Tactile Mapping of Harsh, Constrained Environments, with an Application to Oil Wells

by

Francesco Mazzini

Master of Science in Mechanical Engineering, Georgia Institute of Technology (2006)

Laurea in Ingegneria Meccanica, Università degli Studi di Brescia (2006)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2011

Signature of Author .....
Department of Mechanical Engineering
May 1, 2011

Certified by .....
Steven Dubowsky
Professor of Mechanical Engineering and Aeronautics and Astronautics
Thesis supervisor

Accepted by .....
David E. Hardt
Ralph E. and Eloise F. Cross Professor of Mechanical Engineering
Graduate Program Committee Chair

Tactile Mapping of Harsh, Constrained Environments,
with an Application to Oil Wells

by

Francesco Mazzini

Submitted to the Department of Mechanical Engineering on May 1, 2011
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

Abstract

This work develops a practical approach to explore rough environments when time is critical. The harsh environmental conditions prevent the use of range, force/torque or tactile sensors. A representative case is the mapping of oil wells. In these conditions, tactile exploration is appealing.

In this work, the environment is mapped tactilely, by a manipulator whose only sensors are joint encoders. The robot autonomously explores the environment collecting few, sparse tactile data and monitoring its free movements. These data are used to create a model of the surface in real time and to choose the robot's movements to reduce the mapping time.

First, the approach is described and its feasibility demonstrated. Real-time impedance control allows a robust robot movement and the detection of the surface using a manipulator mounting only position sensors. A representation based on geometric primitives describes the surface using the few, sparse data available. The robustness of the method is tested against surface roughness and different surrounding fluids. Joint backlash strongly affect the robot's precision, and it is inevitable because of the thermal expansion in the joints. Here, a new strategy is developed to compensate for backlash positioning errors, by simultaneously identifying the surface and the backlash values.

Second, an exploration strategy to map a constraining environment with a manipulator is developed. To maximize the use of the acquired data, this work proposes a hybrid approach involving both workspace and configuration space. The amount of knowledge of the environment is evaluated with an approach based on information theory, and the robot's movements are chosen to maximize the expected increase of such knowledge. Since the robot only possesses position sensors, the location along the robot where contact with the surface occurs cannot be determined with certainty. Thus a new approach is developed, that evaluates the probability of contact with specific parts of the robot and classifies and uses the data according to the different types of contact.

This work is validated with simulations and experiments with a prototype manipulator specifically designed for this application.

Thesis Supervisor: Prof. Steven Dubowsky
Title: Professor of Mechanical Engineering and Aeronautics and Astronautics

Acknowledgements

First and foremost, I would like to thank my advisor, Dr. D, for his guidance, help and support. I owe him a good part of what I learn during these amazing years.

I would like to thank the professors that served in my committee, Tomas Lozano-Perez, Nicholas Roy and Sanjay Sarma, for their availability and their sharp contributions.

Furthermore, I thank all my present and past labmates, for their collaboration, their support and their friendship. A particular mention goes to Dan Kettler, for the amazing experimental robot he built. Particular thanks also to Peggy, Amy, Jekan and Leah, for the time spent working together and for the patience when reading my Italian-English.

I would like to thank Schlumberger for its financial support for half of my thesis, and Julio Guerrero for his contribution and his good spirit in all the meetings we had.

Finally, I want to thank my family, Caitlin, and my Friend above, for their irreplaceable support and love.

Contents

Chapter 1

Introduction

1.1. Motivation

The objective of this work is to develop a practical approach to map a man-made environment under harsh conditions in a short time. Existing approaches for autonomous exploration rely on range sensors such as cameras, lasers or sonars, which acquire a large amount of data in a very short time. In situations where these sensors cannot be used, autonomous exploration is still an unsolved problem. These situations arise, for instance, when mapping extremely dark and opaque structures, surfaces covered with a layer of dirt, foggy locations, or environments filled with an opaque fluid. Typical examples are oil wells, mines, caves, nuclear facilities, gas pipes, and sewers. Exploration in these environments presents the following challenges:

- *Few, sparse data*. The lack of range sensors makes data acquisition extremely time expensive. Given the time constraints of the operation, mapping needs to be completed using only few, sparse data, which need to be chosen well and used effectively.
- *Limited sensors*. The harsh environment conditions strongly limit the types of sensors that can be used. Range sensors, force/torque or tactile sensors are unfeasible or too delicate, and should be avoided. Only position sensors are feasible. This implies the use of a robotic agent that carries such position sensors while moving in the environment.
- *Constraining environment*. The agent is highly constrained in its movements by the unknown environment. These constraints need to be recognized and faced with the limited sensorial data available, and without any a-priori knowledge of the surfaces in the environment.
- *Imprecise measurements*. Besides limiting the types of feasible sensors, the environment also imposes strong design and structural constraints to the feasible sensors. Hence, measurements are expected to be imprecise and unreliable. In this work, the exploring agent is assumed to be a manipulator, and manipulator's joint backlash is an example of the problems not considered in less hostile environments.

1.2. Applications

1.2.1. A representative case: oil well junction mapping

A representative example of the applications addressed in this work is the mapping of a junction in an oil well. Oil wells often have a number of junctions where divergent branches leave the main well at unrecorded depths (Figure 1). When most of the oil has been extracted from a reservoir reached by a lateral branch, the branch is abandoned. Rising world oil demand and advanced oil recovery techniques have made it economically attractive to rehabilitate previously abandoned branches, which requires lowering instruments and tools into it. To allow this operation, location and shape of the junction need to be mapped. Alternatively, the junction can be mapped during the logging process right after the lateral drilling. In this case, mapping is desirable to determine the location with greater precision compared to what can be obtained by drilling, and to obtain a complete model of the junction shape. Any maintenance operation in an oil well is very expensive (tens of thousands of dollars per hour), because it requires suspending the production of the well and using expensive equipment such as the oil rig. Therefore, the map acquisition must be done as quickly as possible. Modern oil wells can exceed a depth of three thousands meters, and lateral wells are drilled at any depth in the well. At high depth, temperature and pressure can reach 300 °C and 2000 atm. *Drilling mud,* an opaque fluid that fills the well to avoid its collapse, prevents the use of visual or laser sensors to measure the junction. Ultrasonic sensors have been suggested for this application, but unpublished industrial studies have shown that they do not possess the required performance for down-hole conditions, and suffer from reflections problems. Also, a layer of "mud cake" often obscures the well surface underneath, which is the real surface to be mapped.

1.2.2. Other applications

The oil well industry offers several other examples.

Whenever a cable holding a tool breaks in an oil well, the tool must be removed through a process called *fishing*. The current approach consists of trying to snag the object with a hooked tool: it is very time consuming and often unsuccessful [3]. The ability to map the broken tool would drastically increase the chances of successfully grasping it.

Breakouts and washouts are local openings and deformations in an oil well, caused by pipe corrosion and deformation, erosion of the surrounding rocks, or other natural causes. Breakouts represent a challenge because tractors (tools used to drive other instruments into the well)

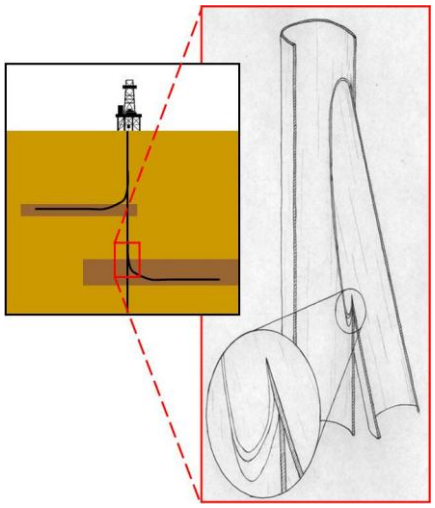

Figure 1. Typical oil well branching structure, with cutaway detail of a junction. Image courtesy of [2]

are often unable to cross them. One of the reasons for this is the lack of information on the dimension and shape of the washouts. Thus, a tool capable of mapping the location in front of the tractor would considerably increase the chances of successfully crossing these obstacles.

Figure 2. Oil spilling from the riser of BP Deepwater Horizon on May 17th, 2010. The oil leaks prevent cameras from determining the shape of the crack [1] .

Spills from deep sea well sites (Figure 2) present numerous challenges due to their extreme depth, which makes manned operations impossible. Underwater Remotely Operated Vehicles (ROVs) have been heavily used in various repair operations. A major reason for failure of many of these procedures is the challenging operating environment, including the complete lack of visibility due to the oil flow from cracks in the wreckage. Lasers and sonars are unreliable due to strong turbulence and methane gas spilling from the leaks. Knowing the shape and dimensions of the cracks and the rate of oil spill provides critical information for the success of the operation.

Other potential applications exist outside the oil well industry, requiring the exploration of a very challenging environment with very limited sensorial information. An example is the need for autonomous operations inside a nuclear reactor [4]. In a nuclear reactor, the intense radiation prevents the use of several sensors, including standard cameras. Tactile exploration and manipulation is a potential solution. Other harsh environments where visual sensors are often unfeasible include pipes, mines, and sewers.

1.3. Previous Work

Mapping challenging environments has been a prominent application in robotics for the last twenty years. Robots have been used to explore pipes [5-7], sink holes [8], volcanoes [9], abandoned mines [10], deserts [11], the Antarctic [12], the moon and Mars [13-15], and underwater wreckage [16]. However, all these studies use range sensors, because the environment, although challenging, does not prevent the use of cameras, lasers or sonars. When visual data are available, the map is often created by remotely controlling a robot, as in most of the studies listed previously. When remote operation is not feasible, the robot needs to move autonomously and plan its own movement: this problem is called *robotic exploration*. Several studies have been published in autonomous robotic exploration (see [17, 18] and Chapter 5); however, virtually all these studies assume the use of range sensors.

When the environment prevents the use of range sensors, the exploration must be carried out by direct contact using tactile sensing. Furthermore, the limited human ability to intuitively

interpret tactile data requires the exploration to be highly autonomous. In these situations, the exploration methods developed for range sensors are not directly applicable. Tactile exploration of harsh environments, discussed in this work, is therefore a new and unexplored concept. Another recent study presents a similar idea, but it is restricted to the simple case of mapping a 2D pipe of known diameter by a serpentine robot [19].

Although tactile exploration of environments is new, similar topics have been proposed in several related areas, including tactile localization or recognition of objects [20-29], haptics and manipulation [30-38], path planning for coordinate-measuring machines [39-41], and computational geometry [42-44].

Several studies in tactile exploration develop techniques for locating and identifying objects among a library of known models. Early works focused on the object identification using as few data points as possible [20-23]. Other works identify local features on the object such as surface curvature or specific symbols [24, 25]. As several branches in robotics shifted their focus to stochastic methods, more recent works introduced stochastic approaches to tactile perception, and developed interesting strategies to guide the robot's movements and reduce the number of points required for identification [26-29, 45]. Some of these studies share the exploration approach developed in this work, and will be described in greater detail in Chapter 5. However, they only identify an object among a set of models, and do not explore an unknown environment.

Other approaches have been developed to map general, unknown objects. Seminal research focuses on the aspect of real time control of a robot [46, 47]. Other works focus on the type of surface description to be used. This is an important aspect in tactile exploration, because tactile data are sparse and time expensive. When exploration time is not an issue, or when the inspected surface is arbitrarily complex, a triangular mesh is often used [23, 31, 43, 44]. Other approaches propose splines [33, 34, 48]. While meshes and splines are effective representations of a general surface, they require dense data to represent a generic object well, and are not efficient when a short exploration time is essential. The same requirement occurs when generalized cylinders are used [49, 50]. An alternative approach represents surface geometry as a composition of geometric primitives, such as quadrics (planes, cylinders and spheres) or superquadrics. These primitives can be determined with curve and surface fitting methods [30, 32, 35, 51], or using differential invariants [52].

While focusing on the representation methods, several of these works also propose interesting heuristic exploration strategies [23, 30, 32]. However, such strategies have been developed to map small, often convex objects touched by a robotic hand, and they are not applicable to the exploration of an unknown environment. Some other studies develop strategies for objects that can be moved or manipulated by the robot [24, 36-38], which is not possible in environmental exploration. The mapping of a larger, static surface has been studied in applications related to coordinate-measuring machines [39, 40, 53]. These works recognize the importance of intelligently selecting the locations in which to collect data for effective

exploration, but the developed strategies are still tied to a two-dimensional grid, which is not time-effective and only applicable to specific objects.

With regard to the physical hardware needed for exploration, the studies mentioned above (except [38]) use either a force-torque sensor or an active tactile probe. These sensors are not reliable in a very harsh environment, where only position sensors can be used (*proprioceptive exploration*[1]). There are some works in the literature to approach proprioceptive exploration [54-56]. However, these works can only be applied to a small, manipulated object, and study only local contact detection and not the issues of exploration and surface model construction. The literature on this topic will be described in greater detail in Chapter 6.

In conclusion, while past works have studied various components of intelligent exploration, characterization of general unknown geometries, and proprioceptive exploration, the problem of exploring an unknown, harsh, internal environment when time is critical remains unsolved.

1.4. Contributions of this Work

This research develops the first approach to tactilely explore harsh environments with a manipulator. The research can be divided in two main parts.

The first part (Chapter 2 - Chapter 4) develops the approach for tactile exploration, and shows its applicability to mapping very harsh environments. The real-time robot control is studied to provide fast and reliable movements and contact detection with only position sensors. A surface representation is developed to quickly describe the environment with only the sparse data available from tactile sensing, while keeping the flexibility required when mapping a generic environment. Simple and effective exploration strategies are proposed to obtain a uniform coverage of a surface and a fast exploration of the environment. The robustness of the approach is validated with extensive simulations and experiments with a three degree-of-freedom prototype manipulator specifically developed for this application in collaboration with Daniel Kettler [2]. The approach was found to be robust to harsh environmental conditions such as rough surfaces, measurement noise, and surrounding viscous fluids. Joint backlash was found to strongly affect the ability to precisely map the surface. Thus, a new method to compensate for backlash has been developed: the robot compensates for positioning errors by simultaneously identifying, during the exploration, both surface parameters and backlash values.

The second part (Chapter 5 - Chapter 6) proposes a strategy to map a constraining environment with a manipulator provided only with position sensors. The main contributions can be summarized as follows:

[1] Proprioception is a term derived from medicine, and it refers to the sense of an individual about the position of the parts of his own body. In robotics, proprioceptive sensing means that measurements refer to internal variable of the robot, such as joint positions and torques.

1. To maximize the use of the acquired data, this work proposes a new hybrid approach involving both workspace and configuration space: data are saved and analyzed in workspace, but motion planning is evaluated in configuration space.
2. Since tactile data are few and sparse, information about the entire environment needs to be inferred from these few points. Here, a method is developed to evaluate the correlation between close points, based on the physical dimensions of the objects in the environments.
3. The amount of knowledge of the environment is evaluated with an information-theoretic approach, and the robot's movements are chosen to maximize the expected increase of such knowledge. In previous studies using an entropic approach the two actions of movement and measurement could be treated separately because data were provided by range sensors. With tactile exploration, the robot movement is the measurement itself, and new information only comes from a movement into unexplored space. Therefore, the knowledge obtained with a path does not depend on the endpoint, but on the path itself. This requires a new approach to the problem.
4. The absence of any sensor besides position encoders makes the determination of the location of contact a challenging problem. Here, a new approach is developed to determine the probability that specific parts of the manipulator are touching the environment, and thus classifying the type of contact. After points have been classified, a novel method is developed to interpret the available tactile data. This method is based on the same surface representation used previously, but extended to the new kind of contacts that arise when the environment is touched with the whole manipulator's body.

Chapter 2

The Approach: Tactile Mapping

This work develops a practical approach to map a surface under very harsh environmental conditions. The objective is to tactilely create a geometric model of the whole environment within a given accuracy in the shortest possible time. The harsh conditions prevent the use of any range sensors, while delicate sensors such as force/torque or tactile sensors are unreliable, and should be avoided.

2.1. The Proposed Solution

This work studies the feasibility of tactile exploration using a robotic manipulator equipped only with position sensors. The approach is as follows: the manipulator is brought in proximity to the surface to be mapped, and its base is fixed with respect to this surface. Figure 3 represents this situation for a very simple manipulator and environment. The problem is described here with a general approach, so it can be easily applied to different situations. The oil-well environment is one of the situations where these conditions occur, see Figure 4.

Once the robot is fixed, the environment is mapped by touching the surface with the manipulator and reading the joint positions during contact. Like a man's arm in a dark hole, the robot uses all the sources of information that it can obtain with its movements to map the environment in the shortest possible time. Since tactilely probing a surface is inherently slow, collected tactile data will be sparse. These data are analyzed in real time to provide a provisional model of the surface. This model is used by the algorithms guiding the robot to choose the robot's path to make the mapping process as efficient as possible.

2.2. Assumptions

The manipulator is assumed to be brought in proximity of the environment by external means. The manipulator's base is fixed on the environment, and used as the reference frame for the process. The surface will be reconstructed with respect of this frame. If the base is relocated,

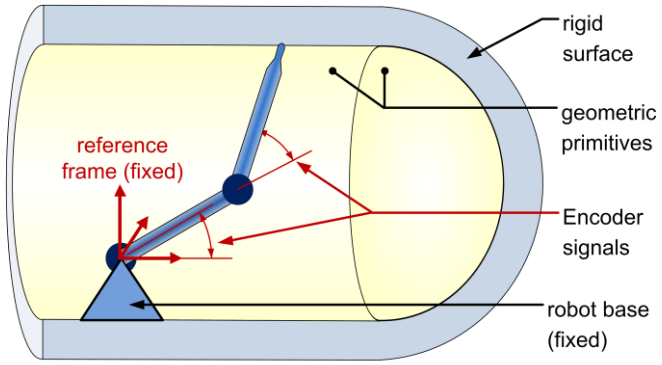

Figure 3. Robot for tactile exploration of an unknown environment.

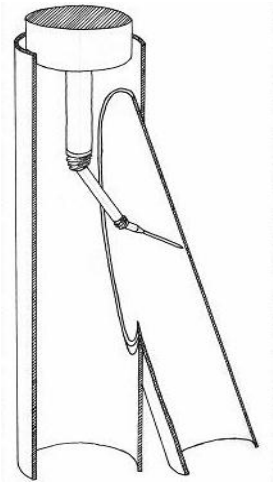

Figure 4. Manipulator for tactile exploration of a junction in an oil well. Image courtesy of [2].

the robot will create a map for each location, and these maps can be registered (joined). Approaches for registration are available in the literature [57].

The manipulator is assumed to possess only position sensors. Since most manipulators are composed of rotational joints, and encoders are the standard joint position sensors, in this work the position sensors are referred to as encoders. No assumption is made on the type of joints in the manipulator. Kinematic non-idealities, link deformations and encoder resolution are assumed to affect the manipulator's precision and create measurement errors. Here, these errors are modeled as random measurement noise. Joint backlash is assumed to be too significant and consistent to be modeled as random noise. However, a specific backlash identification method will be developed to compensate for its error.

This work does not focus on the real-time performance of a specific manipulator. Therefore, the manipulator's speed is assumed to be limited by a maximum value of joint speed. If this holds, a measure of the time required for exploration is given by robot joint movements.

The equivalence between robot travel and exploration time is only valid if the robot never stops during the exploration. However, this research assumes computational time to be negligible. There are several reasons for this. First of all, tactile exploration is intrinsically slow, because moving a manipulator in an unknown, rough environment requires low speeds. Second, tactile exploration collects few, sparse data, which can be processed faster than the large amount of data obtained with range sensors. The mapped space is also limited, because the surface that can be mapped by a fixed manipulator is within the manipulator's workspace. Finally, if there is a specific operation requiring non-negligible computational time, it can be executed during the robot movements. For all these reasons, this research assumes computation time to be negligible, and uses the robot path as an equivalent measure of the exploration time.

In this work, the environment is assumed to be rigid and static. In the case of an oil well, several surfaces are covered with a layer of a highly viscous material called *mud cake*. However, the environment to map is the rigid surface underneath the mud cake.

The environment is assumed to be man-made. In most cases, man-made environments are regular, and they can be easily described by using a combination of a few, geometric primitives. A survey indicates that 95% of man-made objects can be described as the combination of the following primitives: planes, spheres, cylinders, cones and tori [58]. Therefore, this work assumes that the environment is mainly composed of such shapes. To maintain the generality of the approach, local deformed or irregular shapes can be present.

The environment is assumed to be completely unknown. If any a-priori information is available, this can be easily introduced by starting the exploration with a partial map. To be general, this work assumes no a-priori information.

Two parameters are required to guide the exploration and define the required accuracy of the map:

1. The minimum desired dimension of a feature to be found, denoted by B. This can be interpreted as the *characteristic length* of the explored surface, and it controls the trade-off between details of the constructed model and required time. A lower limit on the value of B is the robot accuracy.
2. The maximum allowed distance between a touch point and the constructed surface model, denoted by f. This value controls the approximation when modeling the surface as composition of geometric primitives. It is chosen as a function of robot accuracy and desired surface approximation.

Chapter 3

Feasibility of Tactile Exploration

This chapter describes the feasibility of the tactile exploration approach to map a man-made surface. The overall goal of tactile mapping is divided into three different tasks: real-time robot control, representation of the surface, and exploration strategy. A solution for each of the tasks is proposed. Simulation and experiments demonstrate the feasibility of the approach.

3.1. Three Simultaneous Tasks

The overall approach requires the following tasks to be performed simultaneously:

1. *Real time control and contact detection.* The robot must be reliably controlled in real time while moving in a partially unknown environment. Unpredicted contact with the surface, either by the tip or other parts of the manipulator, needs to be detected and identified.
2. *Surface model construction.* Information from the robot movement and from the contact with the surface is used to construct a model of the surface. This model is refined as information is acquired.
3. *Exploration strategy.* The robot movements need to be chosen in order to minimize the exploration time. The partial reconstructed map is used to guide the robot, to explore the regions where it acquires more information.

This chapter provides a solution for these three tasks. The third task is the most complex, and will be treated in depth in Chapter 5. To prove the feasibility of the tactile exploration approach, a simple and intuitive exploration strategy is proposed. This strategy proved to be effective in simulations and experiments, but no optimality or completeness can be guaranteed.

3.2. Real Time Control and Contact Detection

3.2.1. Background

A robot interacting with the environment can be controlled using two different strategies: hybrid force-velocity control, or impedance control [59]. *Hybrid force-velocity* control see (Figure 5) is used to slide the robot tip along a surface, by controlling the force in the direction normal to the surface and the speed in the two tangential directions [60]. Since these directions are unknown and variable, they need to be estimated at all times with a force-torque sensor. This control architecture is only valid when the robot is already in contact with the environment: another control method is needed to guide the robot in free space.

The second scheme to control a manipulator interacting with the environment is *impedance control* [61]. This is essentially a smart use of a position controller, see Figure 6. The joint torques are controlled to simulate the presence of an impedance between the robotic tip and an arbitrary point in space, called *virtual point*. The virtual point is arbitrarily moved and the robot tip follows it as long as no obstacle prevents its movements. When the virtual point moves beyond the environment surface, the robot cannot follow it and stops, and contact is detected. The position of the probe tip, and therefore the contact point on the surface, can then be determined from the manipulator's joint angles. This controller does not require any control switch between contact and free-space movement, and permits the manipulator to hold its probe against the environment without any force or tactile sensors. There are different implementations of impedance control [62]. In one of its simplest forms, the impedance between virtual point and end effector is simply composed of a spring and a damper. The required motor torques are therefore:

$$T = J^T \left[D\left(\dot{x}_v - \dot{x}_{ee}\right) + K\left(x_v - x_{ee}\right) \right] \tag{3.1}$$

where J is the kinematic Jacobian, D is the damping constant, K is the spring constant, and the subscripts v and ee indicate the virtual point and the end effector.

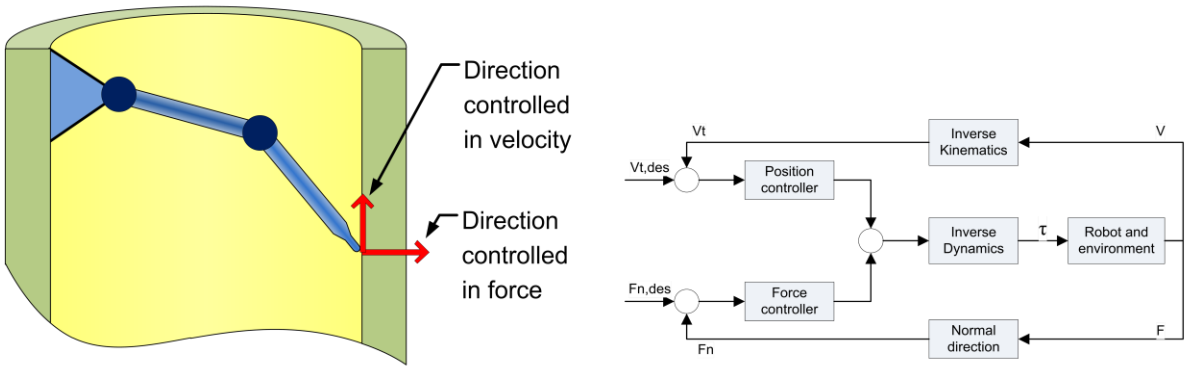

Figure 5. Left: representation of an hybrid force-velocity controller. Right: block diagram.

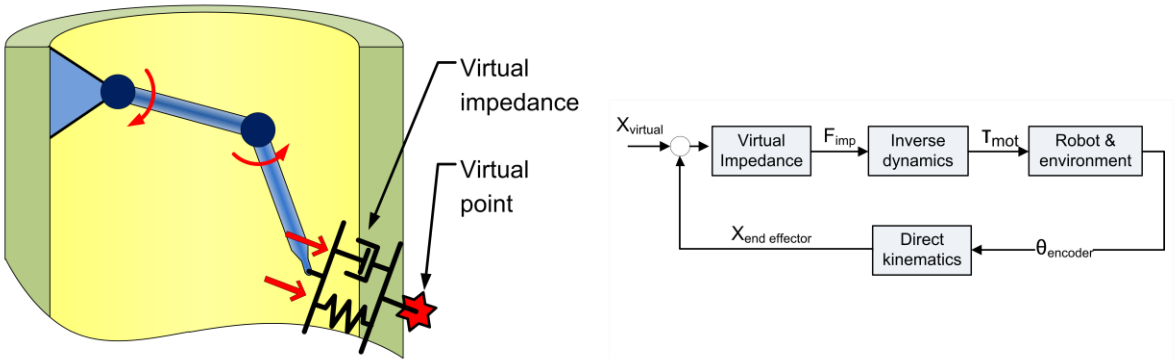

Figure 6. Left: representation of an impedance controller. Right: block diagram.

3.2.2. Choice for tactile exploration

The *impedance control* scheme is best suited for this application, because of the ability to operate without any force or torque sensors, and the robustness during the transition between contact and non-contact situations. The approach is inexpensive, robust, and very suitable for harsh environments. There are two drawbacks with impedance control, but they do not provide any important limitation for this particular application:

1. Impedance control without any force sensors commands the torques in open loop, and it does not allow precise control of the interaction force between robot and environment (mostly because of joint friction). Nevertheless, if the surface is rigid, knowing the precise interaction force is unnecessary, as it does not provide any information.
2. Continuously tracking a surface under impedance control is less reliable than under hybrid force-torque control. Nevertheless, this research showed that discretely probing a surface is more effective than tracing, when mapping a harsh and unknown environment (see section 3.4.1).

Using the position of the tip to determine the contact location assumes that contact with the environment always occurs at the robot tip. This might not be valid when exploring a highly constrained environment (see Figure 7). This assumption is relaxed in Chapter 6.

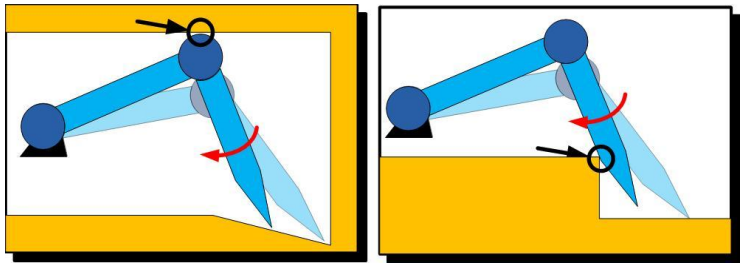

Figure 7. Two situations where the robot touches the environment with its body and not its tip.

3.2.3. Implementation

Laboratory experiments provided the insights to ensure the robustness of the impedance controller under harsh conditions. A simple compliant controller did not prove to be reliable, because joint friction and different robot configurations often cause a false positive in contact detection. Joint friction in a field robot is extremely high because of the harsh environment conditions and the need for pressurized seals surrounding the joints. Friction compensation algorithms have been implemented, including a superimposed dither signal and a feedfoward signal for Coulomb and viscous friction [63, 64]. These methods reduced the problem, but did not solve it. The solution was found by introducing an integral term to the standard real time impedance controller:

$$T = J^T \left[D\left(\dot{x}_v - \dot{x}_{ee}\right) + K\left(x_v - x_{ee}\right) + \int_{t_{\dot{x}0}}^{t} \left(x_v - x_{ee}\right) dt \right] \tag{3.2}$$

where $t_{\dot{x}0}$ indicates the most recent time when the end effector speed was close to zero. By resetting the integral when the tip speed exceeds a small threshold, it is effectively active only when the robot stops. The integral also saturates to a maximum value, to limit the force applied to the environment. The introduction of the integral term drastically decreased the number of false contact detections to a negligible number even in very harsh conditions.

Other techniques have been implemented to provide reliable and precise contact detection. One of these is to "back off" the torques to a minimum value whenever contact has been detected, to reduce the effect of joint elasticity. This is evident when joint torques are plotted as a function of time, as in Figure 24 (section 4.2). The full explanation of this and other implemented techniques goes beyond the scope of this document.

3.3. Surface Model Construction

The objective of surface model construction is to best represent the surface given the current information. There are two main sources of information. The first source comes from the contact detected between robot and surface. In this chapter, since contact is assumed to occur only on the robot tip, contact data are a set of sparse, three dimensional points. In Chapter 6, this approach is extended to include situations where contact occurs on the manipulator's body as well. The second source of information is the volume swept by the robot's movements: the space where the robot has passed needs to be free.

The environment is described with a *boundary representation*, which defines a shape by describing its boundary surface. Tactile data are sparse and time consuming to obtain. To minimize the amount of required data, this work uses a simple and data-inexpensive surface representation whenever possible, and increases the complexity only when necessary (Figure 8). The simple representation takes advantage of the regular structure of most man-made environments, and represents the surface as a combination of the following primitives: planes,

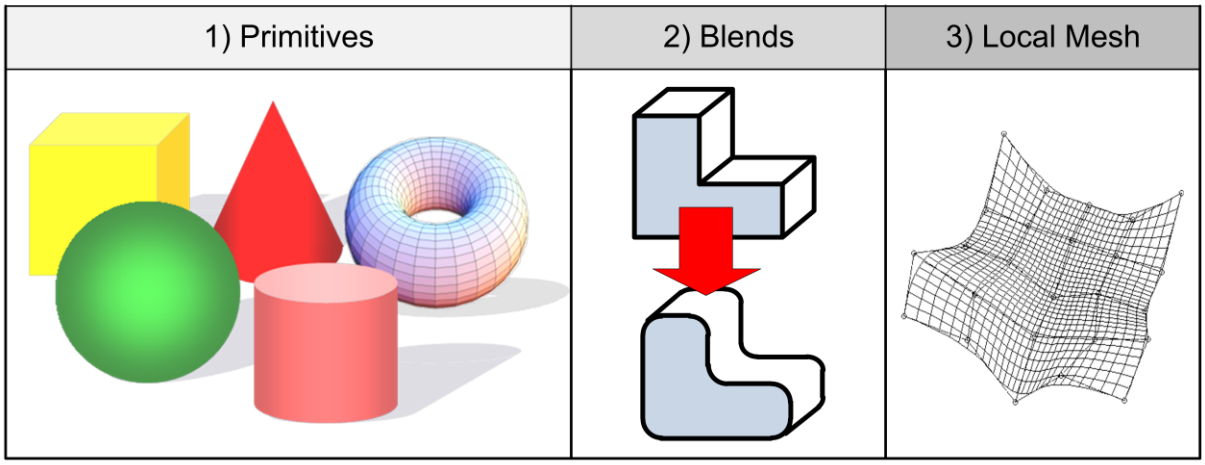

Figure 8. The surface representation used in this work. Whenever possible, the surface is described as the combination of geometric primitives. When needed, blends and local triangular meshes are added.

spheres, cylinders, cones and tori. If the environment is too complex for these primitives, blends between primitives and local triangular meshes are introduced. This representation maintains the generality of the approach, because it can describe any surface, but it also remains very data-inexpensive for standard man-made objects.

The problem of reconstructing a surface made of geometric primitives given points on the surface has been well studied in reverse engineering for large, dense datasets derived from range sensors [65]. This work develops and adapts these techniques to small, sparse datasets from tactile data. Two tasks need to be accomplished simultaneously: determine what primitives (type, location and parameters) are present in the environment according to the existing touch points, and classify the touch points to these primitives. This is called segmentation. The process of determining these primitives requires a method to evaluate how well a set of points fits a specific primitive: this problem is surface fitting. After the classification is completed, the primitives are intersected to produce the complete surface representation. Finally, information from the swept volume is used: if a section of the reconstructed surface lies within a region where the robot has passed, such section is removed.

3.3.1. Surface fitting

Assume that a set of touched points is known to belong to a specific primitive type. Surface fitting is the method that determines the parameters of the primitive that best approximate these points. This is solved with a least squares approach, minimizing the sum of the squared distance between primitive and data points:

$$\theta = \arg\min_{\theta} \sum_{i=1}^{N} d\left(P_i, S(\theta)\right)^2 \tag{3.3}$$

where P_i are the N data points, S is the primitive to which the points belong and θ are the surface parameters. The value of this function represents the goodness of the fit, a measure of

uncertainty of the modeled surface. Different solutions for this minimization problem have been proposed for large dataset of range data [66-68]. Small and sparse datasets present two problems: they are not dense enough to compute a reliable surface normal (typically used to find an initial condition), and they do not provide sufficient redundancy to reduce the effect of random noise. For these reasons, some of the solutions in the literature tend to fail. Tests showed that the method proposed by Jiang and Cheng works the best in this case [68]. It uses geometric reasoning to project the points on special lines or planes, reducing the dimension of the nonlinear search.

3.3.2. Segmentation

In order to correctly fit the data to different primitives, the numbers and types of the primitives need to be detected, and the touch points need to be classified so that each point belongs to one primitive only. This problem is called range data segmentation. Among the several methods proposed in the literature, an approach called *fit and grow* has been chosen and implemented in this work [69, 70]. It is based on the selection of small initial regions to which all the possible primitives are evaluated. Regions that give a good fit are then gradually expanded, otherwise they are discarded. A primitive fit is accepted when the RMS error is lower than the constant *f*, the maximum distance between a point and the approximated surface (see section 2.2). Fit and grow is chosen because it does not require information on surface normal, and it fits well in this particular application, where sparse points are added to the dataset gradually, and where outliers may be present when a primitive has been just partially discovered. Furthermore, this strategy also classifies irregular shapes as outliers, denoting the need of a local triangular mesh.

3.3.3. Intersection and surface cuts

After the primitives are identified, their intersections are computed to produce the complete surface representation [71, 72]. Some primitive parameters need to be modified to ensure that the boundary conditions between primitives are consistent. This is obtained by fitting several primitives in a single minimization while constraining some of their parameters. This is called *constrained fitting*, and it is necessary, for instance, when two different primitives are tangent [73]. It has also been shown that blends between primitives can be determined with relatively few touch points close to the intersections [74].

Finally, the reconstructed surface is check for consistency with the volume swept by the robot. The surface is simply intersected with the swept volume, and the surface inside such volume is deleted. To do this, a map of the locations where the robot has passed needs to be kept in memory. In this work, this is performed by discretizing the volume using voxels, as explained in Chapter 5.

3.4. Exploration Strategy

3.4.1. Discrete sampling vs. continuous tracing

Two approaches can be used to tactilely explore a surface: probe it discretely in separated points or trace continuous lines on it. The former approach has been chosen for this problem, for two reasons. First of all, a continuous trace is not always feasible in a rough environment such as an oil-well. There are two types of oil well surfaces: open-holes, where the borehole is unprotected, and cased-holes, where a steel pipe (called casing) and a layer of cement cover the hole. In open-holes, a thick and highly viscous material called mud cake often forms on the top of the surface. Furthermore, both in open and in cased holes, the surface is very rough. This prevents to trace curves while the probe is in contact with the surface. The second reason is that, even if a continuous trace can be executed, it would not identify the surface better than a discrete sampling. In fact, measurements present noise, surfaces may be rough and locally deformed; thus inferring a primitive from a small local region is intrinsically unreliable, even when hundreds of data points are used. This was first noticed by [52]. As a further proof, the ability to fit a cylindrical rough patch is evaluated in two cases: either by tracing two lines made of 500 points each, or by using 15 random sparse probe points (Figure 9). The cylindrical patch has length 200 mm and radius 100 mm, and it has been corrupted with surface roughness, which is created by a sum of 100 sinusoids with random spatial period between 2 and 10 mm. All the points are also corrupted by random noise with 1 mm variance. The ability to reconstruct the correct cylinder parameters is evaluated with a Monte Carlo simulation. The RMS amplitude of the surface roughness is increased at regular intervals, and 1000 fitting trials are performed for each interval using either the two lines or the sparse points. The percentage of the trials that found the correct cylinder within a 10% tolerance is shown in Figure 10, confirming the superiority of the discrete sampling approach.

3.4.2. A simple discrete sampling strategy

The guidance of the robot based on the continuous interpretation of sequentially-acquired data is a key factor for an efficient tactile mapping. This subject is addressed in detail in Chapter

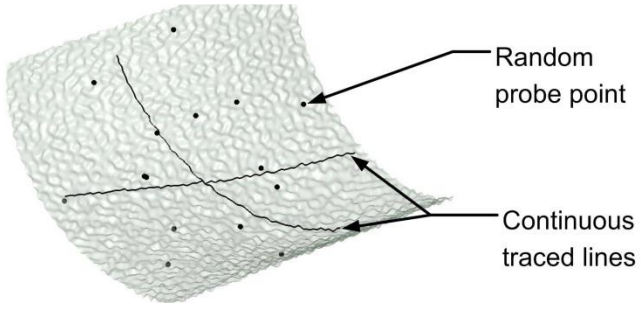

Figure 9. Cylindrical patch with surface roughness, examined with two continuous line traces and 15 random probe points.

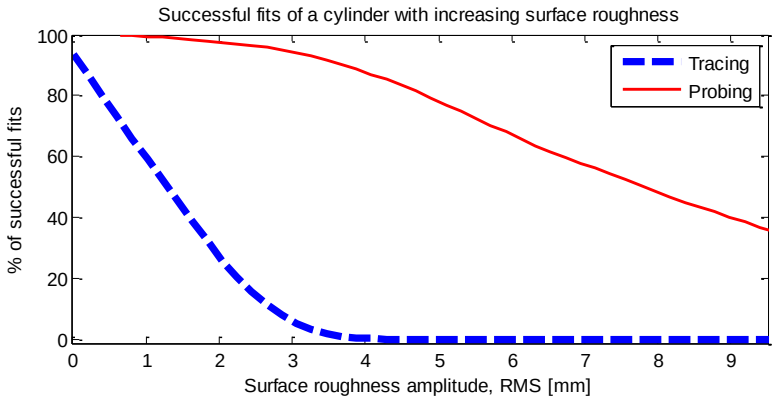

Figure 10. Percentage of trials that return the correct parameters of the cylinder (within 10% tolerance), as a function of surface roughness, when the cylinder is examined with two lines and with 15 sparse probe points.

5. To study the feasibility of the tactile exploration approach, here a simple strategy is proposed, which is well suited for the tactile exploration of rough and constrained environments such as oil wells. It is composed of two main steps, called *Uniform Surface Density (USD)* and *Best Cone* search. The concept is to locally touch a surface until a primitive is identified and then move the robot where the next measurement provides the most information:

1. The robot starts in a random direction.
2. It sequentially probes a surface (*USD* search) until a primitive is identified to a desired accuracy, which is evaluated by comparing the goodness of its fit to a function of the given parameter B (the dimension of the largest surface feature that can be neglected, as defined in 2.2).
3. Then, the robot chooses another direction and moves along a line until it touches a new point or the end of the workspace (*Best Cone* search).
4. If this point belongs to a known primitive, it moves elsewhere as in (3), otherwise it investigates and identifies the surface, as in (2), before continuing the search in a new direction.

3.4.2.1. Uniform Surface Density

Step (2), called *Uniform Surface Density* search, is a strategy that probes points sequentially, one close to the other, on the same surface. It is governed by the parameter B. This method is designed to minimize what is called *dispersion*, the radius of the largest sphere, centered anywhere on the surface, that does not contain touch points. To ensure that no feature larger than B is missed, dispersion on the probed surface should be smaller than $B/2$. On a flat, two-dimensional surface, the structure requiring the fewest number of points for a given dispersion is proved to be a lattice[2] of equilateral triangles [75]. On a curved surface, simple geometric

[2] A group of points is a *lattice* if it can be generated by linear combinations with integer coefficients of a finite number of initial elements, called *base*. In other words, it is an array of regularly-spaced points, that can be constructed by translating existing points using just a few, constant vectors.

reasoning shows that the same lattice satisfies the dispersion requirement (even though optimality is lost). The USD search allows the robot to implement this lattice when the surface is unknown. Given two touch points, the robot tip is moved on a circular path at a distance $\sqrt{3}B/2$ from these points, until it touches the surface (Figure 11). This strategy requires an estimate of the surface normal vector at specific points P. If P belongs to a known primitive, its normal is computed using the primitive itself, providing a reliable result. If the primitive is unknown, the normal is estimated by fitting a plane to its neighbor touch points [76]. Since touch points are sparse, this fit is not always reliable. Best results are obtained using a *moving least squares* approach, where the contribution of each point is weighted with a Gaussian function with width equal to d and centered in P_i. A moving least squares approach allows the increase of the influence of points that are close to P, since they better approximate the tangent plane. If a point does not have enough neighbors to provide a reliable fit (this can be recognized by monitoring the singular values of the plane fit covariance), different points of the lattice are expanded first.

3.4.2.2. Best Cone

Step (3) is critical to reduce the time required to map the entire surface. When a primitive has been identified with low uncertainty, further points on that surface would not provide much information, and the robot changes the location to probe. This new location is chosen to maximize the expected amount of information given by the next measurement. This approach is discussed in Chapter 5. In this section, a simple geometric method is proposed. This method has no proof of optimality but is computationally very fast and performs well in both simulations and experiments with different shapes.

Conceptually, like a man who looks around and chooses to explore where he has not yet been, the robot chooses to move along the direction that is away from all previously touched points (Figure 11). To do this, it computes all the possible circular cones with vertex at the end effector P_{ee}, subject to the constraint that all the probed points P_i are external to the cone. Among these cones, the one with biggest aperture angle is chosen, and its axis N is the next exploration direction:

$$\vec{N} = \arg\max_{\vec{N}} \left[\min_i \left(\vec{N} \cdot \frac{\overrightarrow{P_{ee} - P_i}}{\left\| P_{ee} - P_i \right\|} \right) \right] \tag{3.4}$$

The internal minimization determines, for a given direction N, the largest cone aperture angle that does not include any touch point. The external maximization chooses N to maximize this angle. This evaluation is computationally fast, because it involves a search in just the two variables representing the cone axis. The primitives do not affect this choice. In some applications, the intersections between primitives are important regions and require more accuracy. This is the case at the intersection of main and lateral wells of oil well junction. Greater accuracy can be achieved with detailed exploration along these intersections after the

initial identification, and it can be enforced with the Best Cone search by adding a positive weight to direction passing near these intersections.

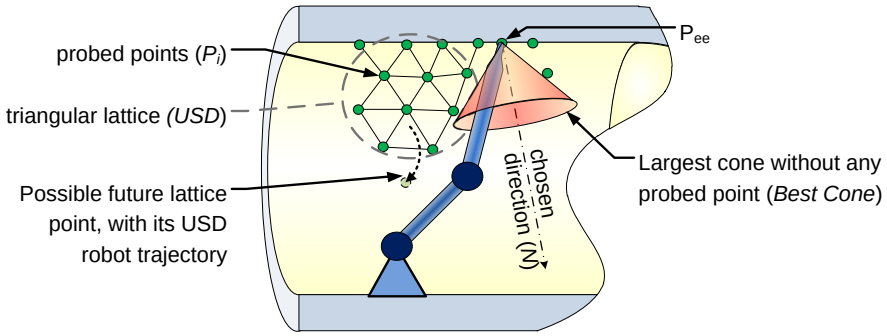

Figure 11. A representation of the Uniform Surface Density *triangular structure and of the* Best Cone *choice of direction.*

3.5. Case Study

To demonstrate the feasibility of the approach, simulations have been performed with the environment represented in Figure 12, composed of a sphere, a cylinder and a plane. This shape, although very simple, includes different primitives and a sharp intersection, and requires constrained fitting between the cylinder and the sphere.

This set of simulations studies the performance of the algorithm in terms of time, or equivalently, length of the traveled path. Shapes are ideal primitives, but measurements are corrupted by random Gaussian noise with 1 mm standard deviation (σ). The robot tip moves inside the environment, constrained by the surface and by its workspace limits. The robot has no initial knowledge of the shape to map. The exploration terminates when the environment is identified within 1% of the correct dimensions.

To provide a comparison for the *Best Cone* search, four strategies have been implemented

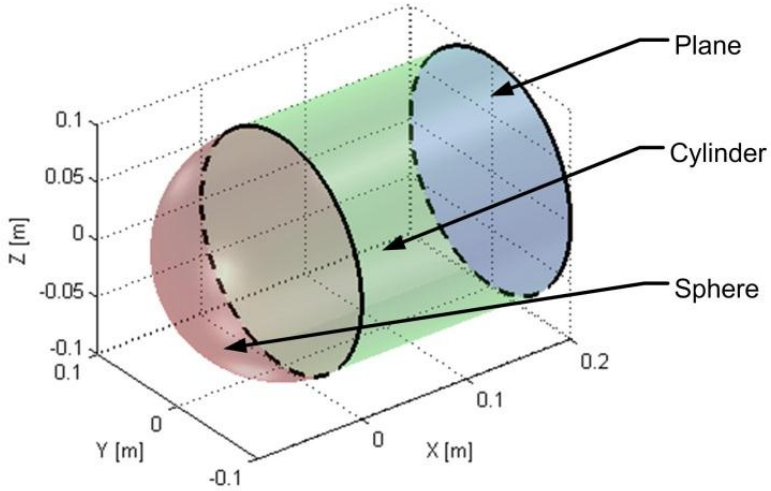

Figure 12. Environment considered in simulation: a reservoir-like shape made or a sphere, a cylinder and a plane.

and tested:

1. *USD search*: the USD strategy is carried on for the whole process, without ever using the Best Cone to change location. This procedure uniformly maps the whole environment with a specific density.
2. *Random search*: every time the robot touches a point, it then moves in a random direction.
3. *Best Cone search*: the complete process described in the previous section.
4. *Semi-random search*: like in the Best Cone search, USD is locally used to identify primitives. However, once a primitive has been identified, the new direction is selected randomly.

For each strategy, 20 trials are run. Differences in trials depend on the initial search direction (randomly chosen), and on measurement noise. When the initial touched points lie close to the intersection of two primitives, a wrong average primitive can be initially fitted. Nevertheless, when more points are touched, the inconsistency is automatically corrected by the segmentation algorithm and the real primitives are eventually found. The mean and standard deviation of the end-effector path lengths are represented in Figure 13, which shows that the *Best Cone* strategy has the best performance.

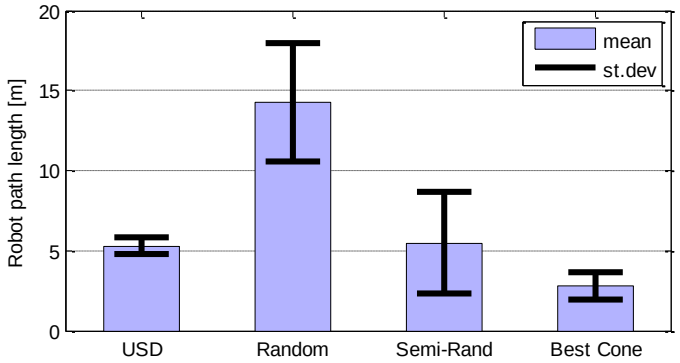

Figure 13. Comparison of the performance of the four strategies: mean and standard deviations of the end-effector path after 20 trials per strategy.

3.6. Application: Oil Well Junction Mapping

The methods described in the previous section have been applied to the map of a junction in an oil well.

3.6.1. Requirements and conditions

The objective of the operation is to provide a map of the junction in the shortest possible time, with accuracy of the order of a millimeter. Given the harsh down-well conditions, the roughness of the drilling process, and the possible presence of debris, the shape can be far from the ideal intersection of two cylinders. Thus, the algorithm needs to be flexible enough to map a generic man-made internal surface.

To reach the junction, the robot will be mounted at the bottom of a cylindrical *wireline tools array*, a system developed by oil well service companies that can carry various logging instruments and is connected to the surface with a wire. This wire serves both as a power and signal transmission, so that no power storage or computational equipment needs to be installed on the tool. This tool array will be lowered into the well and one of its modules will anchor itself to the well case above the junction, whose location is known with an uncertainty of the order of 10 meters. The robot will construct the environment map with respect to this tool, whose location can be estimated using a logging truck winch and an orientation sensor. The dimension of the junction depends on its depth; a reference value for the diameter of the main and lateral cases is 23 cm and 18 cm respectively, with a divergence angle of 5°. The resulting junction is approximately 2 meters long.

The full exploration of this long and narrow junction space requires a redundant manipulator. Kinematic studies were performed by Daniel Kettler, and are covered in greater detail in his Master's thesis [2]. These studies indicate that a 4 degree-of-freedom (DOF) mechanism consisting of a 3 DOF anthropomorphic arm attached to a long prismatic link aligned with the axis of the main well is well suited for this application (Figure 14).

Figure 14. Proposed field system in a cutaway junction. Image courtesy of [2].

3.6.2. Simulations

The effectiveness of the algorithm is first studied in a simulated oil well junction, as represented in the left of Figure 15. The same test described in the previous section has been used: the performances of four different

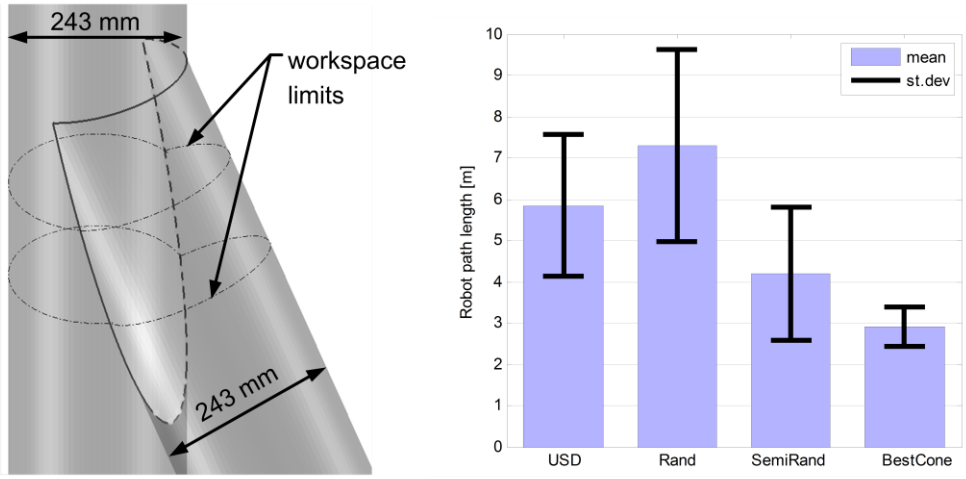

Figure 15. Shape of the simulated junction (left) and simulation results using the four strategies (right).

strategies are evaluated by monitoring the traveled end effector path in 20 trials each. The results are represented in the right of Figure 15, confirming the same behavior showed with the sphere-cylinder-plane environment.

3.6.3. Laboratory prototype

A laboratory system was designed to test the algorithms developed in this work. The system design is explained in details in Kettler's thesis [2]. The experimental system manipulator and environment tank represent the size and kinematic configuration of a well junction field system, given the constraints of the laboratory (Figure 16, Figure 17).

For simplicity, in this experimental system only the 3 DOF arm has been implemented, replacing the first prismatic joint with a mounting ring that can be fixed at different heights. The sizing of the arm links is based on the workspace size and dexterity requirement inside an oil well. The manipulator links have lengths of 8.0 in and 6.0 in (approximately 20.3 and 15.2 cm). The links are stiff enough so that link deformations introduce negligible error in the measured position of the probe tip.

Each joint assembly consists of a motor, gear train, encoder, and associated support bearings. Brushed DC motors are used. The joints are compact in order to minimize the potential for undesirable contact between the manipulator elbow and environment. The joints are sealed by encasing them within rubber bellows, allowing testing of the system while submerged in fluids. Since the manipulator is controlled by a simple impedance controller scheme, no force-torque or tactile sensors are used.

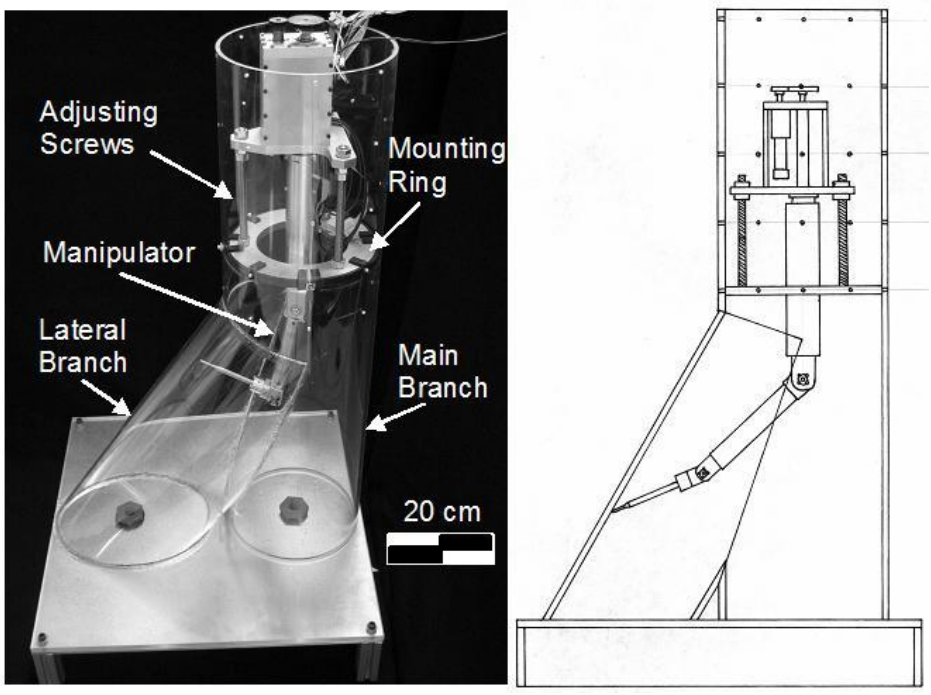

Figure 16. Experimental robotic manipulator in modeled oil well junction.

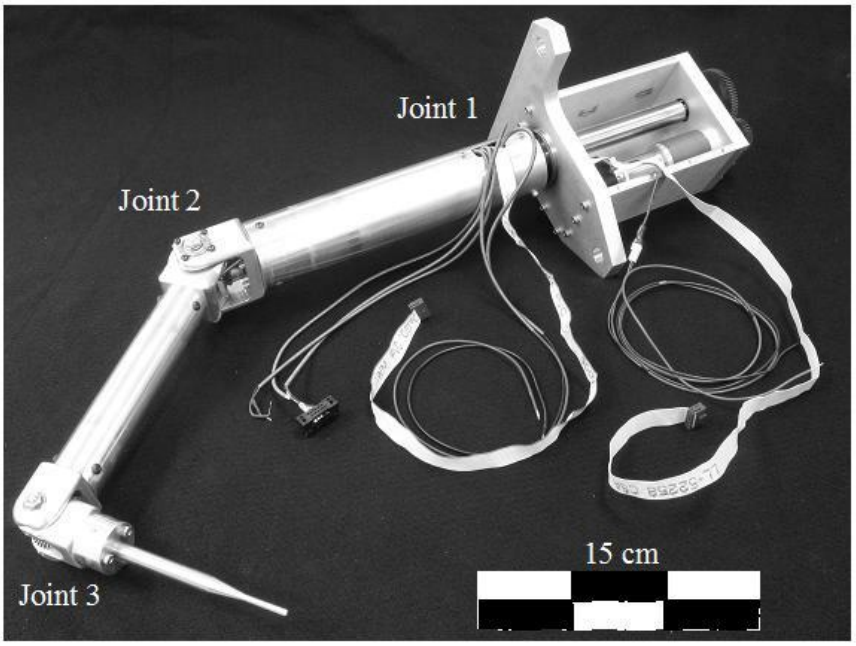

Figure 17. Detail of the robotic arm. The rubber bellows have been removed to show the structure of the joints. Image courtesy of [2].

3.6.4. Experimental results

A set of experiments has been completed to validate the approach proposed in this work. The manipulator performance has been characterized and several search methods have been studied. With an appropriate backlash compensation method (see section 4.3), the manipulator provides sensing errors of roughly 1 mm.

Figure 18 shows a comparison between the pattern of experimental touch points produced by the USD strategy alone and combined with the Best Cone search. The robot has no prior knowledge of the environment or the type of primitives involved, but the search terminates when the algorithm converges to two cylinders with radius within 2% of the real value. Figure 18 also shows the primitives fit to these touch points, as well as the intersection. Table 1 summarizes the results for these two strategies and the semi-random search.

Using the USD search only, the manipulator needs to travel 8.1 m to make these measurements over a period of 311 s. The semi-random search presents similar results. The Best Cone search leads to better performance: the number of points is reduced to 29 and the total distance traveled is reduced by half. Similarly, the required time is reduced to 180 s. Figure 18 clearly shows the reduction in the number of touch points using the Best Cone search to achieve the same accuracy as the other methods.

Table 1. Experimental results

Method	*Number of points*	*Distance traveled (m)*	*Time (s)*
USD	68	8.1	311
Best Cone	29	4.5	180
Semi-random	67	8.9	382

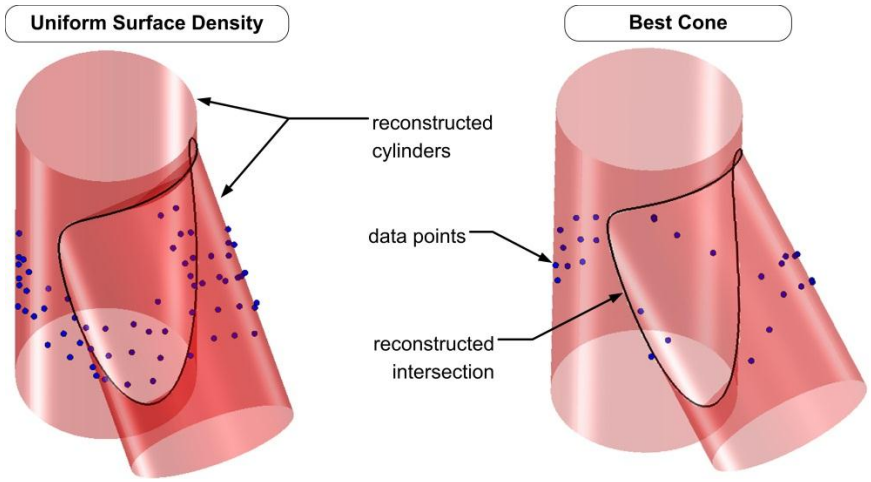

Figure 18. Result of exploration of oil-well junction using the Uniform Surface Density (left) and the Best Cone strategy (right).

3.7. Conclusions

This chapter shows the feasibility of using manipulator-based tactile exploration of environments where conventional range sensing is not feasible. A key challenge is that the data from such exploration are limited and sparse. Here data search and data processing methods are developed to overcome this limitation. This approach only requires a manipulator provided with joint encoders. The approach is applied to the economically important context of mapping the structures of deep-hole oil wells, and in particular mapping the junctions where divergent lateral wells intersect the main well. Simulations and laboratory experiments show that this method is feasible and effective when exploring simple environments, in terms of time, traveled distance and data points needed to map the unknown surface.

This chapter leaves three topics unaddressed. First of all, the robustness of the approach needs to be investigated when operating in very harsh environments. Second, the simple search strategies need to be extended to allow for exploration in more complex and constraining environments, and to ensure exploration speed. Third, the assumption that the manipulator touches the environment with only its tip needs to be removed. These topics are addressed in the following three chapters.

Chapter 4

Harsh Environment Conditions

To be practical, the mapping strategy needs to be robust enough to work in very harsh conditions such as those found in oil wells. Robustness is studied in this chapter, addressing three different topics.

First, the effect of surface roughness and deformations is investigated. In harsh environments, simple shapes might present macroscopic or microscopic deformations, or be covered by a layer of mud or stones. Some regions of the environments can be too complex or too distorted to be represented by simple shapes. The approach needs to be able to cope with these situations, even at the cost of increasing mapping time.

Second, the approach is tested when the robot is submerged in heavy and viscous fluids, such as the drilling fluid filling an oil well. The robot has been sealed and submerged in water and in sucrose solutions with high sugar concentrations, to test the search strategy in such conditions.

Third, the effect of variable joint backlash is studied. Joint backlash often cannot be avoided in such harsh environments, because of the thermal expansion in the transmission gears. This research develops an original method to compensate for backlash positioning error, by simultaneously identifying the unknown surface and the backlash value.

After these three topics are presented, an application of tactile exploration in a very harsh environment is described: the mapping of a crack in an underwater oil-spilling pipe.

4.1. Irregular Surfaces

This approach has been developed to map man-made structures, which can be represented primarily as the composition of simple geometric primitives [58]. However, the harsh environment conditions have a strong impact on the surface shape. The man-made structures can deform, altering the originally regular primitives. Surface corrosion and deposits of dirt or mud yield rough and irregular surfaces. Here, a series of simulations and laboratory

experiments investigate the robustness of the tactile exploration approach to these real-world situations.

4.1.1. Robustness of fitting to surface roughness

This test investigates the ability of surface fitting to determine the underlying shape when a regular geometric primitive is corrupted by local surface roughness.

Numerical simulations were presented in section 3.4.1, to show the superiority of discrete probing compared to continuous tracking (Figure 10). Those simulations also show that the discrete probing approach is quite robust to noise. When the RMS roughness equals 1/20 of the cylinder radius, the cylinder is identified more that 80% of the times. This percentage is even higher in practice, because the fitting procedure is repeated every time a new point is added to the dataset.

The prototype robot and the modeled oil-well junction are used to prove the same concept experimentally. The surface of one of the two junction cylinders has been completely covered with a layer of gravel of size 3 to 10 mm (Figure 19). The robot explores the surface without any prior knowledge, and the number of points required to recognize the approximated cylinder is monitored. A primitive fit is accepted when the RMS error is lower than the constant f (see section 2.2). Ten trials were run with a value of f = 4 mm and different starting locations. The experiment stops when the cylinder is identified with good accuracy (cylinder axis within 3° of the ideal axis and radius within 3% of the average value). The average number of points to reach such accuracy is 16.4, with a standard deviation of 4.1. In average, 1.3 points are recognized as outliers by the segmentation procedure and excluded from the cylinder fit.

Figure 19. Experimental test with a rough surface. The surface of a cylinder is covered with gravel, and the ability of the robot to determine the approximating cylinder is monitored.

4.1.2. Robustness of segmentation to surface roughness

The previous paragraph shows that the fitting procedure can recognize a single primitive when corrupted by roughness. This paragraph studies the effect of several rough primitives in the environment. In this case, the segmentation needs to correctly detect and identify them.

The effect is evaluated in terms of performance of the exploration algorithms. The four exploration strategies compared in section 3.5 (USD, Best Cone, random and semi-random) are evaluated in the same Plane-Cylinder-Sphere shape (Figure 12), where random noise independently affects each point measurement. Noise is increased at constant intervals, and 20 trials were run for each value of noise and each strategy. Since points are touched discretely, the presence of random noise has a very similar effect as surface roughness.

Figure 20 shows the average path travelled by the robot tip when noise variance (σ) increases from zero to 2% of the cylinder radius, a rather large variation. Within these noise values, the algorithm is robust enough to eventually reconstruct the correct surface model. With higher values of σ, the convergence of the fitting and segmentation procedure is not always assured. The figure shows that an increase in noise implies more data points, and therefore higher exploration time for all the strategies. The Best Cone remains the best performing strategy for any noise value.

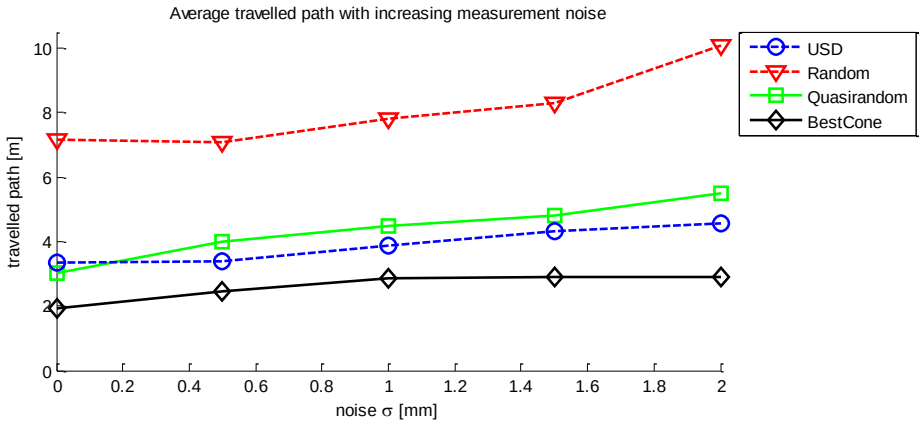

Figure 20. Length of the travelled path for the four strategies with increasing measurement noise.

4.1.3. Deformed primitives

The harsh environmental conditions can often deform surfaces that were originally composed of ideal primitives. This is the case for oil-well junctions, where the steel pipes are deformed by corrosion and by the stress they withstand during casing (insertion of the pipe into the well) and lateral well drilling.

The effect of geometric deformations on the map representation is controlled by the parameter *f*, the tolerance on the distance between a point and the modeled surface. To show this dependency, an elliptical cross section is given to the cylinder in the generic Plane-Cylinder-Sphere environment. When the eccentricity is not extreme, a single cylinder is reconstructed, see Figure 21. When the eccentricity increases, the elliptical shape is interpreted

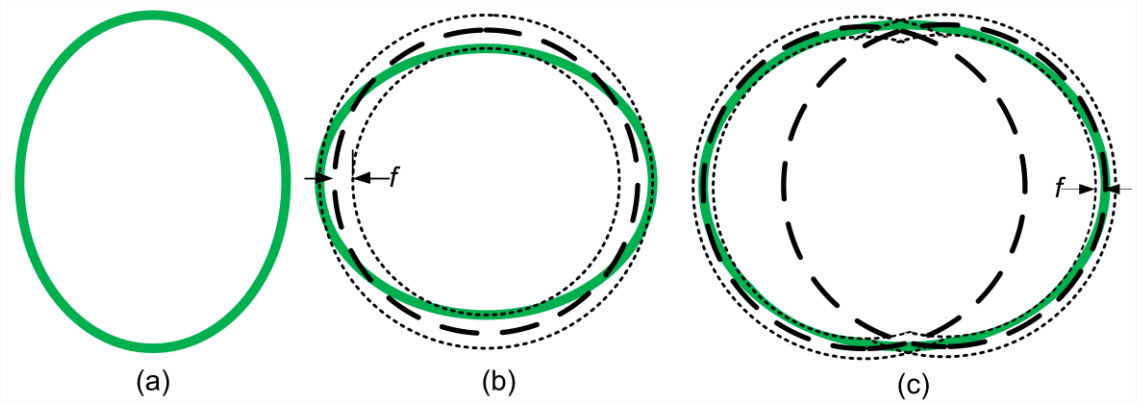

Figure 21. Behavior of the segmentation method for a cylinder with elliptical cross section. (a) Elliptical section. (b) A single cylinder is fit if the tolerance f is high enough. The dashed line is the fit circle, the dotted lines represent the tolerance width. (c) If f is low, more cylindrical patches are formed.

as several patches composed of cylinders with different radii, provided enough points are probed to characterize each patch. For example, with eccentricity 0.5 (i.e. major and minor axes are 107% and 93% of the original radius), the elliptical cylinder is identified as a single primitive in 90% of the trials when f is 5% of the cylinder radius.

4.1.4. Irregular surface

This section shows the behavior of the algorithm when a part of the surface is too complex to be represented as a geometric primitive. When several points are touched close to one another, but no primitive can be fit to them, the segmentation labels the surface as irregular, and represents it with a triangular mesh joining the touched points. A triangular mesh can represent any shape, making the approach feasible for any surface. However, a mesh requires a high number of points to describe a surface, rather than just the dozen points needed to determine a primitive. Thus, a mesh is used only when necessary.

A simulation using the Plane-Cylinder-Cone environment shows the behavior of this approach. The plane is replaced with a very rough and distorted surface, see **Error! Reference source not found.**. Since the parameter f (maximum distance between touch points and reconstructed surface) is set to a value lower than the plane surface roughness, this surface is not recognized as a plane. The algorithm needs to densely probe the surface, and reconstructs it as a triangular mesh. The other primitives are recognized with just a few touch points.

4.2. Viscous Fluids

Several harsh environments are filled with dense and viscous fluids, such as drilling mud in oil wells. This section studies how fluids affect the control algorithm and the search strategy. First, analytical calculations estimate the exchanged forces between a manipulator and the

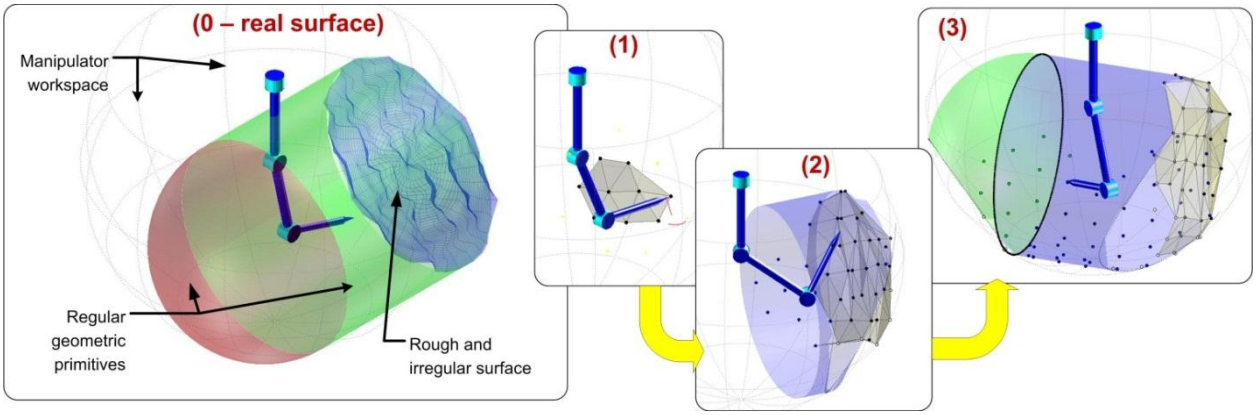

Figure 22. Exploration of an environment including an irregular shape. (0) Surface to be explored, composed of geometric primitives and a generic, rough surface. (1) Beginning of the exploration: the rough surface is represented as a mesh. (2) A primitive is discovered. (3) End of the exploration. The primitives require much fewer points than the mesh to be correctly described.

fluids. Then, the prototype manipulator is tested in hardware experiments when submerged in several viscous liquids.

4.2.1. Background

4.2.1.1. Fluids in an oil well

The oil-well junction mapping provides an example of viscous fluid surrounding the robot. Oil wells are filled with an opaque fluid called *drilling mud*. During the drilling process, the drilling mud has several purposes, including lubrication of the drill bit, transportation of the debris cut by drilling, and stabilization of the wellbore (compensation for the pressure of the surrounding rocks) [77]. When drilling is complete, the well is filled with *completion fluids*, whose purpose is to provide the right hydrostatic pressure to stabilize the well.

Drilling muds and completion fluids are dense opaque and relatively viscous. Their properties change from well to well, and even as a function of depth within the same oil bore. Density ranges from 0.8 to 2.5 kg/l. Viscosity ranges from 1 to 50 cP. As a comparison, water has a density of 1 kg/l and a viscosity of 1 cP, and Italian extra virgin olive oil has a density of 0.9 kg/l and viscosity of 75 cP.

4.2.1.2. Dynamic effects of fluids on a robot

Fluids effects on robots have been studied mainly to predict the behavior of underwater vehicles. There is relatively little literature on the subject [78-81].

Fluid effects could be directly computed using the Navier-Stokes and continuity equations, assuming the fluid is incompressible and Newtonian. This approach is not computationally tractable, except for trivial geometries. A simpler approach will be used here, consisting of separating the total force applied by a fluid into three components:

1. **Buoyancy.** Buoyancy represents the effect of gravity. The force is proportional to the displaced fluid m_f and it is applied to the *center of buoyancy*, the center of the volume of the body, and directed opposite the direction of gravity. Denoting with a the gravity vector:

$$F_b = -m_f\, a \tag{4.1}$$

2. **Added mass.** When a body surrounded by a fluid accelerates, the surrounding fluid particles must accelerate with it, creating a reaction force on the robot surface. Since this force is proportional to the acceleration of the robot, it can be represented as a mass increase on the robot itself, called added mass. For a rigid body with 6 degrees of freedom, a matrix M can be defined relating any of the six accelerations to the six force components. For a cylinder of length L and radius R, such as a link of the experimental manipulator:

$$M = diag\left\{\frac{1}{10}m \quad \rho\pi R^2 L \quad \rho\pi R^2 L \quad 0 \quad \frac{1}{12}\rho\pi R^2 L^3 \quad \frac{1}{12}\rho\pi R^2 L^3\right\} \tag{4.2}$$

3. **Viscous drag.** Drag and lift are forces proportional to the relative velocity of a body with respect to the surrounding fluid. Lift is normal to the velocity, and it is always negligible for a manipulator. Drag is in parallel to the velocity. Given the *drag coefficient* C_D, the contact area A, the fluid density ρ, the speed V, the drag force for a translating body is:

$$D = \frac{1}{2}C_D A \rho V^2 \tag{4.3}$$

A rotating body, such a manipulator's joint, can be divided into thin strips which can be considered as translating bodies. The total force and torque are the sum of the contributions of each strip. Drag coefficients depend on the Reynolds number and are determined experimentally. For a cylinder such as the links of the oil-well prototype, the drag coefficient is approximately 1.2 for a large range of viscosities and robot speed [82].

4.2.2. Dynamic effects on the oil-well prototype

The oil-well prototype is composed of cylindrical links, and joints are surrounded by semi-rigid bellows. To make computations feasible, joints are approximated as spheres. This is acceptable because the joints are small compared to the links. With this approximation, forces can be computed analytically. Here, the results of the calculations for the second link are presented as a representative case. The link has diameter 30 mm, length 302 mm, weight of 0.467 kg, inertia of 0.0077 kg m^2 and travels with an average speed of 10 cm/s. The exchanged forces are summarized in Table 2.

Table 2. Forces exchanged between fluids and the second link of the oil-well prototype.

		Water (ρ = 1 kg/l ; v = 1 cP)	**Dense drilling mud** (ρ = 2 kg/l ; v = 50 cP)
Buoyancy	Displaced mass Buoyancy/gravity ratio	0.14 kg 30%	0.28 kg 60%
Added mass	Link's added mass Added /real mass ratio	0.0019 kg m^2 25%	0.0038 kg m^2 50%
Viscous drag	Reynolds number Total drag force on the link	3x10^4 0.04 N	600 0.05 N

The table shows that these three components reduce gravity by 25% (from buoyancy) and increase link inertia by 30% (from added mass). The increase in damping (from drag) is negligible. Considering that most of the inertia is due to the motors (because of the high gear ratios), and that gravity can be easily compensated for, the total effect of the fluid for this application is not influential.

4.2.3. Experiments

To confirm the results from the analytical study, the performances of the Best Cone algorithm are compared when the experimental tank is filled with several fluids at different densities and viscosities. To provide a fair comparison, the robot movements are constrained to be the same for each trial. The prototype manipulator is sealed by covering the second and third joints with appropriate bellows (Figure 23). Three fluids are used: water, simulating properties of some lower-density oil well fluids, and two sucrose solutions (with 45% and 60% sugar concentration), simulating more viscous and dense well fluids. The addition of sugar increases viscosity and density, as shown in Table 3:

Table 3. Density and viscosity of sucrose solutions

Sugar concentration % *(kg sugar/kg solution)*	**Density** *(kg/liter)*	**Viscosity** *(mPa·s)*
0	1	1
45	1.18	10
60	1.29	55

Figure 24 shows a 5 seconds detail of the torques required by the second link with no fluid, sugar 40% and sugar 60% respectively. Neglecting fluctuations due to noise, no difference can be noticed in the applied torques, at any time of the exploration. Figure 25 shows the RMS value of the torques needed by the last two joints, considering only the time when the robot is moving. The effect of different fluids is negligible, as analytical studies predict. This validates the use of the algorithms proposed in this work when operating in an environment filled with dense and viscous fluids, such as oil wells.

Figure 23. Experimental prototype in the junction tank when submerged in water (left) and in a 40% sucrose solution (right). The red spot on the right of the manipulator in water is the lab pet, a betta fish called Captain.

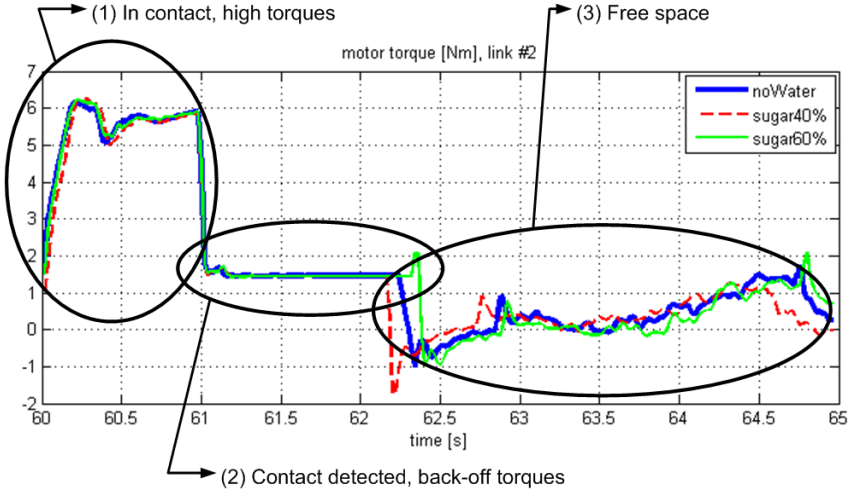

Figure 24. Torques required in a 5-second time interval by the second link in three cases: with no fluid, with sugar at 40% concentration, and with sugar at 60% concentration. No substantial difference can be found. Three moments of the real-time control strategy are highlighted. "Back-off torques" indicates a moment when torques are kept low for a second once contact is detected, to temper a residual dynamic effect of joint elasticity in the position measurement.

4.3. Joint Backlash

This research develops an approach for very harsh environments, with extremely high temperature and pressure swings. In an oil well junction, temperature and pressure reach 300 °C and 1500 atm. The high temperature swing prevents the use of tight gears, which would

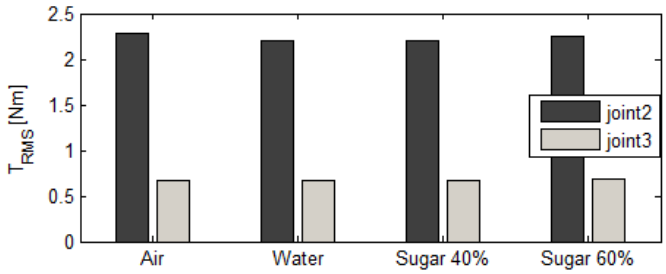

Figure 25. Required motor torques when the tank is filled with fluids with different viscosity. The difference is negligible. Torques for the first joint are not shown since the link is not completely submerged.

wind up because of thermal expansion. The use of loose gears results in large and time-varying joint backlash. However, backlash is a significant source of measurement error, and needs to be identified and compensated for, while the robot is performing its mapping mission Furthermore, the high pressure requires the use of stiff joint seals that produces high joint friction: hence, the effect of friction needs to be taken into account while compensating for backlash.

This work proposes a new technique that identifies backlash while simultaneously mapping unknown surfaces. The robot probes the surfaces in several locations, and computes the contact points using encoder readings and kinematics. Backlash compensation is possible in this application because, when the robot is pressing on the environment, the motor torques force the gears to be in one side of the backlash play. Therefore, estimating the amplitude of the transmission's backlash is sufficient for its compensation. But the amplitude cannot be determined a-priori, given the extreme and unknown down-well conditions. Thus, backlash needs to be identified and compensated while the surface is explored. Here, a least squares minimization is formulated, where both the joint backlash values and the surface parameters are identified. The operation is performed during the surface mapping and repeated when new information is available, improving the parameters of both backlash and surface. The method has been validated in representative case studies and hardware experiments.

4.3.1. Previous work on backlash compensation

Robot calibration has been widely studied in the literature [83-87]. Most often these studies deal with kinematic errors, and not with backlash. They require either some external sensors (open loop methods), or a point or surface as a reference (closed loop methods [83, 85]). Only a few studies directly consider backlash. The standard approach to determine backlash, called static test, manually blocks the output link and measures the movement of the input [88]. This measurement requires manual operations and cannot be performed on-site. An approach that does not require special off-line testing uses signal processing techniques to detect the presence of backlash [89, 90]. One of these studies excites the joints with a band-limited random signal and monitors the coherence function between input and output [89]. Another study exploits the robot's dynamics during its normal operating movements, and detects backlash using the Wigner-Ville distribution and correlation techniques [90]. These methods are very sensitive to

noise, or they require the use of several accelerometers to reduce this noise sensitivity. Moreover, these studies are useful to detect the presence of backlash beyond a threshold, but not to identify its amplitude. Approaches that identify the amplitude of backlash have been proposed. Examples include a method that monitors the change in speed of the input gear due to the impact with the output gear [90, 91]. Another approach uses a neural network to identify and correct backlash positioning error [92]. All these methods require precise models of gear geometry and dynamics, including inertias and frictions, which are difficult to obtain and affected by the harsh environments considered here. In conclusion, the backlash identification techniques developed to date are not applicable to a robot exploring harsh, inaccessible, and unknown environments.

4.3.2. The approach: simultaneous tactile mapping and backlash identification

The manipulator explores the environment by touching the surface with its tip. The position of the probe tip, and therefore the contact point on the surface, is computed from the manipulator's joint angles. If the joint torque is sufficiently high when the robot is pressing against a surface, the gears are forced to be in one side of the backlash play. Thus, backlash error can be compensated by subtracting the joint backlash amplitude from the encoder measurement.

The approach proposed in this research simultaneously identifies both the surface and the backlash amplitude (see Figure 26). A surface primitive can be identified with few touch points. When further touch points are available on the same primitive, this additional information is used by the robot to estimate and compensate for its own backlash. The exploration and the backlash estimation then proceed hand in hand: the backlash estimation improves while a larger portion of the surface is touched; this improvement in turn increases the precision of the robot and therefore helps the correct identification of the surface parameters. The procedure stops when both surface and backlash parameters converge. In several situations for such harsh

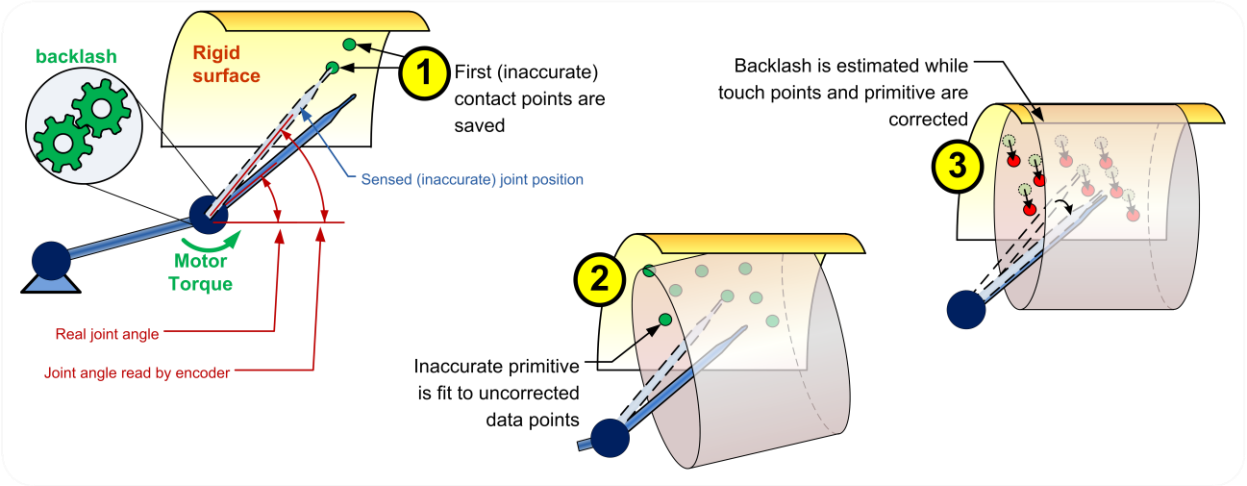

Figure 26. The approach for backlash compensation: simultaneous primitive estimation and backlash identification.

environments, the joint backlash can reach a few degrees per joint and it is the dominant source of measurement error. Other errors, such as imprecisions in robot kinematics, link deformation, and encoder resolution, will be smaller. Here, these secondary errors are treated as noise in the identification process. There are no dynamic effects, because the robot is motionless while touching a point.

4.3.3. Backlash identification

The variables used in this analysis are:

- Q_E Vector of encoder joint angles, corrupted by backlash
- Q_B Vector of joint rotations due to backlash
- Q_R Vector of real joint angle locations
- a Magnitude of joint backlash (vector)
- P Contact points location, in Cartesian coordinates
- β Surface parameters
- T Joint motor torques (vector)
- J Jacobian of the kinematic equations
- μ Vector of joint static friction coefficients

It is assumed that joint encoder angles are known, and a rough estimate of the joint torques is estimated from the commanded torque to the impedance controller. The joint angle measurement error is assumed to be only due to the backlash:

$$Q_R = Q_E - Q_B \tag{4.4}$$

It is also assumed that the dominant error in forward kinematics is due to backlash. The errors due to imprecise kinematics, deformations, and finite encoder resolution are modeled as noise with a uniform Gaussian distribution with zero mean and variance σ_E^2:

$$P = kin(Q_R) + N(0, \sigma_E^2) \tag{4.5}$$

The initial kinematic error due to backlash is assumed to be small enough to allow for the identification of the primitive types using uncompensated touch points. This assumption is valid in practice, because the backlash error is often consistent among close points, and the shape of a small patch is not substantially affected by the backlash. The primitive type defines the distance function d between the primitive, with parameters β, and a touch points P:

$$d = d(P, \beta) \tag{4.6}$$

For each joint j, the backlash model consists of a function f expressing the unmeasured joint rotation due to backlash Q_{Bj} as a function of the torque T_j and of the amount of play in the transmissions a_j. This function depends on the static joint friction, which prevents the joint from reaching the end of the clearance when the torque is low. The direction of backlash error depends on the procedure used to align the joint encoders. Here backlash is assumed to be symmetric and linear in a:

$$Q_{Bj} = \alpha_j f(T_j) \tag{4.7}$$

For negligible joint friction, $f(T_j)$ is simply the sign of T_j. With significant friction, the location of backlash is unknown when the torque is smaller than the joint static friction coefficient μ, and it is modeled as a uniform random number. Denoting *rand(a,b)* as a uniform distribution between *a* and *b*, the *backlash model* (Figure 27) is:

$$Q_{Bj} = \alpha_j f(T_j) = \begin{cases} \alpha_j \, sign(T_j) & \text{if } |T_j| > \mu_j \\ rand(-\alpha_j, \alpha_j) & \text{if } |T_j| < \mu_j \end{cases} \tag{4.8}$$

An estimate of μ is obtained by providing an increasing torque to the arm, and monitoring when the link starts to move. Precision on the value of T and μ is not critical because the backlash function returns one, independently of T and μ, when the torque is sufficiently high.

To estimate the backlash values a, and to refine the primitive's parameters β, the squared distances between the backlash-compensated points and their associated primitives is minimized. The approach is the same as the one introduced for the surface model construction in section 3.3.1. However, here the touch points are not constant anymore, but dependent on the backlash parameters. Using Eq.(4.4)-(4.7) and omitting the zero-mean random term since it does not affect the result, the least squares minimization (3.3) for the N touch points P_i can be expressed as:

$$\min_{\alpha,\beta} \sum_{i=1}^{N} d_i \left(kin\left(Q_{E,i} - \alpha f(T_i)\right), \beta \right)^2 \tag{4.9}$$

This expression is a function of the robot kinematics, the surface (β and the function d_i), and the backlash (a and the function f). The minimization can be solved either by linearizing it or by an iterative nonlinear search.

4.3.3.1. Linearization

Since backlash is relatively small, and an estimate of the primitive parameters is available, the distance function can be linearized around the known values:

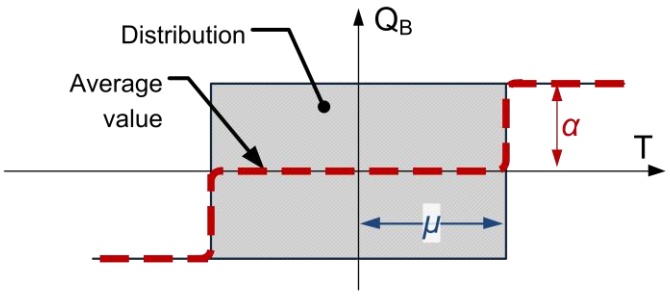

Figure 27. Backlash model, considering the effect of friction. When torque is low, the real position of the backlash is unknown and a uniform probability distribution is used.

$$d_i = d_{0i} + \sum_j \frac{\partial d_i}{\partial \alpha_j} \alpha_j + \sum_k \frac{\partial d_i}{\partial \beta_k} \beta_k \tag{4.10}$$

The linearized least squares problem becomes:

$$\left[\frac{\partial d_i}{\partial \alpha_j} \quad \cdots \quad \frac{\partial d_i}{\partial \beta_k} \quad \cdots \right] \begin{bmatrix} \alpha_j \\ \beta_k \end{bmatrix} = -d_{0i} \tag{4.11}$$

The derivative terms in the above matrix are easily expressed as a function of the surfaces, the kinematic Jacobian and the backlash function. The required computational time for this approach is low. Unfortunately, the estimates obtained this way are biased and very sensitive to noise. This is not surprising, because the minimization of Eq. (4.9) includes the surface parameters, as in the standard primitive fitting of Eq (3.3), which requires specific nonlinear methods to be solved reliably. Thus, the linearized expression cannot be used to solve the least squares problem. Nevertheless, it has been introduced because, as discussed later, it provides a tool to determine which backlash parameters can be identified.

4.3.3.2. Nonlinear formulation

As explained in section 3.3.1, effective methods have been developed to compute a least square fit of geometric primitives [66-68]. To apply these methods, the minimization of Eq. (4.9) is rewritten, separating the backlash and surface contributions:

$$\min_{\alpha} \left[\min_{\beta} \sum_{i=1}^{N} d_i \left(P_i(\alpha, Q_{Ei}, T_i), \beta \right)^2 \right] \tag{4.12}$$

The inner minimization of is only dependent on the surface parameters, while the contact points P_i are constant. Hence, the same method chosen for surface model construction can be used. A few iterations with a local least squares minimizer, such as the Levenberg-Marquardt algorithm, provide a good solution. The outer minimization, seeking the best backlash values, is more complex, because the function presents several local minima. A local minimizer would have the tendency to find only such minima; thus a global minimization algorithm is needed [93]. In this research, the Single-Linkage Multi-Start algorithm, a two-phase stochastic method, provided excellent results. In the first phase, the minimizing function is evaluated in several initial points, and these points are clustered according to their function value and their distance from each other. In the second phase, only one local search for each of these clusters is computed, using again Levenberg-Marquardt, and the best result among these searches is chosen.

4.3.4. Small joint torques

The identification procedure is based on the assumption that the position of a joint in the clearance is known to be in the same direction as the applied torque. When the torque is small this can be not the case, and the real joint position is unknown, as the random term in the

backlash model of Eq. (4.8) describes. This behavior creates two problems. First, the value of the backlash function is required in the minimization, as in Eq. (4.9), and only a probability distribution is available when torques are low. A good approach in this case is to use the mean of the distribution.

The second problem is more complex: points recorded when some joints have small torques are more imprecise than those with higher torques. The approach used in this research consists in weighting the contribution of the points according to their uncertainty. In a general least squares formulation, the *best linear unbiased estimator* is obtained by weighting each measurement with the reciprocal of the measurement variance [94]. In this case, the "measurement" is represented by the distance between point and surface. Thus, the variance of this distance due to backlash uncertainty is computed, and its reciprocal is used as a weight. The minimization becomes:

$$\min_{\alpha,\beta} \sum_{i=1}^{N} \frac{1}{\sigma_i^2} d_i \left(kin\left(Q_{E,i} - \alpha f(T_i)\right), \beta \right)^2 \tag{4.13}$$

The variance is computed in three steps (Figure 28):

1. For each joint j, the probability distribution of the backlash error Q_{Ej} as a function of the torque T_j is defined by the backlash model, Eq. (4.8). To make computations feasible, the distribution of Q_{Ej} is approximated as a Gaussian distribution with the same mean λ and variance σ^2 as the original. For a fixed joint friction coefficient μ_j:

$$\sigma_j^{\,2}(T_j) = \begin{cases} \alpha_j^{\,2} / 3 & \text{if } T_j < \mu_j \\ 0 & \text{if } T_j \geq \mu_j \end{cases} \tag{4.14}$$

If μ_j is not known with confidence, the uncertainty can be represented by describing it with a mean λ_μ and variance σ_μ^2. In this case, the distribution of Q_B smoothly changes as a function of T, and its variance can be expressed, after some algebra, as a function of the Gaussian error function:

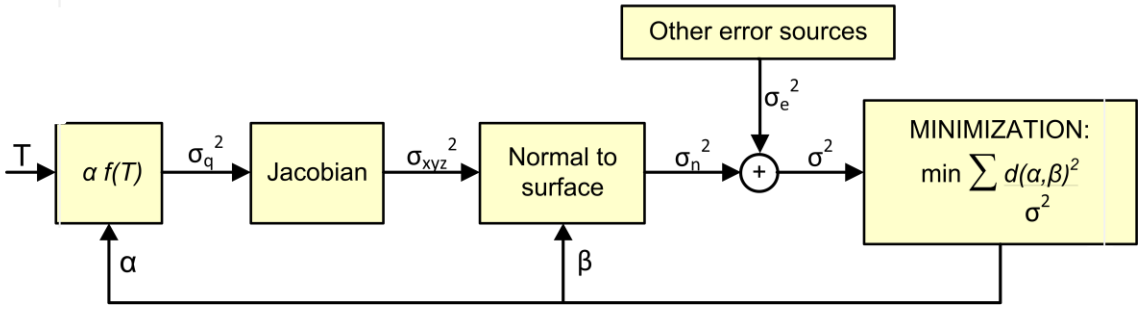

Figure 28. Weighted least squares according to the uncertainty due to small torques and joint friction.

$$\begin{cases} \gamma = erf\left(\dfrac{T_j - \lambda_{\mu j}}{\sqrt{2}\sigma_{\mu j}}\right) \\ \sigma_j^{\,2}(T_j) = \dfrac{1}{12}\alpha_j^2\,(1-\gamma)(5+3\gamma) \end{cases} \tag{4.15}$$

2. The probability distribution in joint space is mapped into Cartesian coordinates using the kinematic Jacobian J. The operation is linear because Q_E is assumed to be Gaussian-distributed: the sum of random variables is the convolution of their distributions, and convolution of Gaussian distributions is another Gaussian with as mean and variance the sum of the means and variances. Therefore, the variance of the distribution in Cartesian space is the sum of the contribution of all the joints, plus the contribution of the random measurement error σ_e that was defined in Eq. (4.5). For the x component:

$$\sigma_x^{\,2} = \sum_{j=1}^{M} J_{xj}^{\,2}\,\sigma_j^{\,2} + \sigma_e^{\,2} \tag{4.16}$$

where $j = 1…M$ represent the degrees of freedom of the robot. The same applies for the y and z components.

3. Since the least squares minimizes the distance between points and surface, the only component of the error affecting the minimization is the component normal to the surface (Figure 29). Thus, only the component of the variance in the normal direction σ_n is used as a weight in Eq. (4.13) . Denoting by n the surface normal vector, this variance is:

$$\sigma_n^2 = n_x^2\,\sigma_x^2 + n_y^2\,\sigma_y^2 + n_z^2\,\sigma_z^2 \tag{4.17}$$

This weighting system is computationally very fast as it involves only a few multiplications. Note that since Eq. (4.14) depends on a and the normal depends on β, the weights are a function of both backlash and surface: they are updated after every iteration of the minimization. The weighting system works as intuitively expected: if some points have small torques in joints that create some normal uncertainty, their variance will be higher and their weight smaller. The variance never goes to zero because of the presence of random measurement noise σ_e in Eq.

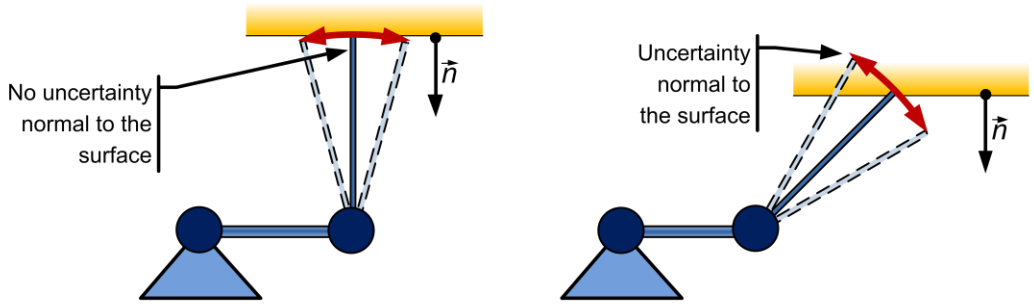

Figure 29. Backlash generates error (and therefore uncertainty) in the distance function only when its effect on the tip position is normal to the surface.

(4.16). If this random term is high compared to the backlash, all the weights have comparable values.

4.3.5. Identifiable parameters

Since the identification procedure is performed simultaneously with the surface exploration, the backlash values are refined as larger portions of the surface are explored. In the initial stage of the search, or when the explored environment is small, not enough data are available to correctly estimate all the backlash values. Given the complexity of the minimizing function in Eq. (4.13), the estimation of a parameter without enough data could potentially lead to a wrong value, and compromise the success of the procedure. Therefore, a method is needed to determine which joints can be estimated, so that the minimization will include only those variables, while freezing the others.

To determine if a parameter should be estimated, the linear model of Eq. (4.11) is used. This model is too simple to provide a precise estimate of the parameters, but it is sufficient to evaluate if estimates would be reliable. Let A denote the normalized version of the matrix with the derivative terms in Eq. (4.11). The normalization is needed because the parameters in A have different units (length and angle). Let $B = A^T A$, and let c_B denote its condition number (the ratio between the largest and the smallest eigenvalues) [3].When c_B is close to unity, all the variables can be found reliably. When c_B is high, instead, some variables do not have enough data point to be estimated. To determine which variables, the eigenvectors of B whose corresponding eigenvalues are zero (or very small, in the presence of noise) are computed. These eigenvectors represent the relationship between the variables that cannot be estimated, given the current data. Therefore, the nonzero elements of those eigenvectors (or large enough compared to the noise) indicate which variables must be held constant in the minimization.

This linear formulation determines whether a parameter can be reliably estimated given the points touched during the exploration. However, identifiability is different: it is the theoretical possibility to obtain the value after an infinite number of measurements. This is not a function of the process or the touched points, but only of environment shape and robot kinematics. Identifiability is useful to determine if a surface is appropriate to estimate all the backlash parameter, or as a check to decide whether to stop the exploration process if no more parameters can be identified. Here, identifiability is computed by evaluating the size of the part of the surface where each joint backlash has an influence. Only if this portion is large, compared to the size of random errors, there is enough information to evaluate the backlash. The influence of a joint's backlash is evaluated with the function $f(T_j)$ used for the backlash model (Eq. (4.8)), where T_j are the joint torques creating a force F_0 normal to the surface and with magnitude defined by the impedance controller:

[3] A computationally faster approach involves the QR factorization of matrix A. But for few, sparse data the difference is negligible, on the order of the microsecond, and this approach is much simpler to implement.

$$T = J^T \cdot F_0 \vec{n} \tag{4.18}$$

The assumption of normal force is consistent with the observation, explained before, that only normal displacements create positioning errors. Therefore, $f(T_j)$ behaves as desired: wherever backlash has influence, the torque T_j is high, and $f(T_j)$ returns 1 (or -1). The final expression for S_j the portion of the surface that can be used to identify backlash in joint j, is a simple integration of this influence function (in absolute value) over the surface reachable by the robot tip, S_{reach}:

$$S_j = \int_{S_{reach}} \left| f\left(T_j\right) \right| ds \tag{4.19}$$

4.3.6. Representative case studies

This section presents the results of the simultaneous mapping and backlash identification for two environments modeled in Matlab™. The robot chosen for these studies has the shape of the laboratory prototype built for oil-well exploration, with all link lengths set to 1 m for simplicity. The environment is completely unknown and is composed of geometric primitives. Each point measurement is corrupted by both backlash and an independently distributed Gaussian noise, as in Eq. (4.5).

The two environments are shown in Figure 30, together with the touch points at the end of an exploration trial. The first shape represents two barrels on a flat floor and it is explored from the outside; the second represents an *L*-shaped junction with a spherical cup, explored from the inside.

In the first environment, the three joints have respectively ±1°, ±1°, and ±2° of backlash. Joint friction is such that in 46% of the sample points at least one of the joint torques is smaller than

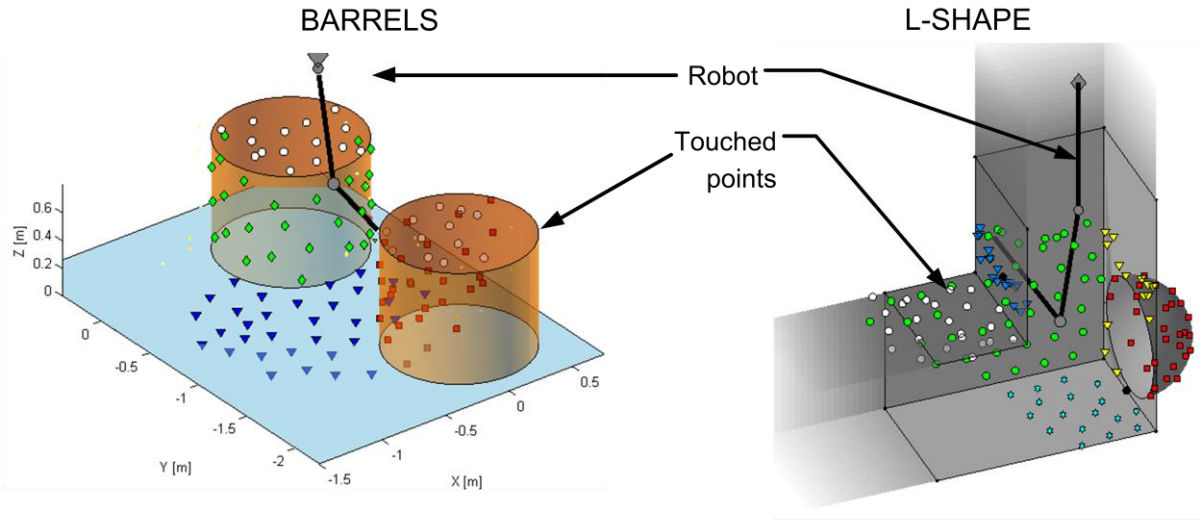

Figure 30. The two environments chosen in the case studies: two barrels, explore from the outside (left), and L-shaped junction, explored from the inside (right).

the friction coefficient, generating a uniform random location of the joint within the backlash play. The value of the friction coefficient is not known, but an estimate is available within 30% precision. Measurement noise is σ = 1 mm.

The robot probes the environment sequentially, until no more points in the workspace can be reached. Backlash is estimated and compensated during the computation, thus increasing the surface precision. Figure 31 shows the estimation of backlash and primitives as a function of the number of points touched on the surface. The top curve describes the error in the estimation of the three joint backlash values. The bottom curve shows the improvement on the parameter estimates of the four primitives (four because the tops of the two barrels are geometrically the same plane). The goodness of the estimated primitives is evaluated with a normalized similarity measure that includes primitive position, orientation and dimensions, divided by a representative dimension. The amount of backlash at the end of the study is estimated with an average error on the three joints of 0.015°, which is 1.3% of the real backlash value. The backlash compensation greatly increases the precision of the reconstructed surface: as an example, at the end of the simulation the radius of the barrels is retrieved with accuracy of 99.8%, against an accuracy of 94.4% without backlash compensation. The change in time shown in Figure 31 is also interesting. The backlash in the first joint could not be estimated before 50 points, and its value was not computed. The precision of backlash and surfaces go hand in hand, and it greatly increases when new surfaces are discovered.

The second case study, in the *L*-shaped environment on the right of Figure 30, has been performed with the same robot, but in different conditions: double noise (σ equal to 2 mm) and smaller backlash (α equal to ±0.5°, ±1°, ±1°). Backlash is therefore harder to identify. As expected, this implies a slower backlash and parameter estimation, as shown in Figure 32. Nevertheless, the final estimates are good: the final average backlash error is 0.03° (3.3% of the

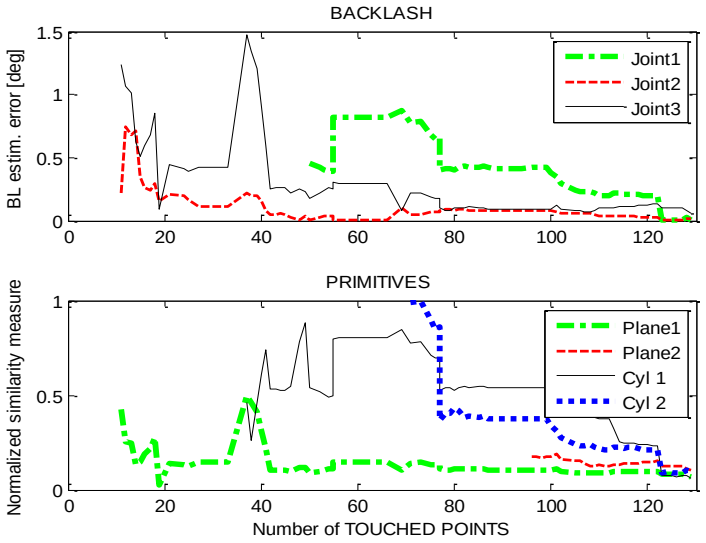

Figure 31. Progress of the backlash (top) and surface (bottom) estimates for the two barrels environment, when more touch points become available. The similarity measure to evaluate the fitted primitives includes position, orientation and dimensions.

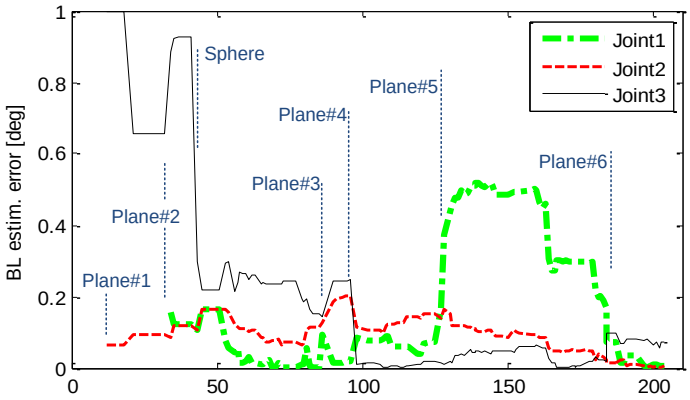

Figure 32 Progress of the backlash estimate for the L-shape while more points are touched. The detection of a primitive is shown by its name on the graph.

real backlash value), and the radius of the spherical cup has a precision of 99.7% relative to the real value.

4.3.7. Experiments

The backlash compensation method has been tested in two sets of laboratory experiments, first with a simple planar manipulator, and then with the spatial oil-well prototype manipulator.

4.3.7.1. Planar Manipulator

In this experiment, a planar robotic manipulator explores an unknown environment made of a single straight line, see Figure 33. Both joints have the same motor-encoder-gearbox unit. Measurements in the joints have very high resolution (250,000 counters per rev), but they present large backlash. This backlash has been measured statically to be 0.97°. This agrees with the manufacturer information. Joints also present significant elasticity, but its effect can be avoided by reducing the applied torques at the moment of recording a contact point location.

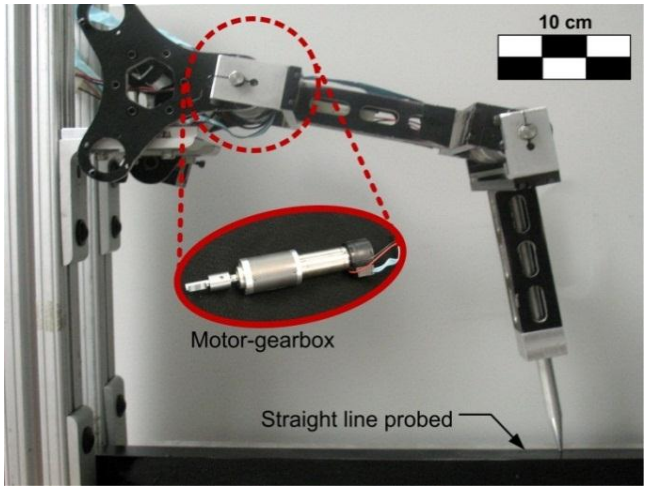

Figure 33. Planar manipulator used in the line probing, and detail of one of the motor-gearbox units.

The static joint friction coefficient is approximately 0.2 Nm.

As the manipulator autonomously explores the environment, the line parameters and the joint backlash are iteratively estimated online. At the end of the exploration, the line is probed in 19 points. The final estimated backlash values are 0.96° and 0.94°, very close to the values measured statically. Figure 34 shows the touched points with and without backlash compensation, together with their best fit line. Compensation greatly improves the precision: the average error of the points from the real line decreases from 2 mm to 0.25 mm.

4.3.7.2. Spatial Manipulator: the Oil-Well Exploration Prototype

The second set of experiments studies backlash compensation with the prototype manipulator for oil-well exploration. For such robot, both backlash play and joint friction are slightly dependent on the configuration, mostly because of the bevel gears in the transmissions. The average backlash value in the three joints is respectively 1.4°, 1.1° and 2.6°, and the average static joint friction is 1 Nm, 0.7 Nm and 0.1 Nm.

First, the algorithm was tested in the modeled oil well junction (Figure 16). In this environment, the robot's first joint axis is aligned with the main cylinder's axis, making most of the environment radial symmetric with respect to the robot. With such geometry, backlash in the first joint is tangent to the surface, and it does not create any error, but it also cannot be identified. This can be shown evaluating the size of the surface where the joint has influence. The total surface that can be touched is 760 cm², while the surfaces S_j where backlash has influence, computed using Eq. (4.19), are respectively 80, 750 and 700 cm² for the three joints. The value for the first joint is very small, and cannot be identified; the other two joints, instead, can be correctly estimated.

Second, to show a situation where backlash in all joints can be estimated, an experiment with an environment composed of three straight planes was performed (Figure 35). The same exploration strategy as in the simulations has been used. Results for the final, mapped surface

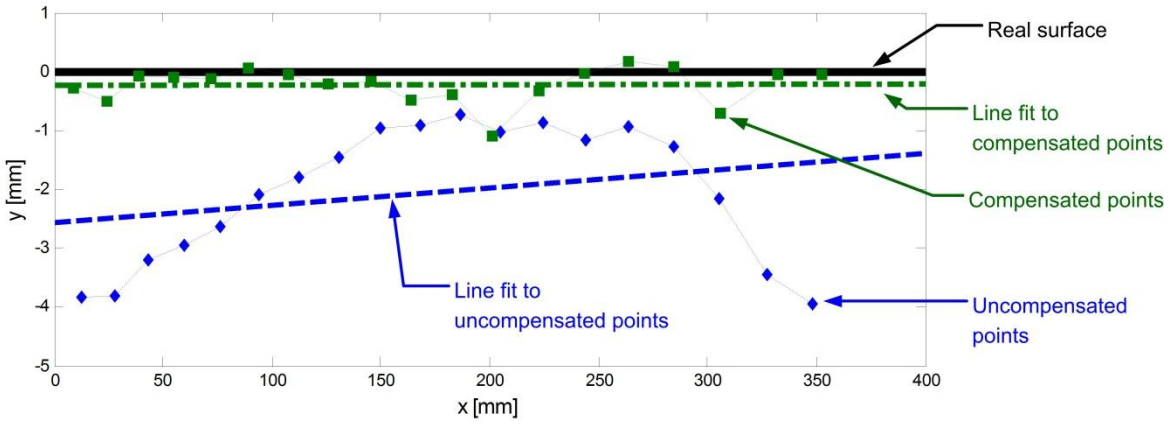

Figure 34. Probed points and best fit lines with and without backlash compensation, showing the improvement obtained with compensation. The two axes have different scales.

and the behavior of the backlash estimates are shown in Figure 35. The final estimated backlash values are respectively 1.38°, 1.15°, and 3.09°, which represent a percentage error of 1.2%, 8.3%, and 18% compared to the statically-measured values. There reason why these values are not as close as in the previous experiment lies on the complexity of backlash in this prototype robot. The precision on the final surface is highly increased. This is evaluated comparing the root mean square distance from the touched points to the planes that they fit: without backlash compensation the RMS distance is approximately 3.2 mm, while with compensation this is reduced to 1.1 mm.

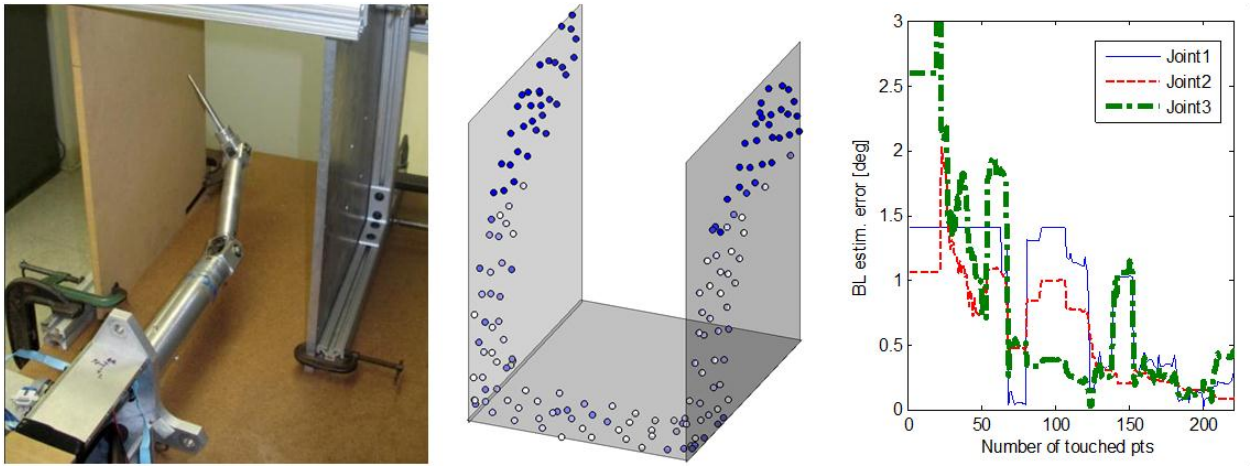

Figure 35. Exploration and backlash estimation of the three-planes environment by the prototype manipulator. Left: robot and environment. Center: the reconstructed surface, with the touched points. The intensity of the points indicates the amount of torque in the first joint, correlated to the identifiability. Right: progress of the backlash estimate while more points are touched.

4.4. Application: Mapping an Oil Spill Site

This paragraph shows the importance of the robustness to very harsh environmental conditions by proposing an application during a critical operation such as the intervention after an underwater oil spill. The application is investigated experimentally using a laboratory-reconstructed underwater environment.

4.4.1. Motivation

The collapse of the offshore oil-drilling platform Deepwater Horizon in the Gulf of Mexico resulted in the largest marine accidental disaster in history [95, 96]. The final closure of the spill was accomplished by drilling a relief well five months after the accident. Temporary solutions to limit the spill heavily relied on Remotely Operated Vehicles (ROVs), and the only solution capable of stopping the spill was the placement of an oil wellhead, three months after the accident. All the temporary attempts before that have failed, mostly because of the harsh operating environment, and fifty thousand barrels of oil per day escaped into the ocean through

three breaks in the pipe that connected the well-head drilling platform and then lay on the seabed.

Knowledge of the shape and dimensions of the cracks would have provided critical information for the repair work. However, vision sensors, normally used for the robot control, were hindered by clouds of escaping oil (Figure 2). Lasers and sonar were unreliable due to strong turbulence and methane gas mixed with the oil. Deposits of dirt, mud and oil composites accumulated on the surface of the wreckage, and the flow noise from the fast moving hydrates and oil made sonic sensors ineffective.

4.4.2. The approach

The approach developed in this research only requires a manipulator provided with joint encoders, which is readily available in most of the ROVs operating underwater. Hence, this approach could have been used during the attempts to control the escaping oil to map the shape of the oil-spilling cracks. This approach requires a manipulator mounted on a stationary base. When exploring an underwater accident site, the base is the ROV itself, which can lie on the ocean floor or anchor itself to an existing fixed structure such as a pipeline (Figure 36). If the ROV cannot be fixed, the method needs to account for base movement. Methods exist in the literature, but this topic is beyond the scope of this research [57, 97].

4.4.3. Experiment

The effectiveness of the approach has been evaluated in a laboratory experiment reproducing sea floor conditions such as those near the broken leaking pipes of the Deepwater Horizon. The environment represented in Figure 37 presents a rusty, truncated steel pipe, two perpendicular planes covered with gravel and dirt, and a stone of irregular shape. This environment is autonomously mapped, without any a-priori knowledge.

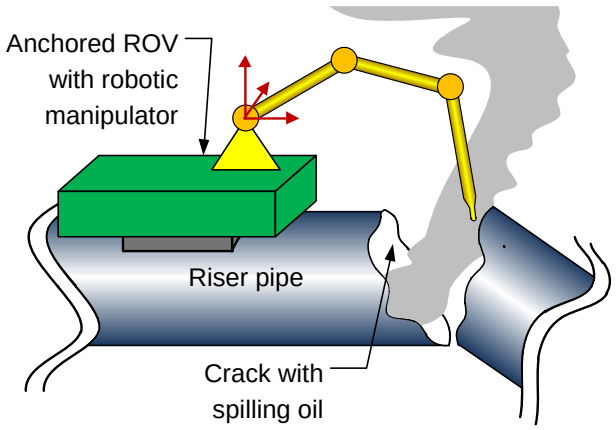

Figure 36. Schematics of the approach to tactilely map the shape of a crack in an oil-spilling underwater pipe using a robotic UAV.

Figure 38 represents the result of an exploration trial. The exploration strategy was able to identify the underlying shape of the structured elements in the environment with few data points: 18 points for the pipe, 12 for the sea floor, and 17 for the lateral graveled surface. Once a shape was identified, the robot moved to a different, unknown location; this allowed the reduction of the total exploration time. The irregular stone could not be described as any combination of primitives, and more data points were needed to represent it as a triangular mesh.

Figure 37. Environment reproducing sea-floor conditions such as those near a broken oil pipe.

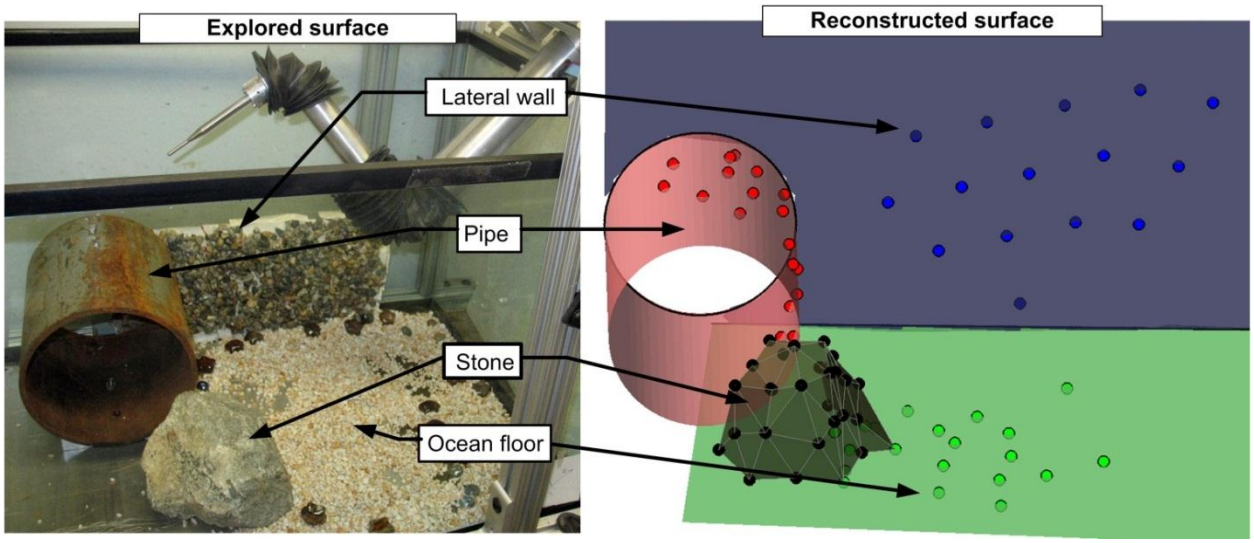

Figure 38. (Left) Manipulator exploring the reproduced submersed crash site. (Right) the reconstructed surface after the exploration, showing the final surfaces and the touched points

4.5. Conclusions

This chapter validated the robustness of tactile exploration when applied to very harsh environments.

Surface roughness and deformations are shown to reduce the speed of the exploration, without compromising the robustness of the operation. The submersion in heavy and viscous fluids was shown not to alter the performance or the requirements of the exploration algorithm. On the other hand, joint backlash could potentially compromise the procedure. This chapter develops method to compensate for backlash, by simultaneously identifying the unknown surface and the backlash magnitude. This method is shown to be effective in both simulations and experiments.

Finally, the robustness to map a very harsh environment has been tested experimentally, by exploring a laboratory reproduction of an underwater oil-spilling pipe.

Chapter 5

Exploration Strategy

The previous chapters show the feasibility of tactile exploration when mapping a harsh environment with a manipulator provided with only joint encoders. They show the importance of the exploration strategy when time is a key performance parameter, and propose a simple strategy, which is easy to implement and performs well in simple environments (section 3.4.2). However, this strategy can be improved. First of all, it does not use all the available information, but only the touched points. Furthermore, it fails when exploring complex environments which require non-trivial motion planning, such as a change in robot configuration to reach partially occluded areas. This chapter reformulates the problem, and develops a new approach to determine the exploration path using concepts from information theory. The key concept is to quantify the knowledge of the environment, and choose the path that maximizes the expected received information, discounted by the path length. Information about the environment is obtained by the touched points, the reconstructed surface, and the robot trajectory, since the volume the robot passes through needs to be free of obstruction.

The problem is approached in two steps. First, the approach is formulated for the case of a point robot in a two-dimensional environment. Second, the same approach is extended to a planar and then a spatial manipulator with joint angle limits and finite thickness, using the concept of configuration space.

5.1. Background

A motion planner is an algorithm that determines the motion required to complete a specified task. The typical example is *path planning*, the determination of a feasible path connecting two locations, the *start* and the *goal*. Exploration is similar to path planning, but it does not have a goal location; the objective becomes to generate a map of the unknown part of the environment. Motion planning has been widely studied in the literature, and it still represents an open research area. Several books and survey papers are available on the topic

[17, 75, 98-104]. This section summarizes the important concepts that are needed to understand the approach introduced in the chapter.

5.1.1. Completeness and optimality

There are two desirable properties for a planning strategy: completeness and optimality [75]. A strategy is complete if it is able to establish whether a solution to the problem exists, and, if it does, to determine it. Completeness is a difficult property to achieve in practical planner, and two weaker definitions have been introduced. A strategy based on the discretization of the environment is *resolution-complete* when it guarantees completeness only if the discretization if "fine enough". A strategy based on random sampling is *probabilistic complete* if it can find a path with high probability when such a path exists; no assurance is implied, if no path exists.

The second important property, optimality, requires a *payoff* (positive) or a *cost* (negative) function to evaluate the goodness of a strategy. Such function can be the length of the traveled path, or the amount of acquired information. A feasible strategy is *optimal* when its payoff function is maximized (or the cost is minimized).

For the case of autonomous tactile exploration in limited time presented here, these properties should be read as follows. A complete strategy will be able to explore every part of the accessible environment. An optimal strategy will do it in the shortest time, and therefore minimize the distance the robot travels.

5.1.2. Configuration space

The configuration space (C-space) is a useful concept for path planning [98, 105]. The configuration of a robot is described with a finite number of variables. For a manipulator, the configuration is represented by its joint angles (and joint velocities, if dynamics is considered). For a mobile robot, the configuration is defined by its position and orientation (and velocity, with dynamics). The configuration space is the space of all the possible robot configurations. A single configuration is represented by a single value of for each robot's variables. Therefore, the robot's configuration is always represented by a single point in C-space, independently of its shape. Similarly, a path in C-space is a continuous set of configurations represented by a one-dimensional curve.

An obstacle in the environment limits the robot's movements, and some robot configurations cannot be reached. The set of all the configurations limited by an object is the map of the object into C-space, and is called C-obstacle. By mapping all the obstacles in C-space, the whole environment can be represented.

Since the robot is a point in C-space, and its path is a curve, motion planning in C-space becomes a simpler and more general problem. Independently of the robot's type, the objective of path planning is to connect the start and goal configurations with a continuous curve, without intersecting any C-obstacle.

The complexity of planning in configuration space is determined by two factors. First of all, the dimension of the space is the number of robot degrees of freedom. Second, mapping the environment into C-space is a very complex and computationally-intense problem [106]. Analytical maps have been developed, but they are often limited to very simple robots and environments. Usually, the mapping is either computed by discretization, or it is never computed explicitly.

5.1.3. Classification of motion planners

Motion planning algorithms are most commonly classified in four categories:

1. *Grid-based planners*. The environment is approximated with a grid, and then standard graph-search methods (depth-first, breadth-first, A* and so on) are used to determine a path. These planners are resolution-complete, and resolution-optimal. Since each dimension needs to be discretized, they cannot be applied when the dimension of the configuration space is high (more than three).
2. *Roadmaps.* Called also "geometric algorithms", these planners capture the connectivity of the environment with one-dimensional curves, called *roadmaps*, joining specific locations in the environment, called *nodes*. The planner moves to the closest node, travels on the roadmap up to the closest node to the goal, and from there to the goal. There are different ways to construct a roadmap, including *visibility roadmaps* [107] and *Voronoi Diagrams* [108]. These algorithms are usually complete and sometimes optimal, but they can be only applied to simple problems.
3. *Potential Fields.* These planners treat the robot as an electrical charge, the goal as a charge of opposite sign, and obstacles as charge of the same sign [109]. The potential field generated by these charges guides the robot through the obstacle towards the goal. These algorithms are not optimal nor complete, as the robot tends to get trapped into local minima.
4. *Sampling-based planners.* These planners use a very different approach [110, 111]. Since mapping the environment in C-space is complex and computationally intensive, sampling-based planners avoid this mapping and do not construct any explicit model of C-space. Instead, the configuration is sampled in several points, and a path between start and goal is constructed by connecting the feasible points with each other. Different strategies to sample the C-space and connect the sampled points exist. Sampling-based planners are usually divided in single-query or multiple-query algorithms, according to the number of operations that are performed on the same environment. Since no explicit representation of the C-space is needed, sampling-based planners can be applied to high-dimensional C-spaces such as those of a manipulator. Most sampling-based planners are probabilistic complete (if they sample randomly), or resolution-complete (if they sample with a grid with increasing density). They are usually not optimal, but some recent approaches developed versions that tend to an optimal solution as the number of samples increase [112].

5.1.4. Planning horizon

Oftentimes, planning does not involve a single movement, but consists of series of separated steps where any following step is affected by the previous movements and by events that can occur between steps. A simple example is a chess game, where a good player plans several moves ahead, also considering the unknown moves of his opponent. The same situation occurs in exploration, where the data acquired after a movement changes the knowledge on the environment, and therefore subsequent movements.

The overall performance of a strategy is evaluated as the final payoff. Therefore an optimal planner should consider all the movements up to the end, to choose the series of steps to maximize the final payoff. When unknown events occur between the steps, the optimal strategy cannot be known a priori. Nevertheless, the planner can compute the expectation of the final payoff, and use that expectation to compute optimality.

This computation becomes very complex with a high number of steps (n) and a high number of choices at each step (*branching factor* - b), because the number of combinations increases exponentially as b^n. Therefore, computing all the possible combinations becomes unfeasible. The number of steps that a planning strategy considers is called *planning horizon*. Often the computation is so complex that only one step is computed; this case is also known as a *greedy* strategy.

5.1.5. Entropy

Entropy is a fundamental concept in information theory, which represents the uncertainty associated with a random variable. For a discrete random variable X with n possible outcomes $\{x_1, \ldots, x_n\}$ and probability function $p(X)$, the entropy is defined as:

$$H(X) = -\sum_{k=1}^{n} p(x_n)\log\left(p(x_n)\right) \tag{5.1}$$

If X is a binary variable, the entropy can be expressed as a function of the probability of one outcomes $P_1 = p(x_1)$:

$$H(X) = -P_1 \log P_1 - \left(1 - P_1\right)\log\left(1 - P_1\right) \tag{5.2}$$

Figure 39 shows the entropy H for a binary variable as a function of P_1. The interpretation of entropy as lack of information is evident in this case. When P_0 is zero or one, the value of the binary variable is known with certainty, and entropy is zero. When $P_0 = 0.5$, the outcome of the binary variable can be any of the two with equal probability, and the lack of information is maximum.

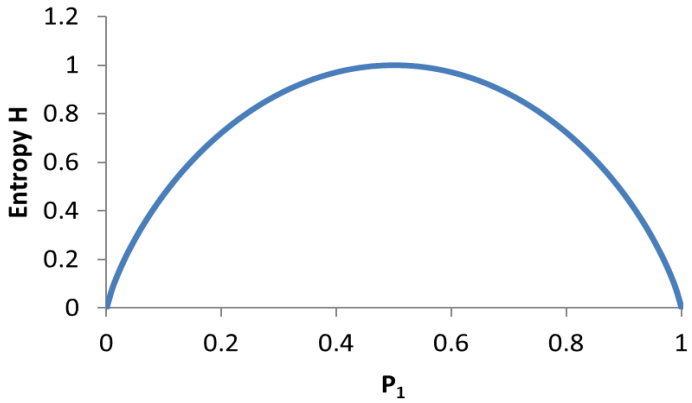

Figure 39. Entropy for a binary variable.

5.2. Previous Work in Exploration

Several techniques have been proposed to guide a robot's movements to build a map of an unknown environment. Seminal works in robotic exploration focused on graph-like environments [113], or on the detection of only specific features in the environment [114]. A common method, called *frontier-based exploration*, determines a frontier between the explored and unexplored regions, and moves the robot to visit the nearest point on the frontier [115]. Another strategy extends this concept by considering the amount of seen and unseen area with a range sensor [116]. Other strategies have been developed for similar problems, such as surveillance or mine sweeping [117].

The idea of maximizing the information obtained with the next action is introduced in [118], where the exploration is modeled as a Markov Decision Process and an optimal solution is sought. The approach has first been applied in computer vision to the problem called "Next Best View", the determination of the next location for a camera to completely scan an object [119-121]. The same concept has been applied to robotic exploration, where the next measurement is chosen to maximize the amount of acquired information by reducing the entropy of the environment [122-127]. The extension to a manipulator in configuration space has been studied [128], by mounting a camera on top of a manipulator.

These concepts cannot be applied to the problem of tactile exploration with a manipulator, for two reasons. First, a range sensor (usually a stereo camera) is always used to acquire information. The range sensor allows them to safely move to a new location, and then take a measurement in such location. This effectively decouples the two problems of path planning and entropy reduction, because the entropy is associated to only the location where the measurement is taken. In tactile exploration, information is acquired only when the robot moves in an unknown location, and there is no measurement at the end of the path. The information gain is therefore associated to the robot path and not to the end point. For instance, several paths to the same locations will provide not only a different movement cost, but also different entropy. Second, the approaches proposed in the literature consider a mobile robot or

a point camera. In such cases, there is no need to use the configuration space for path planning. The use of a manipulator with the same approach has been proposed in [128], but the information gain is completely computed in configuration space, and is missing some important information that would be crucial for tactile exploration, as explained later.

The entropic approach has been also applied to tactile perception [26-29, 45], developing interesting strategies to guide the robot's movements to locate and identify known objects. However, they only identify an object among a set of models and cannot be used to explore an unknown environment.

5.3. Tactile Exploration with a Point Robot

This paragraph introduces the entropy reduction approach for tactile exploration, by applying it to a simple case: a point-robot exploring a planar environment composed of intersecting primitives such as lines and circles.

The objective of the exploration strategy is to determine the best movements to explore the environment in the shortest time. An optimal strategy is impossible to obtain when the environment is unknown: the optimal path can only be computed once the whole environment is known (see [17]). The best that can be done is an optimal strategy given the knowledge acquired so far. However, an exploration horizon greater than one is very complex to implement, because the map can drastically change after each step. Here, a greedy strategy is implemented that considers only one single exploration movement. The approach can be summarized as follows:

1. The whole environment is discretized into cells using an equally spaced grid [129].
2. The state of each environment cell c can be either empty or occupied. Therefore, each cell is assigned an occupancy probability P_C, ranging from 0 to 1. In this work, P_C is defined as the probability of being empty. Two types of information are used to determine cell occupancy: the volume swept by the robot's movements and the partial reconstructed map.
3. Once the probability is known, the entropy of each cell can be determined using Eq. (5.2). Entropy reflects the lack of information on a cell: when P_C is either 1 or 0 the cell is known and $H=0$; when $P_C=0.5$ the cell has no information and H is maximized (see Figure 39).
4. Entropy provides a measure of the lack of knowledge about the environment. A region where cells have low entropy is known to be empty or occupied with high confidence. The total lack of knowledge of the environment is simple the sum of the entropy of each cell. Hence, the best exploration strategy reduces the environment entropy with the smallest movement cost. Since the environment is unknown, only an expected value of this reduction is used. For each candidate path, the amount of expected entropy reduction $(-\Delta H)$ and the movement cost C are computed. The path with highest ratio of entropy reduction over cost is chosen and executed. Their ratio is the payoff function:

$$bestPath = \arg\max_{all\ paths}\left(\frac{E[-\Delta H]}{E[C]^{\alpha}}\right) \tag{5.3}$$

The exponent *a*, called *aggressiveness factor*, weights the effect of movement cost. With a smaller *a*, longer paths will be chosen. This is often desirable, mostly in complex and constraining environments, because it tends to compensate for the lack of long-term planning of a greedy algorithm. In this work, *a* is considered constant. Nevertheless, nothing prevents a more complex strategy which changes *a* dynamically.

The payoff function in Eq. (5.3) uses two separated expectations for the entropy and the cost. Another option for the payoff function is $E\left[-\Delta H / C^{\alpha}\right]$. The two functions are not equivalent since entropy and cost are correlated variables. However, both formulations can be implemented with the same steps and complexity, with very similar results in practice. In this work, the formulation in Eq. (5.3) has been chosen, so that expected entropy and cost can be treated separately.

This approach computes an occupancy map of the whole environment. However, this map is only used for path planning, and not as a representation of the environment. The real environment map is still a boundary representation made of a composition of geometric primitives, as described in section 3.3. This boundary representation is computed every time a point is touched, and it is used by the motion planner to determine the locations where the robot is allowed or not to move, and to calculate the occupancy probability of each cell. Since the occupancy map is not used as a final map but only to compute the best path, the density of the grid does not need to be high.

The following two paragraphs describe the details of concepts mentioned in points 2 and 4: how to obtain the occupancy probability of the environment given the swept region and the touch point measurements, and how to compute the expected entropy reduction, given the occupancy probability of the environment.

5.3.1. Occupancy map definition

This section proposes an approach to obtain the occupancy probability of the environment. At the beginning of the search no information is assumed, therefore the initial value of each cell is $P_C=0.5$, the least informative value. When data on the environment are acquired, the occupancy probability changes accordingly. There are two types of information used to determine cell occupancy. First, a cell where the robot passed without hitting any obstacle must be empty. More specifically, the cell is empty if the robot's position sensors are assumed to be perfect. If the sensors have some uncertainty, then the probability will reflect this uncertainty. The second type of data is the surface reconstructed with the currently available data. Contact points will have $P_C=0$ (or very low, in case of noise). These points are also used to fit the probable shape of the surface, which affects the occupancy probability on the surrounding cells.

Most often in the literature, the occupancy probability of each cell is independent of the probability of any other cell [17, 126, 130]. This assumption highly simplifies the calculations, because each cell can be treated independently. Bayes' rule can be used to update each measurement at each cell, and a stable and fast logarithmic formulation of the binary entropy can be introduced. However, this is a very strong assumption, and it would only be true if the environment were populated by objects smaller than the size of a single cell and uniformly distributed in space, like a cloud. In a real environment, surfaces are continuous, and this makes the occupancy of a cell correlated with the occupancy of neighboring cells. Nevertheless, when range sensors are used, the gain in simplicity and efficiency is enough to justify the loss of information.

This is not true with tactile data. Tactile exploration provides only a little amount of information: touched points are few, sparse, and time expensive. Hence, the cell independency assumption is not acceptable. This work proposes a method to define correlation between neighboring cells based on the physical geometric intuition.

Ideally, the probability of a cell c being empty could be defined as the total number of feasible surfaces where c is empty, divided by the number of all possible feasible surfaces.

$$P_C = \frac{\sum \text{all feasible surfaces } \textit{where} C \textit{ is empty}}{\sum \text{all feasible surfaces}} \tag{5.4}$$

A feasible surface in this case is consistent with all the available measurements: the swept region and the reconstructed surface. The total number of feasible surfaces is impossible to count. To compute an approximated value, this work uses an assumption based on the accuracy required for the reconstructed surface. Section 2.2 defined the parameter B as a "characteristic length" of the environment: the size of the smallest feature to be found in the exploration. Here, B is interpreted as the minimum expected radius of curvature of the unknown surface to be explored. More precisely, it is assumed that the unknown surface can be reconstructed as the envelope of a "solid ball" of radius B, and the free space can be reconstructed as the envelope of a "hollow ball" of radius B (see Figure 40). This condition is slightly stronger than a minimum curvature radius, because no thin walls or thin orifice of size smaller than $2B$ are allowed. The term ball is used because the approach works equivalently in 2D and 3D: balls are circles in 2D and spheres in 3D.

This assumption provides an easy way to count the possible surfaces passing through a point. Since every point is inside either a solid or a hollow ball, one way to estimate the number of surfaces is to count the number of feasible (i.e. consistent with both S and R) hollow or solid balls including the point. The occupancy probability becomes:

$$P_C = \frac{\sum \text{all feasible balls of radius } B \text{ where c is empty}}{\sum \text{all feasible balls of radius } B} \tag{5.5}$$

For each cell, the number of feasible balls is computed by evaluating the area where the centers of these balls can lay:

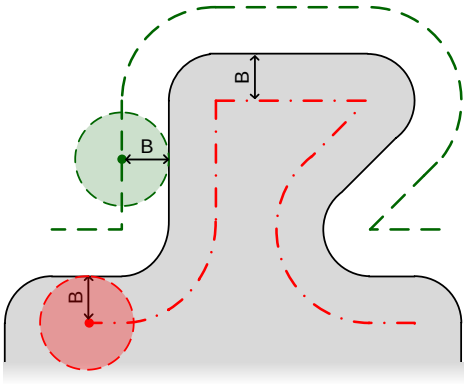

Figure 40. The environment is assumed to be described as the envelope of "solid" and "hollow" balls of radius B.

$$P_C = \frac{A_{feas,solid}}{A_{feas,solid} + A_{feas,hollow}} \tag{5.6}$$

This formula can be computed with only two operations. First, the swept region and the reconstructed surfaces are expanded using a binary dilation with a ball or radius B as a structuring element. The dilation of two sets A and B, also known as Minkowsky sum, is the results of adding every element of A to every element of B (Figure 41):

$$A + B = \{a + b \mid a \in A, b \in B\} \tag{5.7}$$

Second, $A_{feas,empty}$ and $A_{feas,full}$ are computed with a convolution of the expanded map and the same ball of radius B. Details underlying this process are not necessary for the understanding of the following parts, are are reported in Section 5.9 through two examples.

The result of these two operations is a distribution where cells swept by the robot and in front of the surface have P_C=1, cells behind the surface have P_C=0, and the probability of neighboring cells fades nicely with increasing distance from the swept region or the surface. When a cell is further than $2B$ from them, this cell is unaffected and P_C=0.5. Figure 42 represents the distribution obtained with a rectangular swept region and a reconstructed line. The right plot shows the fading effect for a fixed value of x. The fading function depends on the geometry of the swept region and the reconstructed surface. When the frontier of the swept region is a line, as in the lower part of Figure 42, the distribution decays with the following function:

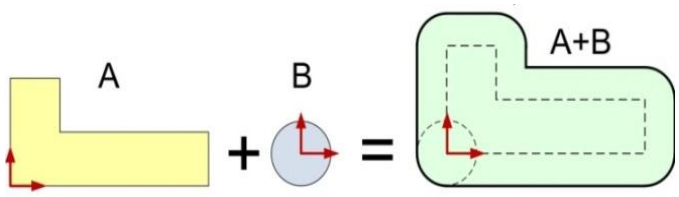

Figure 41. Dilation of A with element B, or Minkowsky sum of A and B.

$$P_C(x) = \cos^{-1}\left(1 - x / B\right) + \left(x - 1\right)\sqrt{2Bx - x^2} \tag{5.8}$$

5.3.2. Expected entropy

Once the occupancy probability on the whole map is determined, its entropy is readily obtained using Eq. (5.2). The value of a path is computed with the payoff function, Eq. (5.3), which needs the expected reduction of entropy and the expected movement cost due to a path. There are two challenges to compute this formula. First, the number of paths to any point is infinite (if there is at least one). In this section, the only paths considered are straight lines starting from the robot location. More complex paths will be considered in the following section.

The second challenge is the calculation of the expected entropy and cost. The robot needs to move into an unknown location to acquire some useful information, and it must expect to encounter an obstacle on its way. The procedure is assumed to stop if the robot either reaches the end of the path, or is blocked by the environment. Therefore, the expected entropy and cost values need to consider all the possible stops at any point in the path. This can be expressed recursively.

Figure 43 represents a discretized path from cell 0 to cell n, and the expected entropy decrease due to the path is computed. Let P_k denote the probability of cell k being empty, and $P_{\bar{k}}$ the probability of being occupied. Since the sum of the probability is one, $P_{\bar{k}} = 1 - P_k$. Starting from cell 1, the expected entropy decrease with the path from cell 0 to 1 is the sum of the

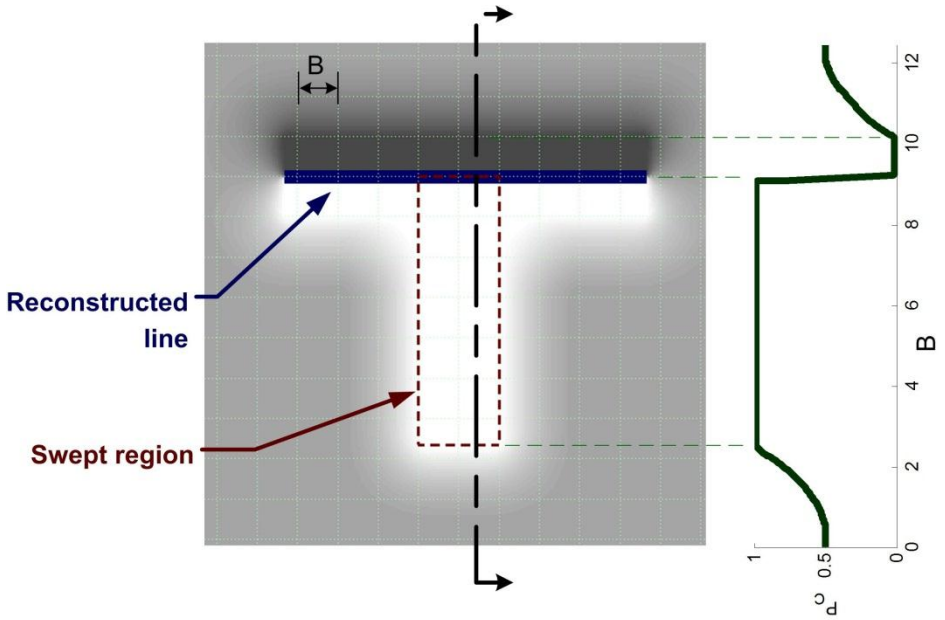

Figure 42. Probability distribution of the cells close to a rectangular swept region and a reconstructed surface line. The intensity represents the probability: P_C=0 is dark grey and P_C=1 is white. The right plot shows the probability distribution for the section indicated with the arrows. Units are normalized to the parameter B.

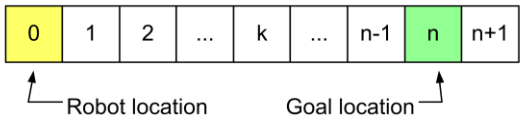

Figure 43, Computation of expected entropy when the robot is moving from its location, cell 0, to the end of a path, cell n.

entropy reduced if cell 1 is empty (ΔH_1^{sweep}) and the entropy reduced if cell 1 is occupied (ΔH_1^{touch}), weighted by the probability of being respectively empty or occupied:

$$\Delta H_1 = P_{\bar{1}}\,\Delta H_1^{touch} + P_1\,\Delta H_1^{sweep} \tag{5.9}$$

If cell 1 is empty, the robot will move to cell 2, and so on until it reaches cell *n*. The final expected entropy is sum of all the cell contributions, discounted by the probability of reaching the cell, and can be written recursively:

$$\begin{aligned} E[\Delta H_n] &= P_{\bar{1}}\,\Delta H_1^{touch} + P_1\left[\Delta H_1^{sweep} + P_{\bar{2}|1}\,\Delta H_2^{touch} + P_{2|1}\left[\Delta H_2^{sweep} + P_{\bar{3}|1,2}\,\Delta H_3^{touch} + P_{3|1,2}[...]\right]\right] \\ &= \Delta H_1 + P_1\left[\Delta H_{2|1} + P_{2|1}\left[\Delta H_{3|1,2} + P_{3|1,2}[...]\right]\right] \\ &= E[\Delta H_{n-1}] + \Delta H_{n|1,\ldots,n-1}\prod_{k=1}^{n-1} P_{k|1,\ldots,k-1} \end{aligned} \tag{5.10}$$

The symbol | indicates conditional probability. For instance, $P_{2|1}$ indicates the probability of cell 2 to be free, given that cell 1 is free. The conditional probability is different from the unconditional probability because of the assumption of correlation between cells (section 5.3.1). The product of the conditional probabilities $\prod_{k=1}^{n-1} P_{k|1,\ldots,k-1}$ represents the probability of reaching cell *n* from cell 0 without hitting any obstacle. This product reduces the effect of cells far from the robot: since those cells have a lower probability of being reached, their expected entropy reduction and cost are low. Increasing the density of the grid, the number of cells between two locations increases and so does the number of elements in $\prod P_{k|1,\ldots,k-1}$. However, the conditional probability $P_{k|1,\ldots,k-1}$ is also dependent on the grid size: it is small for a large cell and tends to one with an infinitely small grid. One could even define the correlation between cells so that the probability of reaching cells is independent on the grid size. However, several assumptions would be needed to apply such distribution to multi-dimensional cases, and this work prefers the use of the formulation introduced in section 5.3.1.

To compute the expected cost, a similar recursion is formulated. Denoting the distance between two cells by Δx, and the cost of touching a point by φ (expressed in a unit comparable to Δx), the total cost is:

$$\begin{aligned} E[C_n] &= \Delta x + P_{\bar{1}}\,\varphi + P_1\left[\Delta x + P_{\bar{2}|1}\varphi + P_{2|1}\left[\Delta x + P_{\bar{3}|1,2}\,\varphi + P_{3|1,2}[...]\right]\right] \\ &= E[C_{n-1}] + \prod_{k=1}^{n-1} P_{k|1,\ldots,k-1}\left[P_{\bar{n}|1,\ldots,n-1}\,\varphi + P_{n|1,\ldots,n-1}\Delta x\right] \end{aligned} \tag{5.11}$$

Once both the expected entropy reduction and the path cost are known, the payoff function can be computed. The path with highest payoff will be executed.

Some simplifications are needed to implement Eq. (5.9)-(5.11). These are treated in details in Section 5.9.

5.3.3. Representative case

As a representative case, a point robot was simulated in a simple environment made of two lines and a circle, see Figure 44. The robot starts in the middle of this environment, without any a-priori information. The robot position is assumed to be corrupted by random measurement noise. This noise creates positioning error. Since the point is assumed to be a manipulator (and not a mobile robot), the positioning error is independent at any time (and not cumulative).

The entropy reduction algorithm is not used at every step, but only once the surface that is currently being explored has been recognized as a primitive. Before that time, the Uniform Surface Density search algorithm described in section 3.4.2.1 is used. This is because one point does not provide enough information to identify the underlying surface. Intuitively, only when a surface has been discovered it is worth spending time to decide where else to go; otherwise new points nearby provide a very high amount of information. However, nothing prevents the use of the entropy approach for the entire tactile exploration.

Figure 44 depicts the following instants of the exploration:

1. The robot starts in the middle of the environment to be explored. The robot movement is also limited by a workspace, represented by the dashed square.
2. The robot chooses a random direction, and probes a few points using the USD algorithm. The shading represents the entropy of the environment. The entropy is nearly zero where the robot has passed.
3. Once a line has been identified, a new exploration direction is chosen using the entropy reduction approach. From the robot location, all the paths along straight lines are evaluated. The intensity of a region represents the payoff of the straight path to that area. The best path is the straight line toward the double circle.
4. Once the path has been chosen, the robot proceeds along it until it either touches the environment or reaches the final point.
5. The second line has been detected using USD and the robot chooses the direction that is expected to minimize the entropy.
6. The process continues until the whole environment is explored. In this last frame, the third primitive (the circle) has been identified.

5.4. Tactile Exploration with a Planar Manipulator

This section extends the entropy approach to a manipulator with nontrivial kinematics and finite thickness. The use of a manipulator instead of a point introduces several challenges. The

manipulator's movements are constrained by the environment in a way that depends on the manipulator's shape and kinematics. Furthermore, a manipulator can reach the same tip location with different configurations. These configurations are not equivalent, because the environment shape might collide with the links in some configurations but not in others. The concept of configuration space introduced in section 5.1.2 becomes a necessary tool.

A simple representative case is considered (Figure 45). The robot is a two-degree of freedom manipulator with its base fixed on the environment. Both joints have limits: (-180º, 0º) for the first, and ±110º for the second. These limits constrain the workspace to the dashed black line in Figure 45. On the right, the configuration space without any obstacles is represented: the axes are the two joint angles, limited by the joint limits; the robot itself is a dimensionless point, representing the joint configuration. The environment to map for this representative case is composed of two lines and a circle. Figure 46 represents this environment and its map into configuration space.

5.4.1. Planning in workspace or C-space

There are strengths to planning the path in C-space and in workspace. The tradeoffs of the different methods are presented in this section.

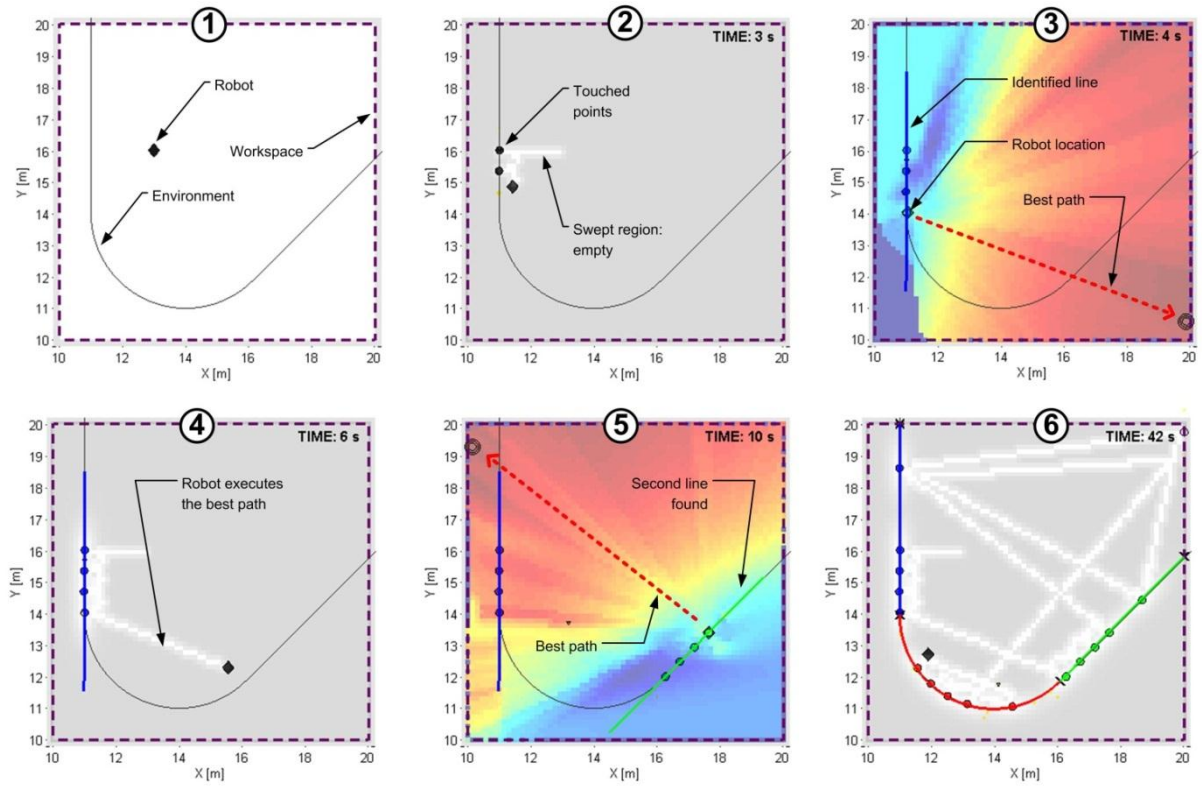

Figure 44. Entropy reduction strategy for a point robot in a 2D environment composed of two lines and a circle.

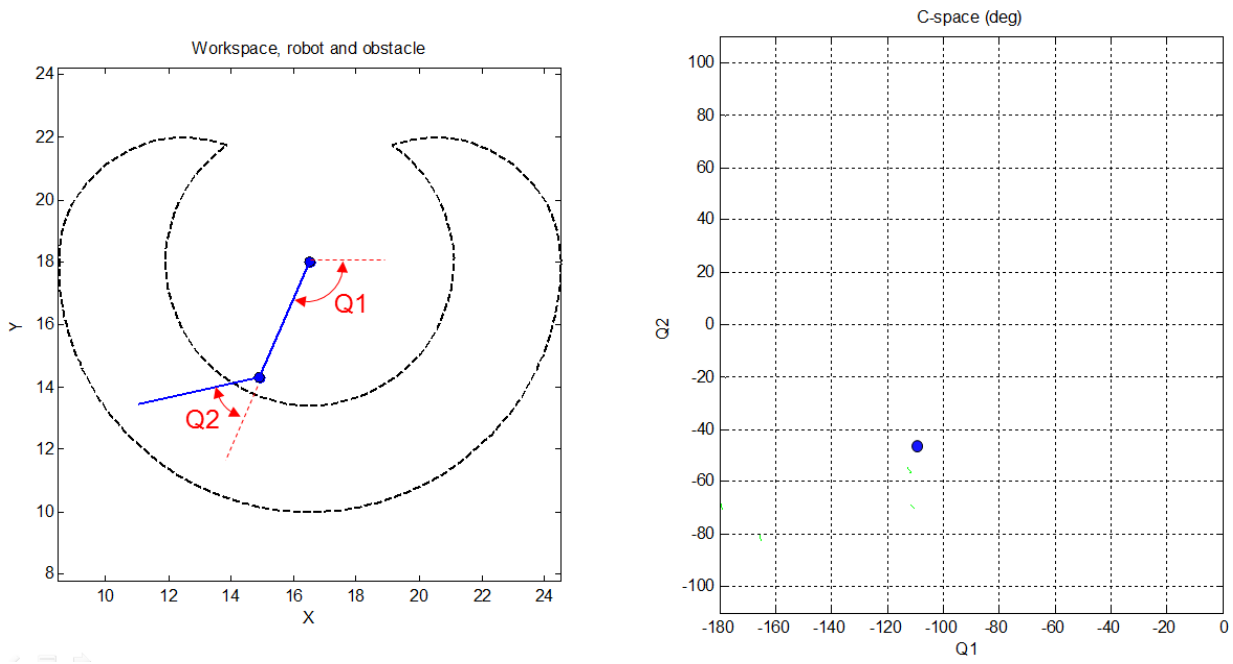

Figure 45. (Left) Planar manipulator in an empty environment. The dashed line is the space reachable by the robot tip, the workspace. (Right) Configuration space representation of the same robot.

5.4.1.1. Strengths of C-space

The C-space is a very powerful tool for path planning. By mapping the robot into a point, problems such as the change in robot configuration are simplified. As an example, the dark (blue) robot configuration shown in Figure 46 cannot easily touch the vertical line on the left in its original configuration. The robot needs to move to the right, change configuration, and come back. This problem is much easier in C-space, where the point-robot only needs to find a path from one configuration to another. This simplification lead to the use of configuration space in the vast majority of works in motion-planning [75, 98, 99].

A second advantage of C-space is particularly useful in the application considered here, tactile exploration with only position sensors. When the robot detects contact, the contact location is not certain in workspace, because it can be occur anywhere over the manipulator's link. On the other hand, the contact location in C-space is known and often extremely informative for a serial manipulator. Let n denote the degrees of freedom of the manipulator. If contact occurs on link n, the C-space constraint is a single point: the configuration in contact. If contact occurs at link $k \leq n$, the constraint is a subspace of dimension n-k: a fixed value of the joints 1...k and each value of the joints k+1…n. For example, in two dimensions, if the first link touches the environment, the C-space constraint is a vertical line, because the first link is in contact for every value of the second joint.

5.4.1.2. Strengths of workspace

The use of the workspace in the entropy approach also has several advantages:

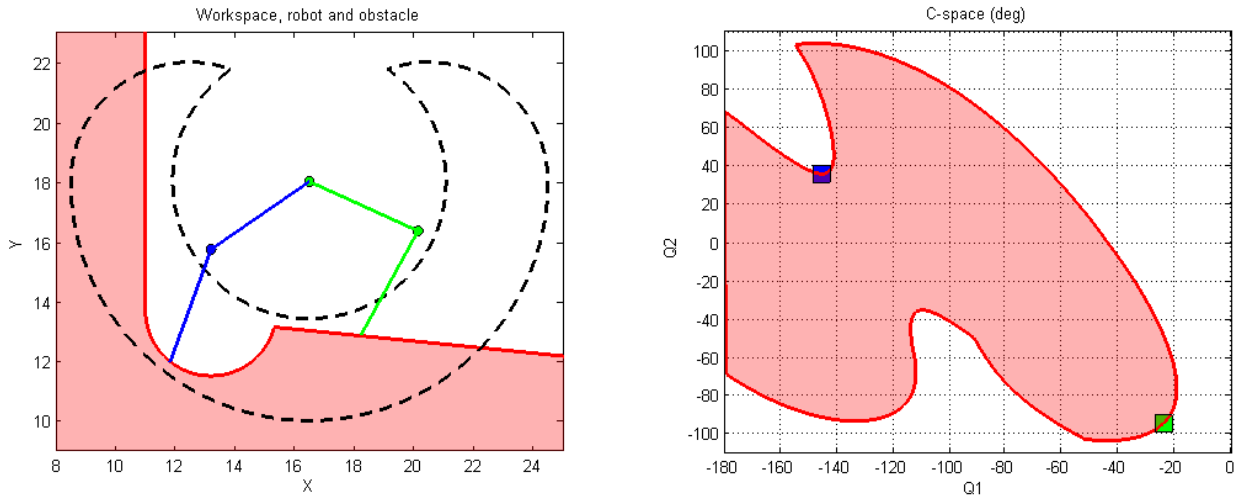

Figure 46. Surface to be mapped (left), and C-space map of such surface (right). The robot in two configurations is also shown, both in workspace and in C-space.

1. The dimension of the C-space is the number of the robot's joints. The discretization in a grid would not be feasible for robots with more than three joints. As an example, the standard industrial 6-dof robot has a six-dimensional C-space, and a rough grid with 50 cells per dimension would require more than 15 billion cells. On the other hand, the workspace has at most three dimensions (except maybe for theoretical physicists), and it can be discretized. The discretization does not need to be fine, because the precision on the surface is given by the touched points, which are independent of the grid.
2. The regularity of the surface in the workspace is not maintained when mapped into C-space. Obstacles in C-space are highly distorted and irregular; hence they cannot be represented by simple primitives. The fitting and segmentation process, which relies on the regularity of the surface to reduce the number of required touch points, need to be computed in workspace. The result of this process can then be mapped in C-space. Note that an explicit map can be only computed if the C-space can be discretized, and in any case it is complex and computationally expensive [106].
3. The objective of the exploration is to determine the physical surface, not its C-space representation. Hence, the amount of information should be evaluated in workspace to ensure optimal exploration. Since a robot can reach the same physical location with several configurations, the amount of explored C-space does not represent the knowledge of the workspace well. For instance, section 5.5 shows a situation where the entire surface in workspace is mapped without reaching half of the C-space. If entropy is computed in C-space, the algorithm could choose paths that would not give any information on the real surface.
4. Since the robot is a dimensionless point and its swept region is a curve, information from the space swept by the robot is hidden in C-space. Figure 47 represents a situation where the robot slides its tip on the vertical line of the environment. The large area swept in the

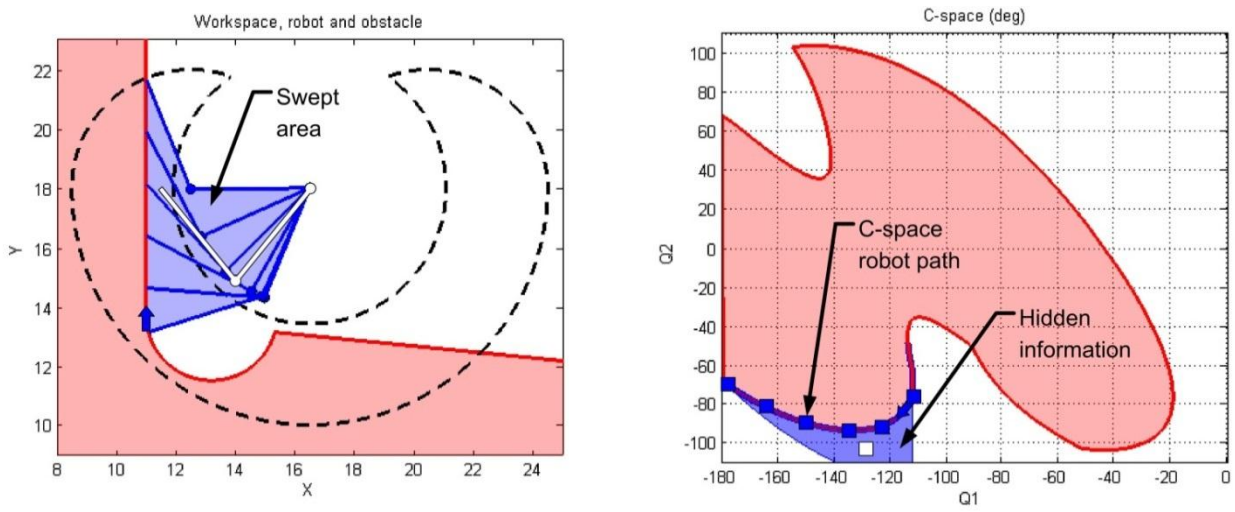

Figure 47. Information from the robot sweep is hidden in C-space.

workspace must be empty. On the right, the C-space swept region is only the curve in contact with the C-obstacle. The darker region beneath the curve is obtained by numerically mapping into C-space all the configurations fitting inside the swept region of the workspace. The white configuration is one of these configurations. This area in C-space is necessarily free, but this information is "hidden", and can be recovered only using the workspace.

5.4.2. A hybrid approach

Both the workspace and the C-space representations have strengths that could be exploited for the application studied in this work. Here, a hybrid approach combining both representations is proposed. C-space is used for path planning, and workspace is used to reconstruct the surface and evaluate the entropy. In particular:

- Information is stored in workspace. The surface is reconstructed through the touched points and swept region. The space is discretized with a grid to store the swept region and to evaluate occupancy probability and entropy.
- The possible robot paths are computed in C-space. Every time the algorithm looks for a new exploration direction, the algorithm generates a series of paths in configuration space from the robot location to any other reachable location. Non-reachable locations are due to the touched points in C-space (more descriptive than their counterpart in workspace) and the C-space map of the reconstructed surface. To find paths in C-space, this work uses a Rapid Exploring Random Tree (RRT), a sampling-based approach that avoids the explicit representation of C-space [131].
- The payoff of any single path is evaluated in the workspace. The RRT produces a series of consecutive segments. The payoff of such segments can be evaluated in workspace by computing the swept region, the expected touched surface, and the expected entropy

reduced by the segment. The payoff of the complete path is the sum of the entropy reduction for all its segments, divided by the total expected cost.

5.4.3. RRT

Sampling-based strategies have an advantage over the other motion planning algorithms listed in section 5.1.3. Sampling-based strategies avoid the direct representation of C-space. This is important, because the map is updated after every touched point. Furthermore, this makes the approach applicable to manipulators with any number of degrees of freedom. Finally, this does not impose any constraint on the type of surface representation adopted in the workspace.

There are two main classes of sample-based algorithms: single-query methods (such as RRT or randomized potential fields) and multi-query methods (such as probabilistic roadmaps). In this work, although the exploration strategy must be executed several times, the known map is continuously updated. Therefore, a single-query algorithm is chosen. In particular, the RRT algorithm [75, 111, 131] is simple to implement and ensures probabilistic completeness. Its original formulation solves the standard path planning problem: connect a starting and a goal position. Nevertheless, the approach can easily be adapted to exploration by removing the goal location so that all accessible points are reached. The approach can be explained in three steps:

1. A random point is generated in C-space. If this is a feasible configuration, the point is saved; otherwise it is discarded, and a new point is generated, until a feasible configuration is found.
2. The distance between this point and the tree is evaluated, and the closest point or segment on the tree is identified.
3. The feasibility of a straight path between the new point and the closest tree location is computed. If no collisions occur, the new segment and the final point are added to the tree. Otherwise the algorithm either discards the point, or creates a branch of the tree until it hits the obstacle.

When more and more points are stochastically added, the tree will eventually cover the whole environment, and each location can be reached by a path along the tree. Figure 48 shows a representative example of the tree applied to C-space. The RRT does not find optimal paths, but it is probabilistically complete, which means that if a path between two points exists, the algorithm will eventually find it. Recent studies proposed

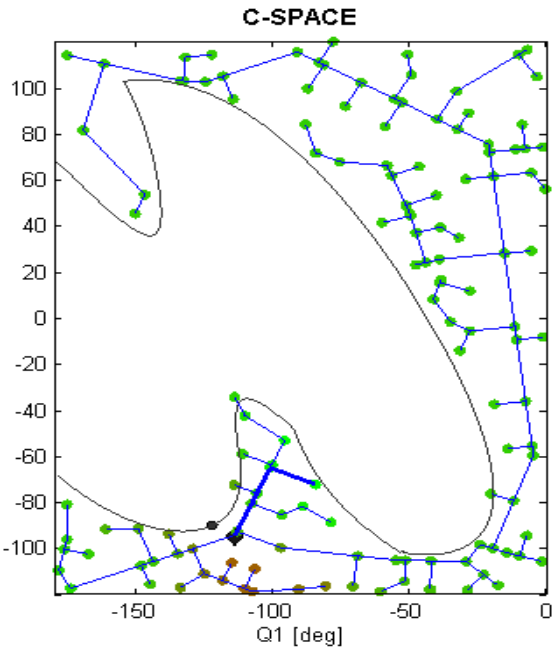

Figure 48. RRT algorithm with entropic exploration, in the C-space of the representative example of . The thick line is the best path.

modifications of the RRT that makes this algorithm asymptotically optimal, by rewinding the tree every time a new branch is added [112]. This modification can be directly applied to the problem in this work to make the exploration algorithm optimal with horizon one. However, this research only implements the standard RRT algorithm, because it simplifies the implementation and does not affect the contribution of the work.

5.4.4. RRT with entropy reduction

The RRT creates a tree of segments from the robot location to any reachable configuration in the C-space. Among these paths, the one with highest payoff should be chosen, as in Eq. (5.3). The approach is simple because Eq. (5.9)-(5.11) have a recursive form. When the RRT adds a new node n in C-space, this node is connected to a previous node $n\text{-}1$ through a straight segment in C-space. The entropy and cost of the path up to n are computed starting from the values at node n-1 and adding the terms relative to the new segment, as shown in Eq. (5.9) and (5.11). This requires the following terms to be available at the previous node:

1. The swept region up to node $n\text{-}1$, S_{n-1}
2. The expected entropy at node $n\text{-}1$, H_{n-1}
3. The expected cost at node $n\text{-}1$, C_{n-1}
4. The product of the conditional probabilities up to node $n\text{-}1$, $\prod_{k=1}^{n-1} P_{k|1,\ldots,k-1}$

Using these terms at the beginning of a segment, Eq. (5.9) and (5.11) can be readily applied to find the same values at the end of the segment. Hence, the entropy and cost of the whole path can be built recursively.

The most expensive term to keep track of is the swept region, since it requires the storage of a full binary map of the environment, or a part of it, for each node. A trade-off between computation and required memory can be obtained here, by storing the partial map only for specific nodes, and every time re-computing the swept region from these nodes.

The synergy between workspace and C-space is evident. The four terms to record at each node are associated with the workspace (except for the movement cost, which can also be defined in C-space). Therefore, even the nodes and segments are added in C-space, the workspace is always used to compute the change in entropy, cost and probability. This requires some attention. The recursive eq. (5.9) and (5.11) are computed discretizing the segment in small parts. Nevertheless, the shape and geometry of the manipulator implies that several cells can swept simultaneously, and some algebra is required to take this into account.

5.4.5. Simulation

The algorithm has been tested using the representative case described at the beginning of this section. The robot has now a finite thickness, which affects the collision with the environment and the swept surface. As in section 5.3.3, the USD algorithm is used to probe an unknown primitive, and the entropy strategy is called when the primitive is recognized.

Figure 49 shows four moments during the exploration process when the RRT is used. The plots use the same conventions as in Figure 44 in terms of touched points, primitives and occupancy map. Parts 1 and 3 show both the workspace and the C-space. Parts 2 and 4 only show the workspace. Each node in the RRT is evaluated, and the goodness of the paths is shown with color intensity. Light (green) segments represent good paths with high entropy/cost ratio, and dark (red) segments represent bad paths, with low entropy/cost ratio. The thick line in C-space represents the best path, and the dashed line in the workspace is the travel of the robot tip for such path.

The three primitives in the environment are discovered after only three runs of the RRT. With this approach, the robot is also able to change configuration to reach the left vertical line. More simulations and a laboratory experiment will be presented in Chapter 6, when strategies to deal with contact all over the manipulator's body will be introduced.

5.5. Tactile Exploration with a Spatial Manipulator

Previous sections have presented the entropic approach for a point robot and a planar manipulator. This section extends the entropic approach to the tactile exploration of a three-dimensional environment with a spatial manipulator, making it applicable to most real-world

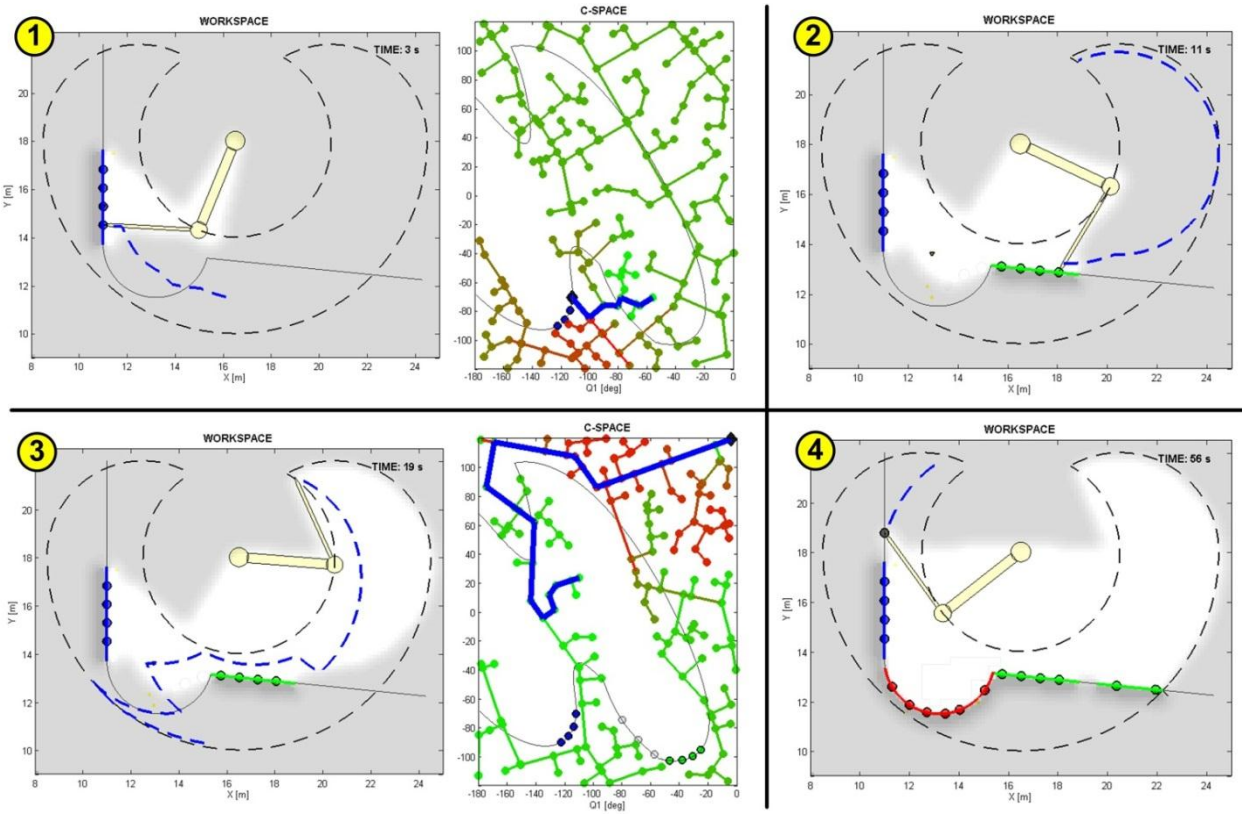

Figure 49. Four steps of the entropy approach with an RRT, showing the progress in the exploration.

problems.

5.5.1. Extension to spatial case

Few extensions are necessary to apply the concepts introduced in sections 5.3 and 5.4 to the spatial case. The method proposed to define the occupancy probability is based on the number of "hollow balls" and "solid balls" that are consistent with the reconstructed surface and the swept region (section 5.3.1). Balls are circles in 2D, and spheres in 3D. The approach is based on an assumption of minimum expected curvature radius. For a surface in space, there are infinite curvature radii, because there are infinite tangent lines to a single point. However, there are two particular directions, called *principal directions*, where the curvature radii have minimum and maximum values. The assumption is extended by assuming that both *principal curvature radii* are expected to be larger than the parameter B.

The computation of the occupancy probability requires a dilation and a convolution, both with a ball of radius B. The convolution is executed in the same way, independently of the dimension of its space. On the other hand, the term dilation was specifically developed in image processing for two-dimensional images. However, the Minkowsky sum (equivalent to a dilation) is defined for any dimension. The easiest way to implement this operation in 3D is to convolute the original map with a solid sphere of radius B, and then flatten the result into a binary map using the function:

$$f(x) = \begin{cases} 0 & if\ x = 0 \\ 1 & if\ x > 0 \end{cases} \tag{5.12}$$

5.5.2. Representative case

The extension is demonstrated with a representative case using a spatial manipulator with the shape of the oil-well exploration prototype (section 3.6.3). The first joint is assumed to have no limits, and can rotate indefinitely. The second and third joints are limited: limits are ±50º and ±150º respectively. Links and joints have finite thickness, affecting the area swept by the robot's movements. Independent random noise affects both joint angles (simulating measurement noise) and the final position of the tip (simulating kinematic and surface imprecision).

The environment to be mapped is shown in Figure 50. The environment is composed of two cylinders, a plane and a sphere. The absence of a limit on the first joint substantially changes the C-space, because an angle of 360º is now equivalent to 0º. Topologically, this represents a homeomorphic space to $\mathbb{R}^2 \times \boldsymbol{S}$, which can be imagined as a deformed parallelepiped where two opposite faces are "glued together", to form a ring. However, Figure 50 represents the C-space as a parallelepiped where the robot can jump from the face Q_1=0º to the face Q_1=360º.

5.5.3. Simulation

The environment is explored using the same approach as before; the robot probes a surface with the USD until a primitive is recognized. Once the primitive is known, the entropy strategy is executed. Figure 51 shows four moments in the exploration of the environment, including the C-space and the RRT. Thus, the manipulator never needs to reach the top part of the C-space to completely explore the whole surface. The four instants in Figure 51 represent the following:

1. The first primitive is detected, and the RRT is computed to guide the robot through the path with most information. Both workspace and C-space are shown. The intensity of the color in the RRT represents the goodness of the paths.
2. The manipulator is probing a corner, and more points are needed to interpret it. While the primitive is not recognized, a triangular mesh is used.
3. All the primitives have been found.
4. The surface is almost completely explored. Several grey points in the lower cylinder represent situation where contact has been detected to occur on the robot's link. This will be treated in the following chapter.

This example shows that the approach developed in this work can be applied to a generic three dimensional environment. The example also shows the importance of evaluating information in workspace: the environment is completely mapped without ever moving into the top half of the workspace (Q_3>0). If entropy were evaluated in C-space, several useless movements would be performed.

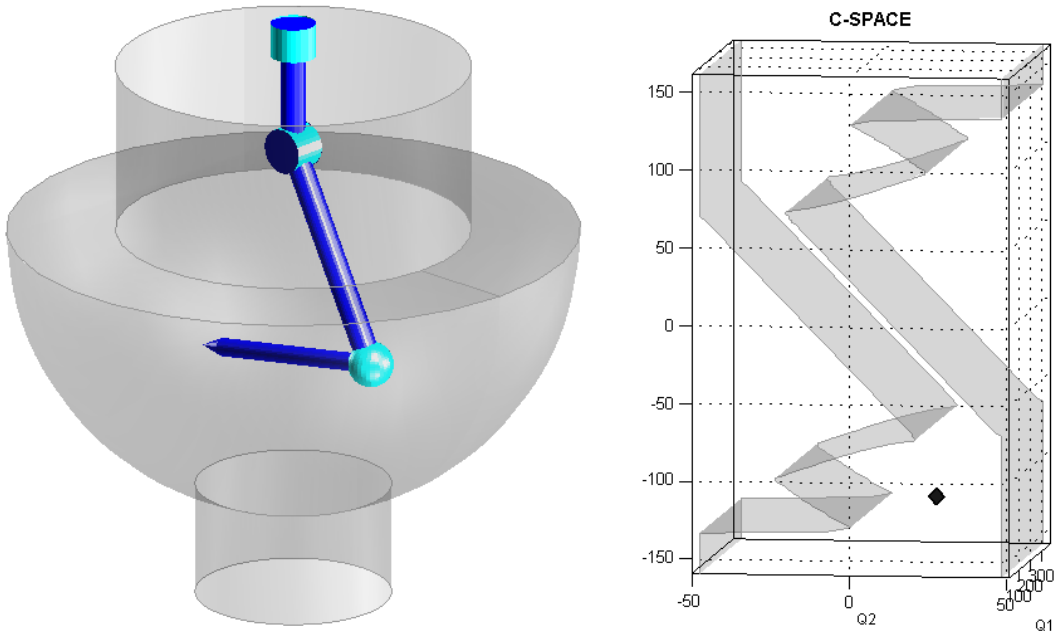

Figure 50. Workspace and C-space of the representative case used in the 3D entropic exploration.

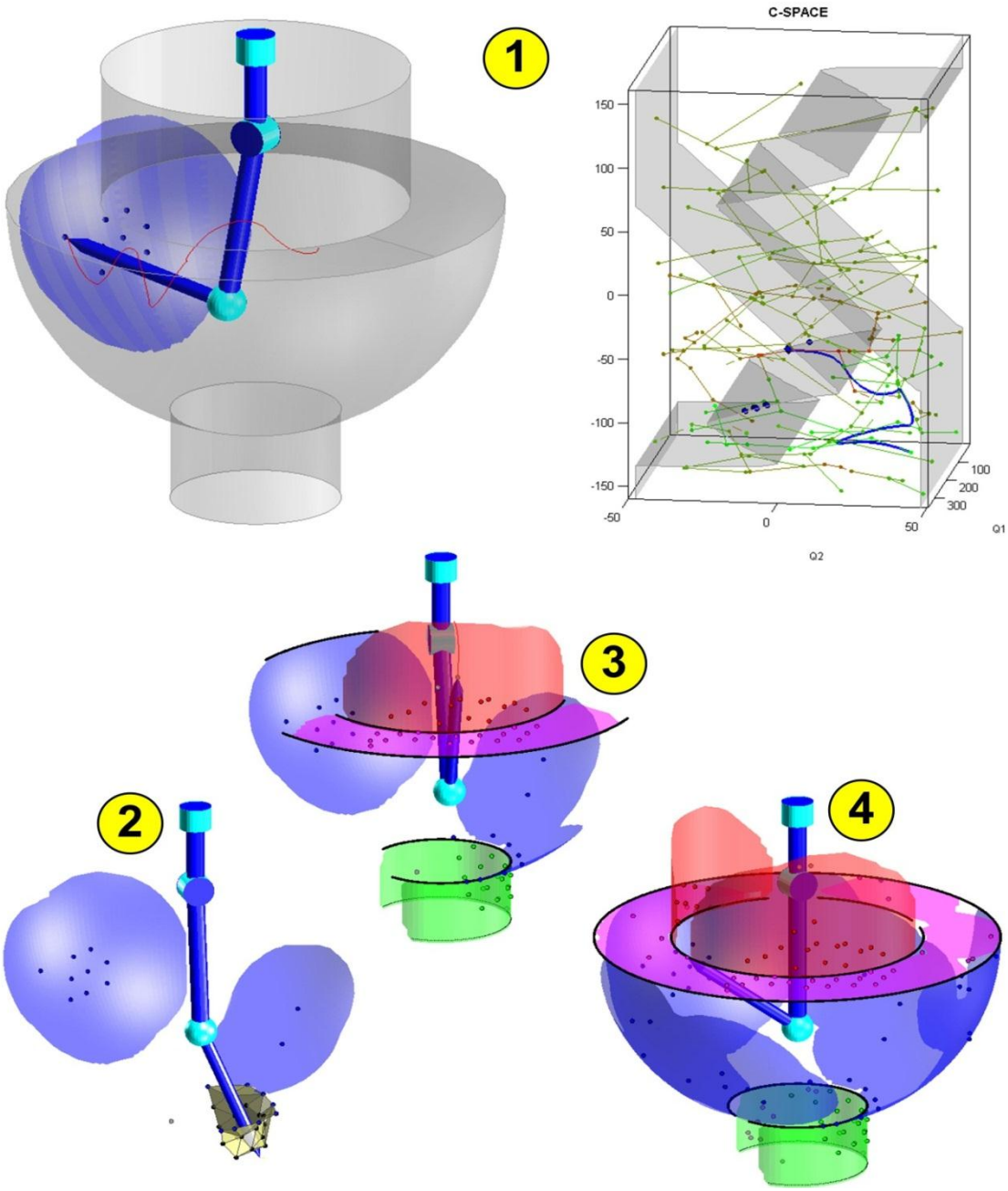

Figure 51. Tactile exploration of a 3D environment with the entropic approach.

5.6. Comparative Study

In this paragraph, the performance of the entropy approach is evaluated by comparing it to other exploration strategies. Figure 53 shows the planar manipulator and the environment to be mapped. The robot's speed is limited such that the Euclidean norm of the joint speeds cannot exceed 40 deg/sec. The exploration strategy is used in a similar way as before: when the robot

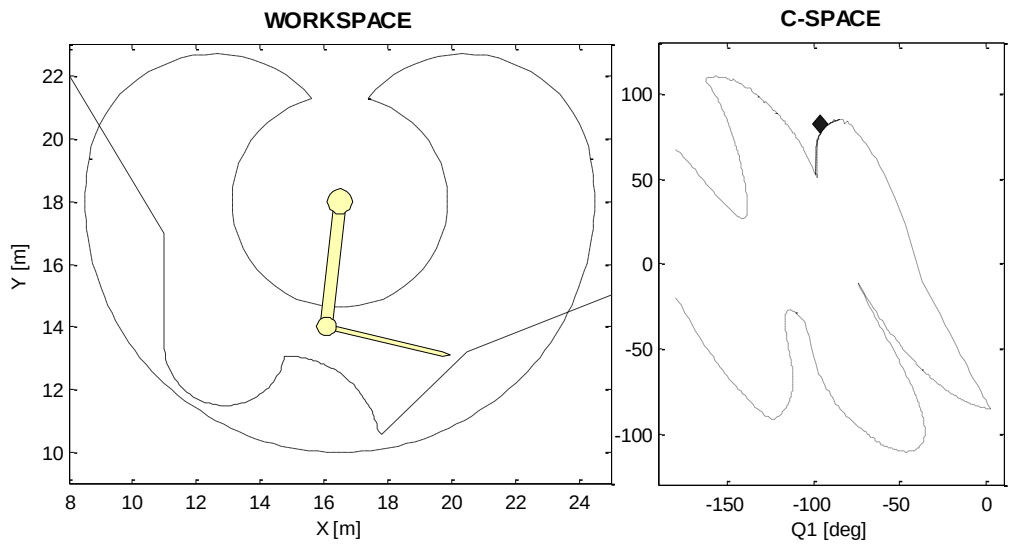

Figure 53. Environment and robot used for the comparison of performances of exploration strategies.

touches an unknown surface, the USD strategy is used to probe close points. Once a primitive is identified, the next location to explore is determined with three different strategies:

1. The *Best Cone* search (section 3.4.2)
2. A straight path along a randomly-chosen direction.
3. The entropic RRT approach.

The performance of the strategies is evaluated by comparing the time required to reconstruct the environment. The decrease in time of the amount of unknown surface is used as a measure of exploration speed. At all time, the percentage of the surface that still needs to be reconstructed is recorded. To make the comparison independent of the initial conditions, one hundred trials are run with independently random initial positions and initial exploration directions, and the average value among these trials is used for comparison. Figure 57 (left) shows the percentage of surface to be explored for each time instant from 0 to 140 seconds.

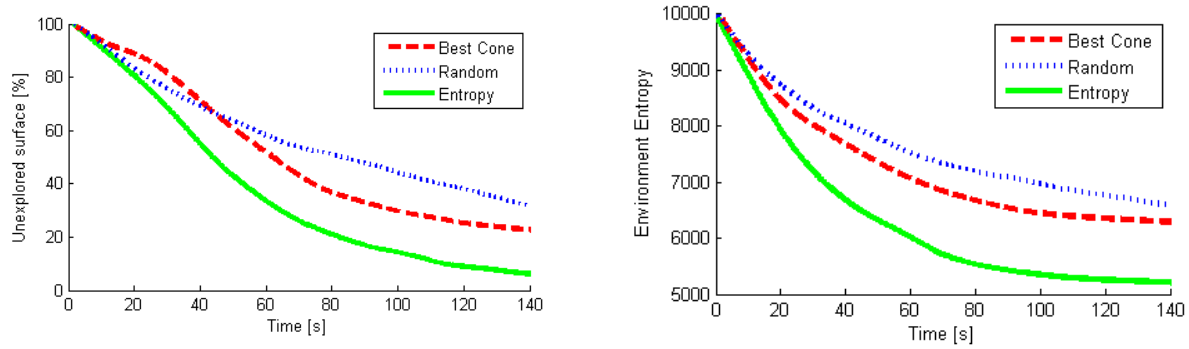

Figure 52. Comparison of the exploration speed with three strategies: Best Cone, Random, and Entropy approach. The left figure shows the decrease in time of the percentage of surface still to explore. The right figure shows the decrease in environment entropy. The curves represent the mean on 100 trials for each strategy.

Figure 57 (right) shows the reduction of environmental entropy, for the same trials and same time interval. These figures show the effectiveness of the entropy approach. At time t = 100 s, for example, the remaining surface to explore is approximately 15% with the entropy approach, 30% with the best cone strategy, and 45% with the random strategy.

5.7. Experiments

5.7.1. Feasibility

The feasibility of the approach is evaluated using a planar-manipulator in a planar environment (Figure 54). The robot links have lengths 156 mm and 215 mm. The two joints have the following limits: ±110º the first joint, and ±118º the second. Using a backlash compensation technique (section 4.3), the robot has approximately a 1mm accuracy. At all times, the end effector speed is limited to 25 mm/s, and both joint speeds are limited to 0.2 rad/s.

The environment to explore is composed of three lines and two circles. Different configurations are required to reach the bottom left line and the internal part of the smaller circle. The robot starts in the upper part of the surface. The robot control is similar to the one used in previous cases: USD until a primitive is recognized, and RRT with entropy reduction once a primitive is determined.

Figure 55 shows three frames of an exploration trial. The first frame shows the beginning of the exploration. Contact between the manipulator's link and the surface occurs, and the method proposed in Chapter 6 is used. The second frame shows an interesting moment during the exploration, when the RRT decides to completely change the configuration of the robot, to touch

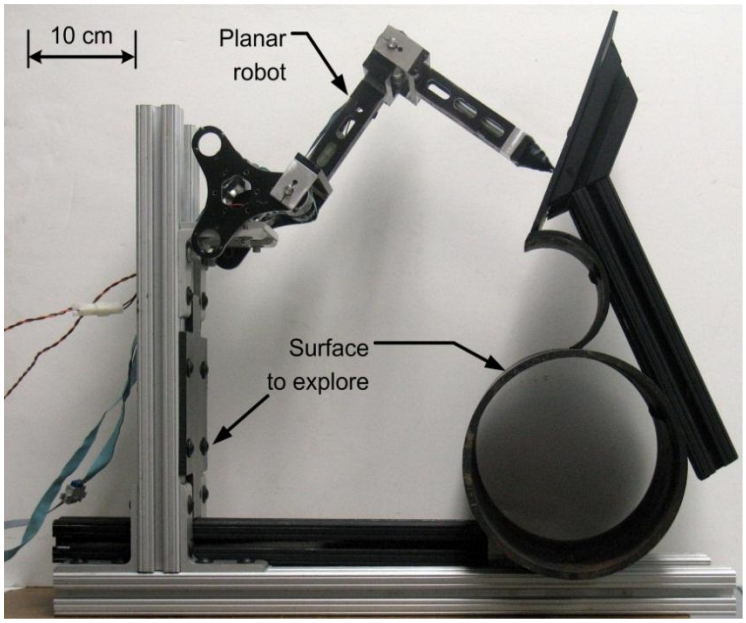

Figure 54. Planar manipulator and environment for tactile exploration.

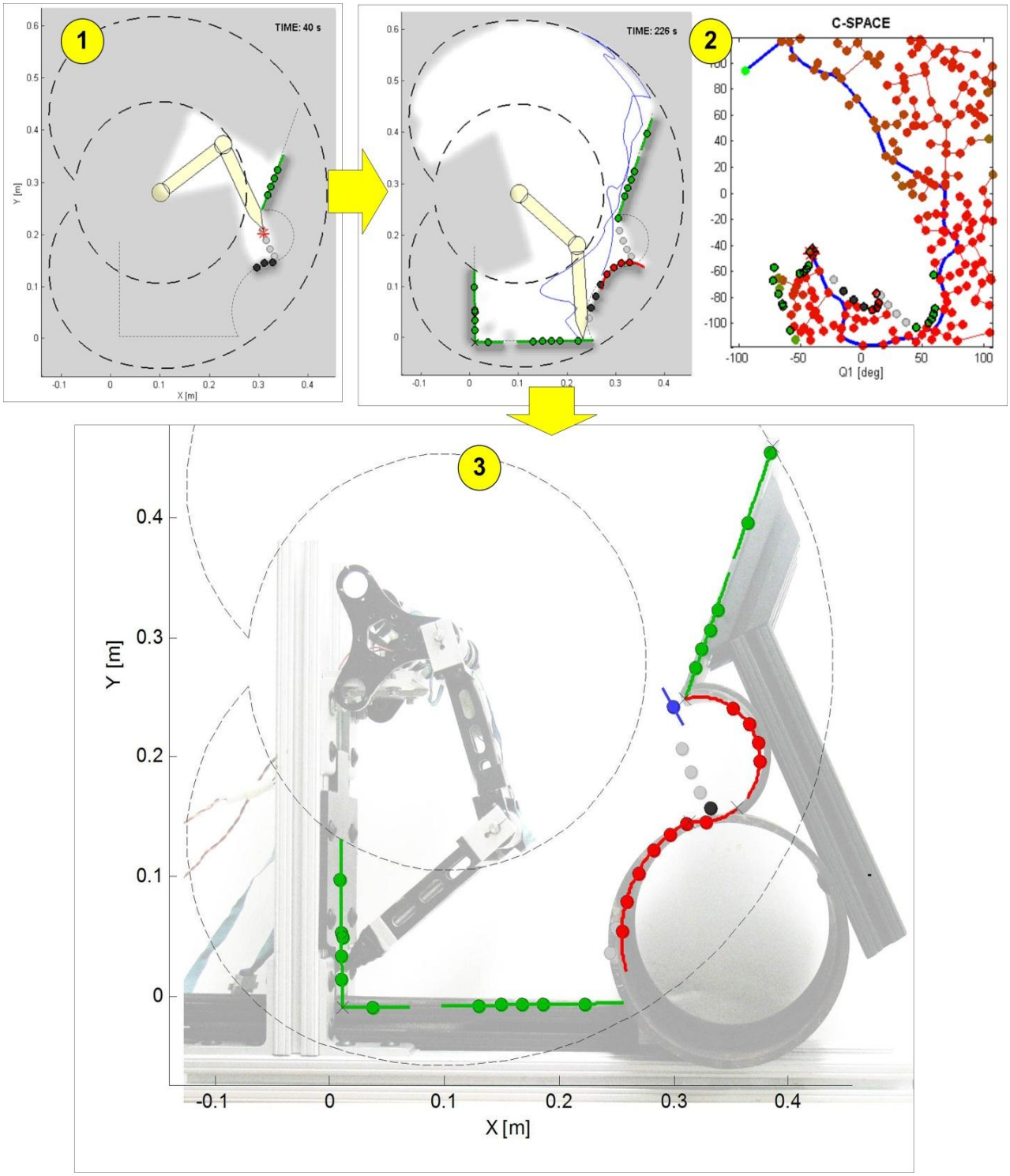

Figure 55. Experimental results of the exploration of a planar environment using the entropic approach.

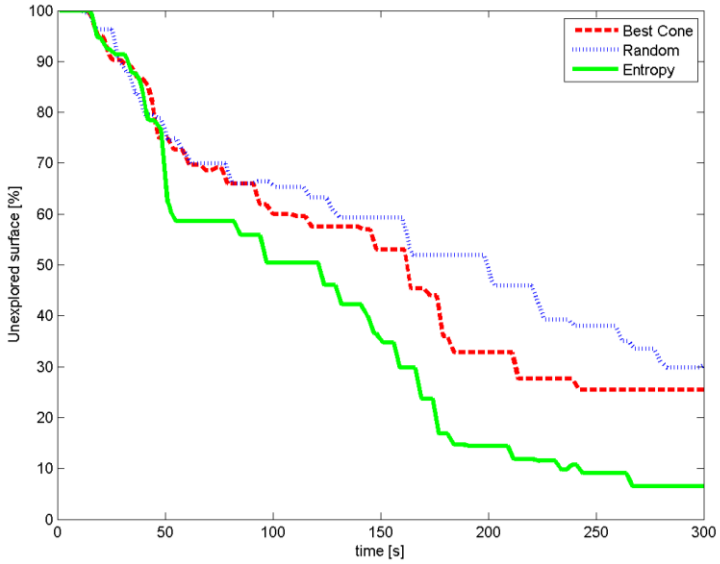

Figure 56. Experimental comparison of the exploration speed with three strategies: Best Cone, Random, and Entropy approach. The graph shows the decrease in time of the unexplored portion of the surface. For each strategy, five trials are run, and the average value is shown.

the side of the larger circle. The third frame shows a superposition of the real robot and the reconstructed surface. The four primitives are correctly detected. The surface is accurately reconstructed (as an example, the diameter of the larger circle is determined with an accuracy of 97%). The discrepancy between the superimposed images is mainly due to perspective in the picture.

5.7.2. Comparative Study

The performance of the entropy approach is evaluated by experimentally comparing it to other exploration strategies. The same approach as in section 5.6 is used: the entropy search is compared to the Best Cone strategy and to a random search. The environment to explore is the same as in the previous experiment (Figure 54). For each strategy, five runs have been performed, and the percentage of unexplored surface is monitored. Figure 56 shows the average behavior for the five runs, for all the three strategies. The figure exhibits approximately the same behavior as the one computed in simulations, showing the superiority of the entropy approach compared to the other strategies.

5.8. Conclusions

This chapter develops a new approach to tactilely explore a constrained environment with a manipulator. The approach uses the concept of entropy to quantify the knowledge of the environment. The environment is discretized with regular cells, and a specific method is proposed to evaluate the occupancy probability of each cell, considering the correlation among

neighbor cells. An RRT is created in configuration space, reaching all the points that are not constrained by the partially constructed map. For every path in the RRT, the expected reduction of entropy and expected movement costs are computed. The path with highest entropy reduction compared to the cost is executed. While the approach creates a discretized occupancy map, the real map used as product of the exploration is still a boundary representation composed of geometric primitives.

The algorithm is easy to implement and computationally feasible in real time. The discretization of the map does not need to be dense because it is not use as the final map. The occupancy probability is computed using two consecutive convolutions. The expected entropy and cost can be computed recursively, so that the values for a segment in the RRT can be evaluated starting of the previous segment.

The approach is evaluated in several simulations, using a point robot, a planar manipulator, and a spatial manipulator. The simulations show the effectiveness of the approach to map very complex environments in short time.

5.9. Computations

This section is optional and it is not required for the understanding of the work. It reports the steps required to compute some elements that have been introduced in this chapter.

5.9.1. Occupancy map

This paragraph describes why the count of "hollow balls" and "solid balls" can be computed with two simple operations: a dilation and a convolution. The process is explained with two examples: the first involving only the swept region, and the second involving both a swept region and the reconstructed surface.

5.9.1.1. Probability of a point given the swept area

The first example explains how to compute the occupancy probability of a cell C considering the area swept by the robot. The example is described through the five drawings in Figure 57:

1. The situation: occupancy probability of point C is evaluated, when a large area has been swept.
2. The "feasible" balls of radius B including the point C are computed. The only constraint in this case is the swept area, which needs to be necessarily empty. Any hollow ball is feasible, but only the solid balls which do not intersect the swept area are feasible. The figure represents a feasible and an unfeasible solid ball.
3. To count the number of possible balls, the area occupied by their centers is evaluated. Hollow balls have no constraints in this case. Thus, the centers of hollow balls are all the points within a distance B from C: a circle or radius B, with area πB^2. Solid balls are limited

by the swept area: the centers of the feasible ones are the points further than B to the swept area (horizontally-dashed in the figure).

4. Since centers of feasible solid balls need to be further than a distance B from the swept region, there is an easy way to compute the area: expanding the swept region by B. More precisely, the area is *dilated* using a circle of radius B as structuring element. Once the swept region is dilated, the area of feasible centers is the part of the circle centered in C outside the dilated region. This area can be computed with a single convolution between the dilated swept area and a circle of radius B and value 1 inside and 0 outside.
5. This figure plots the distribution of the probability P_C as a function of distance from the swept region. It behaves nicely as expected: P_C is 100% inside the region, and it slowly decreases when moving away from it, up to 50% after a distance $2B$.

5.9.1.2. Probability of a point given the reconstructed surface

The same approach can be used to evaluate the effect of the reconstructed surface, which imposes feasibility on both "hollow balls" and "solid balls". Figure 58 represents a region with a small swept area and a reconstructed line.

1. The situation: occupancy probability of a point is calculated given a swept area and a reconstructed wall.
2. Since the surface is assumed to be made envelopes of balls with radius B, an area of thickness $2B$ is necessarily empty in front of this wall; this area is treated in the same way as the swept area. Similarly, an area of thickness $2B$ is occupied behind the wall; this area imposes constraints on the "hollow balls". The two areas are independent because constraints on hollow and solid balls are to be treated separately.
3. As in the previous example, both areas are *dilated* with a circle of size B, as shown in the figure.

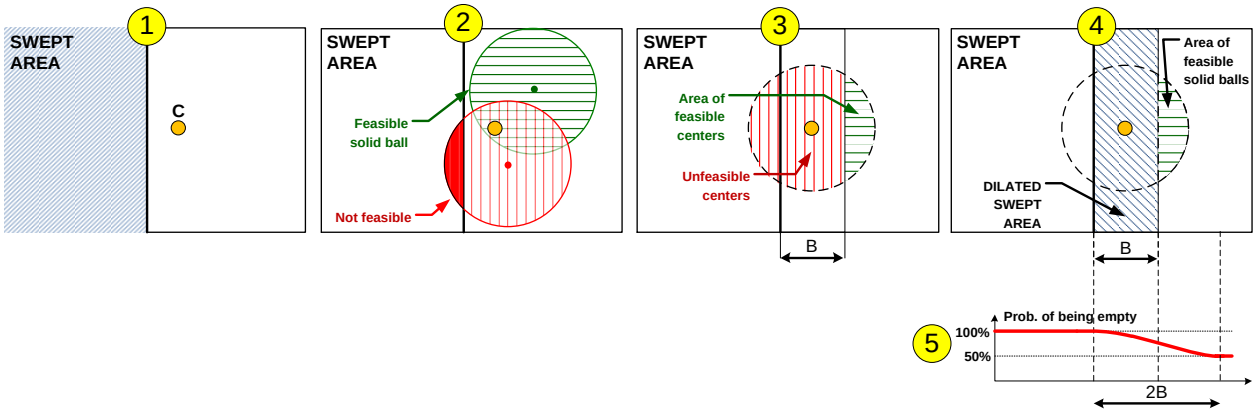

Figure 57. How to compute the correlation between cells with the assumption that the surface has maximum curvature radius B. The four steps represent an example for the computation of Pc at a point C near a large swept area.

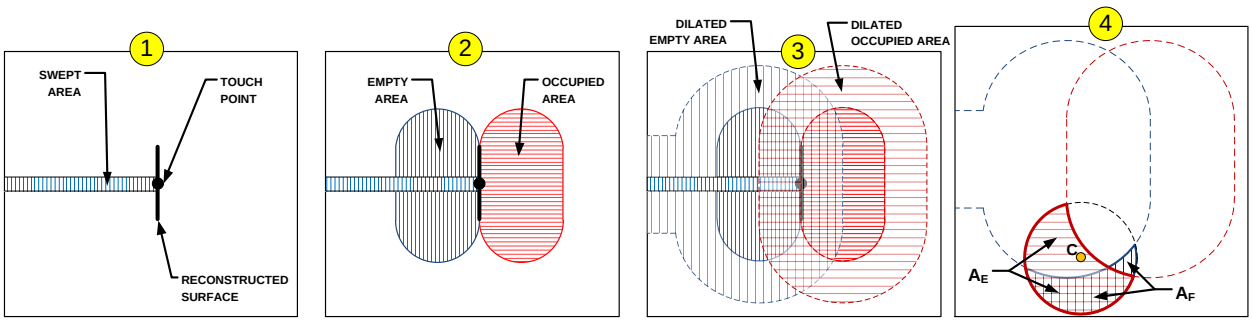

Figure 58. Procedure to determine the occupancy probability at a point near the reconstructed surface.

4. Here, the occupancy probability of a sample point C is evaluated. The area A_E in the figure represents the centers of the feasible hollow balls, while A_F represents the centers of the feasible solid balls. P_C is computed using (5.6) as $A_E/(A_E + A_F)$. As before, both A_E and A_F can be calculated with a convolution between the constraining areas and a circle of radius B.

5.9.2. Implementation of expected entropy

This section describes the implementation of Eq. (5.9)-(5.11), in order to compute the expected entropy of a point-robot tactilely exploring the environment. The equations are repeated here for convenience:

$$\Delta H_1 = P_{\bar{1}}\,\Delta H_1^{touch} + P_1\,\Delta H_1^{sweep} \tag{5.9}$$

$$E[\Delta H_n] = E[\Delta H_{n-1}] + \Delta H_{n|1,\ldots,n-1} \prod_{k=1}^{n-1} P_{k|1,\ldots,k-1} \tag{5.10}$$

$$E[C_n] = E[C_{n-1}] + \prod_{k=1}^{n-1} P_{k|1,\ldots,k-1} \left[P_{\bar{n}|1,\ldots,n-1}\,\varphi + P_{n|1,\ldots,n-1}\Delta x \right] \tag{5.11}$$

These equations contain a recurrent term in the form of the conditional probability $P_{k|1,\ldots,k-1}$. This should be computed by sweeping all cells until cell *k-1* and calculating the effect on the occupancy probability of cell *k*, using Eq. (5.6). However, the term A_{empty} in Eq. (5.6) is constant, because it is not affected by the (expected) robot sweep. Therefore, the value computed for the current entropy can be used. The term A_{occup} can be approximated as either zero, if the cell was already swept, or as influenced by only the closest cell, *k*-1. This can be written as:

$$A_{occup} = \min\left(A_{0\,occup}, \gamma\right) \tag{5.13}$$

where $A_{0\,occup}$ represents the value used for the current entropy, and γ is a constant calculated with Eq. (5.8), assuming that k is a cell surrounded by a completely swept region on one side, and an unknown region on the other side.

The term ΔH_1^{sweep} should be calculated in the same way, considering the effect of the new swept area on the probability distribution. In this work, the same approximation as in Eq. (5.13) is used.

The term ΔH_1^{touch} is more complex. When a point is touched, new information is available for surface reconstruction, and the new surface representation would drastically change the probability and entropy map. Furthermore, the touch point could be on the robot tip, or on the link. The exact computation of ΔH_1^{touch} is beyond the scope of this work, and is likely to require much computation. In this work, a heuristic strategy is used, which estimates the change in entropy by computing the probability of touching with the tip (discussed in Chapter 6) and the expected change in the entropy map according to the possible fit of the point to existing primitives. This estimate was tuned by comparing it to the exact value calculated in simple cases.

Chapter 6

Whole-Arm Exploration

This chapter develops an approach to explore an environment with a robot by touching it with any part of the robot's body. As in the previous part of this work, the robot is assumed to be equipped with only position sensors (*proprioceptive exploration)*.

In this research, the manipulator moves in the environment with an impedance control scheme (see section 3.2). By monitoring the robot's movements, this controller is able to detect when the manipulator is in contact with the environment. Because the robot is not equipped with any force/torque or tactile sensors, there is no easy way to understand where such contact occurs on the manipulator. In the previous chapters, contact was assumed to occur only on the robot's tip, so that the contact location could be determined from the manipulator's joint angles.

However, when a manipulator explores an unknown and complex environment, contact can unexpectedly occur on any part of the robot (Figure 59). The assumption of tip contact in such situations creates a wrong data point, generating an incorrect environment map. Moreover, the lack of a correct map may prevent the exploration strategy from working properly, and the manipulator may not be able to continue the exploration. This is why the approach taken in the previous chapters needs to be generalized to situations where any part of the manipulator comes into contact with the environment.

The easiest solution to this problem is to recognize when contact does not occur on the tip, and discard these data points. However this approach is not ideal, because the contact point contains useful information, which would not be used if the data are discarded. Therefore, this research goes beyond the simple recognition of contact location, and develops a method to equivalently use the touch data whether contact occurs on the tip or on any other part of the manipulator. This allows not only to correctly reconstruct the map, but to significantly accelerate the mapping process. With an analogy

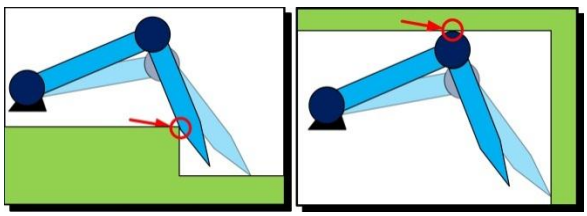

Figure 59. Two situations where the robot touches the environment with its body.

to the concept of Whole Arm Manipulator introduced in [132, 133], this method is called *Whole Arm Exploration*.

The approach is summarized in three steps. The first step estimates what link is in contact, using the applied torques and robot movements constrained by the environments. The second step estimates the probability of touching with the tip or the rest of the link. Finally, the third step uses information from contact on the tip or on the link to determine the environment's surface.

6.1. Previous Work

The earliest ideas to deal with contact between environment and robot's body were introduced by Salisbury with the concept of Whole Arm Manipulation [132, 133], a manipulation system where the robot interacts with the environment not only with the end effector, but with the whole arm. Along the same line, Gordon studied the integration of tactile-force and joint-torque data to control a Whole Arm Manipulator [134]. However, Whole Arm Manipulators detect the contact location with several force-torque sensors or by covering the robot with tactile sensors. Such sensors are too delicate to be reliable in harsh environments, and therefore cannot be used in this research. An approach that does not require distribute sensors was proposed in [135]. By measuring the forces acting on a flexible probe at the end effector, the contact location on the probe is estimated. However, the approach still requires a force/torque sensor, and assumes that contact occurs only along the flexible probe.

Two series of interesting studies introduce robotic proprioceptive sensing, using only position sensors to detect the contact location of a robot in an unknown environment [54, 55]. The first develops a heuristic method called "*self-posture changeability*", to estimate the location of contact between a robot and the environment. The key idea is to lock all the links but two, one of which is moved under positional control, and the other under compliance control. With specific movements, the link is forced to slide on the object, which allows the robot to infer where the rotation occurs, and from that the contact location. The second series of studies develops an interesting approach applicable to a 2D robot with circular joints and rectangular links. For each of the robot's links and joints, two hypothetical contact points per link are first estimated, using position, velocity and torque data. Then, geometric reasoning is used to discard all the hypotheses but one. Finally, a sliding motion of the robot on the environment is analyzed using a Kalman filter, refining the contact location. These two studies introduce interesting concepts, and have recently been used to solve a problem in 2D proprioceptive exploration using a snake robot [19]. However, they have two limitations that make them unfeasible for the exploration of a general harsh environment. First, they can only be applied to a planar robot and environment (the first work is effectively applied to a two-dimensional case, by fixing all the joints but two). Second, they require a specific sliding motion of the

manipulator on the environment whenever contact is evaluated. The sliding motion is time consuming, and it is often not feasible in harsh environments.

Proprioceptive sensing has been used in haptics, to estimate the shape of a object manipulated with a robotic hand [56]. However, such estimation cannot be applied to harsh environmental exploration, because it is based on several assumptions: the robotic fingers are assumed to stay in contact with the object at all time, they can slide on it, and contact with the object only occurs along the last link of the fingers.

As a conclusion, the sensing methods proposed in the literature propose interesting concepts, but they are not applicable to the tactile exploration of an unknown, constraining environment: they require tactile and force/torque sensors, assume a sliding motion of the link on the environment, or can be only apply to simple cases such as planar environments.

6.2. The Approach

This research develops a new approach for this problem, with a different objective. While the literature in the past has focused on the contact location on the manipulator, this research focuses on the surface. The objective of this method is not to determine where the robot touches the environment, but to construct a map of the environment. The contact location is only determined when needed, and treated as one of the many pieces of information that are used to create the map in the shortest time.

To simplify the approach, the manipulator is assumed to have a simple shape. In the planar case, link bodies are assumed to be rectangles, and joints and tip are circles. In the spatial case, link bodies are cylinders while joints and tip are spheres. Joints are assumed to be anchored to the body link before them, and to maintain their shape at any time. The robot tip is treated as the final "joint" of the last link.

Joint position measurements and a rough estimate of the applied torques are available at all times. In this work, the rough torque estimate is obtained using the desired torques from the impedance controller.

The approach is composed of three steps (Figure 60):

1. Data from joint velocity at the moment of contact and desired joint torques provides a tool to determine which link came into contact with the environment.
2. Once the contact link is determined, the partial knowledge of the environment is used to estimate the probability that contact occurs either on the joint/tip or on the link body.
3. Tip and body contact data are used together to determine the surface primitives, in a common procedure of segmentation and least squares minimization.

6.3. Detection of What Link Is in Contact

The first step of this procedure is the determination of what link is in contact with the environment. This problem has been solved in the past with different techniques [55, 132, 134]. Here, these techniques are applied to the problem in this research.

The robot is controlled under impedance control, by moving a virtual point in space and letting the robot follow it. If contact occurs at the n^{th} link, all the links from 1 to n will stop, but the following links will still continue to follow the virtual point because a torque is applied to their joints. Therefore, contact can be detected at the n^{th} link. If the robot configuration is such that the impedance controller does not require any torque to the links after the n^{th}, such links cannot be in contact, because they cannot apply any force to the environment. Therefore, when contact is detected but the links from n+1 to the last have zero torque, contact must occur on the n^{th} link.

In most cases, this is enough to determine what link is in contact with the environment. However, if the environment is particularly rough, the torque signal is extremely unreliable, or very high joint friction affects the movements, it could fail. In such cases, a strategy can be executed to determine the contact location using position constraints and not torque considerations. The manipulator executes small, oscillatory motions by moving back and forth one joint at the time and fixing the others. By monitoring what movements are allowed or constrained, the link in contact can be determined. This technique is more reliable, but it requires considerably more time.

6.4. Probability of Tip Contact

Once the link in contact is detected, the approach is the same for any link, because the simple robot shape makes the robotic tip equivalent to any other joint. Without loss of generality, here it is assumed that contact occurs on the last link.

Even if the link in contact is detected, no information is available on the exact location of

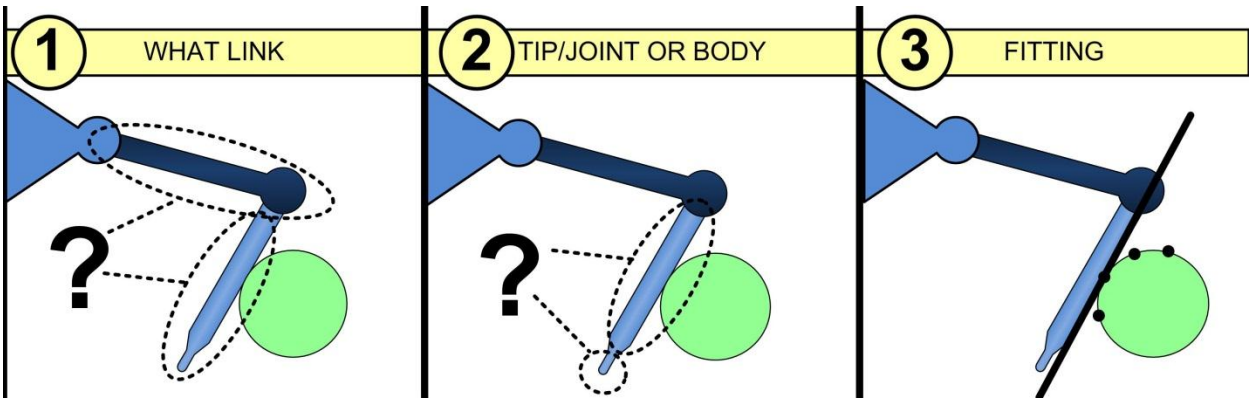

Figure 60. The three steps of the approach for proprioceptive tactile exploration.

contact along the link. There are two different cases, which will be dealt with in different ways: contact on the tip, and contact on the link body. Both cases are useful, because they provide a constraint on the location of the surface; however these two constraints are very different. Hence, it is first necessary to understand whether a data point belongs to one case or the other.

The approach in this research is to evaluate the probability that either of the two cases occurs. Let P_T denote the probability that contact occurs on the tip, called *tip probability*. Let P_L be the probability of contact on the link body, called *link probability*. Since it is assumed from the previous section that either tip or link are certainly in contact, $P_T+P_L=1$. When one of the two is very high, the point is classified to be a "tip contact", or a "link contact". When the two probabilities are comparable, the case is uncertain, and further investigation is needed before reliably using the data.

The tip probability is evaluated considering the robot movements at the time of contact and the partial knowledge of the map. This knowledge relies on the partial reconstructed surface and on the occupancy probability map, as defined in section 5.3.

6.4.1. Tip probability in unknown environment

To introduce the concept, a very simple case is considered first. A two-dimensional robot is moving in a totally unexplored area, where the region around the robot has not been explored, and the probability of any location being empty is uniformly 50%. In such case, the occupancy map is uninformative.

In a similar fashion as in section 5.3.1, the probability of touching the surface with the tip is estimated as the total number of surfaces that can be encountered during the travel of the tip, divided by the total number of surfaces during the travel of the whole link:

$$P_T = \frac{\sum possible\,surfaces\,encountered\,by\,tip}{\sum possible\,surfaces\,encountered\,by\,whole\,link} \tag{5.14}$$

An intuitive way to evaluate this ratio would be considering the area swept by the tip and the rest of the body during its movement (Figure 61):

$$P_T = \frac{A_{swept,TIP}}{A_{swept,TIP+LINK}} \tag{5.15}$$

This formula has an evident problem: if the robot has no thickness, the area swept by the tip is zero, but the tip probability is not. For instance, if a thin arrow is randomly shot inside a room, it will most likely hit an obstacle with its tip, no matter how thin the arrow is. The reason why this happens is because objects in a real environment are made by continuous surfaces. If the environment is discretized in cells, these cells are not randomly

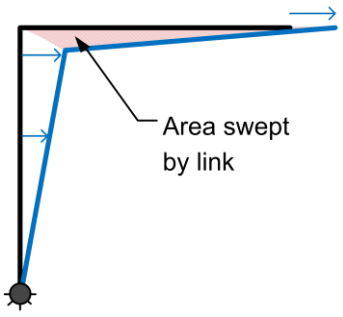

Figure 61. Robot moving in an unknown environment. Contact probability is proportional to the swept area.

and independently empty or occupied, but they are correlated to each other. This is the same situation that occurred when discussing the correlation between cells in the occupancy probability, section 5.3.1. Hence, the same approach is used. The parameter B, indicating the size of the features in the environment, is assumed to represent the minimum expected radius of curvature of the environment. Therefore, to count the number of possible surfaces that could come into contact with the robot during its movement, the number of possible *balls* with radius B is counted. In particular, the number of "solid balls" that become unfeasible due to the robot sweep is counted. This is obtained by separately dilating the shape of the link and the tip, and computing the area swept by the dilated region (Figure 62):

$$P_{TIP} = \frac{A_{swept,\ DIL\ TIP}}{A_{swept,\ DIL\ TIP+LINK}} \tag{5.16}$$

Two clarifications are needed:

1. The dilated area of the tip is a circle or radius B. Nevertheless, only the front half of this circle is not covered by the link, and generates tip contact. This can be imposed excluding, from the area swept by the tip, the area that was swept by any other part of the robot any instant before. As an example, if the movement is opposite to the one in Figure 62, the area swept by the tip is zero, in agreement with the probability of touching something with the tip.
2. Similarly, the area swept by the link needs to consider the area that has been first swept by another link. In Figure 62, only a part of the dilated body creates swept area, because some of the area was swept by the first link.

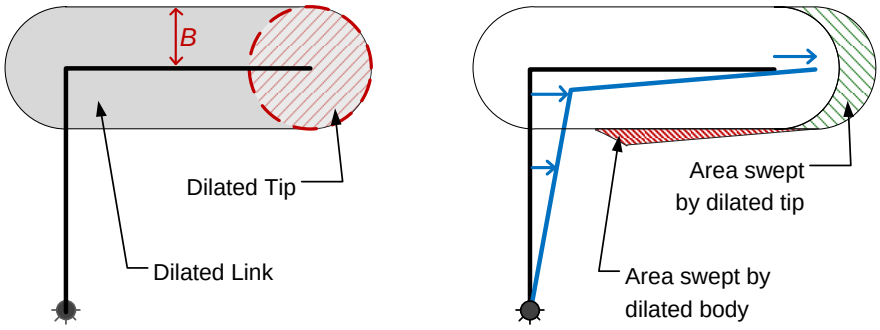

Figure 62. Area swept by dilated robot and dilated tip.

6.4.2. Tip probability using the occupancy map

The approach explained above works when all the regions around the robot have the same probability of being empty or occupied. The situation is different when the robot explores a region where some locations are most likely free or occupied. For instance, if the robot moves over a cell that is most probably occupied, there is higher probability that contact occurs on that cell. To take this into account, the swept area is weighted by the occupancy probability map. Particular care should be taken because the measured area is generated by the dilated robot,

while the occupancy probability refers to the cells swept by the original robot. Therefore, a function $y=g(x)$ is defined to indicate, for each cell x of the dilated area, the part of the robot y that originated the dilated cell x. The probability considered for the cell x is $P_C\left(g(x)\right)$ instead of simply $P_C\left(x\right)$. Since the tip is the only point originating its dilation, $P_C\left(g(x)\right)$ is always equal to the probability of the cell where the tip is located, $P_C(x_{TIP})$. Denoting the probability of a cell being occupied as $P_{\bar{C}} = 1 - P_C$, Eq. (5.16) is generalized as:

$$P_{TIP} = \frac{\iint_{DIL,\,TIP} P_{\bar{C}}\left(g(x)\right)dx}{\iint_{DIL,\,LINK} P_{\bar{C}}\left(g(x)\right)dx} = \frac{P_{\bar{C}}(x_{TIP})\,A_{DIL,TIP}}{\iint_{DIL,\,LINK} P_{\bar{C}}\left(g(x)\right)dx} \tag{5.17}$$

The formula can be simplified by reducing the robot displacement to an infinitesimal step. The area swept by a single point on the robot becomes the dot product between its velocity and the vector normal to the (dilated) robot: $\vec{n}_x \cdot \vec{v}_x$. The integral over an area in Eq. (5.17) becomes a line integral, along the profile of the dilated robot and dilated tip. The integral of dilated tip is to be constrained to only the positive tip side:

$$P_{TIP} = \frac{P_{\bar{C}}(x_{TIP})\int_{DIL,\,TIP}\left(\vec{n}_x \cdot \vec{v}_{TIP}\right)_{>0}\,dx}{\int_{DIL,\,LINK}\left(\vec{n}_x \cdot \vec{v}_x\right)P_{\bar{C}}\left(g(x)\right)dx} \tag{5.18}$$

where $\vec{n}_x$ is the outer normal to the dilated shape, and direction of the axis of the link.

The first of these integrals can be computed analytically, because the probability can be factored outside. With the symbols indicated in Figure 63, it can be expressed as:

$$\begin{aligned}\int\left(\vec{n}_x \cdot \vec{v}_{TIP}\right)_{>0} dx &= \int_{\max\left(-\frac{\pi}{2},\alpha-\frac{\pi}{2}\right)}^{\min\left(\frac{\pi}{2},\alpha+\frac{\pi}{2}\right)} \left[\cos\alpha\cos\theta + \sin\alpha\sin\theta\right] Bv\,d\theta \\ &= Bv\left(1+\cos\alpha\right) = Bv\left(1 + \frac{\vec{v}_{TIP}}{|v_{TIP}|}\cdot\vec{n}_L\right)\end{aligned} \tag{5.19}$$

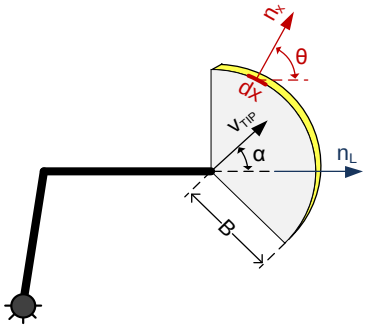

Figure 63. Line integral of the region swept by the tip.

6.5. Segmentation with Tangent Constraints

The approach described so far classifies points in two categories: "tip contact" and "link contact". This section takes as an input these two types of data and uses them to create a model of the surface. The approach is based on the standard surface reconstruction method described in section 3.3. Here, however, the two different types of data are used together in a new approach.

6.5.1. Point and tangent fit

When contact is on the tip, the contact location can be derived from the robot's kinematics. Hence, the reconstructed surface is required to pass by the touched point. This was solved in section 3.3 using a least squares approach, Eq. (3.3). If the tip is not a point but a circle (or a sphere, in 3D) of radius w, the same least square minimization can be used by simply squaring, instead of the distance point-surface d, the difference between d and w (Figure 64). The same approach can be used when contact occurs on a robot joint, as long as the joint is circular/spherical. Thus, contact on a joint or the tip can be treated in the same way, and will be referred as simply "tip contact". The least squares minimization (3.3) to determine the surface parameters becomes:

$$\theta = \arg\min_{\theta} \sum_{i=1}^{N_T} \left[d\left(P_i, S(\theta)\right) - w_i \right]^2 \tag{5.20}$$

When contact occurs on the robot link, the contact location is not known. However, the "link contact" imposes another constraint: the surface must be tangent to the side of the link. The tangent line is known through robot kinematics, and an equation can be written by imposing the surface to be tangent to such line. This is obtained by requiring the distance d between line and surface to be zero. When more data points are used together, a least squares approach minimizes the sum of the squared distances. If the link has no thickness, the line is simply the axis of the link in contact. If the link has a finite thickness w, the same approach as before is used: the line is still the robot axis, and the value to square is the difference between d and w instead of d alone (Figure 65, left). Denoting by L_i the link axes, and N_L the number of "link contacts", the minimization is:

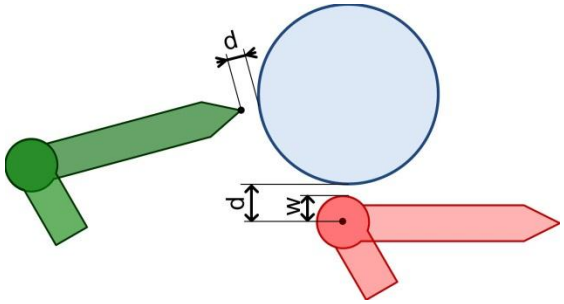

Figure 64. Primitive fitting when contact occurs along a circular (or spherical in 3D) part of the robot, such as the tip or a joint.

$$\theta = \arg\min_{\theta} \sum_{i=1}^{N_L} \left[d\left(L_i, S(\theta)\right) - w_i \right]^2 \tag{5.21}$$

Since both tip and tangent fits can be expressed with a formulation minimizing a squared distance, they can be used together. Whenever a primitive is touched with some tip contacts and some link contacts (Figure 65, right), a common minimization can be written:

$$\theta = \arg\min_{\theta} \left(\sum_{i=1}^{N_T} \left[d\left(P_i, S(\theta)\right) - w_i \right]^2 + \sum_{i=1}^{N_L} \left[d\left(L_i, S(\theta)\right) - w_i \right]^2 \right) \tag{5.22}$$

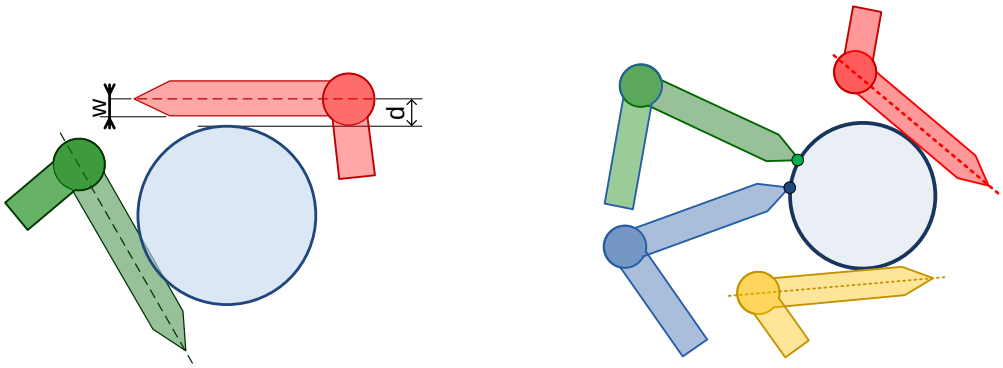

Figure 65. Left: Primitive fitting when contact occurs along the robot's link. Right: Primitive fitting with both point contact and tangent constraint.

6.5.2. Segmentation

Before being able to fit both type of data to a primitive, the number and type of primitives need to be determined, and the data points need to be assigned to such primitives. This problem, called *segmentation*, has been treated in section 3.3 for only contact points. The same approach can be extended to deal with tangent constraints. This extension has three noteworthy changes.

First, the segmentation procedure used in 3.3, fit and grow [69, 70], requires a measure of distance between points in order to expand initial small regions to neighbor points. Tangent constraints do not specify a contact point but a contact line, therefore this distance is undefined. However, the link has a finite width, and the contact needs to occur between the two ends. In this work, the middle point between these ends is used to compute the distance to neighbor points.

Second, the finite width of the links provides another tool to check the correctness of the operation. Once the primitive is fit, the contact point can be found for the tangent lines. If this contact point lies outside the length of the link, the primitive is unacceptable, and it will be discarded.

Third, the new type of contact creates a complex interaction between surface primitives, tangent constraints and intersections. The link can impose a tangential constraint to only concave locations. Hence, all the convex primitives and the primitives with no curvature (lines in 2D and planes in 3D) will never include a "link contact". This must be taken into account during the segmentation process. However, there are other locations that can be touched: the concave intersections between primitives (Figure 67). Therefore, every time two primitives are found, and their intersection is concave, the intersection itself becomes a "primitive" in the segmentation process, because some tangent data points can touch it. If the segmentation assigns "link contacts" to such intersection, the intersection's parameters are affected by the two primitives and by the tangent data point. In other words, the two primitives and the intersection are correlated with each other, and they should be determined with a common minimization, called *constrained fitting* [73].

It is worth noticing that this approach is extremely simple to implement in the planar case, because intersections between primitives are simple points (which can be treated as circles of zero radius). However, in three dimensions primitives are not points but curves. Even for the simple geometric primitives used in this research, such curves can have a complex shape (see [71, 72]), and computing analytically the distance between these curves and the robot is challenging. The approach can be solved numerically, but this goes beyond the scope of this research. This work implements the complete segmentation process only in the planar case.

6.5.3. Computation of the tangent fit

The distance function between a line and a surface is easy to formulate for the geometric primitives used in this research. Lines (in 2D) and planes (in 3D) do not have curvature, and tangent contact is impossible for a robot, because the surface needs to be concave. For a sphere, the distance line-surface can be reduced to the distance between a line and a point; for a cylinder and a cone, to the distance between two lines; for the torus, to the distance between a line and a circle in space. Algorithms to solve such problems can be found in [136]. In the planar case of a circle, it can be expressed as a distance between point and line. However, a linear formulation can also be found. The line parallel to a line $ax + by + c = 0$ at a distance d is given by:

$$ax + by + c - d\sqrt{a^2 + b^2} = 0 \tag{5.23}$$

If the line equation is normalized, and the distance is expressed in terms of the circle radius r and the link width w, with the sign s equal to the sign of the joint torque, the equation becomes:

$$ax + by + c - s(r + w) = 0 \tag{5.24}$$

This can be written in the following matrix form, which allows a linear least squares minimization:

$$\begin{bmatrix} a_1 & b_1 & -s_1 \\ \dots & & \dots \\ a_n & b_n & -s_n \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ r \end{bmatrix} = \begin{bmatrix} s_1 w - c_1 \\ \dots \\ s_n w - c_n \end{bmatrix} \tag{5.25}$$

6.5.3.1. Numerical robustness

This section investigates the numerical robustness to measurement noise and surface irregularity of the fitting process with tangent constraints. This is shown through a case study where a surface is estimated using touch points, tangent points and a combination of both.

Figure 66 represents a planar robot touching a circle in sixteen points: eight points with its tip, and eight with its link. Both tip and link contact points span the same amount of the circle (60^o). The circle is corrupted by random surface roughness, and the encoders are corrupted by random measurement noise. The best least square fit of the circle is computed using some of these corrupted measurements, for three different cases:

1. Using the eight tip contacts;
2. Using the eight link contacts;
3. Using four tip and four link contacts. The two sets of four data points are adjacent, so that the eight contact locations still span the same amount of the circle as in case (1) and (2).

The parameters found using the least square fit are then compared to the original circle values. If the norm of the difference is smaller than a threshold, the fit is considered successful; otherwise it is unsuccessful. This is carried out for increasing value of surface roughness (up to 2.5% of the circle radius) and for increasing values of joint measurement noise (up to a standard deviation on 0.25). For each value of roughness and noise, 1000 trials have been executed, and the percentage of the good fits is shown in Figure 68 and Figure 69. The figures show that, although the link contact by itself is more sensitive to joint measurement noise, the combination of four tip and four link points always give the best results. The addition of link contact to the minimization, therefore, increases the robustness of the surface fit. This also means that the exploration is performed faster by using a combination of tip and link contacts, because fewer data are needed to reach the same precision.

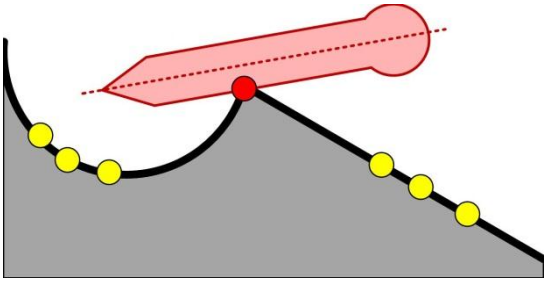

Figure 67. The robot link in contact with the intersection point between two primitives.

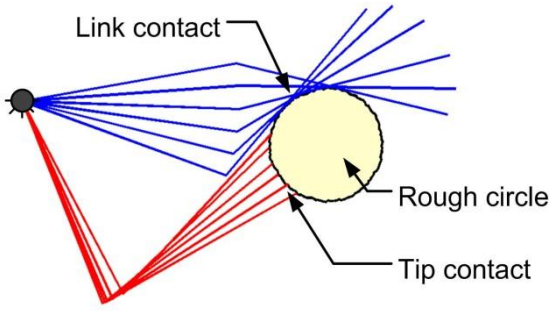

Figure 66. The situation used for the simulation investigating the numerical stability of tangent fit.

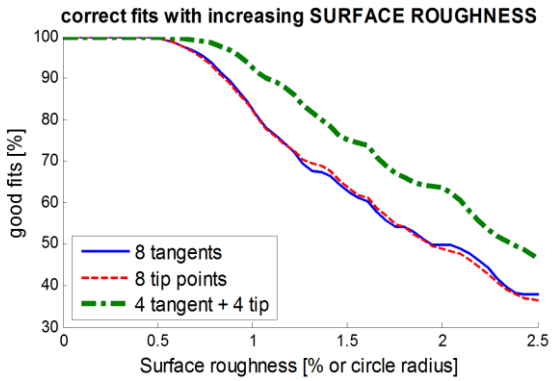

Figure 68. Percentage of fits that are close enough to the original circle, for three data set and for increasing surface roughness.

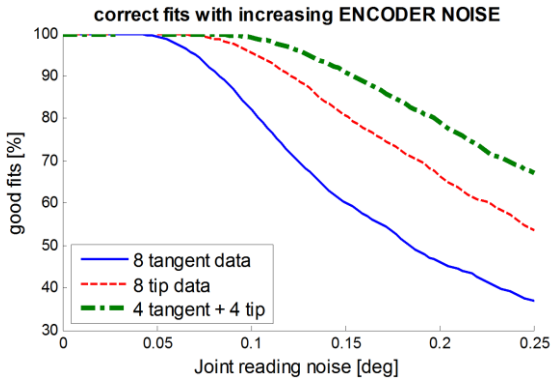

Figure 69. Percentage of fits that are close enough to the original circle, for three data set and for increasing measurement noise.

6.6. Weighted Least Squares

The approach described in this chapter can be summarized as a two-step process: the classification of touch data into "tip contact" and "link contact", and the use of such data to construct the model of the surface. An underlying assumption is that points are classified correctly. If a tip contact is classified as tip or vice versa, the resulting surface fit could be drastically wrong. This limitation is true; however it is mitigated by several factors.

First of all, the surface is updated and the tip probability is re-evaluated every time new information is available. Therefore, an incorrect interpretation can be corrected later, once more data are available.

Second, the more the surface is known, the more the estimation of tip probability improves. A common reason for improvement is the swept region after contact is detected. If later movements sweep the region where the tip was located during contact, the tip probability drops to 0%. Similarly, if later movements sweep all the area around the location where the link was located during contact, the probability rises to 100%. Such movements could be executed on purpose, to increase the confidence of the estimation. This is basically the intuitive idea of "feeling around" to understand what the environment looks like in the proximity of contact. In this work, these movements have not been implemented, because the mere continuation of the exploration is enough to let the probability tend to a value close to 0% or 100%.

For these reasons, the classification usually works well in practice. However, this research implements a further strategy to increase the robustness of the procedure. It is based on the same idea used in section 4.3.4 during backlash estimation: trust the safe data points more than the unsafe. When a point is classified with confidence, that point is more reliable, and should be considered more in the surface reconstruction. This can be formally obtained with a weighted least squares approach. Whenever the uncertainty of measurement points is not constant, the

best linear unbiased estimator is the sum of the squared distances, weighted by the reciprocal of the variance σ² [94]. Eq.(5.22) becomes:

$$\theta = \arg\min_{\theta}\left(\sum_{i=1}^{N_T} \frac{1}{\sigma_i^2}\left[d\left(P_i, S(\theta)\right) - w_i \right]^2 + \sum_{i=1}^{N_L} \frac{1}{\sigma_i^2}\left[d\left(L_i, S(\theta)\right) - w_i \right]^2 \right) \tag{5.26}$$

An estimate of the variance σ² can be obtained from the tip probability in both cases. Here, the computation for tip contact is outlined.

Figure 70 shows the situation for a link of length L, where the probability of tip contact is $P_T=a$. If contact is not on the tip, a uniform probability distribution along the link is assumed (a more precise formulation should weight the distribution according to the occupancy probability). The average value and the variance of the contact location are:

$$\mu = 0 \cdot \alpha + \frac{L}{2}(1-\alpha) = \frac{1-\alpha}{2} L \tag{5.27}$$

$$\begin{aligned} \sigma^2 &= \int (x-\mu)^2 \, p(x)\, dx \\ &= \frac{(1-\alpha)^2}{4} L^2 + \frac{1-\alpha}{L} \int_{x=0}^{x=1} \left[\frac{1-\alpha}{2} L - x \right]^2 dx \; = \frac{L^2}{12}\left(1 + 2\alpha - 3\alpha^2\right) \end{aligned} \tag{5.28}$$

This variance tends to zero when the probability tends to one. This is correct, because the point is known to be a tip contact for sure. However, some measurement noise is always present, and is should be added to the total measurement variance[4]. This prevents the weights in (5.26) from going to zero.

A similar calculation can be performed for the point classified as link contact.

The use of weighted least squares reduces the effect of wrong classification: uncertain points have high variance, and will be discounted more in the least squares fit.

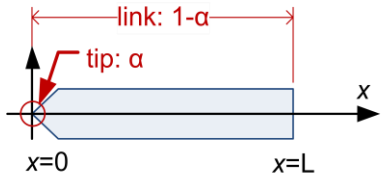

Figure 70. Estimation of the variance for contact on the tip.

6.7. Simulations

A series of simulations have been performed in several structured environments. The exploration strategy described in Chapter 5 has been used: the Rapidly-Exploring Random Tree with maximum entropy reduction. Whenever contact is detected, the tip probability is evaluated, and points are classified in tip or link contact, and the surface is reconstructed

[4] To be precise, the variance of the sum of two variables is the sum of the variances only for independent Gaussian distributions. Here, the distributions are indeed independent and noise can be reliably modeled as Gaussian, but the distribution of the location is not Gaussian. However, a precise computation is extremely complex and the game is not worth the candle.

accordingly. The simulations also show the integration between the RRT-entropy approach and the contact detection.

Three simulation results are shown here. Two of them are performed with a planar robot, and the third with the spatial oil well manipulator. The extended fitting and segmentation method has only been implemented in the planar case. However, the spatial simulation implements the calculation of the tip probability, and shows the feasibility of the same approach in three dimensions.

Figure 71 shows six moments of the exploration of a planar environment made of four lines and two circles, with three concave intersections and a concave circle. The six frames show that the tip probability (in percentage) correctly classifies points into tip and link contacts, and probability improves in time. Link contacts points are fit to an intersection, which is recognized as belonging to a known circle.

Figure 72 shows the exploration of the same environment from another starting location, so that the first contact occurs on the link. The estimation of tip probability improves thanks to the new acquired information. When four tangent contacts are recognized, the circle is correctly fit.

Figure 73 shows four frames from the same simulation run in section 5.5.3, with exactly the same random numbers. This figure, however, focuses on the tip probability, plotting such probability for the points touched with the link. The algorithm correctly classifies the points as tip contacts, and the probability improves when a larger surface is discovered.

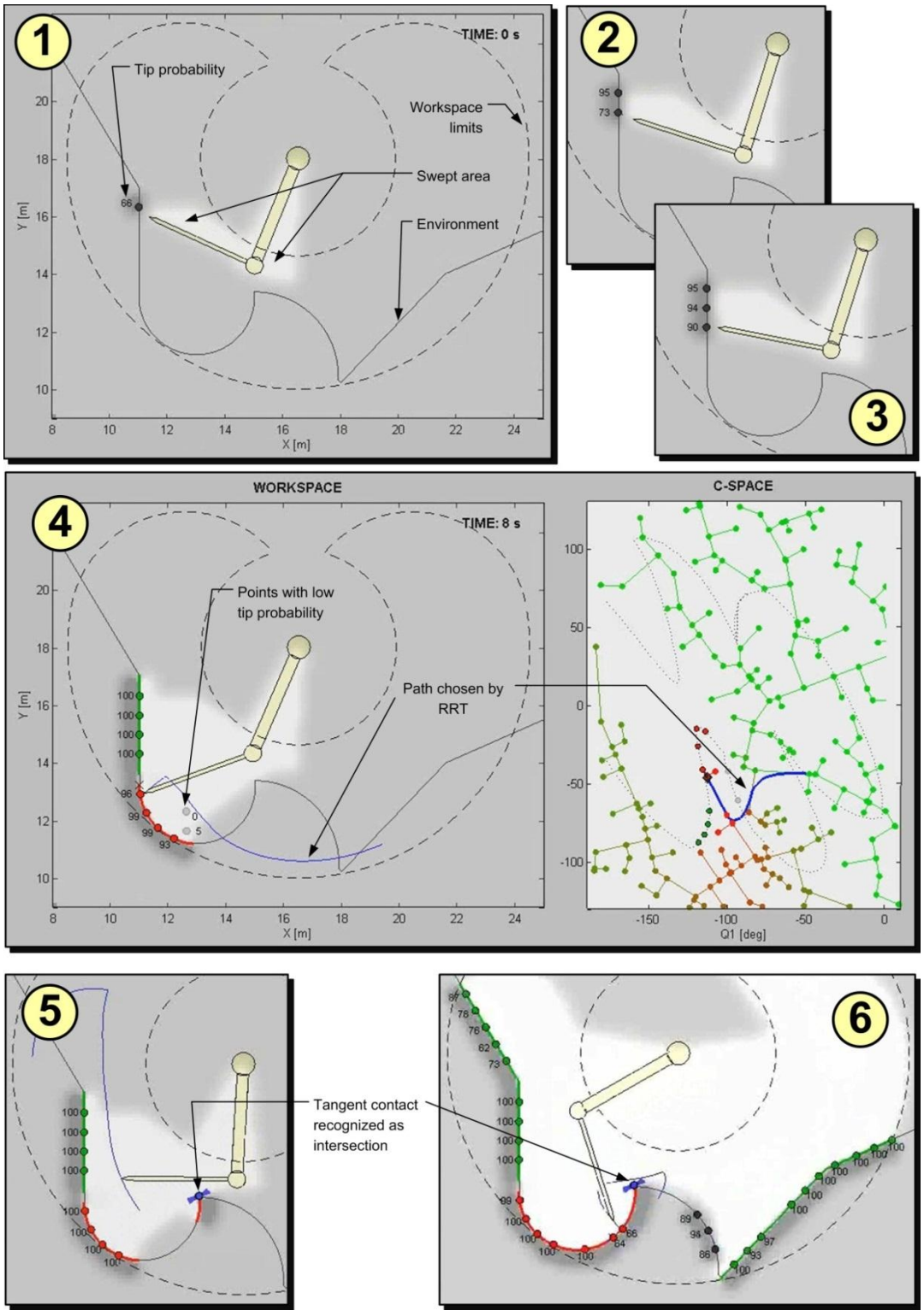

Figure 71. Six representative moments of the exploration with tip probability and tangent constraints. (1) First contact point. The point is correctly classified, but the tip probability is 66%. (2,3) More touch points are available, and the tip probability automatically improves. (4) RRT with maximum entropy reduction. Two non-tip points are correctly recognized. (5) Non-tip points are fit to an intersection, which is recognized to belong to a known circle. (6) Towards the end of the exploration: most tip probabilities tend to 100%.

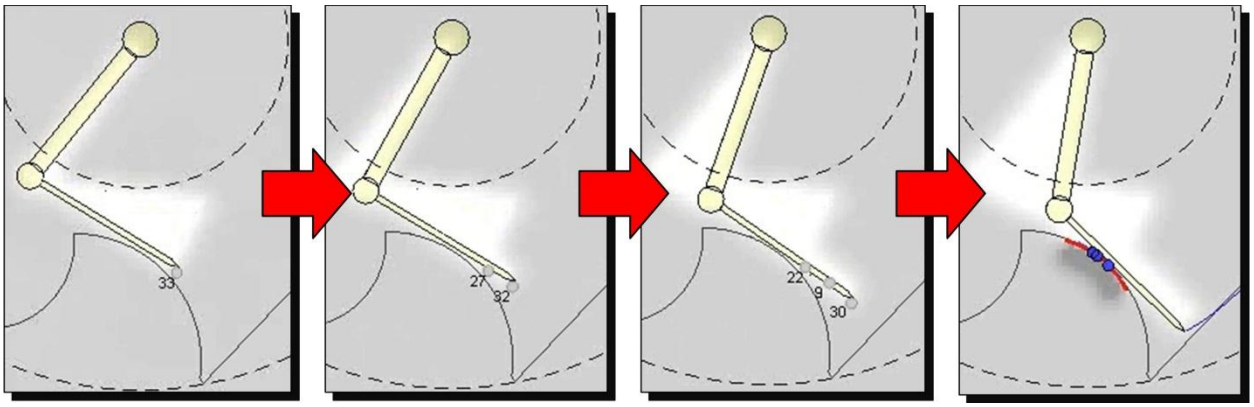

Figure 72. A concave circle is recognized using four consecutive link contacts. The tip probability improves from measurement to measurement. In the last figure, the probabilities (not shown), are 18%, 3%, 3% and 22%. The 3% residual value is due to the finite grid size.

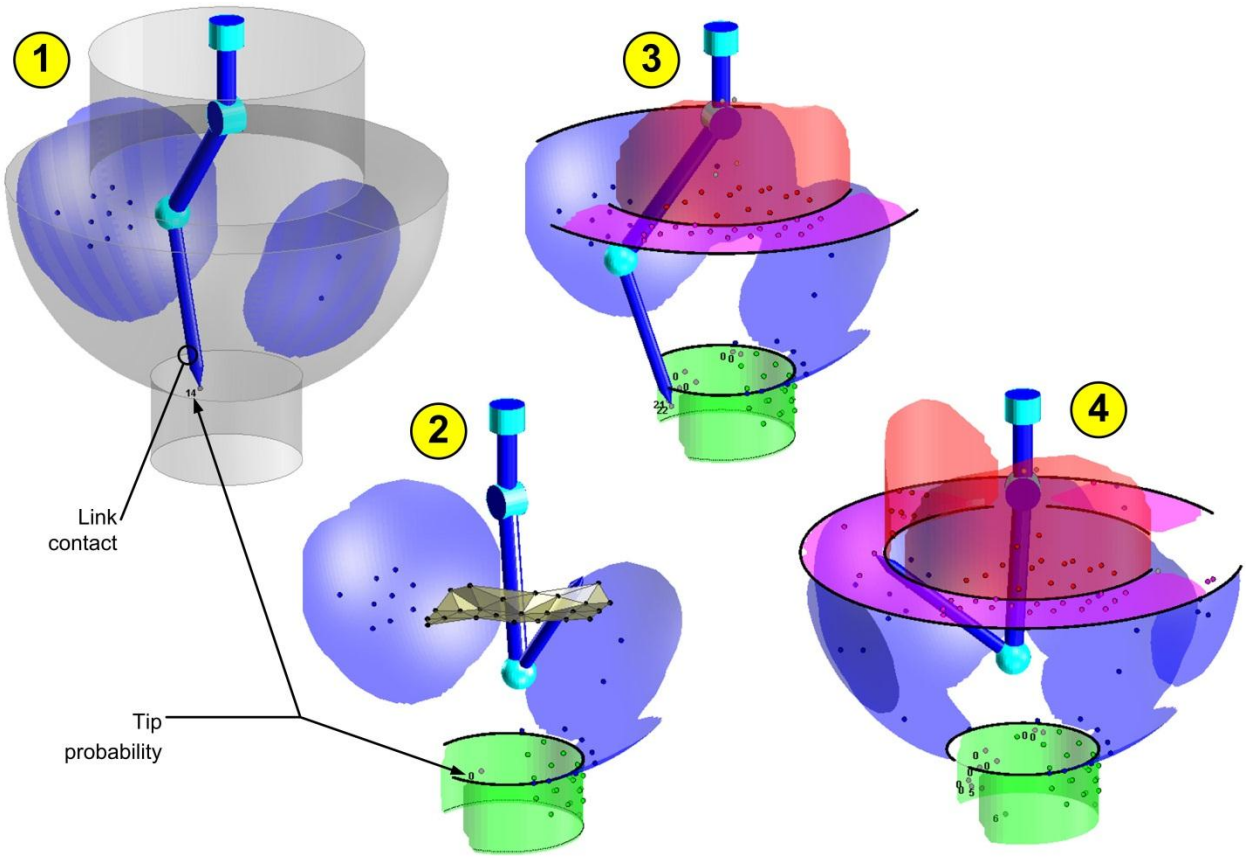

Figure 73. The same simulation run as in Figure 51 is depicted here, showing the tip probability of points in contact with the link. (1) The first link contact occurs, and the tip probability is 14%. (2) When the lower cylinder is detected, the probability correctly drops to 0%. (3) Other link points are detected. Since this work does not implement the segmentation in 3D, these points are not used here. Since no surface is built, the RRT tries several times to go beyond the points with such configuration, creating several link contacts. (4) With more information, all the probabilities of link contacts tend to 0%.

6.8. Experiments

A series of experiments have been performed using the planar robot described in Section 5.7 and shown in Figure 54.

In the first experiment, the same exploration trial shown in Figure 55 is proposed, focusing here on the importance of the detection of tip probability. Several contact points occur at the intersection between the top line and the top circle. The algorithm correctly classifies them as link contact, and uses them to refine the location of the intersection between the line and the circle. In this case, the robot mounts a triangular tip.

In a second set of experiments, a simpler environment is explored, composed of a circle and two planes (Figure 74). The same robot is used, but here a circular tip is mounted. The width of the link and the diameter of the tip are 1-inch. The circular tip provides a way to exclude the case of contact between the environment and a side of the triangular tip, a case which is not considered in this work. On the other hand, the thick tip does not permit the exploration of locations with a higher curvature than the tip radius. This is not a problem in this case, since the environment does not have small features. The environment's shape is such that contact can only occur along the last link and on the tip. The initial condition of the exploration is chosen so that the robot comes into contact for the first time with its link, as in Figure 74. The rest of the exploration is executed using the RRT with entropy reduction described in Chapter 5.

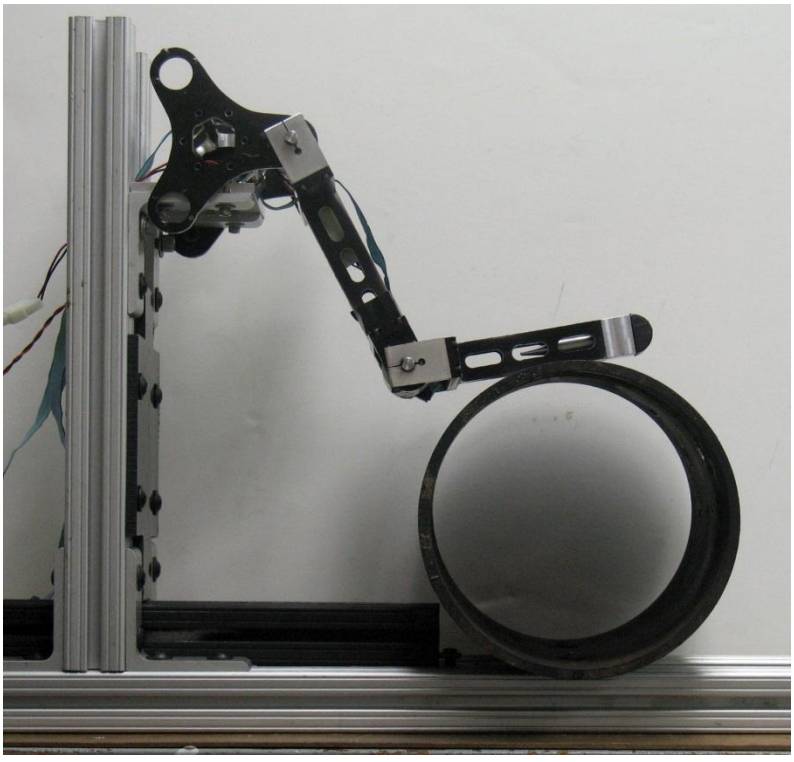

Figure 74.Environment composed of a circle and two planes. The manipulator's tip is a circle of same size as the link's body (1 inch). The figure shows the first contact location.

The whole environment is explored after 22 points: 12 tip contact points, and 10 link contact points. The result of the exploration is shown in Figure 76. The figure shows with different colors the points that are recognized as link or as tip contact on the circle. Using both types of data allows determining the radius of the circle with an error of 3%. If only the four tip contacts are used, the error in the radius increases to 18%, while if only the ten link contacts are used, the error is 5%. The synergy between the two types of data is evident.

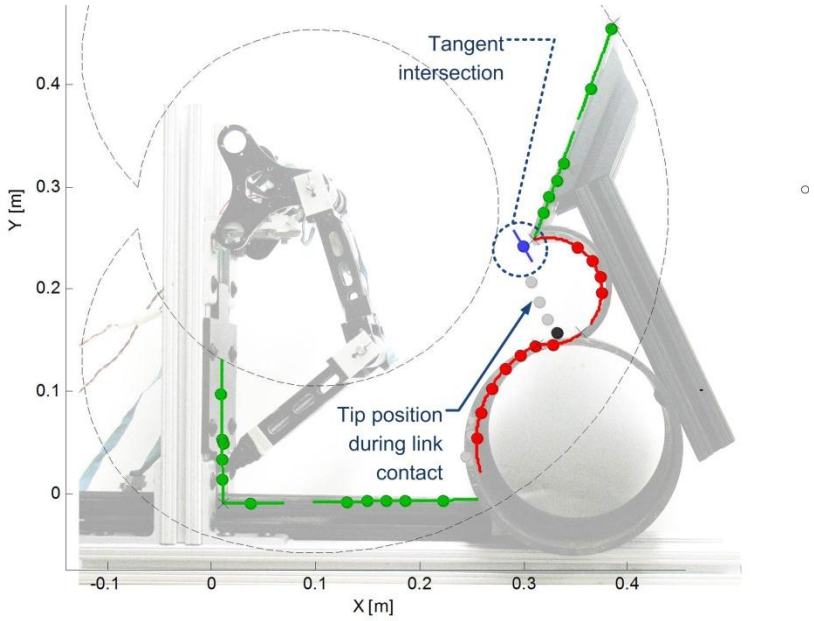

Figure 75. Experimental results of the exploration of a planar environment, showing the detection of "link contact" at an intersection between primitives.

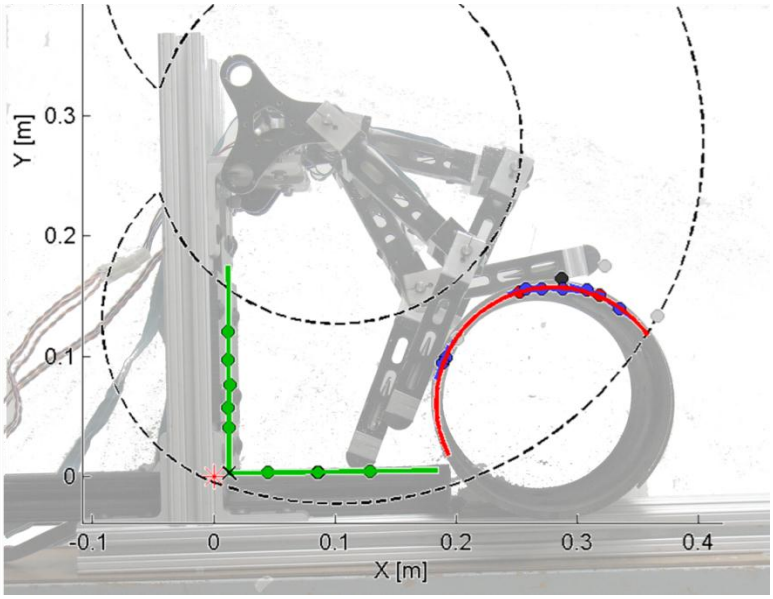

Figure 76. Reconstructed surface, superimposed to the original surface. The circle is reconstructed using 4 tip contacts and 10 link contacts.

6.9. Conclusions

This chapter develops an approach to explore an environment with a robot by touching it with any part of the robot's body. A method is developed, which equivalently uses the touch data whether contact occurs on the tip or on any other part of the manipulator. This allows not only to correctly reconstruct the map, but to significantly accelerate the mapping process.

The approach is summarized in three steps. First, the link in contact is estimated based on the applied torques and robot movements constrained by the environments. Then, the probability of touching with the tip or the link is estimated. Finally, a method is provided to use information from contact on the link in a similar fashion than what is done for contact points on the tip.

Several simulations and experiments demonstrate that the approach correctly determines the type of contact and the location along the manipulator. This makes the exploration of a complex and constraining environment using only position sensors feasible. Moreover, the two types of contact are shown to perform very well when used together to determine the same surface, resulting in a faster exploration of the environment.

Chapter 7

Conclusions

7.1. Summary

This work develops the first approach to tactilely explore rough environments when time is critical. In extremely harsh environments, range, force/torque or tactile sensor cannot be used. A representative case is the mapping of oil wells, where extreme pressures and temperatures and an opaque fluid filling the well make these sensors useless or unreliable, and expenses for oil well inactivity require short exploration time.

To map such environments, this work proposes the use of tactile exploration with a manipulator provided with only joint encoders. The manipulator is brought in proximity to the surface to map, and its base is fixed. The environment is then mapped by touching the surface with the manipulator and reading the position measurements during contact. Since tactilely probing a surface is inherently slow, collected tactile data will be sparse. These data are analyzed in real time to provide a provisional model of the surface and to choose the robot's movements to reduce the mapping time.

The first part of this work demonstrates the feasibility of the approach. Real-time impedance control provides reliable robot movements and the detection of the surface using only joint encoders. A representation based on geometric primitives is introduced, to describe the surface using the few, sparse data available. The robustness of the method is tested against surface roughness and different surrounding fluids. The problem of joint backlash, highly affecting the robot precision, is solved developing a new strategy to compensate for backlash error, by simultaneously identifying both surface and backlash values. The feasibility is validated experimentally with a 3 degree-of-freedom prototype manipulator specifically developed for this application.

The second part of the work proposes an optimal strategy to map a constraining environment with a manipulator. A hybrid approach involving both workspace and configuration space maximizes the ability to interpret the tactile data. The amount of knowledge of the environment is evaluated with an information-theoretic approach, and the

robot's movements are chosen to maximize the expected increase of such knowledge. The algorithm is probabilistically complete and can be made optimal with horizon one. Since the robot only possess position sensors, probabilistic techniques identify the contact location on the robot, and to use any type of contact to acquire knowledge on the surface.

7.2. Extensions

The work presented in this research should be seen as a "complete package". First, an unsolved problem is introduced: the mapping of rough environments in short time. A simple solution is proposed: the use of a manipulator with only joint encoder. Then, the solution is tested and validated experimentally. Finally, the solution is extended to achieve (partial) optimality and to be able to work in any constraining environment. However, there are several research directions where this work could be extended. Some of these directions are as follows:

1. **Mobile base**. This research assumes that the manipulator base is fixed with the surface at all times. However, in some applications the base cannot be held fixed. This is the case when a manipulator is mounted on an underwater vehicle, or when the environment to be mapped does not fit in the robot's workspace and the base needs to be relocated. The displacement of the base is usually hard to measure precisely, and the error introduced this way must be taken into account [57, 97].
2. **Extension to multi-manipulators.** There are applications where more than one manipulator can explore the environment. In a robotic hand, for instance, several fingers can simultaneously explore an object or a surface. As another example, several models of underwater ROVs already possess two robotic arms. Exploration with multiple mobile robots is a known topic in the literature [125, 126, 137], but the use of manipulators has not been studied.
3. **Sensor integration.** This approach has been developed for situations where range sensors are not feasible. However, there are applications where range sensor can be used, but their measurements are not completely reliable in some parts of the environment. In such situations, the integration of range and tactile sensing would considerably increase the exploration speed [138].

7.3. Lessons Learned

This section quickly explains a few concepts that the author learnt during this research. This does not want to describe all the skills and concepts learnt during a PhD thesis, but only a couple of interesting, general concepts regarding tactile exploration.

First and foremost: humans are smart. Tactile exploration with only position sensors is a challenging problem, and oftentimes the solution to some of its aspects is not trivial. However, human beings do it extremely well. The exploration of a dark room is not a good example,

because human beings have very imprecise position sensors (more precisely, proprioceptive sensors), but extremely sensitive and precise contact and tactile sensors. A better example consists in a man inserting a rigid stick into a dark hole, and moving it to understand its shape, while watching the hand outside this hole to determine its position. This analogy provided several ideas that have been used in this work, just by simply mimicking the intuitive movements that everyone would do. For instance, an instinctive movement is waiving the stick in free space, remembering where the stick has passed because there is no surface in such locations. Furthermore, once contact is detected, it is natural to probe discretely in different locations to understand the shape of the hole. Moreover, if the stick is held horizontally and is moved up and down, the man intrinsically assumes that contact occurs along the side of the stick, and can quickly imagine a surface on the side wherever contact is. On the other hand, if the stick is moved back and forth, the man naturally thinks that contact occurs on the tip, and reconstructs in his mind the surface at the tip location. Several concepts introduced in this research are simply intuitive behaviors, described and implemented in a slightly more formal fashion.

Another important lesson learnt about tactile exploration is that the problem is extremely hard if faced in its most general case. Some a-priori knowledge dramatically decreases the complexity of the problem. In this work, the intrinsic knowledge is represented by the assumption that the surface is regular enough to be represented as primitives. Another piece of knowledge helping both the computation of a probability map and detecting the contact location is the given "characteristic length" B. The surface is assumed to change (in terms of curvature) within the limits of B, and therefore a prediction of the neighborhood of a known point is possible. Here, again, human beings do exactly the same, and in a much better way. Humans intrinsically mix exploration and objects recognition, based on some intuitive a-priori expectation. For example, if a person touches a sharp corner at the height of his hips in the middle of a dark room, he will probably assume that the object is a table. This hypothesis might be wrong and will be tested later. However, this dramatically speeds up the exploration. The use of (even only expected) partial knowledge, of physical intuition is a key element to determine a solution to tactile exploration.

Bibliography

[1] *B.P. website: www.bp.com*.

[2] D. Kettler, "Robotic Tactile Oil Mapping: Feasibility, Experimental System and Digital Mechatronic Concepts," Master's thesis, Massachusetts Institute of Technology, Cambridge MA, 2009.

[3] R. Samuel, "Downhole Drilling Tools – Theory and Practice for Students and Engineers," ed: Gulf Publishing, Huston TX, 2007.

[4] M.A. Meggiolaro and S. Dubowsky, "Improving the Positioning Accuracy of Powerful Manipulators with Application in Nuclear Maintenance," *Proceedings of the 16th Brazilian Congress of Mechanical Engineering on Robotics and Control*, pp. 210–219, Rio de Janeiro, 2001.

[5] T. Fukuda, H. Hosokai, and M. Uemura, "Rubber Gas Actuator Driven by Hydrogen Storage Alloy for in-Pipe Inspection Mobile Robot with Flexible Structure," *IEEE International Conference on Robotics and Automation*, pp. 1847-1852, 1989.

[6] Y. Kawaguchi, I. Yoshida, H. Kurumatani, T. Kikuta, and Y. Yamada, "Internal Pipe Inspection Robot," *IEEE International Conference on Robotics and Automation*, pp. 857-862, Kyoto, 1995.

[7] J. Hertzberg and F. Kirchner, "Landmark-Based Autonomous Navigation in Sewerage Pipes," *1st Euromicro Workshop on Advanced Mobile Robots*, pp. 68-73, 1996.

[8] M. Gary, N. Fairfield, W.C. Stone, D. Wettergreen, G. Kantor, and J.M. Sharp Jr, "3d Mapping and Characterization of Sistema Zacatón from Depthx (Deep Phreatic Thermal Explorer)," *Proceedings of KARST08: 11th Sinkhole conference ASCE*, 2008.

[9] J.E. Bares and D.S. Wettergreen, "Dante Ii: Technical Description, Results, and Lessons Learned," *The International Journal of Robotics Research,* vol. 18, p. 621, 1999.

[10] C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D. Hahnel, and S. Thrun, "A Campaign in Autonomous Mine Mapping," *IEEE International Conference on Robotics and Automation*, pp. 2004-2009, 2005.

[11] D. Bapna, E. Rollins, J. Murphy, E. Maimone, W. Whittaker, and D. Wettergreen, "The Atacama Desert Trek: Outcomes," *IEEE International Conference on Robotics and Automation*, pp. 597-604, 2002.

[12] C. Urmson, B. Shamah, J. Teza, M.D. Wagner, D. Apostolopoulos, and W.R. Whittaker, "A Sensor Arm for Robotic Antarctic Meteorite Search," *3rd Int. Conf. Field Service Robotics,*, Helsinki, Finland, 2001.

[13] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, and Y. Shinoda, "Experience with Rover Navigation for Lunar-Like Terrains," *Intelligent Robots and Systems*, pp. 441-446, 1995.

[14] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin, "Mars Microrover Navigation: Performance Evaluation and Enhancement," *Autonomous Robots,* vol. 2, pp. 291-311, 1995.

[15] K. Iagnemma and S. Dubowsky, *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers*: Springer Verlag, 2004.

[16] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually Navigating the Rms Titanic with Slam Information Filters," *Robotics: Science and Systems*, 2005.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*: The MIT Press, 2005.

[18] S. Thrun, "Robotic Mapping: A Survey," *Exploring Artificial Intelligence in the New Millennium,* 2002.

[19] J. Everist and S. Wei-Min, "Mapping Opaque and Confined Environments Using Proprioception," *IEEE International Conference on Robotics and Automation*, pp. 1041-1046, Kobe, Japan, 2009.

[20] W.E.L. Grimson and T. Lozano-Perez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *The International Journal of Robotics Research,* vol. 3, 1984.

[21] J.L. Schneiter and T.B. Sheridan, "An Automated Tactile Sensing Strategy for Planar Object Recognition and Localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, pp. 775-786, 1990.

[22] K. Roberts, "Robot Active Touch Exploration: Constraints and Strategies," *IEEE International Conference on Robotics and Automation*, pp. 980-985, Cincinnati, OH, 1990.

[23] S. Caselli, C. Magnanini, F. Zanichelli, and E. Caraffi, "Efficient Exploration and Recognition of Convex Objects Based on Haptic Perception," *IEEE International Conference on Robotics and Automation* pp. 3508-3513, Minneapolis, MN 1996.
[24] A. Okamura and M. Cutkosky, "Feature Detection for Haptic Exploration with Robotic Fingers," *The International Journal of Robotics Research,* vol. 20, pp. 925-938, 2001.
[25] E. Petriu, S. Yeung, S. Das, A. Cretu, and H. Spoelder, "Robotic Tactile Recognition of Pseudorandom Encoded Objects," *IEEE Trans. on Instrumentation and Measurement,* vol. 53, pp. 1425-1432, 2004.
[26] K. Gadeyne and H. Bruyninckx, "Markov Techniques for Object Localization with Force-Controlled Robots," *10th International Conference on Advanced Robotics*, pp. 91–96, Budapest, Hungary, 2001.
[27] K. Gadeyne, T. Lefebvre, and H. Bruyninckx, "Bayesian Hybrid Model-State Estimation Applied to Simultaneous Contact Formation Recognition and Geometrical Parameter Estimation," *The International Journal of Robotics Research,* vol. 24, p. 615, 2005.
[28] A. Petrovskaya, O. Khatib, S. Thrun, and A. Ng, "Touch Based Perception for Object Manipulation," *Robotics: Science and Systems, Robot Manipulation Workshop*, 2007.
[29] M.A. Schaeffer and A.M. Okamura, "Methods for Intelligent Localization and Mapping During Haptic Exploration," pp. 3438-3445, 2003.
[30] J.S. Bay, "Tactile Shape Sensing Via Single-and Multifingered Hands," *IEEE International Conference on Robotics and Automation*, pp. 290-295, 1989.
[31] M. Moll and M. Erdmann, "Reconstructing the Shape and Motion of Unknown Objects with Active Tactile Sensors," *Algorithmic Foundations of Robotics V, Springer-Verlag,* pp. 293–309, 2004.
[32] P. Allen and P. Michelman, "Acquisition and Interpretation of 3-D Sensor Data from Touch," *IEEE Transactions on Robotics and Automation,* vol. 6, pp. 397-404, 1990.
[33] M. Charlebois, K. Gupta, and S. Payandeh, "Shape Description of General, Curved Surfaces Using Tactile Sensing and Surface Normal Information," *IEEE International Conference on Robotics and Automation*, pp. 2819-2824, 1997.
[34] M. Charlebois, K. Gupta, and S. Payandeh, "Shape Description of Curved Surfaces from Contact Sensing Using Surface Normals," *International Journal of Robotics Research,* vol. 18, pp. 779-787, 1999.
[35] S. Petchartee, G. Monkman, and A. Suebsomran, "3d-Shape Recognition Based Tactile Sensor," *International Conference on Sensing Technology, New Zealand,* 2007.
[36] A. Bicchi, A. Marigo, and D. Prattichizzo, "Dexterity through Rolling: Manipulation of Unknown Objects," *IEEE International Conference on Robotics and Automation*, p. 1583, Detroit, MI 1999.
[37] M. Erdmann, "An Exploration of Nonprehensile Two-Palm Manipulation," *The International Journal of Robotics Research,* vol. 17, p. 485, 1998.
[38] M. Erdmann and M. Mason, "An Exploration of Sensorless Manipulation," *IEEE Journal of Robotics and Automation,* vol. 4, pp. 369-379, 2002.
[39] G.C. Vosniakos and T. Giannakakis, "Reverse Engineering of Simple Surfaces of Unknown Shape with Touch Probes: Scanning and Compensation Issues," *Institution of Mechanical Engineers, Part B: J. of Eng. Manufacture,* pp. 563-568, 2003.
[40] P.N. Chivate and A.G. Jablokow, "Solid-Model Generation from Measured Point Data," *Computer Aided Design,* vol. 25, pp. 587-600, 1993.
[41] P.N. Chivate and A.G. Jablokow, "Review of Surface Representations and Fitting for Reverse Engineering," *Computer Integrated Manufacturing Systems,* vol. 8, pp. 193-204, 1995.
[42] R. Cole and C.K. Yap, "Shape from Probing," *Journal of Algorithms,* vol. 8, pp. 19-38, 1987.
[43] L.P. Chew, "Guaranteed-Quality Mesh Generation for Curved Surfaces," *9th Annual Symposium on Computational Geometry,* pp. 274-280, 1993.
[44] J. Boissonnat, L. Guibas, and S. Oudot, "Learning Smooth Shapes by Probing," *Computational Geometry: Theory and Applications,* vol. 37, pp. 38-58, 2007.
[45] V. Caglioti, "An Entropic Criterion for Minimum Uncertainty Sensing in Recognition and Localization. I. Theoretical and Conceptual Aspects," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, ,* vol. 31, pp. 187-196, 2002.
[46] K. Pribadi, J.S. Bay, and H. Hemami, "Exploration and Dynamic Shape Estimation by a Robotic Probe," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 19, pp. 840-846, 1989.
[47] J.S. Bay, "Tactile Shape Sensing Via Single- and Multifingered Hands," *IEEE International Conference on Robotics and Automation* pp. 290-295, Scottsdale, AZ 1989.
[48] L. Fuming and T. Hasegawa, "Reconstruction of Curved Surfaces Using Active Tactile Sensing and Surface Normal Information," *IEEE International Conference on Robotics and Automation* pp. 4029-4034, Seoul, Korea, 2001.

[49] R.S. Fearing, "Tactile Sensing for Shape Interpretation," in *Dextrous Robot Hands*, pp. 209-238, ed: Springer-Verlag New York, Inc., 1990.
[50] M.D. Berkemeier and R.S. Fearing, "Determining the Axis of a Surface of Revolution Using Tactile Sensing," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, pp. 1079-1087, 1993.
[51] N. Chen, R. Rink, and H. Zhang, "Local Object Shape from Tactile Sensing," *IEEE International Conference on Robotics and Automation*, p. 3496, Minneapolis, MN 1996.
[52] D. Keren, E. Rivlin, I. Shimshoni, and I. Weiss, "Recognizing 3d Objects Using Tactile Sensing and Curve Invariants," *Journal of Mathematical Imaging and Vision,* vol. 12, pp. 5-23, 2000.
[53] S.H. Suh, S.K. Lee, and J.J. Lee, "Compensating Probe Radius in Free Surface Modelling with Cmm: Simulation and Experiment," *International Journal of Production Research,* vol. 34, pp. 507-523, 1996.
[54] M. Huber and R. Grupen, "2-D Contact Detection and Localization Using Proprioceptive Information," *IEEE Transactions on Robotics and Automation,* vol. 10, pp. 23-33, 1994.
[55] M. Kaneko and K. Tanie, "Contact Point Detection for Grasping an Unknown Object Using Self-Posture Changeability," *IEEE Transactions on Robotics and Automation,* vol. 10, pp. 355-367, 1994.
[56] S. Haidacher and G. Hirzinger, "Contact Point Identification in Multi-Fingered Grasps Exploiting Kinematic Constraints," *IEEE International Conference on Robotics and Automation* pp. 1597-1603 vol.2, 2002.
[57] F. Lu and E. Milios, "Globally Consistent Range Scan Alignment for Environment Mapping," *Autonomous Robots,* vol. 4, pp. 333-349, 1997.
[58] A. Requicha and H. Voelcker, "Solid Modeling: A Historical Summary and Contemporary Assessment," *IEEE Computer Graphics and Applications,* vol. 2, pp. 9-24, 1982.
[59] G. Zeng and A. Hemami, "An Overview of Robot Force Control," *Robotica,* vol. 15, pp. 473-482, 1997.
[60] M.H. Raibert and J.J. Craig, "Hybrid Position/Force Control of Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control,* vol. 102, pp. 126–133, 1981.
[61] N. Hogan, "Impedance Control-an Approach to Manipulation. I-Theory. Ii-Implementation. Iii-Applications," *ASME Journal of Dynamic Systems, Measurement, and Control,* vol. 107, pp. 1-34, 1985.
[62] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*: Springer Verlag, 2000.
[63] B. Armstrong-Hélouvry, *Control of Machines with Friction*: Kluwer Academic Pub, 1991.
[64] H. Olsson, K.J. Åström, C. Canudas De Wit, M. Gäfvert, and P. Lischinsky, "Friction Models and Friction Compensation," *European Journal of Control,* vol. 4, pp. 176-195, 1998.
[65] P. Benko, R.R. Martin, and T. Várady, "Algorithms for Reverse Engineering Boundary Representation Models," *Computer-Aided Design,* vol. 33, pp. 839-851, 2001.
[66] S. Ahn, *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*. New York, NY: Springer-Verlag, 2004.
[67] G. Lukacs, R. Martin, and D. Marshall, "Faithful Least-Squares Fitting of Spheres, Cylinders, Cones and Tori for Reliable Segmentation," *Lecture Notes in Computer Science,* vol. 1406, pp. 671–686, 1998.
[68] X. Jiang and D. Cheng, "A Novel Parameter Decomposition Approach to Faithful Fitting of Quadric Surfaces," *Lecture Notes in Computer Science,* vol. 3663, pp. 168-175, 2005.
[69] P. Besl and R. Jain, "Segmentation through Variable-Order Surface Fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 10, pp. 167-192, 1988.
[70] A. Leonardis, A. Gupta, and R. Bajcsy, "Segmentation of Range Images as the Search for Geometric Parametric Models," *International Journal of Computer Vision,* vol. 14, pp. 253-277, 1995.
[71] W. Wang, R. Goldman, and C. Tu, "Enhancing Levin's Method for Computing Quadric-Surface Intersections," *Computer Aided Geometric Design,* vol. 20, pp. 401-422, 2003.
[72] J.R. Miller and R.N. Goldman, "Geometric Algorithms for Detecting and Calculating All Conic Sections in the Intersection of Any 2 Natural Quadric Surfaces," *Graphical Models and Image Processing,* vol. 57, pp. 55-66, 1995.
[73] P. Benko, G. Kos, T. Varady, L. Andor, and R. Martin, "Constrained Fitting in Reverse Engineering," *Computer Aided Geometric Design,* vol. 19, pp. 173-205, 2002.
[74] G. Kós, R.R. Martin, and T. Várady, "Methods to Recover Constant Radius Rolling Ball Blends in Reverse Engineering," *Computer Aided Geometric Design,* vol. 17, pp. 127-160, 2000.
[75] S.M. LaValle, *Planning Algorithms*: Cambridge University Press, 2006.
[76] N. Mitra and A. Nguyen, "Estimating Surface Normals in Noisy Point Cloud Data," *International Journal of Computational Geometry and Applications,* vol. 14, pp. 261-276, 2004.
[77] W. Lyons and G. Plisga, *Standard Handbook of Petroleum and Natural Gas Engineering*: Gulf Professional Publishing, 2005.
[78] J. Yuh, "Modeling and Control of Underwater Robotic Vehicles," *IEEE Trans. on Systems, Man and Cybernetics,* vol. 20, pp. 1475-1483, 1990.

[79] G. Antonelli, *Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems*: Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.
[80] T.I. Fossen and S.I. Sagatun, "Adaptive Control of Nonlinear Systems: A Case Study of Underwater Robotic Systems," *Journal of Robotic Systems,* vol. 8, pp. 393-412, 1991.
[81] C.R. Carignan and D.L. Akin, "The Reaction Stabilization of on-Orbit Robots," *IEEE Control Systems Magazine,* vol. 20, pp. 19-33, 2000.
[82] Y. Nakayama and R.F. Boucher, *Introduction to Fluid Mechanics*: Butterworth-Heinemann, 1998.
[83] S. Besnard, W. Khalil, and G. Garcia, "Geometric Calibration of Robots Using Multiple Plane Constraints," *Advances in Robot Kinematics,* pp. 61-70, 2000.
[84] J. Hollerbach and C. Wampler, "The Calibration Index and Taxonomy for Robot Kinematic Calibration Methods," *The International Journal of Robotics Research,* vol. 15, pp. 573-591, 1996.
[85] M. Meggiolaro, G. Scriffignano, and S. Dubowsky, "Manipulator Calibration Using a Single Endpoint Contact Constraint," presented at the ASME Design Engineering Technical Conference Baltimore, MD, 2000.
[86] B. Mooring, M. Driels, and Z. Roth, *Fundamentals of Manipulator Calibration*. New York, NY: John Wiley & Sons, Inc., 1991.
[87] Z. Roth, B. Mooring, and B. Ravani, "An Overview of Robot Calibration," *IEEE Journal of Robotics and Automation,* vol. 3, pp. 377-385, 1987.
[88] N. Chironis, *Gear Design and Application*: McGraw-Hill, 1967.
[89] N. Dagalakis and D. Myers, "Adjustment of Robot Joint Gear Backlash Using the Robot Joint Test Excitation Technique," *The International Journal of Robotics Research,* vol. 4, p. 65, 1985.
[90] N. Sarkar, R. Ellis, and T. Moore, "Backlash Detection in Geared Mechanisms: Modeling, Simulation, and Experimentation," *Mechanical Systems and Signal Processing,* vol. 11, pp. 391-408, 1997.
[91] J. Stein and C. Wang, "Estimation of Gear Backlash: Theory and Simulation," *Journal of Dynamic Systems, Measurement, and Control,* vol. 120, p. 74, 1998.
[92] D. Seidl, S. Lam, J. Putman, and R. Lorenz, "Neural Network Compensation of Gear Backlash Hysteresis in Position-Controlled Mechanisms," *IEEE Transactions on Industry applications,* vol. 31, pp. 1475-1483, 1995.
[93] R. Horst, P. Pardalos, and N. Thoai, *Introduction to Global Optimization*: Kluwer Academic Publishers, 2000.
[94] R.L. Plackett, "Some Theorems in Least Squares," *Biometrika,* vol. 37, p. 149, 1950.
[95] C. Robertson and C. Krauss, "Gulf Spill Is the Largest of Its Kind, Scientists Say," in *The New York Times*, ed, August 2, 2010.
[96] "Bp Leak the World's Worst Accidental Oil Spill," *The Daily Telegraph,* 03 August 2010.
[97] S. Dubowsky and A. Tanner, "A Study of the Dynamics and Control of Mobile Manipulators Subjected to Vehicle Disturbances," *4th international symposium on Robotics Research*, pp. 111-117, 1988.
[98] J. Latombe, *Robot Motion Planning*: Springer, 1991.
[99] Y.K. Hwang and N. Ahuja, "Gross Motion Planning - a Survey," *ACM Computing Surveys (CSUR),* vol. 24, pp. 219-291, 1992.
[100] J.T. Schwartz and M. Sharir, "A Survey of Motion Planning and Related Geometric Algorithms," *Artificial Intelligence,* vol. 37, pp. 157-169, 1988.
[101] J. Canny, *The Complexity of Robot Motion Planning*: The MIT Press, 1988.
[102] M. Sharir, "Algorithmic Motion Planning in Robotics," *Computer,* vol. 22, pp. 9-19, 2002.
[103] L. Huajun, Y. Jingyu, L. Jianfeng, T. Zhenmin, Z. Chunxia, and C. Weiming, "Research on Mobile Robots Motion Planning: A Survey [J]," *Engineering Science,* vol. 1, 2006.
[104] H.M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*: The MIT Press, 2005.
[105] T. Lozano-Perez, "Spatial Planning: A Configuration Space Approach," *IEEE Transactions on Computers,* vol. 100, pp. 108-120, 1983.
[106] K. Wise and A. Bowyer, "A Survey of Global Configuration-Space Mapping Techniques for a Single Robot in a Static Environment," *The International Journal of Robotics Research,* vol. 19, p. 762, 2000.
[107] E. Welzl, "Constructing the Visibility Graph for N-Line Segments in O (N2) Time," *Information Processing Letters,* vol. 20, pp. 167-171, 1985.
[108] O. Takahashi and R. Schilling, "Motion Planning in a Plane Using Generalized Voronoi Diagrams," *Robotics and Automation, IEEE Transactions on,* vol. 5, pp. 143-150, 2002.
[109] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research,* vol. 5, p. 90, 1986.
[110] L.E. Kavraki, P. Svestka, J.C. Latombe, and M.H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation,* vol. 12, pp. 566-580, 1996.
[111] S.M. LaValle and J.J. Kuffner, "Randomized Kinodynamic Planning," *The International Journal of Robotics Research,* vol. 20, p. 378, 2001.

[112] S. Karaman and E. Frazzoli, "Incremental Sampling-Based Algorithms for Optimal Motion Planning," *Robotics: Science and Systems*, 2010.
[113] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robotic Exploration as Graph Construction," *IEEE Transactions on Robotics and Automation,* vol. 7, pp. 859-865, 1991.
[114] B. Kuipers and Y.T. Byun, "A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations," *Robotics and Autonomous Systems,* vol. 8, pp. 47-63, 1991.
[115] B. Yamauchi, "A Frontier-Based Approach for Autonomous Exploration," *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146-151, Monterey, CA, 1997.
[116] H.H. Gonzalez-Banos and J.C. Latombe, "Navigation Strategies for Exploring Indoor Environments," *The International Journal of Robotics Research,* vol. 21, p. 829, 2002.
[117] H. Choset, "Coverage for Robotics – a Survey of Recent Results," *Annals of Mathematics and Artificial Intelligence,* vol. 31, pp. 113-126, 2001.
[118] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artificial Intelligence,* vol. 101, pp. 99-134, 1998.
[119] P. Whaite and F. Ferrie, "Autonomous Exploration: Driven by Uncertainty," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 193-205, 1997.
[120] S. Wenhardt, B. Deutsch, J. Hornegger, H. Niemann, and J. Denzler, "An Information Theoretic Approach for Next Best View Planning in 3-D Reconstruction," *Pattern Recognition,* vol. 1, pp. 103-106, 2006.
[121] W.R. Scott, G. Roth, and J.F. Rivest, "View Planning for Automated Three-Dimensional Object Reconstruction and Inspection," *ACM Computing Surveys (CSUR),* vol. 35, pp. 64-96, 2003.
[122] S.J. Moorehead, R. Simmons, and W.L. Whittaker, "Autonomous Exploration Using Multiple Sources of Information," *IEEE International Conference on Robotics and Automation*, pp. 3098-3103, Seoul, Korea, 2001.
[123] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte, "Information Based Adaptive Robotic Exploration," *IEEE International Conference on Intelligent Robots and Systems*, pp. 540-545, Lausanne, Switzerland, 2002.
[124] C. Stachniss, G. Grisetti, and W. Burgard, "Information Gain-Based Exploration Using Rao-Blackwellized Particle Filters," *Robotics: Science and Systems*, pp. 65–72, Cambridge, MA, 2005.
[125] C. Stachniss, "Exploration and Mapping with Mobile Robots," PhD thesis, Universitat Freiburg, 2006.
[126] V. Sujan and S. Dubowsky, "Efficient Information-Based Visual Robotic Mapping in Unstructured Environments," *The International Journal of Robotics Research,* vol. 24, pp. 275-293, 2005.
[127] T. Kollar and N. Roy, "Trajectory Optimization Using Reinforcement Learning for Map Exploration," *The International Journal of Robotics Research,* vol. 27, p. 175, 2008.
[128] Y. Yu and K. Gupta, "C-Space Entropy: A Measure for View Planning and Exploration for General Robot-Sensor Systems in Unknown Environments," *The International Journal of Robotics Research,* vol. 23, p. 1197, 2004.
[129] A. Elfes, "Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception," *Autonomous Mobile Robots: Perception, Mapping, and Navigation,* vol. 1, pp. 60–70.
[130] H. Moravec and A. Elfes, "High Resolution Maps from Wide Angle Sonar," *IEEE International Conference on Robotics and Automation*, pp. 116-121, 1985.
[131] S. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Report No. TR 98-11, Computer Science Dept., Iowa State University, 1998.
[132] J. Salisbury, "Interpretation of Contact Geometries from Force Measurements," *IEEE International Conference on Robotics and Automation*, pp. 240-247, 1984.
[133] J. Salisbury, W. Townsend, B. Ebrman, and D. DiPietro, "Preliminary Design of a Whole-Arm Manipulation System (Wams)," *IEEE International Conference on Robotics and Automation*, pp. 254-260, Philadelphia, PA, 1988.
[134] S. Gordon and W. Townsend, "Integration of Tactile Force and Joint Torque Information in a Whole-Arm Manipulator," *IEEE International Conference on Robotics and Automation*, pp. 464-469, Scottsdale, AZ, 1989.
[135] T. Tsujimura and T. Yabuta, "A Tactile Sensing Method for Employing Force/Torque Information through Insensitive Probes," *IEEE International Conference on Robotics and Automation*, pp. 1315-1320, Nice, Francesco, 1992.
[136] M.K. Agoston, *Computer Graphics and Geometric Modeling: Implementation and Algorithms*: Springer-Verlag New York Inc, 2005.
[137] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider, "Coordinated Multi-Robot Exploration," *IEEE Transactions on Robotics,* vol. 21, pp. 376-386, 2005.
[138] P.K. Allen, A.T. Miller, P.Y. Oh, and B.S. Leibowitz, "Integration of Vision, Force and Tactile Sensing for Grasping," *International Journal of Intelligent Machines,* vol. 4, pp. 129-149, 1999.