhood of the source.

5.5 Conclusions

While mobile robot olfaction has great potential for future environmental monitoring application, its development is hindered by the difficulties induced by the turbulent nature of gas propagation. In this chapter, we have proposed an effective solution to robot gas mapping that leverages the concept of stigmergy, and can be implemented



a)

b)

Figure 5.6: Variance map obtained using concurrent seeding: a) seeding phase: the maximum value is adjacent to the source location b) distance map obtained using real paths. It is worth of note that asymptotic values are not gathered yet although the representation is consistent, being the source location properly estimated.

on inexpensive robots navigating on a RFID floor. Our solution combines measurements of gas concentration and fluctuation with distance computation to generate an artificial potential field whose maxima indicate the likely position of the gas source(s). This potential is computed on the RFID-floor rather than in the memory of the robot: this allows the robot to totally ignore its metric location, thus posing only minimalistic requirements on the needed hardware and software. It also enables multi-robot cooperation, by having several robots share the same RFID-floor. Extensive experimental results show that our approach produces usable gas maps even when using very simple robots.

The proposed approach points to several direction for further development. One such direction is to study exploration strategies that use the current information in the RFID floor. Both exploration and exploitation should consider carefully how to deal with local minima in the field stored in the floor. Another interesting extension is to equip the robot with a ring of RFID tag readers, rather than a single reader, in order to estimate the direction of the gradient in one step. Finally, we plan to study mechanisms that can cope with the case in which the position of the gas source changes over time.

Appendix

The proof of the Theorem is based on Assumptions 2. Furthermore we state, without proof, the following propositions (details can be found in (19)):

Proposition 4

- *the values $m(t)$ and $v(t)$ are non decreasing for each cell*
- *At the end of every cycle, the distance counter has the same value of the tag r upon which the robot is.*

With an abuse of notation, denote with $t_k = 0$ the end of the first phase. By this definition, it is possible to infer the following property:

Lemma 5.1 *Each maximum $\bar{x}_i(t), r_i \in \mathcal{R}$, will be equal to its own value at the end of the first phase, i.e. $\bar{x}_i(t_k) = x_i(0), \forall t_k \in \mathbb{N}$*

Proof: the proof directly exploits the update rule presented in Algorithm 3 and Definition 5.1. By contradiction, suppose that $\exists t'_k \in \mathbb{N}$ s.t. $x_i(t'_k) > x_i(0)$. This implies that $\exists r_j, e_{ij} \in \mathcal{E}$, s.t. $x_i(t'_k) = \max(x_i(0), x_j(t'_{k-1}) - \delta) = x_j(t'_{k-1}) - \delta$. Thus, at instant t'_k the neighbor r_j has a greater value than r_i. Hereafter, by Proposition 4 and under the assumption that exploration strategy is complete, being every x non decreasing over the time, x_i will be possibly updated by a neighbor having a greater value and thus it can not be a maximum according to Definition 5.1. □

It is worth to note that each initial value $x_i(0)$, according to Eq.(5.7) and Eq.(5.9), has the form $\bar{x}_i(0) = K_i D$, $K_i \in \mathbb{N}$. Now that it has been demonstrated that asymptotically the maxima will be constituted by a subset of values gathered at the end of the first phase, the Theorem 5.1 can be proved.

Proof: Let $\bar{x}_i(0)$ be a local maximum and denote with p_{iz} the length of the shortest path linking the cell r_i to a generic cell r_z. By contradiction, suppose that exists another maximum $\bar{x}_j(0)$, s.t. $\bar{x}_j(0) > \bar{x}_i(0)$, with $p_{ij} < D$. By Lemma 5.1 the maxima have the form $\bar{x}_i(0) = K_i D\delta$ and $\bar{x}_j(0) = K_j D\delta$, $K_j > K_i \in \mathbb{N}$. According to Algorithm 3, $\exists t'_k \in \mathbb{N}$ s.t.:

$$x_i(t'_k) = max(x_i(0), x_j(0) - p_{ij}\delta) \tag{5.11}$$

but being $x_i(0) - x_j(0) = K_j - K_i \geq D$, if x_j is within the area of radius $D-1$ centered at c_i, i.e. $p_{ij} < D$, x_i can not be a maximum because $x_i(t'_k) = x_j(0) - p_{ij}\delta$. □

It is worth of note that if N multiple maxima having the same value are deployed in a manner such that every maximum has at least one other homologous at distance smaller than D, then no other maxima will be present into an area given by the union of the N areas of radius $D - 1$ centered at maxima locations. The proof is omitted being very similar to the previous case.

Chapter 6

Static Sensor network localization: an initial step toward integration between static and mobile systems

In this Chapter, algorithms concerning localization of static sensor networks are discussed. This task furnishes the knowledge about the nodes location, which is essential in order to integrate mobile and static systems

6.1 Introduction

Differently from previous sections, in this chapter algorithms for Static Sensor Networks are provided. In many contexts, the interaction between static and mobile devices is essential in order to complete a task: mobile devices operate to improve the performances of a preexistent static system, which has been deployed in a certain manner on the operating scenario. Because the deployment can be done in hazardous environments, not directly accessible to the human operators, these devices can be randomly registered, i.e. launched from an airplane, and consequently have to estimate the global configuration of the network autonomously. Thus, knowledge about the likely position of the nodes can be considered a prerequisite for the interaction between static and dynamic Sensor Networks. In such a view, this Chapter can be seen as an initial step toward their integration.

Many sensor network applications require location awareness, as it is essential to know where the information is sensed. Moreover location-based routing protocols could save significant energy by eliminating the need for route discovery (102). However, adding hardware like GPS on each node or manually configuring locations are not cost effective for most sensor network applications. Thus, *localization* problem, i.e. the problem of estimating the spatial coordinates of nodes, has received a great deal of attention from several research groups during the last few years.

As explained in (103) and (104), it is hard to find a solution to this problem and it is not always possible to identify sufficient conditions for a sensor network to be localizable. The approaches proposed in literature can be grouped in centralized and distributed ones, according to the computational issues of localization techniques. Centralized algorithms suppose the availability of a central unit able to perform complex computations, exploiting the information retrieved by nodes. Distributed algorithms spread the computation over the network compensating the lack of knowledge through an intensive collaborative processing. Both of them come with advantages and drawbacks. The centralized approach achieves good performances, but cannot be applied on large networks as they are not scalable. The distributed approach provides sub optimal solution prone to error accumulation, but it results more robust and scalable.

A centralized approach is proposed in (105), where the semi definite programming (SDP) is used. This system is centralized, so that all the information are processed in one station. The gathered measures are used to constrain the node location and an optimization algorithm furnishes the network configuration in metrical terms. The result is a bounding region for each node, representing feasible locations where nodes are supposed to be. An improved SDP able to cope with noisy measurements is provided in (106). The idea is to take advantage of an additional technique to mitigate the inaccuracy of the solution provided by the SDP formulation. Authors proposed a distributed formulation, which is the result of a clusterization and a local execution of the algorithm within each subset. Therefore, the approach still remains almost centralized, while the computational load is reduced using a limited number of nodes.

A distributed approach is adopted in (107) where an algorithm focused on providing more robust local maps is developed. The idea is to split the problem into a sub-set of smaller regions in which the localization is performed by means of *robust quadrilaterals*. A robust quadrilateral is a set of four fully-connected nodes whose relative displacements lead to a configuration which is difficult to miss. After that, a proper optimization algorithm can refine the estimated configuration.

In (108), Authors describe a 2 steps localization algorithm. In the first stage, the ranging, a geometric algorithm like trilateration, is used. During the second step, the positioning, a classical *least squares* approach, is used to minimize the quadratic error about position estimate; this approach achieves better results than the classical *min-max* approach which is anyway more suitable in devices which have serious hardware constrains. A third step can be possibly executed to refine the final estimate: in this case the position of each node is refined taking into account both the information about its neighbors and the one provided by the agents which can be achieved with by a multi-hop communication protocol.

A probabilistic way is explored in (109), where the proposed algorithm aims at minimizing a functional depending on the quadratic error of distances between nodes and the prior knowledge provided by anchors, i.e., nodes whose locations are known.

In the present chapter, some well known results in robot localization (110) has been used. A probabilistic point of view is assumed and the localization problem is mapped into a stochastic estimation problem. Several Kalman-Filter based algorithms are proposed to discover the location of the nodes. These procedures work in a distributed fashion to reduce or balance the computational load. A comparison between the developed algorithms is proposed to analyze their effectiveness.

6.2 Probabilistic localization

Localization problem in a sensor network can be regarded as a stochastic estimation problem for a system described by the following equations

$$\begin{array}{rcl} q_k & = & f(q_{k-1}, u_k, w_k) \\ z_k & = & h(q_k, v_k) \end{array} \tag{6.1}$$

where q_k is a stochastic variable representing the location of the nodes, u_k is the control input, w_k and v_k are noises that affect the system, while $f(\cdot)$ and $h(\cdot)$ are mathematical relations that characterize the state transition and the observation z_k respectively.

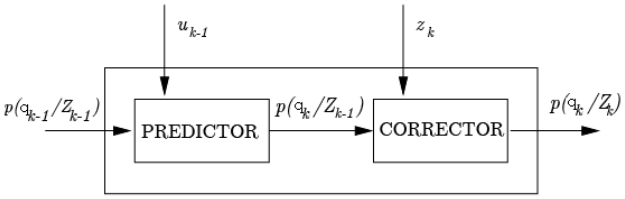


Figure 6.1: Bayesian filter.

From a Bayesian perspective, the localization problem consists in recursive computation of the probability distribution $p(q_k|Z_k, U_k)$, which describes the *joint posterior density* of the sensor locations (q_k) given the recorded observations (Z_k) and control inputs (U_k) up to time k. Assuming that the probability density function (pdf) $p(q_0|z_0, u_0)$ is available, in principle, the pdf $p(q_k|Z_k, U_k)$ may be obtained recursively, in two stages, exploiting the *state transition model*, the *observation model* , and applying the Bayes filtering (see Figure 6.1). The recursive propagation of the posterior density is only a conceptual solution in the sense that in general it cannot be determined analytically. Only in a restrictive set of cases the posterior density can be exactly and completely characterized, resorting to additional assumptions.

In (111) a linear or linearised system model and additive noises are required

$$\begin{array}{rcl} q_k & = & A_k q_{k-1} + B_k u_k + w_k \\ z_k & = & H_k q_k + v_k \end{array} \tag{6.2}$$

where $w_k \sim \mathcal{N}(0, \Xi_k)$, $v_k \sim \mathcal{N}(0, R_k)$, $q_0 \sim \mathcal{N}(\hat{q}_0, P_0)$ are mutually independent Gaussian variables for each pair of time instant (k, k'). In this way all the density function involved are Gaussian and, hence, parameterized by a mean and covariance. The mode ($\hat{q}_k$) of the joint posterior $p(q_k|Z_k, U_k)$ yields the current positions of the nodes, and the variance (P_k) represents the current uncertainty. As only these two parameters have to be computed to propagate uncertainty, there is no need to discretize the state space. In this way the prediction becomes

$$\begin{array}{rcl} \hat{q}_{k|k-1} & = & A_k \hat{q}_{k-1|k-1} + B u_k \\ P_{k|k-1} & = & A_k P_{k-1|k-1} A_k^T + \Xi_k \end{array} \tag{6.3}$$

while the correction requires the computation of the well known Kalman gain matrix

$$K_k = P_{k|k-1} H_k^T \left[H_k P_{k|k-1} H_k^T + R_k \right]^{-1} \tag{6.4}$$

before update the estimate

$$\begin{aligned} q_k &= q_{k|k+1} + K_k(z_k - z_k) \\ P_k &= P_{k|k-1} - K_k\big[J_k P_{k|k-1} J_k^T + R_k\big] K_k^T \end{aligned} \tag{6.5}$$

The advantage of Kalman filter lies in its efficiency and in the high accuracy that can be obtained: however, it is not able to cope with high nonlinear system and multimodal distributions.

Several probabilistic global methods have been proposed to overcome these drawbacks relaxing Gaussian assumption and introducing the discretization of the space state. As only Kalman filter is used in the sequel, these techniques are not reported, however a complete review can be found in (112).

6.3 Problem Setting

In this work, a group of N static nodes deployed on a planar environment is considered. The nodes are equipped with rangefinder sensors to sense distances from other team mates and with radio devices in order to exchange information. Nodes are able to exchange data, i.e., their estimated positions and the related uncertainty, whenever a node i is in the coverage area of the node j. By this way, node i acquires information about its relative position with respect to node j, and vice versa. Absolute positioning devices are mounted on few nodes, whose position is not computed as they play the role of anchors in the network. Therefore their location is assumed to be *a priori* known.

Localization problem is regarded in the framework of Bayesian filtering. The state to be estimated is represented by the positions of the nodes

$$\mathbf{q}_k = \left[q_k^{(1)T}, q_k^{(2)T}, \cdots q_k^{(N)T}\right]^T \tag{6.6}$$

where $q_k^{(i)} = [p_{x,k}^{(i)}, p_{y,k}^{(i)}]^T$ is the position of the i-th node in a global reference frame. The measurements of the system are the relative distances retrieved by range-finders, while all uncertainty sources are assumed to have Gaussian distribution. Several approaches based on Extended Kalman Filter (EKF) are applied to estimate the location of the nodes.

As nodes are assumed to be static, the state transition model can be implemented as

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \mathbf{w}_k \tag{6.7}$$

where $\mathbf{w}_k \in \mathbb{R}^{2N}$ is a zero mean white noise vector with covariance matrix Ξ_k. The covariance update can be computed according as

$$P_k = P_{k|k-1} + \Xi_k \tag{6.8}$$

The observations of the system are represented by rangefinder measurements of all nodes. For each node i the measurements are composed by m_i^n sub-vectors

$$z_k^{(i,j)} = h^n(q_k^{(i)}, q_k^{(j)}) = \sqrt{(p_{x,k}^{(j)} - p_{x,k}^{(i)})^2 + (p_{y,k}^{(j)} - p_{y,k}^{(i)})^2} \tag{6.9}$$

where m_i^n is the number of nodes in the viewing area of the i-th node. When an anchor is involved in the measurement, the observation can be written as:

$$z_k^{(i,j)} = h^a(q_k^{(i)}, \mathcal{M}) = \sqrt{(a_x^{(j)} - p_{x,k}^{(i)})^2 + (a_y^{(j)} - p_{y,k}^{(i)})^2} \tag{6.10}$$

being m_i^a the number of anchors in the viewing area of the i-th node and $(a_x^{(j)}, a_y^{(j)})$ the position of an anchor. It should be noted that $h^a(\cdot)$ depends on a map $\mathcal{M}$, here represented by a list of anchors position $(a_x^{(j)}, a_y^{(j)})$.

Due to the non linearity of the mapping, the Jacobian (J_q^h) of the map $h(\mathbf{q}_k)$ with respect to $\mathbf{q_k}$ has to be used in the computation of Kalman gain

$$K_k = P_{k|k-1} J_k^{h^T} \left[J_k^h P_{k|k-1} J_k^{h^T} + R_k \right]^{-1}. \tag{6.11}$$

The Jacobian is composed by m_i^n sub vectors representing the Jacobian of the maps $h^n(\cdot)$ and m_i^a sub vectors representing the Jacobian of the maps $h^a(\cdot)$. Below is reported the Jacobian of the map $h^n(\cdot)$

$$\begin{aligned} j^h(q_k^{(i)}, q_k^{(j)}) &= \left[\dots \frac{\partial h^n(\cdot)}{\partial x_i} \ \frac{\partial h^n(\cdot)}{\partial y_i} \dots \frac{\partial h^n(\cdot)}{\partial x_j} \ \frac{\partial h^n(\cdot)}{\partial y_j} \dots \right] \\ &\quad \left[\mathbf{0} \ \ \frac{dx}{z_k^{(i,j)}} \ \frac{dy}{z_k^{(i,j)}} \ \ \mathbf{0} \ \ \frac{-dx}{z_k^{(i,j)}} \ \frac{-dy}{z_k^{(i,j)}} \ \ \mathbf{0} \right] \end{aligned} \tag{6.12}$$

where $dx = (p_{x,k}^{(j)} - p_{x,k}^{(i)})$ and $dy = (p_{y,k}^{(j)} - p_{y,k}^{(i)})$.

As it can be intuitively argued, to avoid ambiguous configuration, a number of at least 3 non-aligned anchors is needed.

6.4 Distributed Localization Algorithms

In this section three different algorithms based on extended Kalman Filter for localization in a sensor network are reported. All the proposed algorithms aim at distributing the computational load of the KF over the network, exploiting the decoupling state transition model of the system. In the update step, different strategies are adopted to deal with the interaction in the measurement process between nodes.

6.4.1 Distributed Complete EKF

The Distributed Complete Extended Kalman Filter (DCKF) is devoted to determine an estimate for the whole system state given in eq. (6.6). The estimate is retrieved applying Kalman Filter by means of distributing the computational load over the sensor network. It is easy to notice, indeed, that the state transition model given in eq (6.7) is suitable for a distributed implementation, since it is linear and fully decoupled. Unfortunately, during the correction step, the computation of the expected measurement according to eq. (6.9) and the update of the covariance matrix $P_{k|k}$ feature some couplings. In order to reduce the computational burden, it is possible to split the covariance matrix in N^2 sub matrices, having dimension $[2 \times 2]$. Each sub-matrix P_{ij} represents the relation between a couple of nodes i and j; according to this notation, sub-matrices P_{ii} are the ones describing the statistical features about the estimate of a single node.

$$P_{k|k-1} = \begin{bmatrix} P_{ii}^{-} & \dots & P_{ij}^{-} & \vdots \\ \vdots & \ddots & \dots & \vdots \\ P_{ji}^{-} & \ddots & P_{jj}^{-} & \vdots \\ \vdots & \dots & \dots & \ddots \end{bmatrix}. \tag{6.13}$$

As far as only one measurement between two nodes i-j is considered at each instant of time k, it is easy to show that, due to the observation model:

- $S_k = \left[J_k^h P_{k|k-1} J_k^{h^T} + R_k\right]$ is a scalar and depends only on the Jacobian of the output map and the sub matrices of $P_{k|k-1}$ related with the nodes involved in the measurement, i.e. P_{ii}^{-},P_{ij}^{-} and P_{ji}^{-};
- the m–th element of the vector K_k depends on the Jacobian of the output map, S_k and the cross-correlation matrices P_{mi}^{-} and P_{mj}^{-}.

In this way, each node is able to perform the correction of the estimate using only the Jacobian of the output map, a subset of the matrix $P_{k|k-1}$ and S_k. Moreover, due to the symmetric structure of $P_{k|k-1}$, the computation of the covariance matrix can be reduced, too. Supposing that each node computes its own auto correlation matrix P_{ii}^{+}, there are still other $\frac{N}{2} \cdot (N-1)$ cross correlation sub matrices to be computed. As shown in (113), the computational load can be split in a way such that every node computes about $\frac{N-1}{2}$ sub matrices. However, this operation requires that a computational balance step is performed at the beginning to assign a subset of sub matrices P_{ij}^{-} to each node i. At each instant k, nodes share information about their estimated positions and the subset of cross-correlation matrices, producing a high level of traffic over the net.

6.4.2 Local EKF

In the case of Local Extended Kalman Filter (LKF), the whole state of the system given in eq. (6.6) is decomposed into N sub vectors each of which containing the position of a sensor: each node computes an extended Kalman filter to estimate its own location. During the update, every unit needs to communicate only with its neighbors to know their configurations, therefore the traffic over the network is considerably reduced. Moreover the computational load is lowered since the cross correlation terms are neglected. For the same reason, the convergence of the filter is quickly achieved, whilst the estimate results quite inaccurate. It is worth noting that this filter is used only to gather a lower bound about performances of the other two algorithms.

6.4.3 Compressed EKF

The Compressed Extended Kalman Filter (CKF) has been introduced by (110) to solve simultaneous localization and map building problem, however it can be also applied to localization in sensor networks. At each time k, nodes collaborate to compute an estimate for the state given in eq. (6.6), according to the full Extended Kalman Filter equations (6.7)–(6.12). The computational load is split over the network as presented in DCKF, however only few nodes, called *preserved nodes* are involved in the update process, while the state and the cross covariance matrices of the remaining nodes, *discarded nodes* ($\mathcal{S}$), are not updated. When a new observation has a significant information contribution for a node, such a node is considered in the computation of cross-correlation terms, otherwise it increases the set of neglected nodes $\mathcal{S}$. The eq. (6.5), using a more compact form, can be written as:

$$P_k = P_{k|k-1} - \delta P.$$

In order to quantify the significance of the information that a certain observation furnishes, the diagonal elements $P_{k|k-1}(i,i)$ of $P_{k|k-1}$ and $\delta P(i,i)$ of δP are considered. If they are enough small, the cross correlation terms between two nodes, that respect this statements, are not computed as it gives an irrelevant contribution to the estimate compared to computational costs. In other terms, denoted by $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{S}$ three sets such that:

- $q^{(i)} \in \mathcal{S}_1$ if $P(i,i) < \alpha, \quad \alpha > 0$
- $q^{(i)} \in \mathcal{S}_2$ if $\delta P(i,i) < \beta \cdot P(i,i), \quad \beta > 0$
- $q^{(i)} \in \mathcal{S}$ if $i \in S_1 \quad or \quad i \in S_2$

if two elements $(q^{(i)}, q^{(j)})$ belong to the set $\mathcal{S}$, then the cross correlation correction term $\delta P(i,j)$ is not computed. Choosing properly α and β, a compromise between performances and computational load can be achieved. As it is shown in (110), this approach leads to a sub optimal solution, due to the neglecting terms in the correction step; on the other side, the traffic generated on the net is considerably reduced with respect to DCFK.

6.5 Experiments

In this thesis, the proposed algorithms have been widely tested both in simulation and by means of trials in real world environment. The simulations have been carried out using a framework developed in order to stress the algorithms under different operating conditions. The real world experiments have been carried out acquiring data retrieved by a MICAz sensor network from Crossbow and processing them using MatLab.

To emphasize the interaction between nodes, an ”all to all” topology has been considered. Although this topology is not common in real sensor network, it can be associated to clusters of nodes and exploited to decompose the localization algorithm in a hierarchical fashion.

6.5.1 Simulation tests

A network of 30 sensors deployed in a square environment of 10 × 10 units has been adopted for the simulation tests (see Figure 6.2). To prove the reliability of the proposed algorithms , two sets of experiments have been considered: the sets differ in the number of the anchors, respectively 3 and 4, located on the corners of the square. The results retrieved using the two different configurations are summarized in Tables 6.1 and 6.2.

The output of the simulations has been averaged over 20 trials, each one composed by a fixed number (15) of ”all to all” communication steps. Initial conditions were corrupted by a white noise, as detailed in the same tables, while measures have been perturbed by a random noise with standard deviation equal to 0.2 units. The Euclidian distance between the real position of a node and the estimate has been adopted as the metric for the estimation accuracy. During the simulation, it was supposed that a node i retrieves its own measurements and performs $N-1$ update steps: this rigid policy has been chosen because it reflects the used procedure in real world experiments.

According with the results in Tables 6.1 and 6.2 , DCKF and CKF are able to estimate the position of the sensors in the network with the same accuracy. In Figure 6.4 the evolution of the position error of a node during a trial is shown. As one can notice, the CFK rapidly converge to a small error by means of a smooth trajectory, whilst the convergence of DCKF is slower. This behavior can be explained considering the coupling between nodes. In DCKF a small change in the estimate of a node is propagated all over the network and affects all the nodes. During an iteration of CKF only few nodes are coupled and new poor information is automatically discarded.

In Figure 6.3 the cardinality of the set $\mathcal{S}$ in a trial is reported. When the algorithm starts, all nodes are coupled, as the uncertainty on the initial position is high and any new information is considered reliable. As soon as the estimate is firmed, the cardinality of $\mathcal{S}$ increases and only the interaction with a small subset of neighbors is evaluated. During the trial, the number of preserved nodes increases if the estimate becomes inaccurate: in this sense the CKF algorithm shows a self adaptive behavior.

LKF is reported to provide an upper bound in distributing the complete filter over the network. It shows bad performances due to the inaccurate estimate process. In Figure 6.4 the evolution of the filter changes only when the sample node performs measurement, according to the fully decoupled approach of the algorithm.

6.5.2 Experimental tests

Experimental results have been performed to validate the proposed approaches in a real context. The network has been deployed using MICAz (MPR2400) platform from Crossbow Technology. The MPR2400 (2400 MHz to 2483.5 MHz band) uses the Chipcon CC2420, IEEE 802.15.4 compliant, ZigBee ready radio frequency transceiver integrated with an Atmega128L micro-controller. The MICAz platform is designed to

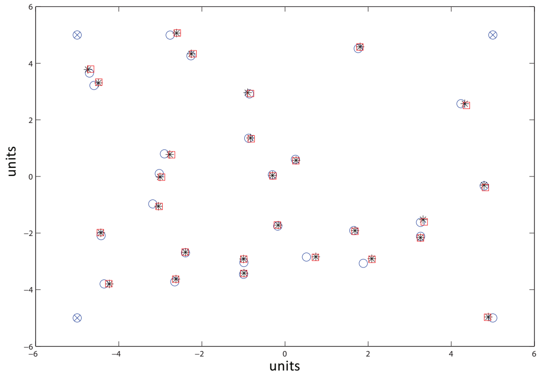


Figure 6.2: Network deployment: anchors (crossed circles), nodes (circles), GCKF estimate (asterisks), CFK estimate (squares).

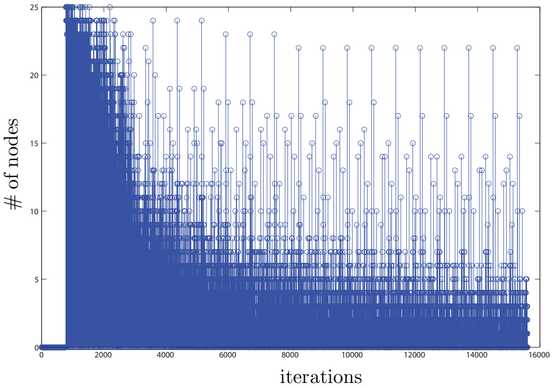


Figure 6.3: Cardinality of set D during a trial.

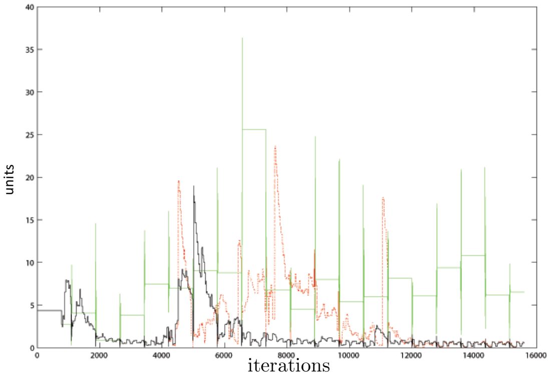


Figure 6.4: Trajectory error of a node, simulation results: CKF (black solid line), DCKF (red dashed line), LKF (green solid line).

Table 6.1: Simulative test with square area and four anchors.

	Cov $(unit^2)$	$\mathbf{e_{mean}}$ $(unit)$	$\mathbf{e_{min}}$ $(unit)$	$\mathbf{e_{max}}$ $(unit)$
DCKF	0.0028	0.1008	0.0166	0.2136
CKF	0.1742	0.1585	0.0187	0.7092
LKF	16.1544	10.0622	3.9647	18.4137
Init.Cond.	8.5221	4.2349	0.093	3.4345

Table 6.2: Simulative tests with square area and three anchors.

	Cov (m^2)	$\mathbf{e_{mean}}$ (m)	$\mathbf{e_{min}}$ (m)	$\mathbf{e_{max}}$ (m)
DCKF	0.1549	0.0955	0.0197	0.2265
CKF	0.1627	0.0964	0.0197	0.2239
LKF	2.7763	11.0066	4.5781	20.4064
Init.Cond.	1.0473	2.5388	0.4722	5.6635

mount several sensor boards, like the MTS310 used in these tests. MICAz platform comes along with TinyOS, an open source event-driven operating system developed for wireless embedded sensor networks. TinyOS component library includes network protocols, distributed services, sensor drivers and data acquisition tools which can be further modified to adapt them to specific performances.

A ranging technique based on the Time of Arrival (ToA) principle was exploited to compute inter-node distances. The implementation consists of a node sending first a RF packet and emitting an acoustic signal right after. The propagation of the RF packet is assumed to be instantaneous and the signal is used to trigger a timer on the receiving nodes. Such timer stops when the acoustic pulse is received. Measuring the time of propagation, it is possible to relate this time with the distance traveled by the acoustic pulse.

A network composed by 7 nodes and 3 anchors has been deployed over a squared area (2 × 2m), as Figure 6.5 shows. The collected data has been processed off-line, using the proposed algorithm and different initial conditions to set up different trials. The results, averaged over 20 trials, are collected in Tab. 6.3.

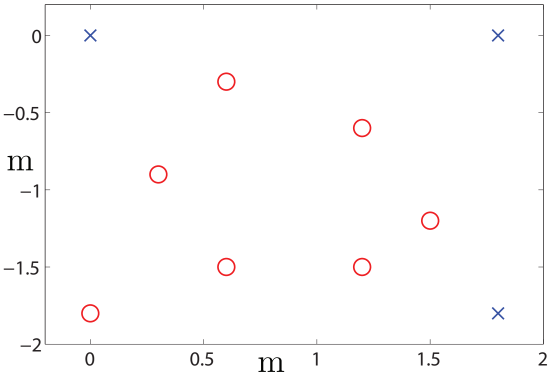


Figure 6.5: Network deployment: nodes (red circles), anchors (blue asterisks).

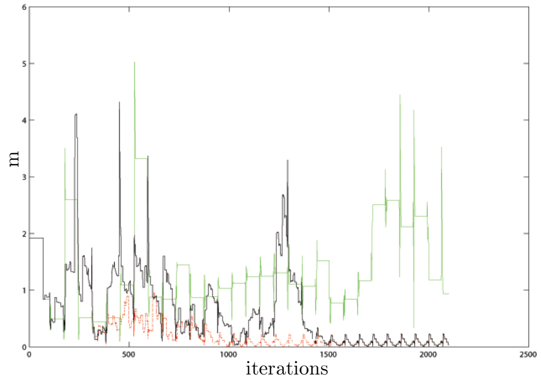


Figure 6.6: Error trajectory for a node, real experiments: DCKF (black solid line), DCKF (red dashed line), LKF (green solid line).

This table shows that DCKF and CKF achieve the same results in terms of accuracy, however, as mentioned before, the DCKF is able to tune the coupling between nodes, reducing the overall computational load. According to the experimental results for this configuration, as expected, LKF presents the worst performances. This can be explained by the fact that DCKF and CKF exploit the coupling between nodes, while LKF does not take any advantage by this knowledge. Moreover, due to the coupling between nodes in DCKF and CKF, an inaccurate initial condition can be recovered, whilst LKF tends to converge toward an alternative admissible solution due to the symmetrical layout of the network as shown in Figure 6.6.

Table 6.3: Real tests with square area and three anchors

	Cov $(unit^2)$	$\mathbf{e_{mean}}$ $(unit)$	$\mathbf{e_{min}}$ $(unit)$	$\mathbf{e_{max}}$ $(unit)$
DCKF	0.0444	0.1032	0.0400	0.1661
CKF	0.0031	0.1454	0.0458	0.2866
LKF	1.1714	1.8820	0.8323	3.6220
Init.Cond.	0.1129	0.6558	0.2823	1.1355

6.6 Conclusion

In this chapter, three different localization algorithms, based on Extended Kalman Filter, to solve localization problem in a sensor network have been proposed. The algorithms, well known in robotics, work in a decentralized cooperative manner so to distribute the computational load on each node of the network.

To prove the effectiveness of the approaches, extensive simulation tests have been carried out using both simulative and experimental tests.

Although the results obtained are promising, several interesting challenges still remain for future works. A low computational load procedure for topology discovery and load balancing could be investigated and implemented on the nodes of the network. A comparison with the information form of Kalman filter could be useful to understand different strategies to reduce the computational burden in the update step.

Regarding large networks, a probabilistic clusterization induced by the discarded and preserved nodes could be analyzed with respect to the technique achieved by spatial classification.

Chapter 7

Conclusions

This dissertation has been focused on multi robot systems. In particular 3 topics have been treated.

The first is related to the hardware prototyping of mobile robots. In particular the realization of a multi-robot system has been proposed. This platform has been successfully designed and several tests have been conducted to demonstrate its effectiveness.

However this project can be considered only at initial stage: although the core software has been developed, the system is still susceptible of improvements. A structured scheme for the implementation of higher level tiers has to be introduced. Furthermore the integration with standard sensor networks seems to be the next logical step. The experimental validation of these hybrid systems could lead to very interesting results.

Concerning the encircling problem, two approaches have been demonstrated to be effective on real robots. The major contribution of this work is the realization of a fully autonomous system which is able to execute the task without any adjunct external sensor.

Further directions concern the study of algorithms which execute the encircling task taking into account collision avoidance constraints: the proposed approaches show some promising properties although the complexity, i.e. the switching dynamics, makes it an hard problem.

Regarding the data fusion, the major contribution is related to the extension of the Transferable Belief Model to distributed systems. Currently the proposed technique is suitable only for tree topologies: a future work direction is to encompass also graph topologies.

In environmental monitoring, which is a rather new field in robotics, several directions could be taken. In this dissertation a stigmergic approach focused on the low-cost map-building has been proposed; however this task, in general, is susceptible to improvements. In particular the dynamics of the gas dispersal constitute the major issue: this complexity has to be coupled with robotic architectures which are constrained by computational and power resources. Future directions take into account models which can help the robots both to model the gas advection and to fuse information in an effective manner. One of the main limitations actually present in the majority of works is the weak responsiveness of the estimation system both with respect to sudden variations and to the complexity of the operating scenario.

This thesis describes a work pointing toward multiple research directions. This feature implies pros and cons. Pros are constituted by the fact that many aspects related to the design of multi-robot systems have been analyzed and validated with several tests: the thesis describes the realization from scratch of a platform (SAETTA) for a multi-robot system first, and then several techniques which fit the characteristics of the SAETTA platform. In some cases these techniques are new, in others they have been adapted to the specific context so they constituted the basis for scientific publications. In every chapter, at least one real experiment has been described and SAETTA robots have been used (with the exception of Chapter 3 and Chapter 6).

At the same time, the amount of experimental work, coupled with the heterogeneity of the faced topics, inevitably hindered the depth of the analysis referred in the various chapters. For example, compared to a thesis focused on a single topic, this thesis provides a shorter bibliography and a less than exhaustive comparison with respect to the state of the art for each topic.

Bibliography

[1] M. Di Rocco and G. Ulivi, “An efficient implementation of a particle filter for localization using compass data,” in *7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy, 09/2010 2010. 4

[2] M. Di Rocco, F. La Gala, and G. Ulivi, “Saetta: a small and cheap mobile unit to test multi-robot algorithms,” *Submitted to IEEE Robotics and Automation Magazine*, 2010. 4

[3] A. Franchi, P. Stegagno, M. Di Rocco, and G. Oriolo, “Distributed target localization and encirclement with a multi-robot system,” in *7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy, 09/2010 2010. 5

[4] M. Di Rocco, A. Priolo, and S. Panzieri, “Formation control through environment pattern recognition for a multi-robot architecture,” in *European Conference on Mobile Robots (ECMR 2009)*, Milini/Dubrovnik Croatia, September 2009. 5, 69

[5] M. Di Rocco, F. Fiorini, A. Gasparri, and S. Panzieri, “A networked transferable belief model approach for distributed data aggregation - static version,” in *49th IEEE Conference on Decision and Control, CDC10*, 2010. 5, 102

[6] ——, “A networked transferable belief model approach for distributed data aggregation - dynamic version,” in *49th IEEE Conference on Decision and Control, CDC10*, 2010. 5

[7] C. Carletti, M. Di Rocco, A. Gasparri, and G. Ulivi, “A distributed transferable belief model for collaborative topological map-building in multi-robot systems,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010. 5

[8] M. Di Rocco, F. Fiorini, A. Gasparri, and S. Panzieri, "A networked transferable belief model approach for distributed data aggregation," *Submitted to IEEE Transactions on Systems, Man and Cybernetics*. 5

[9] M. Di Rocco, M. Reggente, and A. Saffiotti, "Gas source localization using inexpensive robots and stigmergy in domestic environments," in *Submitted to Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011. 5

[10] M. Di Rocco and F. Pascucci, "Probabilistic localization in sensor networks using distributed kalman filter," in *7th IFAC Symposium on Intelligent Autonomous Vehicles*, Lecce, Italy, 09/2010 2010. 5

[11] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *Int. Journal of Robotics Research*, vol. 23, pp. 415–436, 2004. 8

[12] J. Mclurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," in *in 7th International Symposium on Distributed Autonomous Robotic Systems (DARS*, 2004. 8

[13] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo, "The cooperation of swarm-bots: Physical interactions in collective robotics," 2005. 8

[14] R. A. F. P. Yang and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Transactions on Automatic Control*, to appear. 8

[15] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion with limited communication," *IEEE Trans. Automat. Contr.* 8

[16] B. P. Douglass, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. 13

[17] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B. Seo, and Y. Cho, "The PEIS-ecology project: vision and results," in *Proc of the*

IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS), Nice, France, 2008, pp. 2329–2335, online at http://www.aass.oru.se/˜asaffio/. 20, 126

[18] R. Olfati-saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proceedings of the IEEE*, 2007, p. 2007. 20, 24

[19] R. Johansson and A. Saffiotti, "Navigating by stigmergy: A realization on an RFID floor for minimalistic robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Kobe, JP, 2009, online at http://www.aass.oru.se/˜asaffio/. 20, 117, 120, 123, 124, 126, 133

[20] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *11th International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, Jun. 2003. [Online]. Available: citeseer.ist.psu.edu/gerkey03playerstage.html 21, 69

[21] M. Broxvall, B. S. Seo, and W. Y. Kwon, "The peis kernel: A middleware for ubiquitous robotics," in *Proceedings of the IEEE/RSJ International Workshop on Ubiquitous Robotic Space Design and Applications*, 2007. 21

[22] M. Bordignon, J. Rashid, M. Broxvall, and A. Saffiotti, "Seamless integration of robots and tiny embedded devices in a peis-ecology," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 200t. 21

[23] D. P. Bovet and M. C. Ph, *Understanding the Linux Kernel, Third Edition*, 3rd ed. O'Reilly Media, November 2005. 22

[24] A. D. Luca and G. Oriolo, "Local incremental planning for nonholonomic mobile robots," in *In Proceedings of 1994 International Conference on Robotics and Automation*, 1994, pp. 104–110. 24

[25] J. Handshin, "Monte carlo techniques for prediction and filtering of non-linear stochastic processes," pp. 555–563. 28

[26] P. Katz, *Digital Control using Microprocessors*, P. Hall, Ed., 1981. 31

[27] A. Oppenheim and R. Schafer, *Digital Control using Microprocessors*, P. Hall, Ed., 1981. 31

[28] V. Gazi and K. Passino, "Stability analysis of social foraging swarms: combined effects of attractant/repellent profiles," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3, 10-13 Dec. 2002, pp. 2848–2853vol.3. 38

[29] N. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. of the 40th IEEE Conf. on Decision and Control*, vol. 3, 4-7 Dec. 2001, pp. 2968–2973. 38

[30] R. Sepulchre, D. Paley, and N. E. Leonard, "Stabilization of planar collective motion with all-to-all communication," *IEEE Trans. on Automatic Control*, 2007. 38

[31] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of planar collective motion with limited communication," *IEEE Trans. Automat. Contr.* 38

[32] N. Moshtagh, N. Michael, A. Jadbabaie, and K. Daniilidis, "Vision-based, distributed control laws for motion coordination of nonholonomic robots," *IEEE Trans. on Robotics*, 2008, accepted. 38

[33] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The entrapment/escorting mission for a multi-robot system: Theory and experiments," in *Proc. of the 2007 IEEE/ASME Int. Conf. on Advanced intelligent mechatronics*, 2007, pp. 1–6. 38

[34] N. Ceccarelli, M. Di Marco, A. Garulli, and A. Giannitrapani, "Collective circular motion of multi-vehicle systems," *Automatica*, vol. 44, no. 12, pp. 3025–3035, 2008. 39

[35] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. on Automatic Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005. 39

[36] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Trans. Automatic Control*, vol. 50, no. 1, pp. 121–127, 2005. 39

[37] R. Saber and R. Murray, "Consensus protocols for networks of dynamic agents," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 2, June 4-6, 2003, pp. 951–956. 39, 41

[38] W. Ren, "Consensus seeking in multi-vehicle systems with a time-varying reference state," in *2007 American Control Conf.*, 9-13 July 2007, pp. 717–722. 39

[39] ——, "Consensus strategies for cooperative control of vehicle formations," *Control Theory & Applications, IET*, vol. 1, no. 2, pp. 505–512, March 2007. 39

[40] G. Nitschke, "Co-evolution of cooperation in a pursuit evasion game," *Proc. of the 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 2037–2042, Oct. 2003. 39

[41] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002. 39

[42] S. D. Bopardikar, F. Bullo, and J. P. Hespanha, "A cooperative Homicidal Chauffeur game," *Automatica*, Jul. 2008, provisionally Accepted. 39

[43] B. P. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *Int. J. Rob. Res.*, vol. 25, no. 4, pp. 299–315, 2006. 39

[44] A. Franchi, P. Stegagno, and G. Oriolo, "Mutual localization in a multi-robot system with anonymous relative position measures," in *2009 IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, St. Louis, MO, USA, Oct 2009. 40, 53, 54, 62

[45] A. Franchi, G. Oriolo, and P. Stegagno, "On the solvability of the mutual localization problem with anonymous position measures," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, submitted. 42, 43

[46] A. Franchi, P. Stegagno, and G. Oriolo, "Mutual localization in a multi-robot system with anonymous relative position measures," *In 2009 IEEE/RSJ Int. Conf. on Intelligent Robot & Systems*, October 2009. 43, 96

[47] ——, "Probabilistic mutual localization in multi-agent systems from anonymous position measures," in *49th IEEE Conference on Decision and Control*, Atlanta, GA, USA, 12/2010 2010, pp. 6534–6540. 43

[48] G. Oriolo, A. D. Luca, R. D. Luca, and M. Vendittelli, “Wmr control via dynamic feedback linearization: Design, implementation, and experimental validation,” 2002. 43, 44

[49] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, “Closed loop steering of unicycle-like vehicles via lyapunov techniques,” pp. 27–35, 1995. 43

[50] M. Fliess, J. Lvine, and P. Rouchon, “Flatness and defect of nonlinear systems: Introductory theory and examples,” *International Journal of Control*, vol. 61, pp. 1327–1361, 1995. 44

[51] A. Franchi, P. Stegagno, M. D. Rocco, and G. Oriolo, “Distributed target localization and encircling with a multi-robot system,” *7th IFAC Symposium on Intelligent Autonomous Vehicles*, 2010. 45

[52] A. Benzerrouk, L. Adouane, P. Martinet, and N. Andreff, “Multi lyapunov function theorem applied to a mobile robot tracking a trajectory in presence of obstacles,” in *European Conference on Mobile Robots (ECMR 2009)*, Milini/Dubrovnik Croatia, September 2009. 52

[53] M. R. Ghahroudi and R. Sabzevari, *Sensor and Data Fusion*, N. Milisavljevic, Ed., 2009. 63

[54] A. Doucet, N. De Freitas, and N. Gordon, Eds., *Sequential Monte Carlo methods in practice*, 2001. 63

[55] J. Schwertfeger and O. Jenkins, “Multi-robot belief propagation for distributed robot allocation,” in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*, july 2007, pp. 193 –198. 63

[56] J. Schiff, E. Sudderth, and K. Goldberg, “Nonparametric belief propagation for distributed tracking of robot networks with noisy inter-distance measurements,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 1369 –1376. 63

[57] G. Shafer, *A Mathematical Theory of Evidence.* Princeton: Princeton University Press, 1976. 64, 100

[58] P. Smets, “Information content of an evidence,” *Int. J. Man Machine Studies*, vol. 19, pp. 33–43, 1983. 64

[59] P. Smets and R. Kennes, “The transferable belief model,” *Artificial Intelligence*, vol. 66, pp. 191–234, 1994. 64, 66

[60] *Fast Algorithms for Dempster-Shafer Theory.* London, UK: Springer-Verlag, 1991. 64

[61] T. Denoeux, “Inner and outer approximation of belief structures using a hierarchical clustering approach,” *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 9, no. 4, pp. 437–460, 2001. 64

[62] Y. Denoeux, “Approximating the combination of belief functions using the fast moebius transform in a coarsened frame,” *International Journal of Approximate Reasoning*, vol. 31, no. 1-2, pp. 77–101, 2002. 64

[63] P. Smets, “The application of the transferable belief model to diagnostic problems,” *Int. J. Intelligent Systems*, vol. 13, pp. 127–157, 1998. 64

[64] F. Delmotte and P. Smets, “Target identification based on the transferable belief model interpretation of Dempster-Shafer model.” *IEEE Transactions on Systems, Man and Cybernetics*, vol. A 34, pp. 457–471, 2004. 64

[65] *State estimation using interval analysis and belief function theory: Application to dynamic vehicle localization.*, 2009. 64

[66] *Risk Assessment Based on Weak Information using Belief Functions: A Case Study in Water Treatment*, 2006. 64

[67] L. A. Zadeh, “On the validity of dempster rule of combination,” in *In Memo M 79/24*. Univ. of California, Berkeley, 1979. 66

[68] P. Smets, “The combination of evidence in the transferable belief model,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 447–458, 1990. 66, 78, 102

[69] G. Oriolo, G. Ulivi, and M. Vendittelli, “Real-time map building and navigation for autonomous robots in unknown environments,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, pp. 316–333, 1999. 67, 95

[70] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2508–2530, 2006. 72, 73

[71] K. Johansen, U. Jørgensen, S. Nielsen, S. Nielsen, and S. Skyum, "A distributed spanning tree algorithm," *Distributed Algorithms*, pp. 1–12, 1988. [Online]. Available: http://dx.doi.org/10.1007/BFb0019790 73

[72] Y. Dalal, "A distributed algorithm for constructing minimal spanning trees," *Software Engineering, IEEE Transactions on*, vol. SE-13, no. 3, pp. 398–405, March 1987. 73

[73] S. Thrun, "Robotic mapping: a survey," *Exploring artificial intelligence in the new millennium*, pp. 1–35, 2003. 95

[74] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Hybrid simultaneous localization and map building: a natural integration of topological and metric," *Robotics and Autonomous Systems*, vol. 44, no. 1, 2003. 95

[75] S. Thrun and A. Buecken, "Integrating grid-based and topological maps for mobile robot navigation," in *AAAI 1996*, 1996. 95

[76] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE RAS*, vol. 13, no. 2, pp. 99–110, June 2006. 95

[77] D. Fox, F. S., and P. H., "Voronoi random fields: extracting topological structure of indoor environments via place labeling," in *Proc. of the International Joint Conference on Artificial Intelligence*, 2007. 95

[78] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *Int. J. Rob. Res.*, vol. 25, no. 12, pp. 1243–1256, 2006. 95

[79] S. Panzieri, D. Petroselli, and G. Ulivi, "Topological localization on indoor sonar based fuzzy maps," in *Intelligent Autonomous System 6*. IOS Press, 2000. 95

[80] D. Dubois and H. Prade, "Possibility theory, probability theory and multiple-valued logics: A clarification," *Annals of Mathematics and Artificial Intelligence*, vol. 32, pp. 35–66, 2001. 95

[81] S. Yadlapalli, S. Darbha, and K. Rajagopal, “Information flow and its relation to stability of the motion of vehicles in a rigid formation,” *IEEE TAC*, vol. 51, no. 8, pp. 1315 –1319, 2006. 96

[82] L. Krick, M. Broucke, and B. Francis, “Stabilisation of infinitesimally rigid formations of multi-robot networks,” *International Journal of Control*, vol. 82, no. 3, pp. 423 – 439, 2009. 96

[83] P. Hough, “Method and means for recognizing complex patterns.” U. S. Patent 3, 069654, December, 18 1962. 100

[84] B. Shraiman and E. Siggia, “Scalar turbulence,” *Nature*, no. 405, pp. 639–646, June 2000. 116

[85] C. H. Zah and S. Fezer, “Embedded sensor scout flooring system by interface flor,” nov. 2008, pp. 106 –110. 116

[86] S. Han, H. Lim, and J. Lee, “An efficient localization scheme for a differential-driving mobile robot based on rfid system,” *Industrial Electronics, IEEE Transactions on*, vol. 54, no. 6, pp. 3362 –3369, dec. 2007. 116

[87] K. Kodaka, H. Niwa, Y. Sakamoto, M. Otake, Y. Kanemori, and S. Sugano, “Pose estimation of a mobile robot on a lattice of rfid tags,” sep. 2008, pp. 1385 –1390. 116

[88] A. J. Lilienthal, M. Reggente, M. Trincavelli, J. L. Blanco, and J. Gonzalez, “A statistical approach to gas distribution modelling with mobile robots the kernel dm+v algorithm,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 11 – October 15 2009, pp. 570–576. 117, 118, 120, 122, 128

[89] E. Bonabeau, “Editor’s introduction: Stigmergy,” *Artificial Life*, vol. 5, no. 2, pp. 95–96, 1999. 117

[90] O. Holland and C. Melhuish, “Stigmergy, self-organization, and sorting in collective robotics,” *Artif. Life*, vol. 5, no. 2, pp. 173–202, 1999. 117

[91] A. T. Hayes, A. Martinoli, and R. M. Goodman, “Distributed odor source localization,” *IEEE Sensors Journal, Special Issue on Electronic Nose Technologies*, vol. 2, no. 3, pp. 260–273, 2002. 117, 118

[92] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario, “Explorative particle swarm optimization method for gas/odor source localization in an indoor environment with no strong airflow,” dec. 2007, pp. 841 –846. 118

[93] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli, “A comparison of casting and spiraling algorithms for odor source localization in laminar flow,” in *ICRA*, 2008, pp. 1138–1143. 118

[94] T. Lochmatter and A. Martinoli, “Theoretical Analysis of Three Bio-Inspired Plume Tracking Algorithms,” in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, 2009. 118

[95] H. Ishida, T. Nakamoto, and T. Moriizumi, “Remote sensing of gas/odor source location and concentration distribution using mobile system,” *Sensors and Actuators B*, vol. 49, pp. 52–57, 1998. 118

[96] A. H. Purnamadjaja and R. A. Russell, “Congregation behaviour in a robot swarm using pheromone communication,” in *Proc. ACRA*, 2005. 118

[97] P. Pyk, S. Bermúdez I Badia, U. Bernardet, P. Knüsel, M. Carlsson, J. Gu, E. Chanie, B. S. Hansson, T. C. Pearce, and P. F. J. Verschure, “An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search,” *Auton. Robots*, vol. 20, no. 3, pp. 197–213, 2006. 118

[98] A. Lilienthal and T. Duckett, “Building gas concentration gridmaps with a mobile robot,” *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 3–16, 2004. 118

[99] M. Reggente and at al, “The dustbot system: Using mobile robots to monitor pollution in pedestrian area,” in *Proc. of NOSE*, 2010. 118

[100] M. Di Rocco and G. Ulivi, “An efficient implementation of a particle filter for localization using compass data,” in *IAV ’10*, 2010.

[101] I. Horswill, “Analysis of adaptation and environment,” *Artificial Intelligence*, vol. 73, pp. 1–30, 1995. 127

[102] J. W. Martin Mauve and H. Hartenstein, “A survey on position-based routing in mobile ad-hoc networks,” *IEEE Network Magazine*, 2001. 136

[103] B. Jackson and T. Jordán, “Connected rigidity matroids and unique realizations of graphs,” *J. Comb. Theory Ser. B*, vol. 94, no. 1, pp. 1–29, 2005. 136

[104] T. Eren, D. Goldenberg, W. Whitley, Y. Yang, A. Morse, B. Anderson, and P. Belheumer, “Rigidity computation and randomization of network localization,” in *Proc. IEEE Conf. Computer Comm. (Infocom 2004)*, March 2004. 136

[105] L. Doherty, L. E. Ghaoui, and K. S. J. Pister, “Convex position estimation in wireless sensor networks,” in *Proceedings of Infocom*, 2001. 136

[106] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, “Semidefinite programming approaches for sensor network localization with noisy distance measurements,” in *IEEE Transaction on Automation Science and Engineering*, October 2006, vol. 3. 136

[107] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements,” in *SenSys ’04: Proceedings of the 2nd international conference on Embedded networked sensor systems.* New York, NY, USA: ACM Press, 2004, pp. 50–61. 136

[108] K. Langendoen and N. Reijers, “Distributed localization in wireless sensor networks: a quantitative comparison,” *Comput. Networks*, vol. 43, no. 4, pp. 499–518, 2003. 137

[109] J. A. Costa, N. Patwari, and A. O. H. III, “Adaptive distributed multidimensional scaling for localization in sensor networks,” 2006. 137

[110] J. Guivant and E. Nebot, “Optimization of simultaneous localization and map building algorithm for real time implementation,” *IEEE Trans. on Robotics and Automation*, vol. vol.17, no. no 3, 2001. 137, 142, 143

[111] R. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions ASME Journal of Basic Engineering*, vol. 82, pp. 35–44, 1960. 138

[112] A. Doucet, de Freitas J.F.G., and Gordon N.J., *Sequential montecarlo methods in practice.* NewYork: SpringerVerlag, 2001. 139

[113] M. Di Rocco and F. Pascucci, “Sensor network localisation using distributed extended kalman filter,” in *Advanced intelligent mechatronics, 2007 ieee/asme international conference on*, Sept. 2007, pp. 1–6. 141