POLITECNICO DI MILANO

School of Industrial and Information Engineering

Master of Science in Automation and Control Engineering

Obstacle avoidance in variable admittance control for robot manual guidance

Supervisor:	Prof. Paolo Rocco
Co-supervisors:	Ing. Davide Bazzi
	Prof. Andrea Maria Zanchettin

Master Thesis dissertation of:

Giorgio Priora

Id: 921112

Academic year 2019-2020

Alla mia famiglia

Ringraziamenti

Ringrazio la mia famiglia per avermi sempre sostenuto e spronato a dare il massimo durante questi anni, è grazie a voi e al vostro supporto che oggi riesco a raggiungere i miei traguardi, dedicandomi con passione a ciò che amo.

Un immenso grazie lo dedico a te, Francesca, per tutto l'amore che mi hai donato e continui a donarmi ogni giorno. Per la pazienza che hai avuto nei miei giorni pesanti e per la felicità regalatami in quelli leggeri. Per avermi dato la carica quando ero spento. Per esserci stata sempre, in ogni momento, bello o brutto, di questo capitolo della mia vita, e per avermi sempre fatto sentire a casa, nonostante tutto il resto.

Un ulteriore ringraziamento è rivolto ai Professori Paolo Rocco e Andrea Maria Zanchettin per avermi proposto questo progetto di tesi, permettendomi di approfondire le mie conoscenze teoriche e capacità pratiche.

Un doveroso ringraziamento a Davide Bazzi, per il tempo dedicatomi durante questi mesi in laboratorio, per i suggerimenti, per le correzioni in fase di scrittura e per la pazienza con la quale ha sopperito alla mia inesperienza. Non avrei potuto trovare guida migliore.

Infine, ringrazio Andrea Car, Andrea e Martina Volonté, Alessandro Molteni, Valerio Rucconi, Fabio Canazza e tanti altri, per tutti questi anni passati insieme, per i bei momenti condivisi, anche in questo periodo difficile e soprattutto per avermi donato la gioia di un'amicizia incondizionata.

Contents

List of Figures

Abstract

Modern manufacturing is characterized by small-batched, rapidly varying production mixes, requiring some repetitive operations, that should be executed by industrial robots, and others that are too complex for them. Despite their speed and accuracy, traditional robots are not well suited for this new kind of tasks, since they lack the flexibility and soft skills of the human beings.

The Industry 4.0 paradigm aims at solving this problem through collaborative robotics. *Cobots* are endowed with particular sensors and control systems that allow them to work alongside human workers, without the need of safety barriers that prevent a direct interaction with them.

One important field of collaborative robotics is physical human robot interaction, where the operator is physically in contact with the mechanical structure of the manipulator, usually in correspondence of its end-effector. A particular branch is formed by *manual guidance* operations, where a robotic manipulator and a human worker collaborate in the transportation of heavy and usually bulky objects that a human is not able to move alone. The robot compensates for the gravitational load, and the operator guides the end-effector to the correct target, applying a force on an appropriate handle. This solution allows great flexibility in terms of different types of transported object and multiple destinations, thanks to the combination of the robot strength and human cognitive capabilities.

However, there might be cases where the bulky size of the transported object obstructs the human field of view, increasing the risk of accidental collisions during the movement and reducing the accuracy of the final positioning/insertion phase. Hence the overall performance would considerably decrease while the risks would increase.

In the present work, a new variable admittance control strategy, which is able to provide a directional haptic feedback to the operator in a *manual guidance* task, is proposed. Thanks to this feedback, the user is able to navigate in cluttered working spaces, reaching the predefined target position with very low final positioning error, even with closed eyes.

To achieve this objective, an *environment exploring structure* is constructed. Then, at each time instant, an optimal motion direction is determined based on the end-effector position with respect to the exploring structure and on the user force direction, which is a symptom of his/her motion intention. The admittance parameters are suitably varied according to the optimal movement direction, in order to realize an intuitive and effective directional haptic feedback for the operator.

The performance of the developed algorithms is evaluated through three set of experiments, which were executed by multiple volunteers assisted by a Comau Smart Six manipulator.

Sommario

L’industria manifatturiera odierna è caratterizzata da una produzione in piccole quantità e ad elevata variabilità, che richiede sia operazioni ripetitive eseguite da robot industriali, sia azioni per loro troppo elaborate. Nonostante l’elevata velocità e accuratezza, i robot tradizionali non sono adatti a questo tipo di compiti, poiché mancano della flessibilità e delle soft-skills proprie dell’uomo.

Il paradigma dell’Industria 4.0 suggerisce la robotica collaborativa come soluzione. I *Cobot* sono dotati di sensori e sistemi di controllo particolari che permettono loro di collaborare con i lavoratori umani, eliminando le barriere di sicurezza che impediscono l’interazione diretta tra le due parti.

Un importante campo applicativo della robotica collaborativa è l’interazione fisica tra uomo e robot, dove l’operatore è fisicamente in contatto con la struttura del manipolatore, solitamente in corrispondenza dell’organo terminale del robot. Una particolare categoria è costituita dalle operazioni di *guida manuale*, dove un manipolatore robotico e un operaio umano collaborano nel trasporto di oggetti pesanti e spesso voluminosi, che l’uomo non sarebbe in grado di sollevare da solo. Il manipolatore compensa il peso del carico, mentre l’operatore guida l’organo terminale del robot verso la destinazione corretta, applicando una forza su un’opportuna maniglia. Grazie alla combinazione tra forza del robot e capacità cognitive dell’uomo, questa soluzione permette una grande flessibilità, sia in termini di tipi di oggetto diversi trasportabili, sia di destinazioni variabili.

Tuttavia, ci sono casi in cui le dimensioni dell’oggetto trasportato sono tali da ostruire il campo visivo del lavoratore, aumentando il rischio di collisioni accidentali durante il movimento e peggiorando l’accuratezza del posizionamento/inserimento finale. Di conseguenza, l’efficienza dell’operazione diminuirebbe mentre i rischi aumenterebbero.

In questa tesi viene presentata una nuova strategia di controllo ad ammettenza variabile, in grado di fornire un feedback tattile all'operatore durante la *guida manuale*. Grazie a questo riscontro, l'utente è in grado raggiungere la destinazione con un minimo errore sul posizionamento finale, attraversando un ambiente di lavoro ricco di ostacoli, anche ad occhi chiusi.

A questo scopo, viene generata una *struttura di esplorazione dell'ambiente*. Ad ogni istante di tempo, viene determinata una direzione di moto ottimale basandosi sulla posizione dell'organo terminale rispetto alla struttura e sulla direzione della forza esercitata dall'utente, che indica la sua intenzione di moto. I parametri dell'ammettenza vengono modificati in base alla direzione di moto ottimale, così da realizzare un feedback tattile direzionale intuitivo ed efficace per l'operatore.

La performance dell'algoritmo sviluppato è state valutata con tre set di esperimenti, eseguiti da più volontari, assistiti da un manipolatore Comau Smart Six.

Chapter 1

Introduction

When the first *Unimate* robot was installed inside one of the General Motor automotive plants in 1961, a revolution sparked in the manufacturing industry. Until that moment, the workers were responsible for each operation in the production chain. Instead, with the introduction of *industrial robots*, human presence in the line gradually decreased. Early major applications included manipulating heavy spot-welding guns, manning spray-painting equipment, and many other jobs that resulted unpleasant or dangerous for the human workers.

In the following sixty years, constant hardware and software improvements granted these machines higher levels of speed and precision: robots can handle heavy loads, work continuously and efficiently, without the restriction of bad environmental conditions, fatigue and stress. Their ability of executing series of cyclical operations with high speed and precision boosted the productivity of large assembly and production lines. The number of workers required for certain tasks was reduced, and, in many cases robots completely took over manual labor, like in the case of Tesla automated factories, depicted in figure 1.1. The worker figure gradually adapted from machine operator to machine supervisor.

Figure 1.1: Tesla car automated assembly line.

However, in the past ten years the manufacturing product mixes have undergone some dramatic changes. Producers are moving to smaller batch sizes, made to respond to the rapidly varying demands of their customers. Manufacturers are also intervening to drive down the cost of production of materials and of logistics in the factory. Automation is an answer to these demands, but *traditional robotics* is not well suited to this task [1]. Classical robotic work cells, endowed with traditional manipulators, lack versatility and flexibility, making it difficult to adapt to dynamic environments or to efficiently fabricate small-batched production. Some of the actions which must be performed in a production line are still too complex for a robot and each minimal change in the operation sequence, dictated by high levels of customization in the final product, would require continuous reconfiguration of the workstation. A solution to this problem is to reintroduce the flexibility and cognitive soft skills of humans in the line. Nevertheless, safety of the workers in the vicinity of *traditional robots* would result in strict regulations requiring rigorous safety provisions, like steel cages around the robot workspace.

Then, it would be impossible for humans and robots to work on the same object at the same time.

Industry 4.0 attempts to fulfill the new manufacturing requirements by introducing the concept of collaborative robotics. This new type of robots should be designed to work alongside humans rather than in their own space. In a sense, they become like co-workers, which are able to solve simple and repetitive tasks with high precision. By sharing the same workspace, it is possible to provide the production lines with huge benefits. Robots force, speed and accuracy mixed with human intelligence and manual skills improve not only the industrial processes efficiency, but also the health of the worker, whose fatigue and alienation associated with cyclic tasks are thus reduced.

1.1 Collaborative robotics

The idea of collaborative robots, namely *cobots*, is strictly bonded to the advancements of automation, but it differs in how the robot is seen inside the production chain: it is not just a mechanical object, executing a series of cyclical tasks, but a device that can *learn* from humans and *act* along with them.

Figure 1.2: Example of a collaborative task.

The concept of collaboration between humans and robots is not new, with its first appearance in publications dating back to 1996, but it boomed only in 2017, both in the literature, with more than 700 articles published, and in the industrial world, with a market value of $600 millions that year and a forecast of $7.6 billions for 2027 [2].

The main difference between *traditional robots* and *cobots* is in the direct interaction with the workers. The first require safety barriers, such as metal cages, in order to protect the operator from their elevated speed and power. The *cobots*, instead, are characterized by a lightweight design and by intrinsic safety features: recent hardware advances of the robotic platforms enabled the implementation of various control techniques for improved interactions with humans and unstructured environments[3]. Hence, *cobots* can co-habit with human workers in the same environment, as portrayed in figure 1.2.

The classical division of labor, still today prevalent on factory floors, is overcome: the worker and the robot are able to collaborate in the production line, combining the speed and strength advantages of automation with the human innate flexibility, the ability to adapt to unforeseen events and the capacity of maintaining strong decision making skills also in dynamic and complex environments. The machines perform all sorts of high-precision tasks. Humans might be capable of accomplishing the same jobs alone, but by delegating the repetitive actions to *cobots*, they can be freed-up for more essential duties.

Additionally, the use of collaborative robots in industrial processes is proven to be beneficial considering the fact that they can be managed and taught through more intuitive interfaces. Instead of relying on a programmer to tell them what to do, *cobots* are often taught by example. An operator physically moves the robot thanks to a lead-through control, running it through the necessary tasks keypoints. The robot can then remember different tasks and perform them again and again with perfect recall and execution. On the contrary, traditional industrial robots require expert engineers to program them [4].

The introduction of *cobots* in the production lines not only improves the efficiency, but also provides multiple benefits to the workers. In some instances, these machines take over when tasks are not ergonomic, relieving the operator from fatigue and bio-mechanical stress accumulation and protecting them from injuries. Moreover, the most repetitive tasks may be assigned to the robotic co-workers, reducing the alienation of the operator and improving job satisfaction.

Thanks to their flexibility and efficiency, *cobots* can be beneficial in a wide variety of operations. In particular, an important sub-set is constituted by Physical Human-Robot Interaction (*pHRI*) tasks, which includes the activities where humans enter in direct physical contact with the robots.
One example is represented by quality inspection operations, where the finished part is compared with the manufacturing specifications to verify the absence of defects and imperfections. These tasks usually involve multiple high resolution cameras to take pictures of the part for its analysis, or probes that must be positioned with high accuracy and held still during the tests. Potential mental fatigue can cause inspectors to miss problems, but collaborative robots can decrease their alienation, consequently improving the efficiency of this repetitive task and the overall quality of the batches.

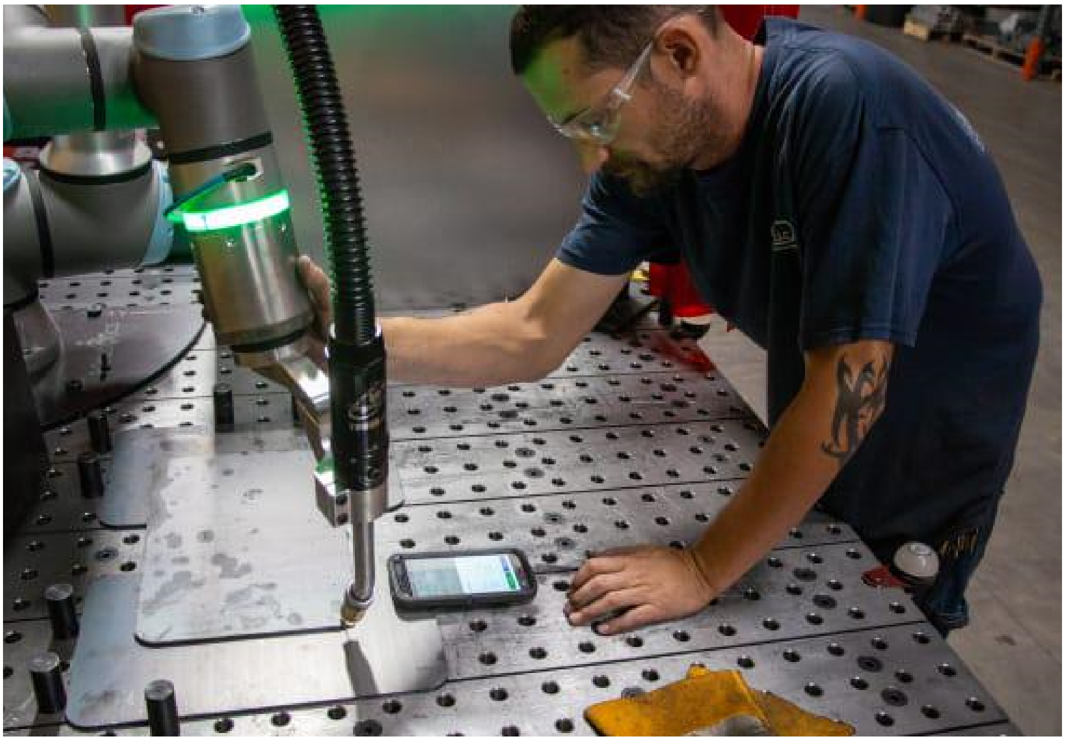


Figure 1.3: Example of a collaborative welding operation.

Moreover, *cobots* can be of particular use in manufacturing operations such as welding or polishing of parts. In fact, a completely automated process may not be able to handle errors from pre-machining and distortions induced by heat correctly. As it can be seen in figure 1.3, in a collaborative set-up the complexities are tackled by the human skills and flexible mindset, while the robot is responsible to relieve the operator from keeping the heavy-duty tools in position by hand, improving the task ergonomics and the precision of the welds.

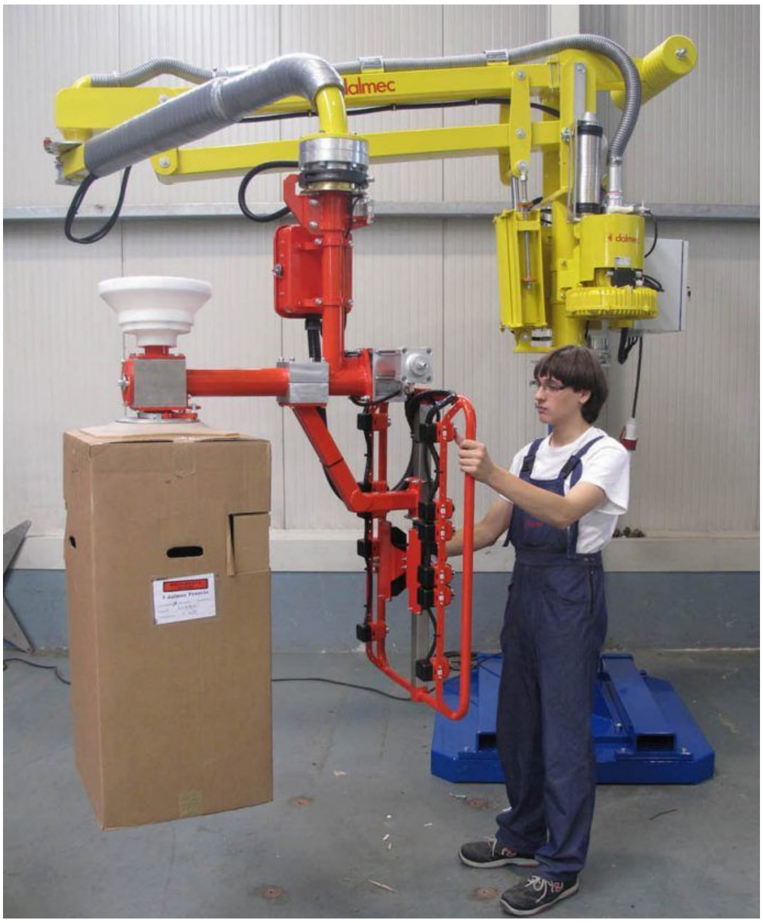


Figure 1.4: Example of a Manual Guidance Application.

Another important *pHRI* application is Pick and Place operations. These consist in the transportation of objects, such as raw materials for processing or finite parts for packaging and pallettizing, across the workspace. Manual transport jobs require a lot of repetition, that may often lead to mistakes and inefficiency along the production chain and can inflict workers with strains and injuries. A remarkable case study, portrayed in figure 1.4, is manual guidance: a task where the *cobot* and the human concur in moving heavy or bulky items, the types of which the worker might not be able to transport alone. In this case, the robot is responsible for the compensation of the object weight, while the operator plans the correct path to reach the unloading position. The limitations of a fully automatic system are overcome by the human smart thinking and adaptation capabilities, and the operator is relieved from fatigue and alienation deriving from this job. Collaborative manual guidance is the focus of this thesis.

1.2 State of the art

In the last 40 years, continuous hardware improvements, faster controllers and more affordable and accurate force measuring devices sped up the development of Physical Human Robot Interaction techniques. In particular, the recent drive towards collaborative working cells, which are able to improve the flexibility of the production chain, boosted innovation in the hand guidance sub-field. In this type of *pHRI* operation, the user manually interacts with the manipulator to transport an object across the working space. The robot has to deduce the intention of the user from the force exerted on the end-effector and comply with it. When a robot interacts with the environment (the human hand in this specific context), reaction forces occur and have to be managed in the correct way: failing to do so would result in an undesired behavior and eventually would damage the robot and hurt the operator. Classical position control is not suitable in this case, since the contact forces would cause a deviation from the nominal trajectory that the controller would try to compensate for, increasing their intensity instead of managing them. For this reason, a peculiar set of interaction control laws must be adopted.

Hogan [5] provided a vital contribution to interaction control with the introduction of the impedance control law. He demonstrated that, if the reaction forces between the robot and the environment are known, either by measuring them with a force/torque sensor or by estimating them, then it is possible to simplify the dynamics of the interaction to a set of decoupled impedance relations, each one representing a mass-spring-damper system.

There exists also a dual strategy, known as admittance control, that can be implemented to impose the desired impedance relation. Depending on the causality of the control problem at hand, an impedance or admittance control should be adopted. Impedance control extract a force reference from a displacement. It is more suitable to render lower inertia but it is sensible to disturbances and approximation errors in the manipulator dynamic model. It is commonly adopted in teleoperation tasks.

On the other hand admittance control [6] computes a displacement reference from the interaction force. As a result, it is possible to make the end-effector naturally compliant with the force exerted by the human, which is preferable in a hand guidance operation. The force applied by the operator is measured with a Force/Torque sensor placed on the end-effector, then, it is converted into a position reference, which is tracked by the fast low level position controllers of the robot. Such conversion is performed through an *admittance filter*, whose equation in Laplace domain is:

$$x_d(s) = \frac{1}{s(ms+d)} f_H(s)$$

where $x_d(s)$ is the new position reference, $f_H(s)$ is the force applied on the end-effector, m and d are the virtual inertia and damping perceived by the human during the motion. In such a case, the elastic term is not used, since in manual guidance applications there is not a fixed equilibrium position.

However, the adoption of an invariable admittance filter, meaning that the virtual mass and damping coefficients are constant during task execution, entails a trade-off between performance in the traveling phase and precision in final positioning. When the user accelerates or travels at cruise speed, a low damping factor is more suitable to lower the physical effort. On the contrary, when the user reaches the target position, the end-effector must slow down: in that case, a high damping value is preferable in order to improve his/her positioning accuracy.

It is straightforward to notice that, adapting the filter coefficients based on the working situation, improves the quality of the interaction and reduces the human effort required to execute the task. For this reason, various researchers focused on developing strategies, collected under the name of Variable Admittance Control, to estimate the user intention and vary the virtual parameters accordingly.

Ikeura et al. [8] proposed the adoption of two different admittance filters: a high virtual damping value is used for lower speed and a low damping parameter is applied when the absolute value of the velocity overcomes a certain threshold. The virtual inertia is kept at a low, constant value.

In [9] the damping coefficient is varied according to the velocity module. At low speeds, a constant high damping value is enforced, then it is decreased according to an inversely proportional law until the minimum value admissible for stability is reached. Additionally, high damping is enforced at high speed as a safety feature.

In [7], the cooperative characteristics of the human is modeled as a pure damping. Then, a set of experiments, where the operator has to drive a slider along a linear path, is performed. During such experiments, the time varying viscosity component is estimated through recursive least squares method. An optimal, time varying, damping equation is fitted to the obtained estimates. A variable admittance control strategy, based on the aforementioned optimal damping characteristic equation, is adopted for the cooperative transportation task. The outcomes show that the proposed optimal variable admittance approach produces a trajectory very close to the minimum jerk one.

In [10] the virtual damping is exponentially decreased based on the velocity. Moreover, different admittance filter configurations are tested: the best performances are obtained with a low, constant virtual mass and a variable damping.

Duchaine et al. [11] propose a variable admittance model based on the online estimation of the user intention. An increase in force magnitude in the direction of the velocity is interpreted as the desire to accelerate, hence the damping is lowered. On the contrary an increase in the opposite direction corresponds to a deceleration and the damping is increased. In [12] the cooperation task is subdivided into 4 sections based on a passivity index, which measures the amount of energy exchanged between the robot and the human. When this index crosses a predefined threshold, the movement beginning phase is detected and the mass and damping are minimized. Then, during the motion at cruise speed, the virtual parameter are kept constant and minimal. A negative passivity index signals the beginning of the deceleration phase, hence the damping is increased. Afterwards, if the average of the passivity index returns normal, the virtual mass and damping are reset for another movement.

In general, the interaction with the human arm is identified as an instability source for admittance controllers. Its behavior is often modeled as a pure stiffness. In [13] an admittance model is applied to control the interaction with a robotic exoskeleton. The stiffness of the arm is estimated from the derivative of the measured force, then it is used to evaluate an appropriate virtual damping, while the inertial component is kept at a minimum constant value.

In [14] and [15], some methods to detect the instability generated by the stiff contact with the environment are reported. When a deviation from the nominal behavior is detected, the virtual inertia is increased in order to regain stability.

In more recent studies, researchers proposed variable admittance techniques based on machine learning algorithms. In [16] and [17] a Fuzzy Model Reference Learning Controller is proposed, which is able to compute the correct virtual damping based on the input force and velocity. The models are trained according to a learning function which minimizes the error between the velocity and the minimum jerk velocity.

In [18] a multi-layer perceptron is trained with a series of repetitive movements in order to compute the correct admittance parameters in future hand guidance tasks.

An important contribution to interaction control in manual guidance operations is provided by Bazzi et al. [19]. For the first time, a geometrical interpretation of the admittance filter is presented and the variation of the virtual parameters according to the spatial direction of the applied force is examined. According to this new interpretation, Goal-driven variable admittance control can be used to provide a directional haptic feedback to the operator, leading him/her along a specific direction without the need of an explicit visual indication. This approach becomes fundamental in situations where the bulky size of the transported object blocks the operator field of view, like in the case depicted in figure 1.4. Thanks to the varying resistances perceived by the human during the motion, it is possible to passively drive him/her towards a specific target in the working space, even in case this destination is invisible.

However, the assumption that the motion happens in an empty environment limits the applicability of this approach. In fact, a realistic collaborative work cell contains furniture, machines and other obstacles that prevent the user from following simple paths. It would be interesting and valuable to endow this new variable admittance control with obstacle avoidance capabilities in order to allow the human to reach even invisible target positions in such cluttered environment, overcoming the obstacles that lie behind the transported object, outside his/her field of view.

Obstacle avoidance is a fundamental problem of path planning. Over the years, many researchers proposed strategies for the automatic computation of collision-free paths, to allow traditional industrial manipulators to reach a specific target position in a cluttered workspace. Usually, the path planning problem is addressed in the configuration space (*C-space*) of the robot: for industrial manipulators, the common choice adopted is the joint space. However, solving the problem in *C-space* requires all obstacles to be mapped from the workspace to this new space, which usually is a complex and computationally expensive process. For this reason, its not uncommon to address the problem directly at workspace level, which is a low dimension space: a path is planned for the end-effector, then an optimization problem is solved to find a collision avoiding configuration for the rest of the robot arm.

In [20] the path is planned across different Tool Center Point (*TCP*) through an iterative algorithm: the kinematic inversion problem is solved as an optimization process which minimizes the squared distance from the starting position to the target. Additionally a Gaussian therm, representing the estimated risk of collision is added in the cost function. Then a new *TCP* is evaluated along the gradient of the objective function. This process is repeated to compute the complete path. In [21] path planning in the workspace is reinterpreted as a disjunctive programming problem, solvable with modern optimization algorithms. The joint positions at given time instants become decision variables and the Kinematic and Dynamic Constraints of the manipulator, as well as collision against obstacles, are treated as constraints for the problem. The average kinetic energy is assumed as the optimization function to be minimized.

In [22] an artificial potential field-based solution to the path planning problem in operational space is reported. Such field is obtained from the superposition of an attractive field propagating from the target position, and multiple repulsive fields propagating from the obstacles that are present in the workspace. To compute a complete path, an artificial force, defined as the negative gradient of the potential field, is evaluated and used to compute a new position for the end-effector. The artificial force can also be computed for any point belonging to the manipulator, preventing collisions of the rest of the robot structure. Unfortunately, this strategy does not always produce a valid solution, since it is sensitive to local minima of the potential field that can be originated from peculiar obstacles dispositions.

In [23] a countermeasure to the local minima of artificial potential fields is proposed. The problem is addressed with a two step algorithm: first the oscillations around the minimum are detected with an observer, then a virtual obstacle is placed in the direction opposite to the target in order to produce a repulsive field, erasing the minimum.

In [24] a hybrid strategy for path planning in the configuration space is proposed. At first Probabilistic Road Map, a sampling-based approach which is very efficient in high-dimensional spaces, constructs a graph of interconnected samples of *C-space*. Then Q-learning method is used to plan a path. An agent continuously attempts to find collision-free paths from the starting configuration to the final one. When a path is found, an Action value Q is assigned to each graph edge belonging to the path: higher Q corresponds to paths which are shorter and travel further from the obstacles. Then, the optimal path is selected as the one that minimize the sum of all Q values of the connections.

In [25] the end-effector path is computed using the Rapidly exploring Random Tree algorithm. The workspace free-space is randomly sampled to construct a tree, rooted at the starting position and growing progressively towards the objective. When one tree node is in proximity of the target, the algorithm terminates. The output is a sequence of nodes, representing a feasible path from the starting to the final position. Following the same principles Yuan et al.[26] propose heuristics to improve the efficiency of RRT algorithm: the probability of sampling the free-space in the target direction is increased and the new samples are projected towards the destination, hence fewer nodes are needed to reach it. Moreover, the Douglas-Peucker algorithm is used to reduce the number of control points obtained from RRT, then a final smooth trajectory is computed using B-spline fitting algorithm.

In [27] another sampling-based path planning strategy is presented. The *C-space* is evenly sampled according to an Halton sequence pattern. Then, a path from the starting configuration to the target is constructed using the A* algorithm. The adopted heuristic is a function of the distance from the objective, the total connection cost, and the mean value of the danger field along each connection.

In recent times, the development of machine learning techniques lead to new path planning solutions. As an example, in [28] obstacle avoidance is treated as a minimization problem, where the target end-effector pose and the manipulator kinematic chain are enforced as constraints. A genetic algorithm is used to solve it: a sequence of joint configuration is created, then the path is checked for collisions. If no collision happens, then the fitness function assumes a high value and the sequence is likely to be copied in the next generation, otherwise the fitness function return a low value and the sequence is discarded.

Moreover, in [29] a locally connected Neural network is used to represent the topology of the *C-space*. A positive excitation input is provided by the target position, while an inhibitory one comes from the obstacles. The dynamic state of the neural network is gradually updated with a gradient ascent approach, hence the inputs are propagated inside the map. At any time it is possible to produce the next configuration/position by finding the maximum state value between the connected nodes.

1.3 Thesis purpose

The purpose of this work is to develop a variable admittance control strategy for manual guidance in a cluttered working environment, where the presence of obstacles forces the operator to follow complex trajectories. The obstacle distribution and shape is considered known and fixed, but the operator field of view may be obstructed by the bulkiness of the transported cargo. Hence he/she acts based on a partial knowledge of the surroundings. The risk of collisions with the obstacles is high and the goal position is out of sight.

Through variable admittance, the developed approach must provide a haptic feedback to the operator about the correctness of the selected motion direction. This makes the operator aware of which are the available direction and which, instead, lead to collisions.

Moreover, it is desirable to introduce an adaptive behavior that tries to accommodate the human motion intention in order to reduce wasted energy and avoid collisions in case the human intention is slightly different from the optimal path determined by the robot.

The result will be a control algorithm that is able to help the human operator in directing towards an unknown target position with high accuracy, allowing him/her to overcome all the obstacles, minimize the collisions, and reduce the physical effort and the execution time.

1.4 Thesis contribution

In this thesis, the goal-driven variable admittance control [19], which is functional in obstruction-free spaces, is adapted to the navigation inside realistic working environments, where the presence of obstacles forces the user to follow complex paths. This form of assistance has its application field in situations where the user has to transport an object with large dimensions and/or inertia. In these cases the operator field of view may be partially or totally obstructed, hence it is easy for him/her to loose orientation: in the best case he/she will proceed slowly, wasting time and energy during the transportation task, in worst cases he/she may cause an accidental collision that would ruin the cargo. By applying this innovative type of variable admittance control it is possible to substitute the visual feedback, which is compromised, with a tactile one, provided by the variable resistances to motion of the end-effector.

The damping parameter of the admittance filter varies according to the misalignment between the input force and a defined optimal direction. Such direction is selected in real time according to the obstacles, the goal position and an estimate of the user intention. Minimal effort is perceived by the operator if he/she drives the end-effector along a feasible path to reach the target, while high fatigue would make him/her aware of the directions that lead to collisions. As a result, the human is led along a complex path that avoids all the obstacles.

Mutual adaptation strategies between the worker and the robot are able to reduce the execution time and the energy required for the operation. Moreover, since the intention of motion is kept in consideration, the user remains in full control of the transported object path: in case multiple routes to reach the target position exist, he/she is able to choose his/her preferred one, and the manipulator will comply with his/her choice and reduce the effort required, as long as the chosen path is feasible and collision-free.

A first approach to achieve this objective is based on the discretization of the space by multi-layers grid. This aims at describing the working space with higher resolution in correspondence of the obstacles and unsafe areas, and lower resolution in safe free-space zone. A strategy to select the local optimal direction (minimum effort one) is defined.

Due to the drawbacks shown by this first approach, another more powerful and general technique is developed. This exploits a tree-based description of the free space. A random tree is constructed from the goal position and covers the entire

working space, providing optimal reference paths that lead to the target position from whatever admissible position.
To evaluate the performance of the developed approaches, an experimental campaign was conducted, where multiple volunteers performed manual guidance tasks in collaboration with a Comau Smart Six manipulator. The outcomes of the experiments showed that the tree-based approach is able to overcome the drawbacks of the grid-based method, producing far better results in therms of intuitiveness of the haptic feedback, efficiency and collision avoidance. When tree-based variable admittance control was in use, the inexpert users always managed to complete the task, even in case of complex and articulated working environments, where the obstacles and the unloading position are invisible and unknown to the human. In 70% of the experiments, inexpert users managed to avoid all collisions, and the average final positioning error with respect to the target position was under $2\,cm$. Moreover, according to the post-task survey answers, all users found themselves comfortable interacting with the robot, and agreed on the effectiveness and intuitiveness of the developed control strategy.

1.5 Thesis structure

The rest of this thesis is organized as follows:

- **Chapter 2**: the working principles of admittance control are described in this chapter. At first, an overview on classical admittance control is provided. Then, the drawbacks of a constant parameter filter are highlighted and the importance of a variable strategy in manual guidance is remarked. Finally, goal-driven variable admittance control (GDVAC), which represents the starting point of this thesis, is presented.

- **Chapter 3**: the problem of manual guidance in a realistic working environment is presented, highlighting the limitations of the basic GDVAC solution. The focus is set on the importance of providing a clear directional feedback to the operator. Eventually, two modified damping shapes, that are able to improve the user sensibility, are presented.

- **Chapter 4**: all the preliminary concepts that are necessary to understand the proposed algorithm are presented. Then, the two developed approaches are analyzed in detail. The first one (grid-based) represents a more reactive strategy for manual guidance, while the second one (tree-based) is a more complex solution to the problem, also implying a global optimization on the planned trajectories. Afterwards, the complete control strategy is thoroughly described in all its steps.

- **Chapter 5**: the two developed variable admittance control algorithms are compared to the classical invariant one. At first, the experimental setup is presented. Then, three sets of experiments and their outcomes are analyzed to evaluate the effectiveness of the proposed strategies. The first set is executed in a completely known environment, where the operator is aware of the surroundings. In the second one, the map is hidden and the user has to rely on the feedback provided by the robot. The third set validates RRT-based manual guidance for the navigation in a complex workspace and investigates the relationship between the level of autonomy left to the operator and the quality of the interaction.

- **Chapter 6**: a final analysis of the developed approaches is provided. Moreover, possible future developments are discussed.

Chapter 2

Goal driven variable admittance control in manual guidance

2.1 Introduction

The vast majority of the robotics applications requires compliance between the robot and the environment that surrounds it. In this kind of situations the implementation of a pure position control, imposing a predefined trajectory to the robot, may lead to unstable behavior and, eventually, damages to the environment or the manipulator itself. This is due to the fact that a traditional robot controller does not make the manipulator compliant, namely it does not take care of the possible interactions with an external environment, but it has to follow a predefined position and velocity reference whatever happens. Therefore, if the robot, during the motion, enters in contact with the external environment, then it cannot track the predefined trajectory and, instead of becoming compliant modifying the nominal path, it continuously increases the actuation torques in order to overcome the obstacle by means of brute force.

In most of the insertion tasks, where the interaction between robot and environment consists in the insertion of a piece held by the robot inside a hole with small tolerances (*peg-in-hole* problem) a simple, passive approach can easily be a solution. A Remote Center of Compliance (*RCC*), is a mechanical device capable of deforming elastically; if applied at the end-effector of the manipulator, then it nullifies the disturbances originated from the environment irregularities by naturally complying with them.

Nevertheless, there are applications, such as machining or collaborative-robotics, where it's fundamental to quantify the energy and take into account the forces generated by the interaction between robot and environment.

Control approaches that consider interaction forces can be generally subdivided into two branches: *hybrid force-position* control and *impedance/admittance* control. In the first case, the degrees of freedom of the manipulator end-effector are well divided in position controlled and force controlled directions. The first ones have to follow a predefined trajectory while the second ones have to realize a desired reference force profile on the environment. This approach is best suited when a precise force profile is needed and a clear separation between position and force controlled directions is identifiable, like in a milling operation on a predefined working piece.

Instead, the impedance/admittance control does not make any distinctions among end-effector degrees of freedom and it does not allow to impose a predefined force profile on the environment. On the contrary, it just imposes a desired dynamic relation between the end-effector position error and the external forces acting on it in every direction. Usually, this desired dynamic relation is chosen as the one that links the position of a mass-spring-damper system with the external force exerted on it. The success of this relationship stands in its simplicity and intrinsic stability. This control method is also adopted in manual guidance operations, where the robot is used to lift a heavy object and the operator pushes the end-effector to move the cargo inside the workspace. In these cases it is mandatory to introduce a control technique that makes the manipulator compliant with whatever external force applied on the end-effector by the environment (corresponding to the human in manual guidance applications).

Specifically, impedance and admittance control, while based on the same general idea, are opposed concerning the causality between outputs and inputs of the filter they implement. In Impedance control, the end-effector position and speed displacements are converted into torque references for the motors, in order to simplify its motion to the one of a simpler mass-spring-damper system. However, this approach has many limitations: it requires low uncertainty in the robot model, perfect cancellation of its non-linearities and needs to access the motor torques in the controller which, usually, it is impossible for industrial manipulators. On the opposite, Admittance control filters the forces measured on the end-effector and generates compliant position displacements. These displacements can then be fed to the low level position controllers of the robot, avoiding the impedance control limitations while obtaining the same results. For these reasons, admittance control can be identified as the best suited control method for the framework proposed in this thesis, as it will be further described in the next chapter.

2.2 Introduction to admittance control

As mentioned in the introduction, the purpose of an admittance controller is to simplify the dynamics of a system interacting with the environment: the resulting dynamic is the same as a mass-spring-damper system, which naturally complies to external forces with a damped harmonic type of motion.
To explain how this method can be applied on a manipulator, it's useful to make an example with a simpler model and then generalize on the full robot dynamic model.

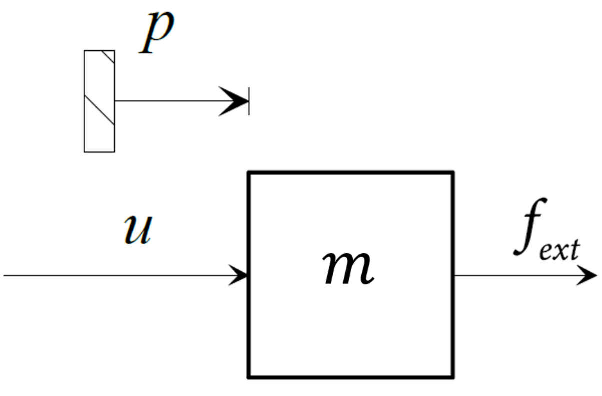


Figure 2.1: Representation of a 1 degree of freedom system.

The system depicted in figure 2.1 consists of a mass m subjected to an external force f_{ext} and a control action u. Its dynamic equation is:

$$ma = u + f_{ext} \tag{2.1}$$

where a represents the acceleration of the mass m.

If we apply a control law obtained from a linear combination of a proportional action k_1 on the speed v and k_2 on the position p of the mass:

$$u = -(k_1 v + k_2 p) \tag{2.2}$$

the new dynamic equation, obtained by substituting 2.2 in 2.1, becomes:

$$ma + k_1 v + k_2 p = ma + d_d v + k_d p = f_{ext} \tag{2.3}$$

which can be interpreted as a mass m attached to a spring $k_d = k_1$ and damper $d_d = k_2$, subjected to an external force f_{ext}, with the stiffness and damping parameters completely tunable.

If, additionally, a force measurement is available, the control law can be modified as:

$$u = \frac{m}{m_d}(-k_d p - d_d v + f_{ext}) - f_{ext} \tag{2.4}$$

where m_d represent an arbitrary virtual mass coefficient, and the following equation is finally obtained:

$$m_d a + d_d v + k_d p = f_{ext} \tag{2.5}$$

where it's also possible to tune the mass parameter m_d.

Equation 2.5 defines a mechanical impedance relation between the position of the mass and the external force applied on it.

To generalize this approach to a manipulator, the generic robot dynamic model must be introduced:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\boldsymbol{q} = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h} \tag{2.6}$$

were $\boldsymbol{q}$ is the vector of the joint coordinates, $\boldsymbol{B}$ is the inertia matrix, $\boldsymbol{C}$ represent the centrifugal and Coriolis effects, $\boldsymbol{g}$ represents the gravitational term, $\boldsymbol{\tau}$ contains the actuator joint torques and $\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}$ represent the effects of the external wrench $\boldsymbol{h}$ (forces and moments) applied to the end-effector, mapped on the joints torques through the geometrical Jacobian $\boldsymbol{J}$.

Implementing the so called inverse dynamic control law in the operational space of the manipulator:

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})\boldsymbol{q} + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h} \tag{2.7}$$

with desired joint acceleration $\boldsymbol{y}$ selected as:

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}\boldsymbol{M_d}^{-1}(\boldsymbol{M}_d\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} - \boldsymbol{M}_d\dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{h}_A) \tag{2.8}$$

where $\boldsymbol{M}_d$, $\boldsymbol{D}_d$ and $\boldsymbol{K}_d$ are diagonal positive definite matrices representing the desired virtual inertia, damping and stiffness for the end-effector motion, $\boldsymbol{J}_A$ is the analytical Jacobian of the robot, $\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}$ represents the Cartesian position error of the robot end-effector and $\boldsymbol{h}_A$ is the wrench expressed in the global reference system (base of the robot), the dynamic model of the robot simplifies in:

$$\boldsymbol{M}_d\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{D}_d\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_d\tilde{\boldsymbol{x}} = \boldsymbol{h}_A \tag{2.9}$$

The result is constituted by a set of six decoupled equations representing monodimensional impedance relations corresponding to simple mass-spring-damper systems. In particular, the first three equations correspond to *translational impedance*, meaning that the error $\tilde{\boldsymbol{x}}$ is the difference between the end-effector position in the operative space $\boldsymbol{p}$ and the desired position $\boldsymbol{p}_{ref}$: the translation of the end-effector is bonded to the forces acting on it in each component of the Cartesian space. The other three equations correspond to *rotational impedance*, where $\tilde{\boldsymbol{x}}$ stands for end-effector orientation error expressed in a minimal representation and $\boldsymbol{h}_A$ embodies the torques exerted at the robot end-effector.

The framework of this thesis is manual guidance operations where the end-effector is guided inside the working space from a predefined starting position to the unloading one, and no constraint is enforced on its orientation. Therefore, the focus, from now on, will be only on *translational impedance*.

As mentioned before, due to the impossibility of a direct access to the motor torques of the manipulator, instead of generating a force reference it's more

convenient to exploit the low-level position controllers provided by the robot manufacturer. This is achieved with an *Admittance Filter*, whose equation is obtained from the causality inversion of the impedance relation 2.9. The external force applied to the end-effector is measured or estimated, then through the admittance filter a reference displacement is generated. This reference is fed into the low-level controllers, resulting in the manipulator interacting as a mass-spring-damper system.
This control method takes the name of *Admittance Control* (since the admittance is the inverse relation of the impedance) and it is portrayed in figure 2.2.

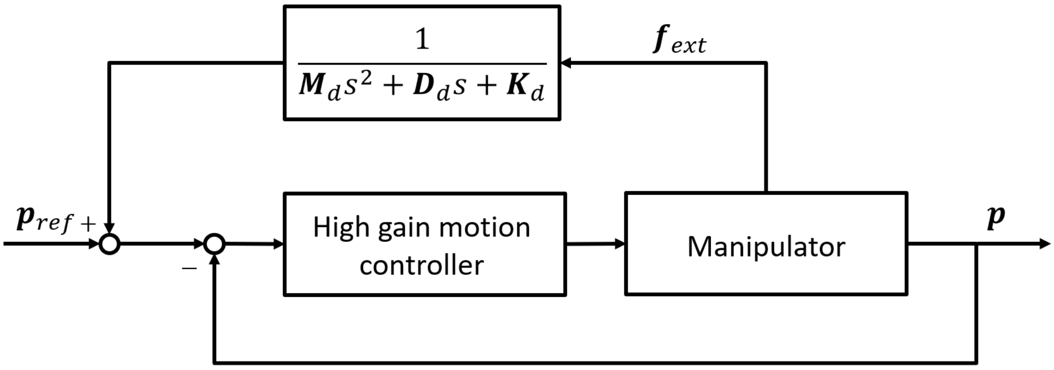

$\frac{1}{\boldsymbol{M}_d s^2 + \boldsymbol{D}_d s + \boldsymbol{K}_d}$
$\boldsymbol{f}_{ext}$
$\boldsymbol{p}_{ref}$
+
+
−
High gain motion controller
Manipulator
$\boldsymbol{p}$

Figure 2.2: Admittance control: block scheme representation.

In a manual guidance task, the lack of a predefined trajectory, namely of a reference position $\boldsymbol{p}_{ref}$, makes the virtual stiffness parameter $\boldsymbol{K}_d$ useless. Therefore, the admittance filter equation in Laplacian domain reported in figure 2.2 can be simplified to a first order filter generating a speed reference, $\boldsymbol{v}_{ref}$:

$$\boldsymbol{v}_{ref} = \frac{\boldsymbol{f}_{ext}(s)}{\boldsymbol{M}_d s + \boldsymbol{D}_d} = \frac{1/\boldsymbol{D}_d}{\frac{\boldsymbol{M}_d}{\boldsymbol{D}_d} s + 1} \boldsymbol{f}_{ext}(s) = \boldsymbol{H}(s)\boldsymbol{f}_{ext}(s) \tag{2.10}$$

Moreover, it can be observed that the virtual damping coefficient affects the steady state value of the response, while the virtual mass to damping ratio influence the bandwidth of the filter.

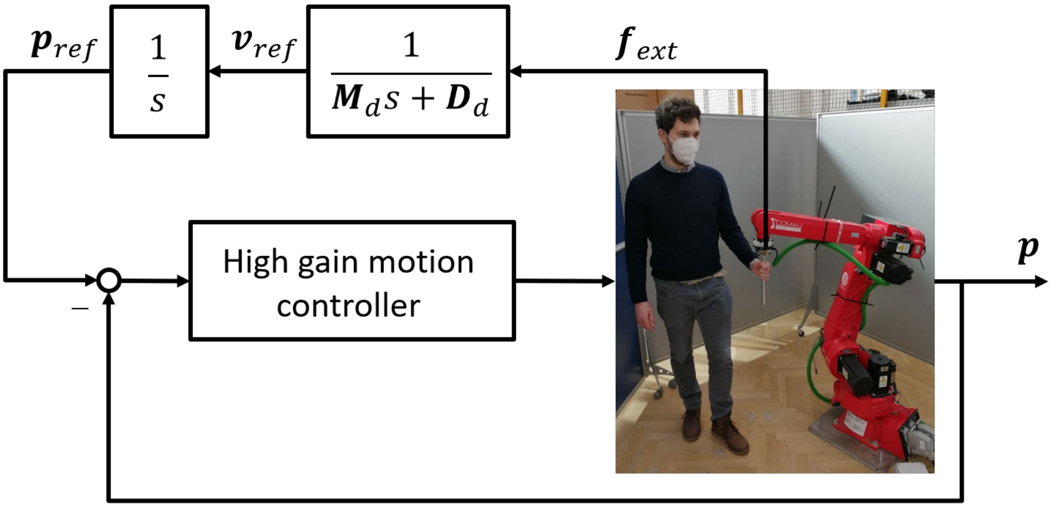


Figure 2.3: Simplified admittance control scheme, adopted in manual guidance.

Focusing on the *translational impedance* concept, each Cartesian direction is represented with a decoupled impedance relation: the dynamic of the displacement in that direction is dominated only by the corresponding component of the force applied at the end-effector. Therefore, an independent Admittance filter could be implemented for each component, but their parameters should be selected equal in order to preserve a natural engagement between human and robot. The three separate components can be rewritten in a unique equation in vector form:

$$\boldsymbol{v_{ref}} = \frac{1}{m_d s + d_d} \boldsymbol{f} \tag{2.11}$$

where $\boldsymbol{f} \in \mathbb{R}^3$ represents the force applied by the operator, and $\boldsymbol{v_{ref}}$ is the resulting end-effector speed reference.
From this relation it is possible to notice that, when an admittance filter is used, the output is scaled with respect to the input, but the direction is left unchanged. Hence, the velocity reference obtained is aligned with the direction of the input force.

2.3 Variable admittance control

Considering a *manual guidance* task, where an operator gently guides the end-effector towards the unloading station, it is fundamental that the movements of the robot indulge the intention of the human. The resulting interaction would feel more natural and produce better results in terms of transportation time, accuracy and spent energy.

Until now, an admittance filter with constant coefficients was considered: the dynamics of the end effector is reduced to the one of a body with fixed inertia m that is decelerated according to a predefined damping d during its motion. Small virtual parameters entail a low effort and a reduced transportation time, but do not provide the user with accuracy capabilities. On the contrary, high parameters allow for precise positioning motion but induce high effort and large completion time. Therefore, it is not convenient to adopt constant parameters, but it is better to online change their values in an appropriate way during the different phases composing the task, in order to facilitate the human in accomplishing each of them.

It is possible to identify three sequential phases of the motion, depicted in figure 2.4: first the body, which is idling at zero velocity, is accelerated until it reaches a certain cruise speed, then the human drives the cargo at that speed across the workspace and, finally, near the destination, a deceleration phase concludes the movement, followed by small adjustments to precisely place the transported cargo. In the acceleration phase it is convenient that the operator feels a light mass since the inertial effects prevail over the damping: with a small effort the transported object quickly reaches cruise speed. During the second phase, corresponding to motion at cruise speed, the damping parameter must be reduced in order not to waste energy, while the mass is not important due to the lack of meaningful accelerations or deceleration. Finally, in the third phase, which corresponds to a deceleration in the unloading position proximity, a high damping factor facilitates the deceleration of the transported load and a low mass is advisable, since otherwise it would counteract the braking effect with inertia. Additionally, high damping allows for more precise and slow movements, necessary to reach the positioning goal of the cargo.

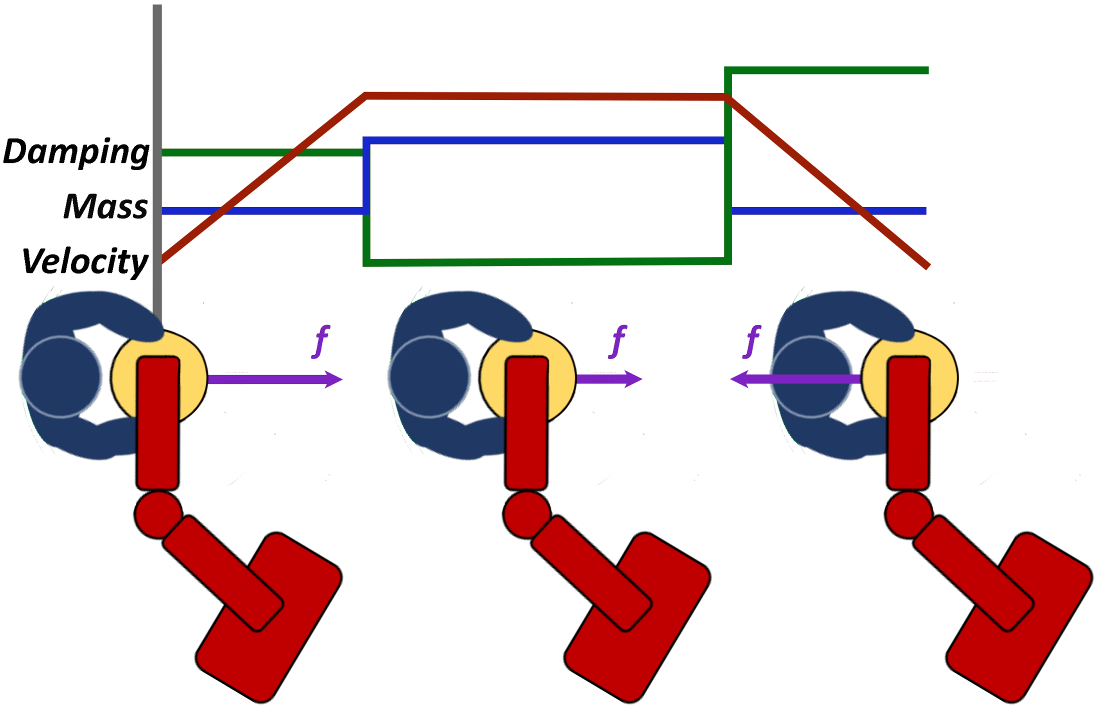


Figure 2.4: Qualitative representation of the three phases of a manual guidance task and the coefficients values prescribed by the literature.

As mentioned before, manual guidance is not a new, nor unaddressed issue in the literature: many researchers presented possible solutions to enable variability of the admittance filter parameters: some introduced hard thresholds on the speed and accelerations in order to apply different filter configurations for different phases of the motion, others proposed estimators in order to predict user intentions and vary the mass and damping accordingly. Only recently, the dependency of the admittance filter coefficients on the spatial direction of human force has been investigated. Indeed, a new physical interpretation of the admittance filter vectorial equation presented in 2.11, and in particular of its parameters, was introduced in [19].

If an optimal motion direction is defined, then the inertial and damping factors can be modified according to the angle between such optimal direction and $\boldsymbol{f}_{ext}$, representing the force exerted by the operator on the end-effector. If the best direction and force vectors are aligned, then the user is traveling in the correct direction, and the virtual parameters should be low, easing his/her motion. However, if the two vectors are opposite, then the user is proceeding in the wrong way, and the admittance coefficients should be high, providing him/her a haptic feedback. Moreover, the virtual parameters rate of change between minimum and maximum value should be a continuous increasing function of the misalignment angle, in order to guarantee a smooth and intuitive interaction between the human and the robot.

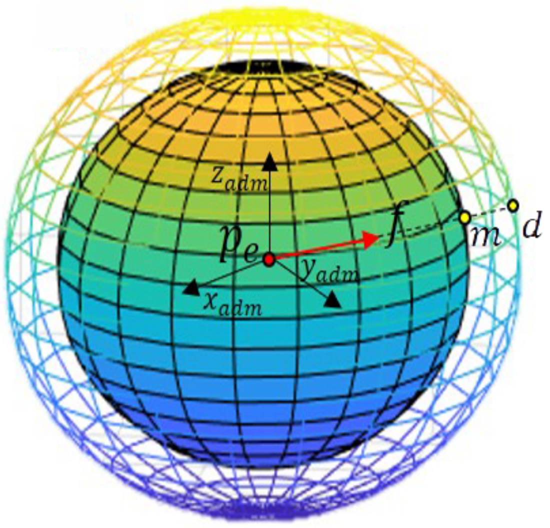


Figure 2.5: 3D Geometrical interpretation of the traditional Admittance control with constant virtual parameters.

Based on the proposed angular dependency, each parameter of a constant admittance filter can be seen in the Cartesian space as a geometrical sphere centered in the end-effector with radius equal to its constant value. This translates the fact that the value of each parameter is equal in any spatial direction (sphere shape) and constant. Generally speaking, the value of each parameter can be seen as the distance between the center of the sphere and the point of intersection between a semi-line directed along the applied force and the sphere itself, as it can be seen in figure 2.5.

Furthermore, whatever variable admittance technique in literature can be seen as a strategy that changes in time the radius of such spheres based on estimators or hard thresholds. Again, the coefficients will be left unchanged regardless of the direction towards which the operator is pushing the end-effector.

However, if the virtual mass and virtual damping spheres are not only modified in radius, but also in shape, the effect of the filter varies also with respect to the direction of the human force. This paves the way to more complex variable coefficients strategies that makes the admittance perceived by the human force direction dependent. In this way, admittance also assumes a Cartesian space direction dependency (shape): for example the virtual parameters can vary with respect to the relative position of the end-effector and the unloading zone. This method takes the name of Goal driven Variable Admittance Control and it will be thoroughly described in the next section.

2.4 An evolution of variable admittance control: goal driven variable admittance control

In the previous section, a new parameter interpretation of both invariable and variable admittance control approaches has been introduced. The virtual mass and virtual damping parameters are associated to two spheres centered at the end-effector position. The actual value of each coefficient is computed as the distance between the center of the sphere and the intersection point between the sphere itself and a semi-line directed as the human force vector.

As mentioned before, it's possible to implement parameters variability, intended in a classical way, by varying the radius of the two spheres based on the intensity of the force, velocity and accelerations of the end-effector or estimators of the user intention. However, if the shape associated to the parameters is changed, their value will also vary with respect to the direction of the input force. This implementation becomes particularly useful in situations where a bulky cargo is being transported: the size of the object can obstruct the view of the operator, hence, some kind of directional cues about the best route towards the destination position is required.

This is the main idea behind the Goal Driven Variable Admittance Control strategy presented in this section: the spheres representing the mass and damping parameters are converted into different shapes that allow to perceive the correct motion direction. In [19], the spheres are substituted with ellipsoids, which have three orthogonal axes of symmetry of independent length and arbitrary spatial orientation. The ellipsoids are centered in the end effector position. The first axis is directed as the connecting line between the end-effector and the goal position (unloading station) and has the minimum length, while the other two axis have much higher values and belong to the plane normal to the first axis.

This way, the more aligned is the operator force with the objective, the smaller will be the virtual mass and damping parameters of the admittance filter and, consequently, the lower will be the energy demanded to guide the robot. Therefore, the operator would perceive the direction towards the goal, i.e. the first axis of the ellipsoid, as the minimum effort one.

The result of this natural feedback method will be a more precise reaching of the goal position, even if the bulkiness of the transported object prevents the operator to clearly see the destination. Since the work presented in this thesis is an evolution of the Goal Driven Variable Admittance Control approach, a more detailed explanation of this method, complete with mathematical description, will be given in the following subsections.

2.4.1 Basic goal driven variable admittance control

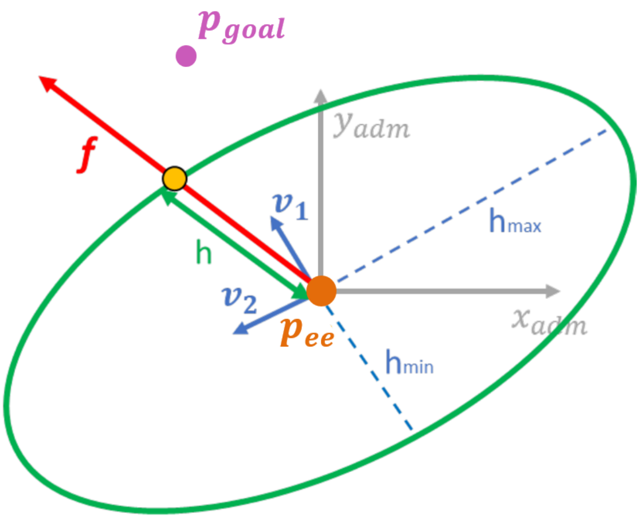


Figure 2.6: Representation of the ellipsoid that defines the parameters space dependency.

In the approach presented in [19], the shape representing the spatial relationship of the virtual mass and damping parameters is modeled as an ellipsoid, centered at the end-effector position and oriented according to the target, like the one depicted in figure 2.6.

A generic ellipsoid is defined as a set of points $\boldsymbol{x} \in \mathbb{R}^3$ satisfying the following equation:

$$\dot{\boldsymbol{x}}^T \boldsymbol{A} \dot{\boldsymbol{x}} = 1 \tag{2.12}$$

where $\boldsymbol{A}$ is a 3x3 positive definite matrix whose eigenvectors $\boldsymbol{v_i}$ represent the principal axes directions and whose eigenvalues λ_i stand for the square inverse of the semi-axes length. In the Goal-Driven strategy, the eigenvectors are chosen according to the previous reasoning:

- the direction of the first main axis, $\boldsymbol{v_1}$, is set coincident to the direction of the line connecting the actual position of the end-effector $\boldsymbol{p_e}$ with the goal position $\boldsymbol{p_G}$;
- the direction of $\boldsymbol{v_2}$ is set perpendicular to $\boldsymbol{v_1}$, in such a way that it belongs to an horizontal plane parallel to the ground;
- the direction of $\boldsymbol{v_3}$ completes a Cartesian frame, according to the right-hand rule.

Hence, the directions of the three main axes are computed as:

$$\begin{cases} \boldsymbol{v_1} = \dfrac{\boldsymbol{p_G} - \boldsymbol{p_e}}{||\boldsymbol{p_G} - \boldsymbol{p_e}||} \\ \boldsymbol{v_2} = \dfrac{\boldsymbol{z} \times \boldsymbol{v_1}}{||\boldsymbol{z} \times \boldsymbol{v_1}||} \\ \boldsymbol{v_3} = \boldsymbol{v_1} \times \boldsymbol{v_2} \end{cases} \tag{2.13}$$

where $\boldsymbol{z}$ is the z-axis and all quantities are expressed in the global reference system.

The three eigenvectors can then be arranged in a 3x3 matrix $\boldsymbol{Q}$, in such a way that the i-th column of $\boldsymbol{Q}$ is $\boldsymbol{v_i}$.

Since $\boldsymbol{Q}$ contains three orthogonal and unitary vectors, reporting the direction of the three main semi-axis of the ellipsoid, it also represent the orientation of such ellipsoid with respect to the global reference system. Furthermore, even though the two parameters of the admittance filter require two different ellipsoid, the rotation matrix $\boldsymbol{Q}$ can be reused for both, since the the two shapes are oriented in the same way.

The length of the first semi-axis (the desired minimum effort direction) is set to the minimum value of the corresponding parameter, while the length of the other two axes is set to a much higher value. In order to set the specified lengths of the semi-axes for the generic admittance parameter h (with h referring to both m and d), that can assume values between h_{min} or h_{max}, the eigenvalues of matrix $\boldsymbol{A}$ are chosen. According to the rules stated before, the eigenvalues are selected in the following way:

$$\begin{cases} \lambda_1 = \dfrac{1}{\left(h_{min}\right)^2} \\ \lambda_2 = \dfrac{1}{\left(\bar{h}\right)^2} \\ \lambda_3 = \dfrac{1}{\left(\bar{h}\right)^2} \end{cases} \tag{2.14}$$

where $\bar{h}$ is a value $\in [h_{min}, h_{max}]$ much higher than h_{min}. These eigenvalues can be collected inside a 3x3 diagonal matrix $\boldsymbol{\Lambda}_{\boldsymbol{h}}$.

The latter matrix represents, as it is, an ellipsoid with each semi-axis having the desired length, but directed as the global reference system axes. To rotate the shape accordingly we must apply the following transformation:

$$\boldsymbol{A}_{\boldsymbol{h}} = \boldsymbol{Q}\boldsymbol{\Lambda}_{\boldsymbol{h}}\boldsymbol{Q}^{-1} = \boldsymbol{Q}\boldsymbol{\Lambda}_{\boldsymbol{h}}\boldsymbol{Q}^{T} \tag{2.15}$$

Hence, two matrices, $\boldsymbol{A}_{\boldsymbol{m}}$ and $\boldsymbol{A}_{\boldsymbol{d}}$, representing the two ellipsoids corresponding to the two virtual parameters of the admittance filter can be obtained.

At each time instant, the ellipsoid orientation is updated and the value assumed by the virtual mass and damping coefficients is calculated as the distance from the center of the ellipsoid to the interception point between the surface of ellipsoid and the line through the end-effector position, directed as the input force vector. This is numerically evaluated, for the generic parameter h, as:

$$h = \frac{1}{\sqrt{\boldsymbol{u}_f^T \boldsymbol{A}_h \boldsymbol{u}_f}} \tag{2.16}$$

where $\boldsymbol{u}_f$ is the unit vector directed as the input force $\boldsymbol{f}$ and $\boldsymbol{A}_h$ is the matrix defining the ellipsoid.

To sum-up the described method, two ellipsoid have been computed, one corresponding to the virtual mass and the other to the virtual damping; each one has the first semi-axis directed towards the objective and has length equal to the minimum value assumed by the corresponding parameter. The other spatial directions determine higher values of the parameters. Therefore, if the operator exerts a force directed towards the destination position, then the filter parameters assume the lowest admissible value: this direction will be perceived as the minimum effort one. Otherwise, if the operator pushes sideways, the energy required for the motion increases accordingly.

The control algorithm provides a natural feedback sensation to the user, who becomes able to perceive which directions of motion would bring him/her closer to the unloading zone, even if his/her field of view is obstructed by the transported cargo.

2.4.2 Damping ellipsoid

A problem arises with the basic ellipsoid shape introduced in Section 2.4.1: due to its symmetry, the user would perceive minimum resistance both in the direction towards the goal and in the opposite way. Therefore, a modification to the shape must be introduced.

According to the literature (see Section 2.3 and figure 2.4), a low mass is desirable both in the acceleration phase, when the operator is pushing towards the goal, and in the deceleration phase, when he/she is pulling in the direction opposite to the goal, since a high velocity change can be obtained with a little effort. Instead, a low damping value is preferable when the operator has to start the movement and reach cruise speed relatively fast, while in the final phase a high damping coefficient is required, both to aid the human in braking the heavy load and to ensure a more accurate positioning.

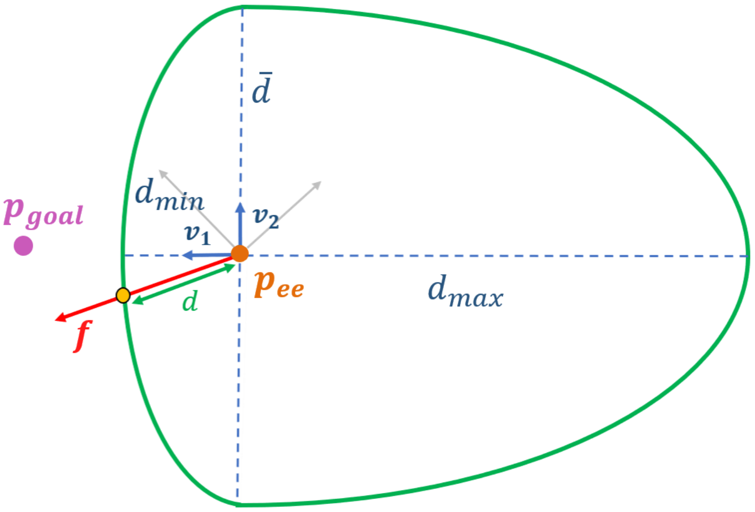


Figure 2.7: Representation of the improved shape for the damping constituted by the union of two half-ellipsoids.

In order to obtain such behavior, the virtual damping ellipsoid $\boldsymbol{A}_d$ has to be defined in a different way with respect to the basic one presented in the previous subsection. The new definition must ensure that at the increasing of the angle between the human force and the minimum effort direction, the damping also monotonically increases. The new shape is obtained composing two half-ellipsoids, divided by the plane identified by $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ as depicted in figure 2.7.

If the half space containing the first semi-axis unit vector $\boldsymbol{v}_1$ is considered, no change is required with respect to the method presented in Section 2.4.1. Whereas, concerning the other half space, a new ellipsoid is constructed with an increased length value for the first semi-axis. The second half differs from the first one only in the length of the first axis of the ellipsoid, which is selected as d_{max} instead of d_{min}.

The result is an asymmetric but continuous shape, having the same advantages stated before if the input force is directed towards the objective and the additional braking effect if the force is exerted in the opposite way. Moreover, thanks to this deformation, the minimum effort direction becomes unique.

Chapter 3

Goal driven variable admittance control: a new damping shape

3.1 Adapting to a realistic and articulated working environment

In a manual guidance task, the operator moves a heavy load inside the working area with the aid of a manipulator. In the near future, the most organized and technological working area will have no barriers but just flexible working isles. In case of production of heavy items, the working isles will be interconnected by means of larger manipulators helping the operators in handling the semi-finished products. The workspace will contain pallets, shelves, machinery, and the path from the starting position to the unloading zone could be complex, and so prone to possible collisions.

The goal-driven variable admittance control (GDVAC) approach introduced in Chapter 2 solves the problem of guiding the human towards the goal position, but it assumes that the motion happens in an obstacle-free environment. In an ideal case this method is a good solution, but it would fail when applied to highly structured working environments. Indeed, consider the workspace depicted in figure 3.1, where a single pallet is placed between the start position $\boldsymbol{p}_{start}$ and the goal one $\boldsymbol{p}_{goal}$: it's straightforward to notice that following a linear path would lead to a collision. Since the direct path towards the objective is obstructed, the operator has to work around the obstacle. However, finding a path that avoids all collisions is not a simple task.

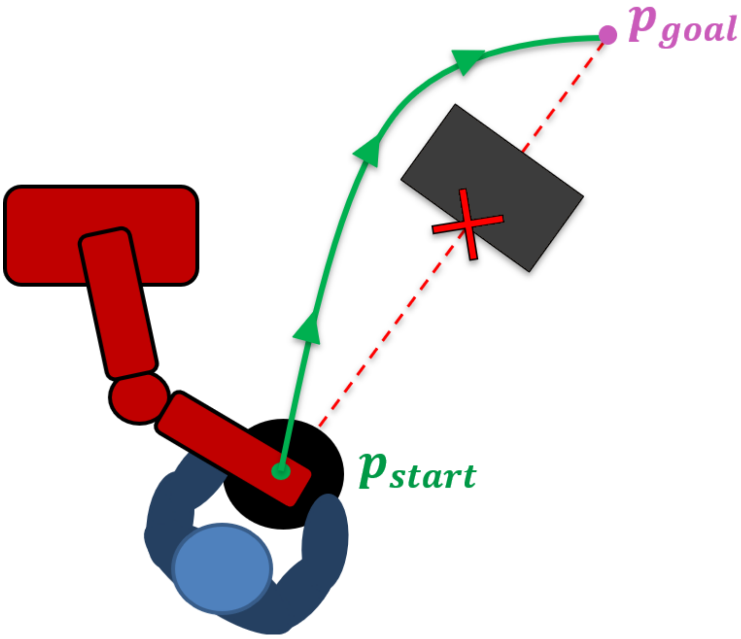


Figure 3.1: Example of a realistic workspace: a box obstacle is placed between the starting position and the objective.

In conventional manual guidance, the path planning task is assigned to the human: however, the framework of this thesis consists in scenarios where the operator field of view could be partially or totally obstructed by the bulkiness of the transported object. Size, position and shape of the obstacles that are hidden behind the lifted cargo are unknown, hence a totally human-based guidance becomes very dangerous and highly stressful. The operator needs some kind of assistance to navigate in the working environment and reach the unloading station in a safe way without any kind of collisions.

In order to preserve manual guidance core principles, the robot should not provide an active support, meaning that it can not move autonomously in the workspace, but at most it can decide to stop the motion in case of an imminent collision. The aid, instead, should be provided in the form of a intuitive directional haptic feedback: the main idea is that the user will be encouraged driving along the optimal path towards the goal position and penalized when his/her actions lead to a collision.

If basic GDVAC was in use, the human would perceive minimum effort when the cargo is pushed directly towards the objective and higher motion resistance in every other direction. Therefore, the operator would follow the shortest path, namely the linear one, towards the goal and would collide with the pallet.

Moreover, the user has no feedback on how far the end-effector should be moved sideways to clear the obstacle safely. He/She might chose to move quickly and based on his/her memory or to proceed with caution and have a look behind the cargo: in the first case the risk of an accidental collision with the obstacle is elevated, in the latter case the movement is slowed down and more time is required to reach the destination.

Therefore, it is evident that the basic GDVAC has to be generalized in case of realistic working environment where some obstacles may exist. The control algorithm has to be endowed with some Obstacle Avoidance strategy. This is the main core of this thesis, i.e. the adaptation of goal-driven variable admittance control to a highly structured working environment. The physical interpretation of the admittance filter presented in Chapter 2 will be enriched with path planning algorithms. In particular, the selection of the minimum effort direction and the design of the damping shape will be rephrased and generalized. The operator will drive the robot end-effector in complex environments, avoiding the obstacles even with closed eyes. In this way, the developed strategy would be effective even in case his/her view is obstructed, due to the dimensions of the transported object.

3.2 Improving the human perception of the optimal motion direction

Considering the problem of manual guidance in presence of obstacles, it is fundamental to point out to the operator which is the best motion direction towards the objective and which are the ones leading to a collision. The human perception and its "resolution" are established by the shape of the admittance parameters. In GDVAC [19], an ellipsoid shape is adopted: it represents the spatial dependency of the damping value of the admittance filter in function of the alignment between the minimum effort direction and the direction of the human force exerted on the end-effector. Minimum effort, i.e. minimum damping d_{min}, is perceived by the user when he/she pushes along the optimal direction, while higher damping factor is selected when the user pushes in the opposite direction or sideways.

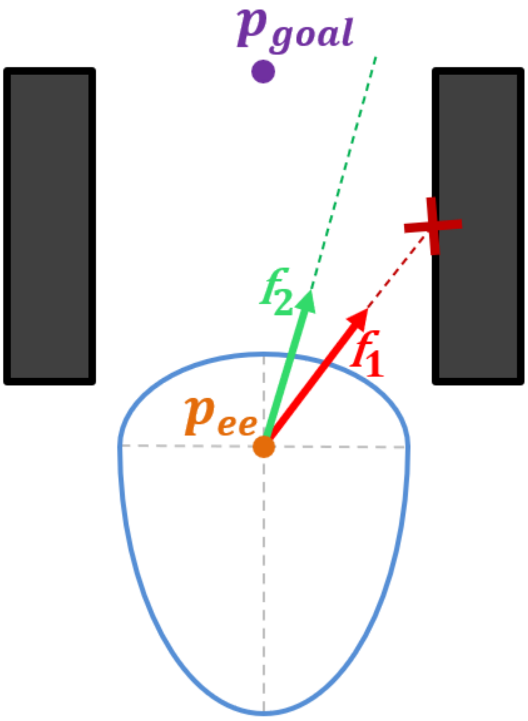


Figure 3.2: Example of a scenario where two obstacles form a corridor.

However, empirical results highlights that, in case of the ellipsoid shape previously described, human fails to clearly distinguish the optimal direction from the other ones in its neighborhood (of almost 30° for each side). This means that the ellipsoid adopted in [19] does not provide a very precise directional haptic feedback. This is not imputable to the developed strategy but to the intrinsic human difficulty in perceiving the tiny and progressive damping differences provided by the ellipsoid around the optimal direction. This aspect is problematic in the obstacle avoidance scenario since it could lead the human to commit some collisions unconsciously. To better understand this concept, the environment portrayed in figure 3.2 is considered. Two obstacles form a corridor and the optimal path is a linear path towards the goal position.

Unfortunately, due to the ellipsoidal shape used to design the damping parameter, if the user pushes slightly sideways, he/she would not perceive a sufficient change in the damping value to make him/her aware of the misalignment. Therefore, he/she is probably going to continue pushing in the slightly wrong direction until a collision happens.

This issue cannot be solved by increasing the difference between the minimum and maximum damping values. Therefore, the damping shape must be modified to give the user a clearer directional haptic feedback about the optimal motion direction by means of a sharper damping variation around $\boldsymbol{v_1}$.

To achieve this purpose, the frontal semi-half of the damping shape (the one that includes the first semi-axis $\boldsymbol{v_1}$) is divided into three zones, depicted in figure 3.3, depending on the angle α with respect to the optimal direction ($\alpha = 0$ when the considered direction is aligned with the optimal one):

- *HIGH DAMPING ZONE* ($\alpha_{high} \leq \alpha \leq \frac{\pi}{2}$): a high damping factor is applied to increase operator effort.

- *LOW DAMPING ZONE* ($0 \leq \alpha \leq \alpha_{low}$): it includes the directions that are almost aligned to the principal axis $\boldsymbol{v_1}$, which coincides with the optimal direction, where the lowest damping factor is applied.

- *TRANSITION ZONE* ($\alpha_{low} < \alpha < \alpha_{high}$): this zone avoids the creation of a discontinuity between the other two zones that would result in an unpleasant haptic feedback, due to abrupt changes in the damping value instantly slowing down or easing the motion. The damping value is selected according to a continuous function $f(\alpha)$.

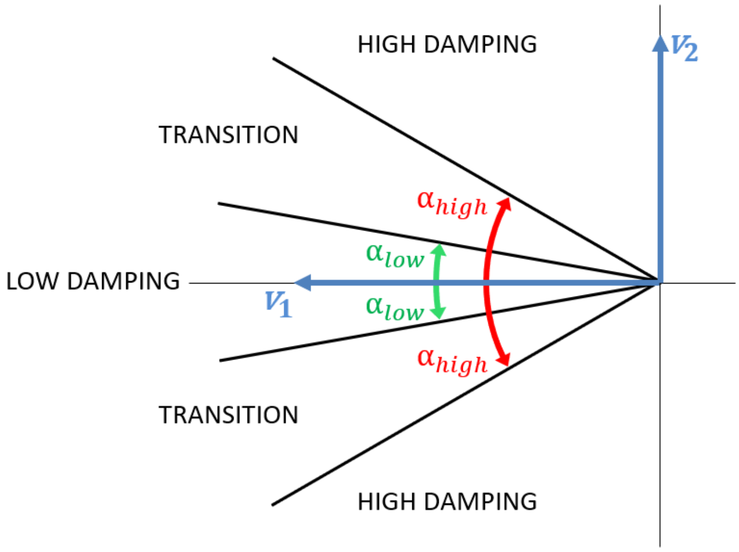


Figure 3.3: Representation of the three zone partition of the frontal semi-space.

Specifically, two types of modified shape, based on this space division are developed in the following subsections.

3.2.1 Concentric spheres

The first proposed solution adopts constant damping values inside the low and high damping zones: $d = d_{min}$ in the first one while $d = \bar{d} >> d_{min}$ in the second one. Since in these sectors the damping value does not depend on the angle α, its shape is represented by pieces of concentric spheres, as it can be seen in figure 3.4. Consequently, feedback to the operator is simplified: constant low damping is perceived if he/she pushes along the correct direction, while higher effort is experienced if the force is misaligned of a tunable angle.
A 3rd degree polynomial interpolation is used to select the damping value inside the transition zone:

$$d = f(\alpha) = c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3 \tag{3.1}$$

where the coefficients c_i are computed imposing the following initial and final conditions:

- $f(\alpha_{low}) = d_{min}$
- $f'(\alpha_{low}) = 0$

- $f(\alpha_{high}) = d_{max}$
- $f'(\alpha_{high}) = 0$

where f' is the first derivative of f with respect to alpha.

This is equivalent to solving the linear system $\boldsymbol{Ax} = \boldsymbol{b}$ with:

$$\boldsymbol{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 1 & \alpha_{low} & \alpha_{low}^2 & \alpha_{low}^3 \\ 1 & \alpha_{high} & \alpha_{high}^2 & \alpha_{high}^3 \\ 0 & 1 & 2\alpha_{low} & 3\alpha_{low}^2 \\ 0 & 1 & 2\alpha_{high} & 3\alpha_{high}^2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} d_{min} \\ d_{max} \\ 0 \\ 0 \end{bmatrix}$$

The established polynomial interpolation also ensures the continuity in the derivative of $f(\alpha)$: a smooth change is expected both when transitioning from the high damping zone to the low one, and vice-versa.

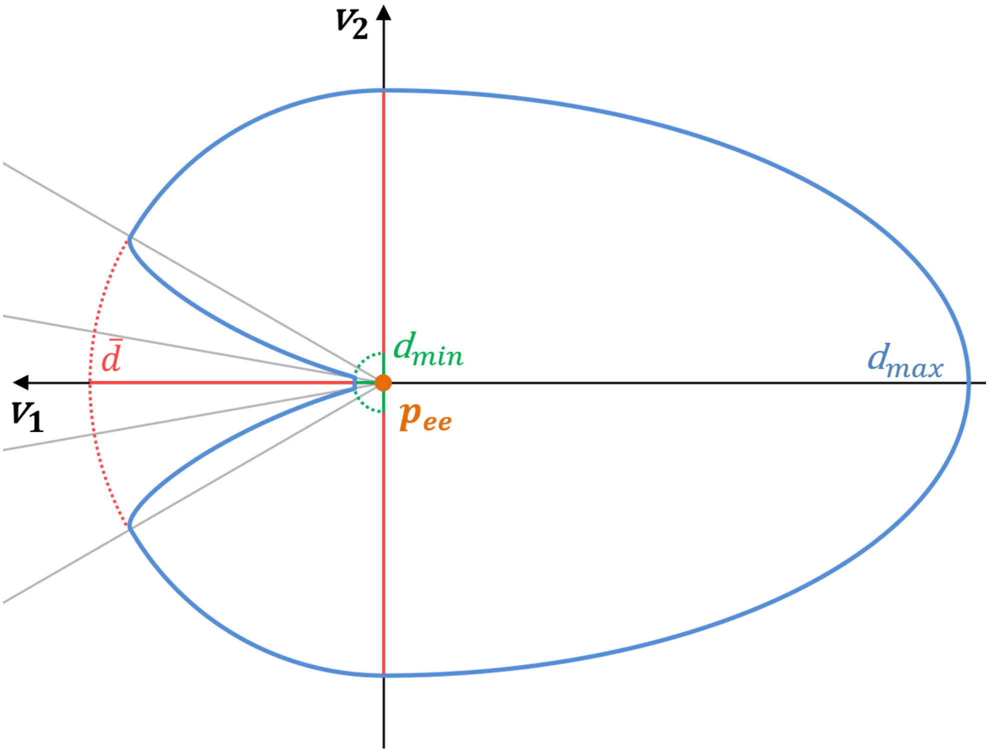


Figure 3.4: 2D representation of the modified damping shape (blue line) obtained from the combination of a high damping sphere (in red) and a low damping one (in green).

This first shape modification enhances the difference between correct and wrong directions. However, no directional feedback is possible inside the high

and low damping zones. In fact, the monotonic rate of change which is characteristic of the ellipsoid is sacrificed in spite of the improved distinction between the two zones.
As a result, this shape is more suited for beginner users, who needs a stronger distinction between advisable and not advisable motion directions. For more experienced users a continuous increase in the damping factor is preferable, since they will be able to perceive smaller differences and perform finer control.

3.2.2 Ellipsoids interpolation

The second proposed damping shape is obtained from the adoption of two different ellipsoids for the high and low damping zones, as portrayed in figure 3.5.
In particular, the ellipsoid adopted in the low damping zone has d_{min} as length of the first semi-axis and d_{high} as the length of the other two axes. Instead, the ellipsoid used in the high damping zone has length of $\boldsymbol{v_1}$ equal to d_{low} and size of $\boldsymbol{v_2}$ and $\boldsymbol{v_3}$ set to $\bar{d}$. The two ellipsoids matrices are evaluated using the same method presented in Chapter 2.
The damping values are chosen in such way that:

$$d_{min} < d_{low} <= d_{high} < \bar{d}$$

Regarding the transition zone, the damping value is obtained by means of an interpolation between the low and high effort ellipsoids. The interpolating ellipsoid is constructed using the following eigenvalues:

$$\begin{cases} \lambda_1 = \dfrac{1}{\left(d_{min} + (d_{high} - d_{min})\left(\frac{\alpha - \alpha_{low}}{\alpha_{high} - \alpha_{low}}\right)^n\right)^2} \\ \lambda_{2,3} = \dfrac{1}{\left(d_{low} + (\bar{d} - d_{low})\left(\frac{\alpha - \alpha_{low}}{\alpha_{high} - \alpha_{low}}\right)^n\right)^2} \end{cases} \tag{3.2}$$

where n is a positive integer exponent used to modify the slope of the transition. If n is set to 1, then a linear transition is implemented and the user feels a continuous growth between α_{low} and α_{high}. If $n > 1$ the transition from low to high

damping is smoother, while a harder transition is perceived when the user pass from high to low damping. The opposite happens with $n < 1$.
Experimental results showed that the operator prefers an immediate response when transitioning from a unadvisable direction (high damping zone) to a suitable one (low damping zone), and a smoother deceleration when he/she is deviating away from the optimal path. Hence a value $n > 1$ is preferable.

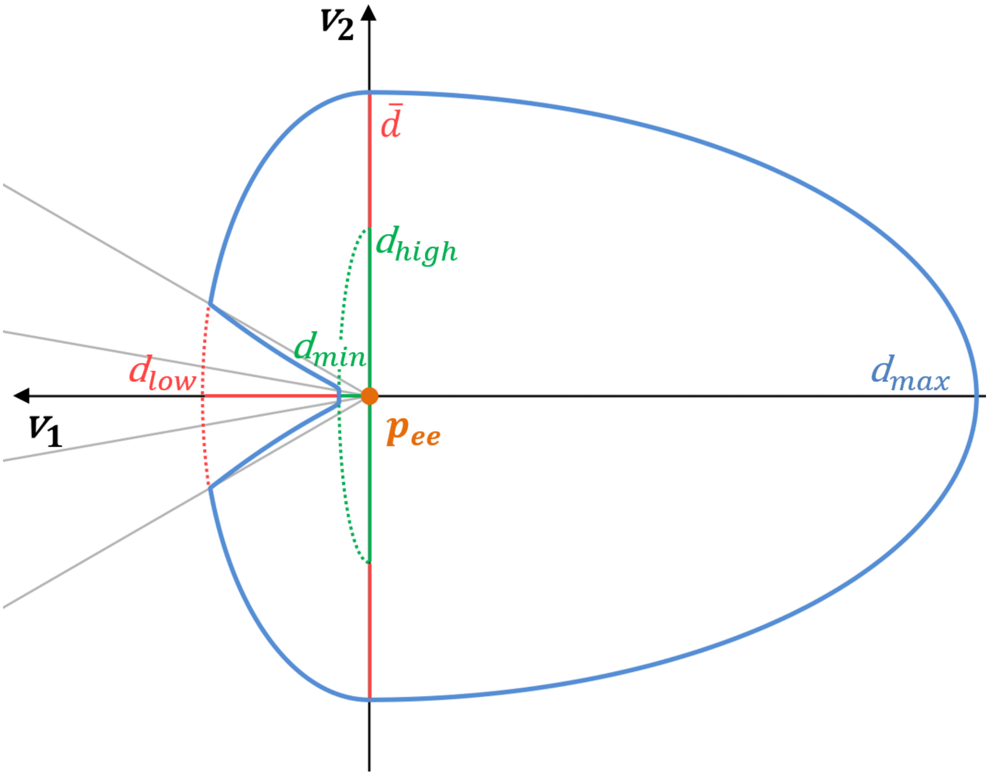


Figure 3.5: 2D representation of the modified damping shape (blue line), obtained from the combination of a high damping ellipsoid (in red) and a low damping one (in green).

Comparing this shape to the concentric spheres, although more complex calculations are required, no feedback resolution is lost when pushing along the optimal direction or sideways. There are no constant damping zones: the coefficient is a monotonic increasing function of the angle α between the applied force and the optimal direction.

Chapter 4

Obstacle avoidance in manual guidance

4.1 Introduction

As stated in Chapter 3 the main purpose of the developed algorithms is to assist the operator during a manual guidance task in a realistic working scenario. Usually, the work cell has a specific layout and many components: pallets, machines and other objects become hurdles that the operator has to avoid while reaching the desired destination.
An intuitive directional haptic feedback can be provided to the operator thanks to the developed technique described in Chapter 2 and 3. At each time instant, the optimal direction $\boldsymbol{v_1}$ has to be determined in a suitable way. Indeed, its basic formulation, provided in eq. 2.13, would fail in presence of a cluttered working environment because it would suggest to the operator the straight line from the actual position to the goal one as the optimal direction, regardless of the presence of some obstacles. Therefore, a new and robust method, that establishes the minimum effort direction taking into account also the obstacles and the dimensions of the robot working space, has to be set-up. Another remarkable achievement would be taking into account also the motion intention of the user, in case more than one collision free path towards the goal position exists.

4.2 Working environment model

The term map will be used to refer to the largest projection on the horizontal plane of the robot working space, namely the largest set of the end-effector position reachable by the robot in an horizontal plane. The external perimeter of the map is only determined by the robot mechanical structure and possibly by the dimensions of the workspace, depending on which one is more restrictive. Both rectangular and circular maps were considered in the development of this thesis (see Fig. 4.1 for an example of a sector of circular map).

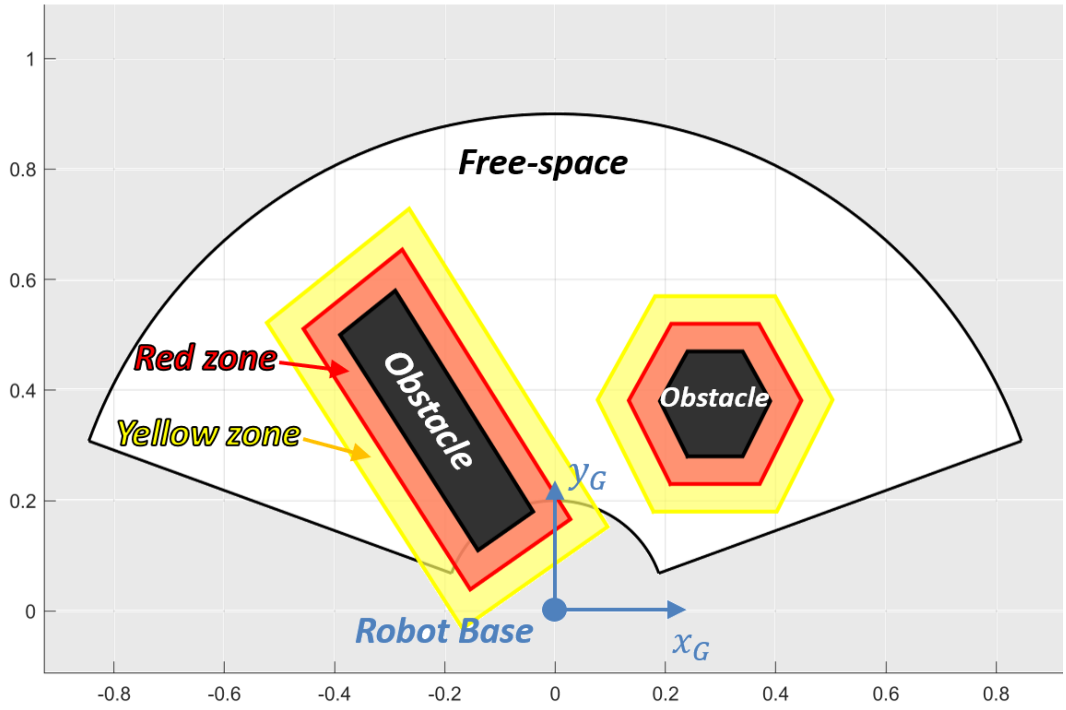


Figure 4.1: Example of a 2D circular map with 2 obstacles (in black), each one surrounded by its *red zone* and *yellow zone*.

Its worth mentioning that only a bi-dimensional simplification of the environment was used for testing, but the developed algorithms can be easily scaled to a three-dimensional map.

Inside the bi-dimensional map, every obstacle is represented by a forbidden area (like the two black areas in Fig. 4.1). The robot end-effector should never collide against any of the obstacles to avoid potential damages. Obstacles are approximated by means of generic convex polygons. This choice allows to implement simpler algorithms for free-space quantification, collision detection and distance measurements. Nevertheless, more complex-shaped obstacles can be created by clustering convex polygons together.

Two zones are constructed around each obstacle. These zones are concentric with respect to the obstacle polygon and completely surround it, as it can be seen in figure 4.1. The inner *red zone* has a safety purpose: its area is not considered as free-space and should never be crossed by the robot end-effector. This zone enforce that, at any time, a minimal safe distance is kept from the real obstacle. The outer *yellow zone*, instead, prevents the algorithm to set path key-points too close to the red-zone. However, a path can be planned across it.

A virtual wall is implemented around each obstacle perimeter as a final collision counter-measure. Its purpose is to slow down the end-effector motion and stop it before impact. The virtual wall is completely included inside the obstacle *red zone*.

Once the working environment has been modeled, a starting position and a goal one have to be placed on the free-space of the established map. Then, an automatic procedure generating a static structure exploring the entire free space of the working environment has to be set-up. Eventually, an online algorithm that, based on the previously defined structure, establishes the optimal (minimum effort) direction $\boldsymbol{v}_1$, at each time instant, has to be defined.

Two types of *environment exploring structures* will be presented in the next two sections. The first one is based on a grid structure while the second one on a tree architecture. For each one, both the creation process and the online analysis is discussed.

4.3 Recursive grid based obstacle avoidance

A first approach to the map discretization problem, namely to the definition of the *environment exploring structure*, is based on a grid architecture. An example is depicted in figure 4.2.

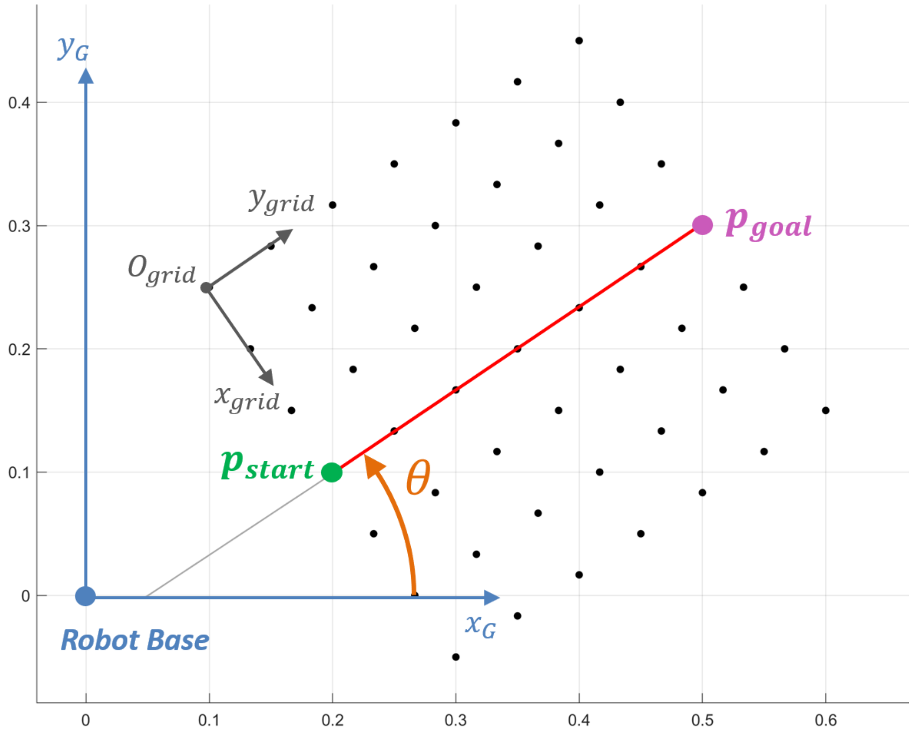


Figure 4.2: Graphical representation of an orthogonal grid of points, used to discretize the space between $\boldsymbol{p}_{start}$ and $\boldsymbol{p}_{goal}$.

A uniform discretization is achieved by means of an orthogonal grid of points. The grid is aligned with the line connecting the starting position and the destination, so that its rows represent a gradual progression from the start to the end of the motion. θ is defined as the angle between the line connecting $\boldsymbol{p}_{start}$ to $\boldsymbol{p}_{goal}$ and the global (robot base) x-axis.

To guide the user towards the goal, it would be sufficient to align $\boldsymbol{v}_1$ to a key-point belonging to a row which is closer to the objective than the one corresponding to the end-effector position and, possibly, directed towards the goal. The operator is continuously led forward until the objective row is reached, then it can be redirected towards the exact unloading position.

It is worth noting that, in obstacle proximity, a small grid *step* would be preferable in order to perform more precise movements around the obstacle, while in free-space, away from any obstruction, only a few samples, taken at larger distance from each other, are required. Therefore, a variable grid *step* strategy, to achieve the required resolution requirements, occurs to be defined.

This is addressed with recursion: the cells of the grid that are close to the obstacles are further discretized by generating a smaller grid inside them. This process can be repeated recursively on the sub-grid cells until the desired level of detail is obtained.

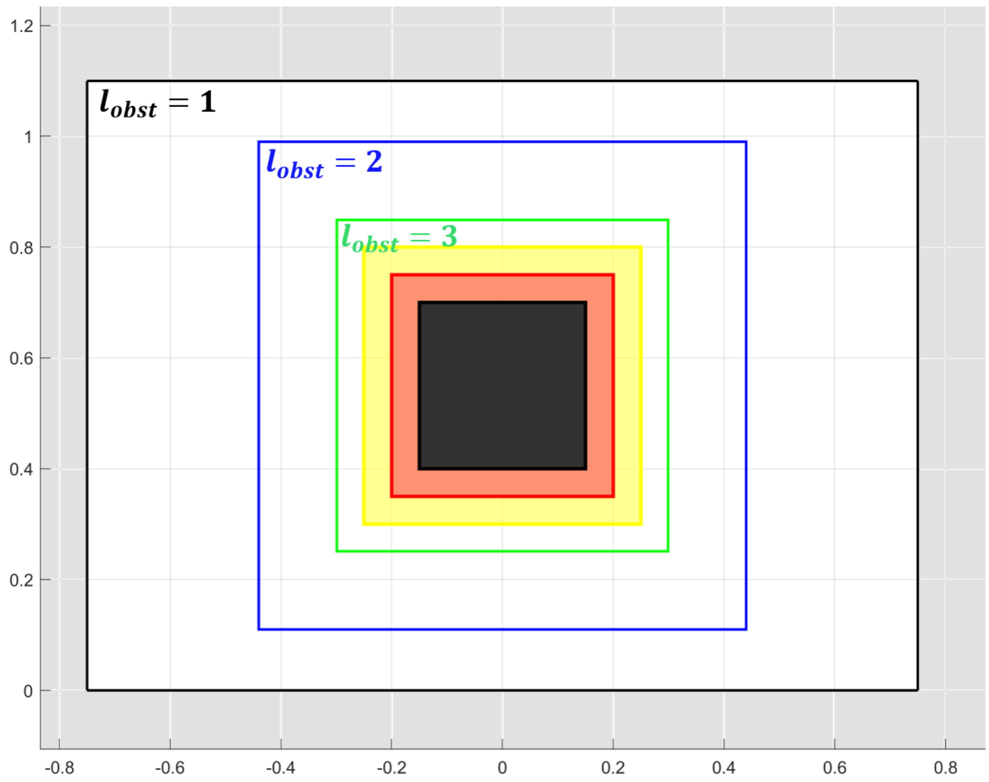


Figure 4.3: Example of a squared map with one central obstacle. Concentric zones are added around it with increasing detail l_{obst}, up to $l_{obst} = l_{max} = 3$.

To determine how many nested sub-grids are necessary in a certain location, the obstacles are provided with additional concentric zones (see figure 4.3). A level of detail l_{obst} is assigned to every area: its maximum value l_{max} is assigned to the zone that surrounds the *yellow zone*. Every other concentric zone is assigned a gradually decreasing level until the minimum value of 1 is reached, corresponding to the main grid detail. A complete recursive grid, constructed according to the concentric zones, is portrayed in figure 4.4.

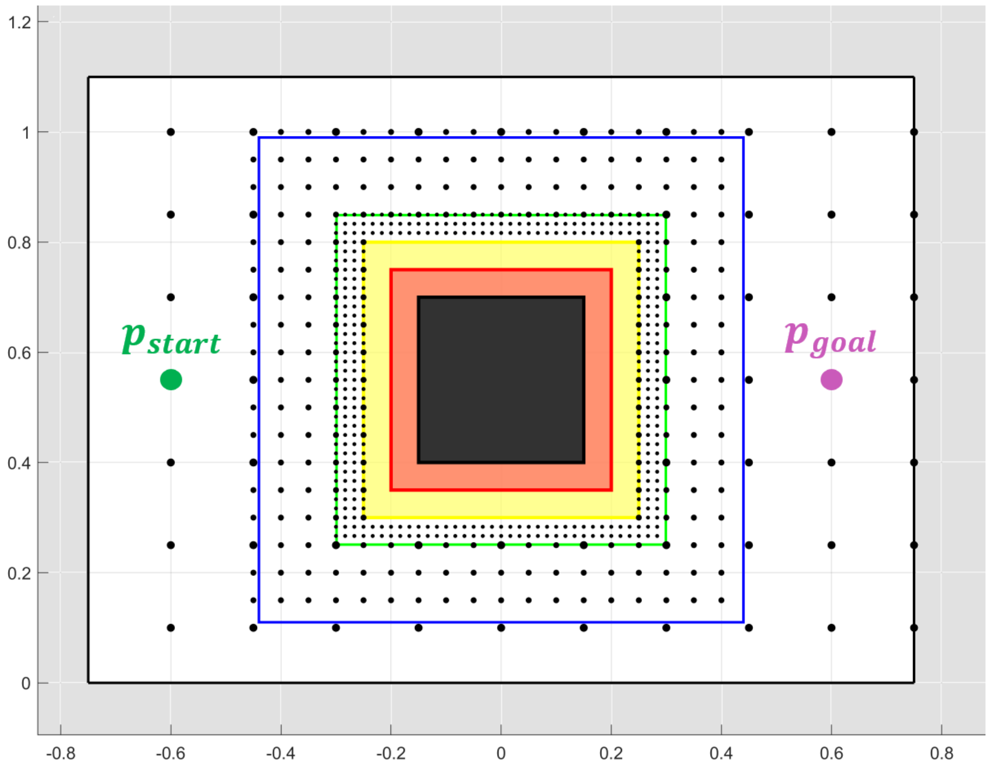


Figure 4.4: Example of a complete recursive grid, constructed according to the layout and concentric zones depicted in figure 4.3.

4.3.1 Recursive grid creation

To define the main grid it is necessary to assign the starting position $\boldsymbol{p}_{\boldsymbol{start}}$, objective position $\boldsymbol{p}_{\boldsymbol{goal}}$ and the grid base dimension n_b, which must be an odd natural number. The base grid is composed by n_b^2 points, equally distributed in the space between the two end-points of the motion.

Moreover, it is possible to specify two parameters $n_{lateral}$ and n_{over}: the first one adds additional columns of nodes at each side of the main grid, the second adds rows at the end of the grid over the objective row, to address possible overshooting of the goal position. Therefore, the complete main grid is composed by $n_b + n_{over}$ rows and $n_b + 2n_{lateral}$ columns, as it can be seen in figure 4.5.

Each node can be identified by a set of coordinates (i, j), representing its column and row inside the grid. The columns are numbered from the left-most (with respect to the line connecting $\boldsymbol{p}_{\boldsymbol{start}}$ to $\boldsymbol{p}_{\boldsymbol{goal}}$) to the right-most, and the rows are indexed starting from the one including $\boldsymbol{p}_{\boldsymbol{start}}$, increasing towards $\boldsymbol{p}_{\boldsymbol{goal}}$.

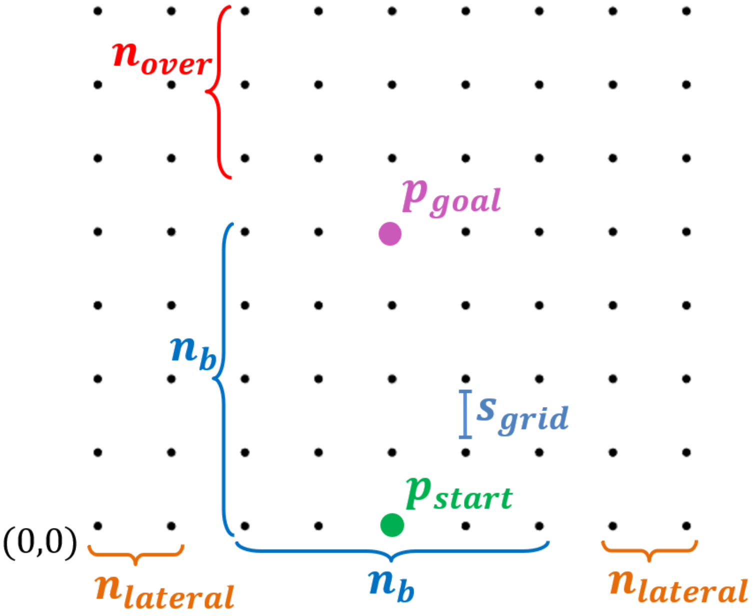


Figure 4.5: Example of a main grid structure with $n_b = 5$, $n_{lateral} = 2$, $n_{over} = 3$.

To compute the coordinates of each node, it is necessary to evaluate the angle of inclination of the main grid θ, the grid step s_{grid} and the grid origin $\boldsymbol{O}_{\boldsymbol{grid}}$, which correspond to the position of the node at coordinates $(0,0)$.

The grid step is computed by measuring the distance between $\boldsymbol{p}_{\boldsymbol{start}}$ and $\boldsymbol{p}_{\boldsymbol{goal}}$ and dividing it by $n_b - 1$, which is the number of grid cells separating the first row, where the starting position is located, from the n_b-th row, corresponding to the goal.

A translation is applied to compute the grid origin, knowing that the grid is symmetric and the starting position is located in the middle column:

$$\boldsymbol{O}_{\boldsymbol{grid}} = \boldsymbol{p}_{\boldsymbol{start}} - s_{grid}(\frac{n_b - 1}{2} + n_{lateral})\begin{bmatrix} sin(\theta) \\ -cos(\theta) \end{bmatrix}$$

Once the main grid dimensions and core elements are established, the map is sampled to define the state of each node. The position of a node in the grid located at generic coordinates (i,j) is given by:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \boldsymbol{O}_{\boldsymbol{grid}} + s_{grid}\begin{bmatrix} sin(\theta) & cos(\theta) \\ -cos(\theta) & sin(\theta) \end{bmatrix}\begin{bmatrix} i \\ j \end{bmatrix}$$

If the point at (x, y) is out of the free-space, then the corresponding node of the grid will not be considered an *available node*. Instead, if the point belongs to the free-space, two different scenarios may occur, depending on its relative position with respect to the obstacles. If the grid cell corresponding to node (i, j), namely the area defined by the nodes (i, j), $(i+1, j)$, $(i, j+1)$ and $(i+1, j+1)$, does not intersect any obstacle concentric zone of level $l_{obst} > 1$, then it is considered an *available node* and no further discretization of the corresponding cell is needed. Otherwise, if the grid cell intersects an obstacle concentric zone of level $l_{obst} > 1$, a finer space discretization is required. Therefore, a subgrid of $n_{sg} \times n_{sg}$ nodes is generated inside the cell, having as origin the position of the node (i, j), inclination θ equal to the main grid and *step* $s_{sg} = \frac{s_{grid}}{n_{sg}}$. The subgrid is populated in the same manner as the main grid: for each node a new point of the map is sampled and, based on its position, it can be an *available node* or not. Additionally, this procedure is recursively applied on the sub-cells, depending on their distance from the obstacles, up to the maximum level of detail l_{max}. The estimated computational complexity of the grid creation process is $\mathcal{O}(n^{2+2l_{max}})$, where $n = n_b$. An example of a grid structure, generated in a circular map containing only one obstacle and with three levels of detail ($l_{max} = 3$), is depicted in figure 4.6.

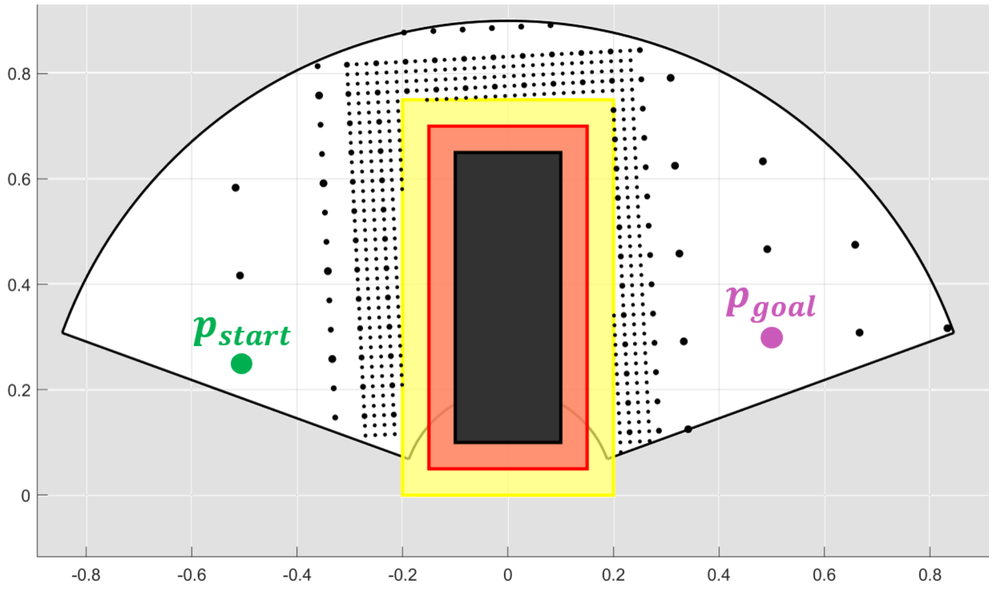


Figure 4.6: Example of a grid-based environment exploring structure with $l_{max} = 3$, $n_b = 7$, $n_{lateral} = 2$, $n_{over} = 2$, $n_{sg} = 3$, generated inside a circular map.

4.3.2 Online recursive grid analysis for optimal direction selection

In order to proceed towards the objective a new point must be chosen from a row which is closer to the objective row j_{goal} than the one corresponding to the end-effector j_{ee}.

The first step is to find the coordinates $\boldsymbol{c_{ee}} = (i_{ee}, j_{ee})$ of the node associated to the grid cell which contains the end-effector position. To do so, a linear system must be solved:

$$\begin{bmatrix} i_{ee} \\ j_{ee} \end{bmatrix} = \left\lfloor s_{grid}^{-1} \begin{bmatrix} sin(\theta) & -cos(\theta) \\ cos(\theta) & sin(\theta) \end{bmatrix} (\boldsymbol{p_{ee}} - \boldsymbol{O_{grid}}) \right\rfloor$$

If the coordinates (i_{ee}, j_{ee}) correspond to a node which is origin of a subgrid, then the procedure is recursively repeated by substituting $\boldsymbol{O_{grid}}$ with the origin of the subgrid. The final result are the coordinates $\boldsymbol{c_{ee}}$ of the end-effector position down to the lowest recursive grid level.

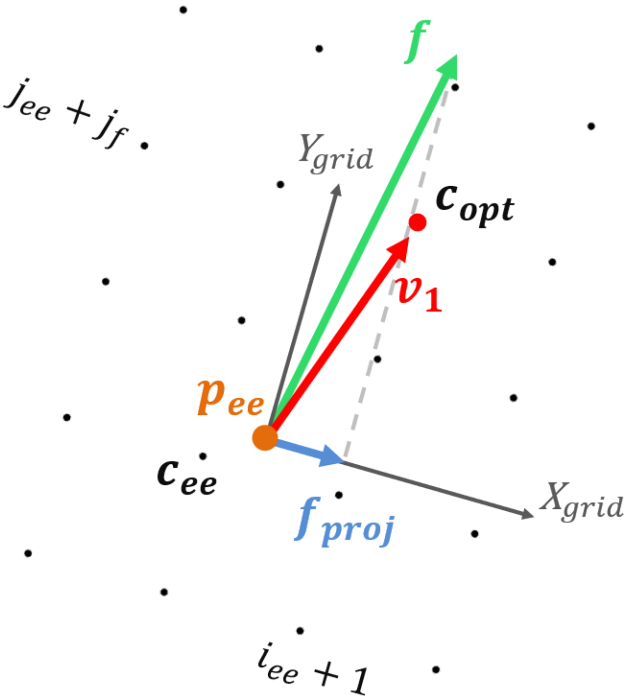


Figure 4.7: Graphical representation of the optimal direction selection process for the grid-based approach, with $j_f = 2$.

The second step is to find the coordinates $\boldsymbol{c_{opt}}$ of the optimal *available node*, namely the node that allows to perform the shortest local path towards the next grid row avoiding the obstacles. The human intention of motion to overcome the obstacle on the left or right side is also taken into account. Then, the direction from the actual end-effector position $\boldsymbol{p_{ee}}$ to the location of coordinates $\boldsymbol{c_{opt}}$, is selected as the optimal direction $\boldsymbol{v_1}$, namely the minimum effort direction for variable admittance control.
The j-th coordinate of $\boldsymbol{c_{opt}}$ is chosen as the one of $\boldsymbol{c_{ee}}$ increased by a predetermined integer number $j_f > 1$, representing how many rows ahead with respect ot the current one the locally optimal node is searched. The greater j_f, the more the operator is pushed forward, but the less precise will be the movements near the obstacles. The i-th coordinate of the optimal node, is determined based on the human intention of motion, as it can be seen in figure 4.7.
Indeed, the projection $\boldsymbol{f_{proj}}$ of the applied human force $\boldsymbol{f}$ on the direction of the x-axis of the grid, which is normal to the line connecting $\boldsymbol{p_{start}}$ to $\boldsymbol{p_{goal}}$, is considered:

$$i_{opt} = \begin{cases} i_{ee} + 1, & \text{if } \boldsymbol{f_{proj}} \cdot X_{grid} \geq 0 \\ i_{ee}, & \text{otherwise} \end{cases}$$

In obstacles proximity, there might be cases where the optimal coordinates $\boldsymbol{c_{opt}}$ correspond to a node which is outside of the free-space or the path from the current end-effector position $\boldsymbol{p_{ee}}$ to $\boldsymbol{c_{opt}}$ is not collision-free. In these cases, the new candidate node for being the optimal one is selected as the nearest to $\boldsymbol{c_{opt}}$, on the same row j_{opt} and in direction of $\boldsymbol{f_{proj}}$. If also the new candidate optimal node is not available, then the searching procedure of a new optimal node is repeated until a fixed maximum number of lateral steps is reached. If no available route has been found yet, j_{opt} is decreased until it becomes equal to $j_{ee} - j_f$, while the lateral steps are kept at the maximum value. If no node has been selected still, the lateral steps are gradually decreased with $j_{opt} = j_{ee} - j_f$.
The result, as it can be seen in figure 4.8 is a complete exploration of the directions aligned with the user intention. If no available node is found this way, then

the user effort will be high in the current force direction, prompting him/her to reverse his/her push and explore the other half-grid.

Finally, when j_{opt} equals j_{goal} the user has reached the objective row. At this stage the obstacles have been cleared and it is necessary to reach the goal position. User autonomy is reduced: the force projection is no longer taken into consideration, only the coordinates of the end-effector relative to the goal is examined. The operator is then guided towards the objective coordinates to obtain an accurate final positioning.

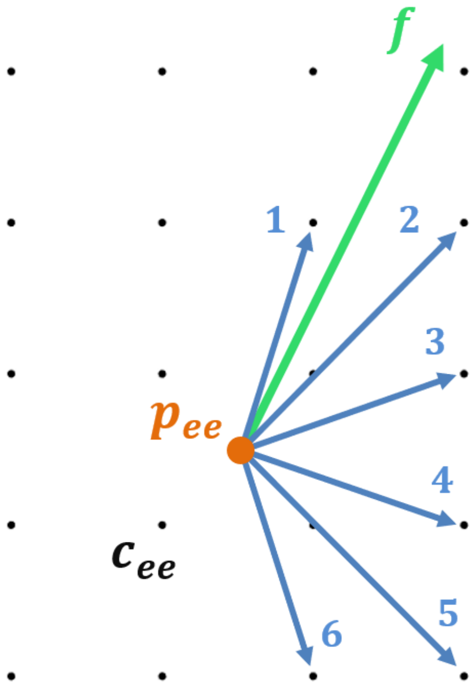


Figure 4.8: Depiction of the the locally explored directions with $j_f = 2$ and a maximum of two lateral steps, ordered from the first to the last analyzed.

If, at any point, the end-effector position belongs to a subgrid, then it is possible to exploit the recursive properties of the grid to improve the quality of the optimal direction choice. In particular, two possible strategies can be applied regarding j_f, as depicted in figure 4.9.

The first one consist in keeping the step forward length-wise constant (see figure 4.9a). The number of steps forward at the main grid level $j_f^{(1)}$ is converted to a number of equivalent sub-steps $j_f^{(l_{ee})}$, depending on the maximum level of detail l_{ee} available at the end-effector location:

$$j_f^{(l_{ee})} = j_f^{(1)} (n_{sg})^{l_{ee}}$$

The new node coordinates $\boldsymbol{c_{opt}}$ are computed at the deepest level of detail (l_{ee}). As a result the distance between the end-effector row $j_{ee}^{(l_{ee})}$ and the new point row is kept constant despite the change in detail level.

The second strategy, instead, is to impose a constant number of forward steps, independent of the change in detail (see figure 4.9b). $j_f^{(l_{ee})} = j_f^{(1)}$ is adopted, meaning that the distances among the current row and the optimal one reduces with the increasing of the resolution of the grid.

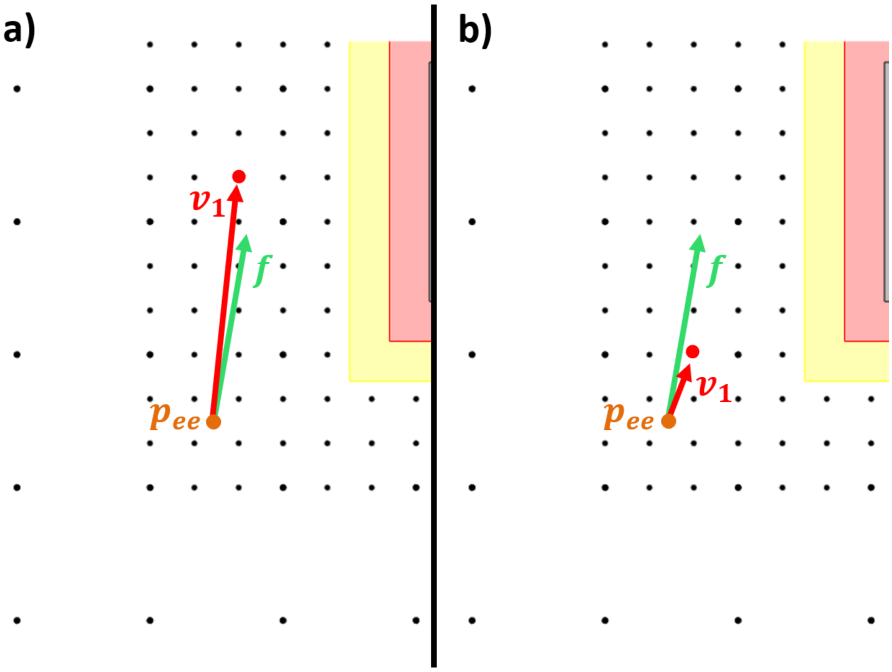


Figure 4.9: Comparison between the two possible subgrid exploration approaches with $j_f = 2$. To the left the *constant length* strategy, to the right the *constant number* one.

In the event that the new coordinates $\boldsymbol{c_{opt}}$ correspond to a node with $l_{new} < l_{ee}$, it is impossible to find the optimal direction while keeping the current detail. The forward steps conversion is repeated, ignoring the deepest level of detail and using l_{new}. However these events occur when the end-effector is moving away from the obstacles, hence a loss of detail is acceptable.

Both strategies have advantages and drawbacks. In fact, adopting the first approach is equivalent to maintaining a constant prediction horizon. The map is examined at a greater distance in front of the end-effector and the chosen key-point is far from $\boldsymbol{p_{ee}}$. Thanks to the large prediction horizon, the presence of an obstacle in the direction of the motion will influence the choice of $\boldsymbol{v_1}$ early, hence, through the haptic feedback, it is possible to make the operator aware of the obstacle when the end-effector is still far from it. However, this strategy is not preferable when multiple obstacles are close to each other, forming corridors, since less nodes are available for the choice: the fewer are the points, the less is the probability of finding a collision-free key-point at a long distance from the end-effector.

On the contrary, the second solution sacrifices the longer prediction horizon in spite of a better exploitation of the recursive sub-grids. The user is less forced to take a specific path since the key-point choice happens at a finer level of detail, where more nodes are available. However, if an obstacle is in the direction of the motion, then the algorithm would react slower when compared to the first strategy.

In the experimental campaign described in Chapter 5, only the second grid analysis strategy was put to the test. The working environments proposed in the experiments often required the operator to cross narrow passages and overcome multiple obstacles. As mentioned before, the first strategy is not suitable in such cases. The second strategy, instead, allows for a finer exploration of the available space, which is fundamental in complex and highly structured environments.

Since only a determined and localized portion of the grid is explored to select the optimal direction, the computational complexity of the grid analysis phase depends only on the complexity of the algorithm chosen for collision detection, which, in our case, is $O(n_{obstacles})$.

4.3.3 Recursive grid limitations

The recursive grid structure presented above allows to find the locally best direction to reach the objective, considering the end-effector current position, the map and the user intention of motion. However, some intrinsic limitations are hereby presented.

The first consideration is that the grid approach is only reactive, meaning that the user is made aware of the obstacle presence only when the end-effector is close to it and he/she is pushing towards it. In free space, away from any obstacle, the operator is only guided forward towards the goal: no path optimization is implemented, the only objective is to proceed inside the grid and the only feedback is reactive to the obstacles.

Moreover, since the structure analysis is only local, this method is sensitive to local minima. If the user drives the end-effector in a dead-end, this algorithm guides him/her forward until an obstacle is in proximity, despite the fact that no feasible path towards the objective is available by following that route.

Additionally, only a rectangular portion of the map is explored, the one comprised between the starting and goal position, hence this approach is not well suited for maps having a complex shape or a complicated obstacle layout.

Furthermore, a known initial position is necessary for the grid creation, hence, a new grid must be created each time a new movement with different end-points needs to be executed.

Therefore, a second approach aiming to solve all these drawbacks has been conceived and will be described in the next section.

4.4 RRT based obstacle avoidance

The second proposed *environment exploring structure* solves most of the Recursive Grid drawbacks, highlighted in the previous section. The orthogonal grid structure, made of single nodes independent of each other, is substituted with a set of randomly sampled points that are interconnected in a meaningful way.
In fact, it is possible to create an ordered graph, where each node corresponds to a point in the free-space and each edge represent the collision-free path to follow to reach the goal position. This kind of graph is commonly named *tree* and it is created starting from one given point, called the *root*. The peculiarity of the *tree* is that it is always possible to reach the *root* from whatever node belonging to the tree by following the nodes edges. The creation of such structure is inspired from the Rapidly exploring Random Tree (RRT) algorithm, a well known path planning algorithm adopted in traditional robotics to find a collision free path form a starting robot configuration to a final one.
Normally, once a path is found, the robot cyclically repeats exactly the same path in an automatic way. Instead, in manual guidance, since the human drives the robot end-effector, a certain path cannot be repeated exactly. Moreover, the current position will always be out of the optimal path and the initial position may vary a lot even if the unloading position remains the same. Therefore, the classical obstacle avoidance problem adopting a tree structure has to be formulated to face all these differences. The tree will have to cover all the working environment starting from the goal position regardless of the initial position. Then, it will be used as a reference to follow to reach the goal independently by the starting or current position.
In this adaptation of RRT, the generated tree is *rooted* in the objective position and its branches spans across the whole map, exploring all the feasible path around the obstacles. Moreover it is possible to re-rout the connections according to different cost functions, allowing for different path optimizations. For example, the operator may prefer a safer path, that keeps him/her away from the obstacles, or a more direct one, optimized only on distances between the nodes.

The structure is generated before the motion, hence the complex computation and optimizations are performed offline. Only the structure analysis, which consists in choosing the next target key-point and hence the minimum effort direction, is executed online. Additionally, since the structure creation requires only a known goal position, it's possible to use the same structure for many movements even with different starting positions, as long as the objective does not change.

4.4.1 RRT in the literature

Given a path planning problem, with known starting ($\boldsymbol{q_{init}}$) and final ($\boldsymbol{q_{goal}}$) configurations, Rapidly-exploring Random Tree algorithm aims at the creation of a tree structure, rooted at the initial configuration and built incrementally to reach the goal. A complete tree represents a set of feasible paths to the goal configuration. A pseudo-code description of basic RRT can be found in algorithm 1.

Algorithm 1 RRT

```
$V \leftarrow \{\boldsymbol{q}_{init}\}$;
$E \leftarrow \emptyset$;
for $i = 1..N$ do
    $\boldsymbol{q}_{rand} \leftarrow SampleFree$;
    $\boldsymbol{q}_{nearest} \leftarrow Nearest(G, \boldsymbol{q}_{rand})$;
    $\boldsymbol{q}_{new} \leftarrow Steer(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{rand})$;
    if $CollisionFree(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{new})$ then
        $V \leftarrow V \cup \{\boldsymbol{q}_{new}\}$;
        $E \leftarrow E \cup \{(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{new})\}$;
    end if
end for
return $G = (V, E)$
```

At each iteration, the *SampleFree* function extract a new configuration $\boldsymbol{q}_{rand}$, sampling it from the free-space (figure 4.10a). Then $\boldsymbol{q}_{nearest}$, which is the nearest tree node to $\boldsymbol{q}_{rand}$, is found (figure 4.10b). Afterwards, the *Steer* function extracts

a new node configuration, $\boldsymbol{q}_{new}$, in the direction of $\boldsymbol{q}_{rand}$, trying to locally optimize the path by steering towards the goal. If the path between $\boldsymbol{q}_{new}$ and $\boldsymbol{q}_{nearest}$ is collision-free, $\boldsymbol{q}_{new}$ is added to the tree and it is connected to $\boldsymbol{q}_{nearest}$ (figure 4.10c). From now on, $\boldsymbol{q}_{nearest}$ will be referred to as *parent* of $\boldsymbol{q}_{new}$, namely the node to follow from $\boldsymbol{q}_{new}$ in order to reach the root of the tree.
This loop is repeated until the objective configuration is connected to the tree or the maximum number of iterations N have been executed.

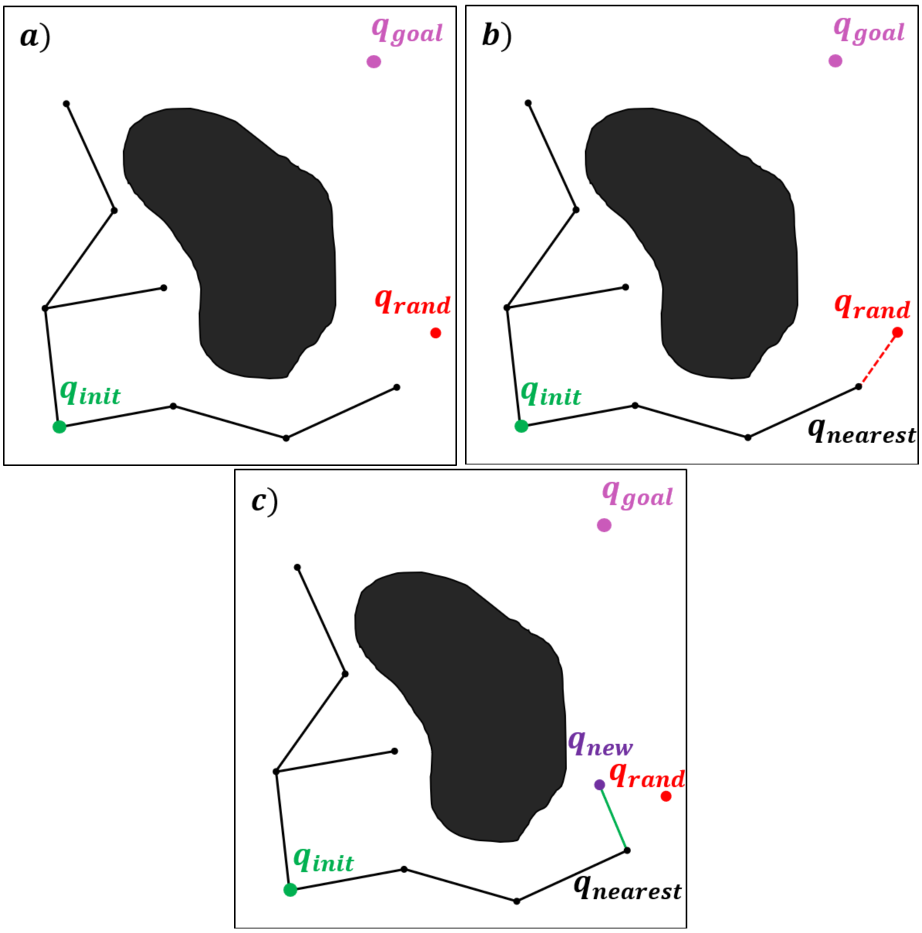


Figure 4.10: Graphical representation of the RRT main procedures: Sampling (a), Finding the nearest node (b), Steering (c).

However, the basic RRT algorithm utilizes a non-exact steering function and computes only a generic path from initial to final configuration. It can be improved by assigning a monotonic cost function to the nodes and minimizing it during tree creation. The generated path is optimized according to the specified cost. An optimal version of RRT, named RRT*, is reported in Algorithm 2.

Algorithm 2 RRT*

```
$V \leftarrow \{\boldsymbol{q}_{init}\}$;
$E \leftarrow \emptyset$;
for $i = 1..N$ do
    $\boldsymbol{q}_{rand} \leftarrow SampleFree$;
    $\boldsymbol{q}_{nearest} \leftarrow Nearest(G, \boldsymbol{q}_{rand})$;
    $\boldsymbol{q}_{new} \leftarrow Steer(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{rand})$;
    if $CollisionFree(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{new})$ then
        $Q_{near} \leftarrow Near\Big(G, \boldsymbol{q}_{new}, \min\Big\{\gamma_{RRT^*}(\log(card(V))/card(V))^{1/d}, r_{max}\Big\}\Big)$;
        $V \leftarrow V \cup \{\boldsymbol{q}_{new}\}$;
        $\boldsymbol{q}_{min} \leftarrow \boldsymbol{q}_{nearest}$;
        $c_{min} \leftarrow Cost(\boldsymbol{q}_{nearest}) + Cost(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{new})$;
        for each $\boldsymbol{q}_{near} \in Q_{near}$ do
            $c_{new} \leftarrow Cost(\boldsymbol{q}_{near}) + Cost(\boldsymbol{q}_{near}, \boldsymbol{q}_{new})$;
            if $CollisionFree(\boldsymbol{q}_{near}, \boldsymbol{q}_{new}) \wedge c_{new} < c_{min}$ then
                $\boldsymbol{q}_{min} \leftarrow \boldsymbol{q}_{near}$;
                $c_{min} \leftarrow c_{new}$;
            end if
        end for
        $E \leftarrow E \cup \{(\boldsymbol{q}_{min}, \boldsymbol{q}_{new})\}$;
        for each $\boldsymbol{q}_{near} \in Q_{near}$ do
            $c^*_{near} \leftarrow Cost(\boldsymbol{q}_{new}) + Cost(\boldsymbol{q}_{near}, \boldsymbol{q}_{new})$;
            if $CollisionFree(\boldsymbol{q}_{near}, \boldsymbol{q}_{neew}) \wedge c^*_{near} < Cost(\boldsymbol{q}_{near})$ then
                $\boldsymbol{q}_{parent} \leftarrow Parent(\boldsymbol{q}_{near})$;
                $E \leftarrow (E \setminus \{(\boldsymbol{q}_{parent}, \boldsymbol{q}_{near})\}) \cup \{(\boldsymbol{q}_{new}, \boldsymbol{q}_{near})\}$;
            end if
        end for
    end if
end for
return $G = (V, E)$
```

In RRT*, the *Steer* function takes into account the robot kinematic constraints and the actuators limits when the steered sample $\boldsymbol{q}_{new}$ is computed. Moreover, the new node is not connected to its nearest anymore. Every node belonging to a neighborhood around the new configuration is analyzed: a new edge is created towards the node that does not cause a collision and minimizes the total cost, which is the sum of the new *parent* cost and the connection cost (figure 4.11).

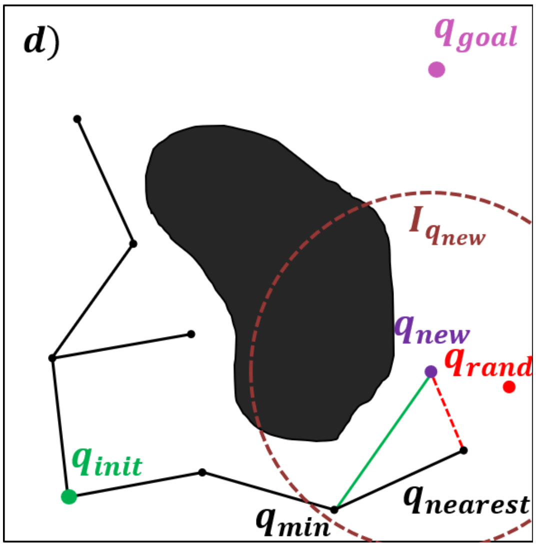


Figure 4.11: Graphical representation of the RRT* cost minimization phase.

The radius of the neighborhood is based on the optimal RRT* radius γ_{RRT^*}, which is:

$$\gamma_{RRT^*} > 2(1+1/d)^{1/d}(\mu(\mathcal{Q}_{free})/\zeta_d)^{1/d}$$

where d is the dimension of the configuration space, ζ_d is the volume of the unit ball and $\mu(Q_{free})$ is a Lebesgue measure of the obstacle-free space. The rate of decay of the radius is a function of the sample dispersion of a set of n random points sampled uniformly and independently, which is $\mathcal{O}\Big((\log(n)/n)^{1/d}\Big)$. The more nodes are added to the tree, the more the radius is reduced.

Additionally, a tree rewiring phase is executed after the first optimization: the new node and edge may open new paths for the nearby nodes. A new connection is attempted from any node belonging to the neighborhood of the new node. If a collision-free path that reduces the cost of a neighbor is available, a new edge towards the center node replaces the old connection. The cost of each children is updated accordingly (see figures 4.12).

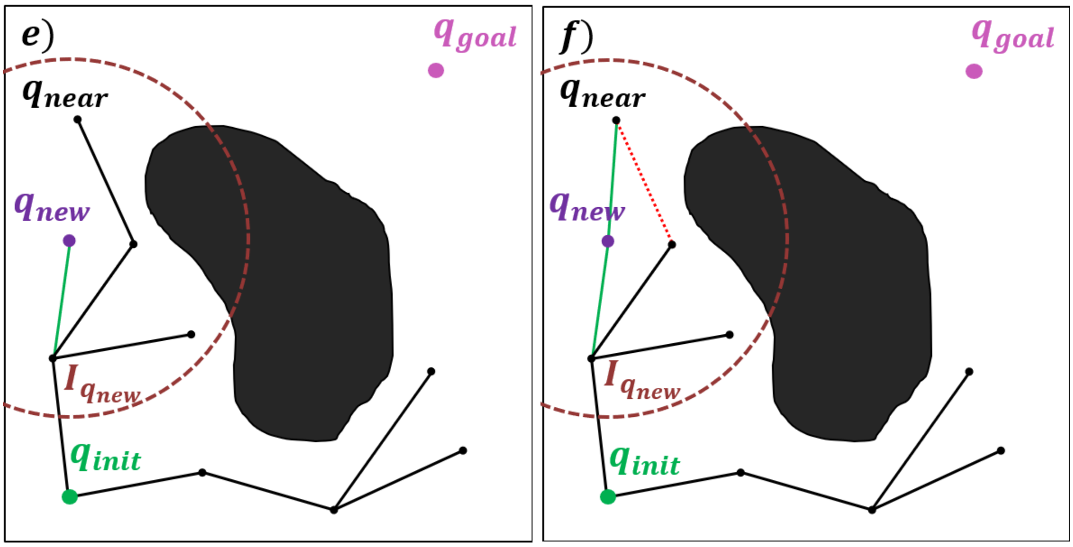


Figure 4.12: Graphical representation of the RRT* rewiring phase: a new sample is extracted (e), then the nodes in the neighborhood are rewired in order to minimize the cost (f).

The algorithm runs until it's possible to connect the target configuration to the tree or the maximum number of iteration N has been reached. The final tree contains a set of feasible paths towards the objective, fully optimized according to the cost function.

4.4.2 A new implementation of RRT* for goal driven variable admittance control

The developed *environment exploring structure* for manual guidance is largely based on the RRT* algorithm, applied to find feasible paths for the robot end-effector in the Cartesian space. However, a few implementation differences can be identified.

The tree is not constructed from a generic starting position, but from the goal. The reason behind this choice is twofold. First, in manual guidance applications the operator is not forced on a predefined collision-free path but he/she drives the robot end-effector where he/she desires. Secondly, the same unloading station (goal position) could be reached starting from whatever point of the free-space. Hence, it is fundamental to produce a structure with well defined paths leading to the final position, independently of the current and starting end-effector location. The tree generation must be based only on the specific map and on the known target p_{goal}, while its branches will be constructed not to connect a specific initial position to the goal one, but to cover the entire map. In this way, regardless of the current and initial position, a reference collision path leading to the target can be found everywhere in the free working space. An example of tree-based structure is portrayed in figure 4.13. Notice how the tree branches spans across the entirety of the working environment, and how different branches of the tree explore different paths around the obstacles.

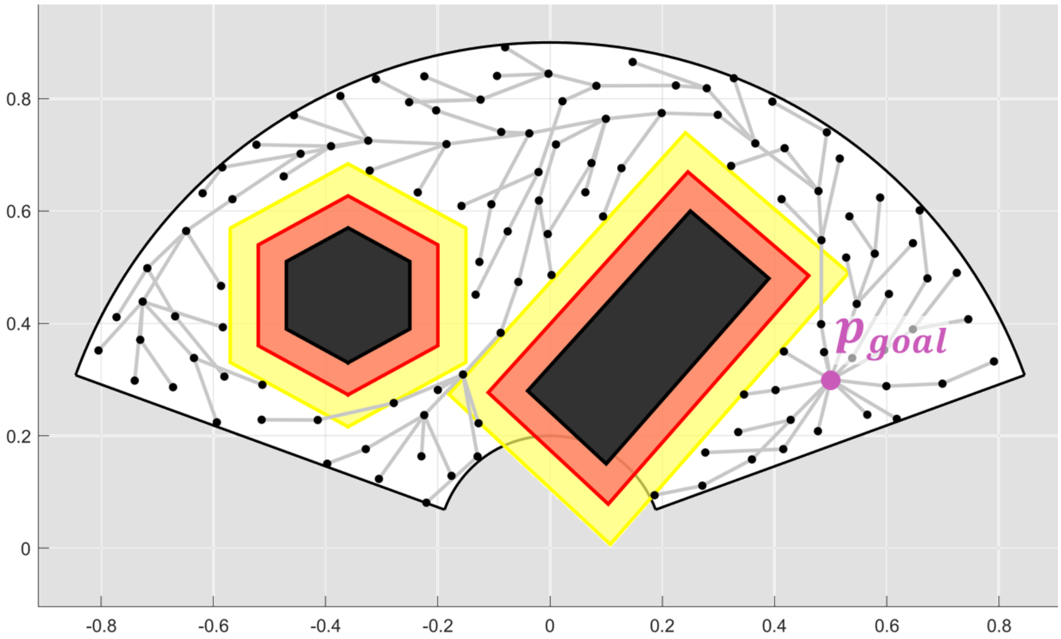


Figure 4.13: Example of an *environment exploring structure* based on the RRT approach, constructed in a circular map with multiple obstacles.

Since the objective of the tree is no more to connect initial and final position, but becomes to cover the entire working space starting from the goal position (root of the tree), the tree nodes must be evenly spread across the map. To achieve this purpose, a maximum and minimum distance boundaries between samples is prescribed. Specifically, a maximum value d_{step} is assigned, and the minimum boundary is set to $\frac{d_{step}}{2}$.

Moreover, a different terminating condition must be established for the algorithm. The total number of nodes n_{total} must be estimated before the creation since it is not possible to determine when the map is fully explored. A heuristic approach is used to solve this problem that is interpreted as a *circle packing* problem. Indeed, the nodes can be seen as circles, whose radius is the prescribed minimum distance of $\frac{d_{step}}{2}$, and the map is the container: finding n_{total} is equivalent to determine the maximum number of circles that can be packed inside the container. The free space is measured by evaluating the map area and subtracting the space occupied by the obstacles and the *yellow zones* around them, taking into account possible overlapping between different obstacles.
The final estimate is:

$$n_{total} = \left\lfloor \frac{A_{map} - \sum A_{yellow} + \sum A_{overlap}}{\pi(\frac{d_{step}}{2})^2} \right\rfloor$$

The algorithm adds a new node at each iteration, sampling its position from the free-space. As already mentioned, the sampling is no more aimed at reaching a known point like in classic RRT: the entirety of the map has to be explored.

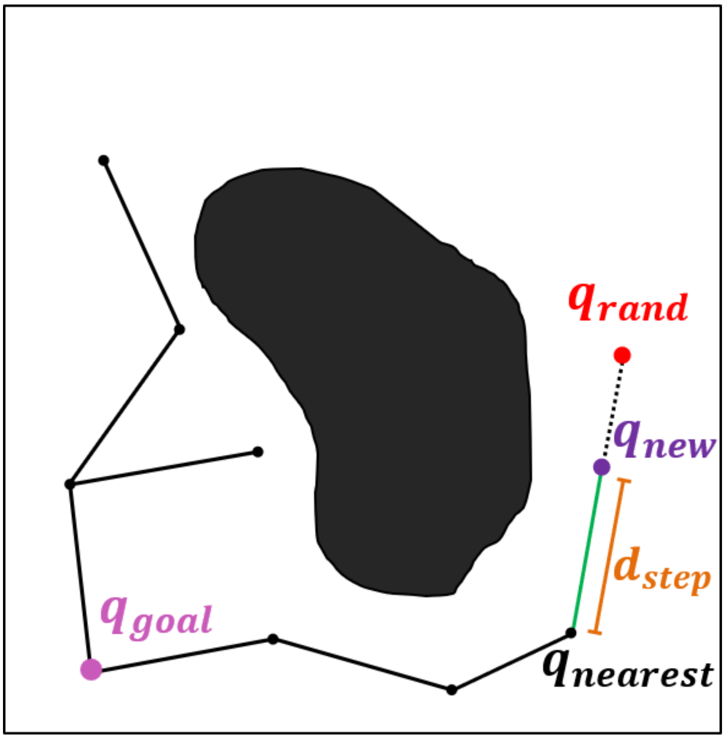


Figure 4.14: Graphical representation of the reinterpreted *NewSteer* function effects on new samples.

RRT *Steer* function is reinterpreted, as there is no need to deviate towards an objective. Instead, the *NewSteer* function ensures that the prescribed maximum (d_{step}) and minimum ($\frac{d_{step}}{2}$) distance boundaries are respected. If the random sample is too close to a tree node, it is discarded. Vice versa, if the distance between the random sample and its nearest node in the tree is higher than d_{step}, then a new sample at distance d_{step} is taken along the line connecting the nearest node and the random sample, as depicted in figure 4.14.

When the new point has been defined, it is compared with all the nodes inside a neighborhood of fixed radius $r = \frac{3}{2}d_{step}$ around it. The new node is connected to the node inside the neighborhood that minimizes its cost c_{opt}, provided that the new edge is not in collision with any obstacle *red zone*. The minimum cost neighbor becomes the *parent* of the new node (see figure 4.15). There is no use in defining a variable neighborhood radius strategy like in RRT*, since the map is sampled at predetermined distances and the maximum number of nodes n_{total} is assigned.

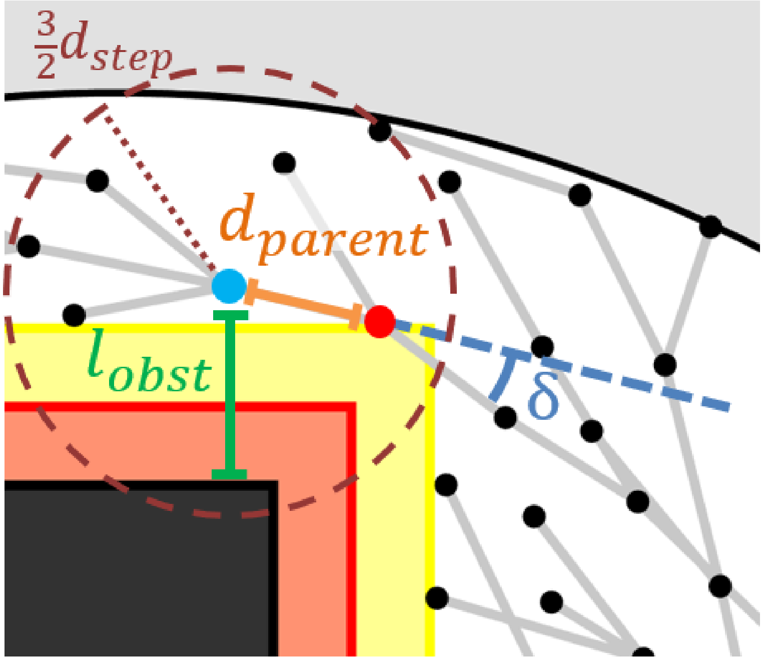


Figure 4.15: Geometrical representation of the three components of the connection cost. The new sample (in blue) is connected to its parent (in red), which is the node inside the neighborhood (brown dashed line) that minimizes its total cost.

The cost function of the connection c_{conn} is chosen as a linear combination of three quantities:

$$c_{conn} = a * d_{parent} + b * acos(\delta) + c * l_{obst}^{-1} \tag{4.1}$$

where d_{parent} is the distance between the node and its *parent*, δ is the angle between the edge connecting the new node to the *parent* and the edge between the *parent* and its parent, l_{obst}^{-1} is the reciprocal of the distance from the new node to the closest obstacle. The coefficients a, b and c represent the weight of each term. The total cost of the new node c_{opt} is obtained like in classical RRT* by adding the connection cost c_{conn} to the parent cost c_{parent}.
All three weighted quantities are positive and monotonically increasing functions, therefore, minimizing this cost function is equivalent to, respectively, reducing the total travel distance, creating a smoother path with less deviation between connected edges and increasing the overall distance from the obstacles. Since these quantities are weighted, the operator can choose to optimize his/her path according to one or more of them and how much they impact on the total cost.

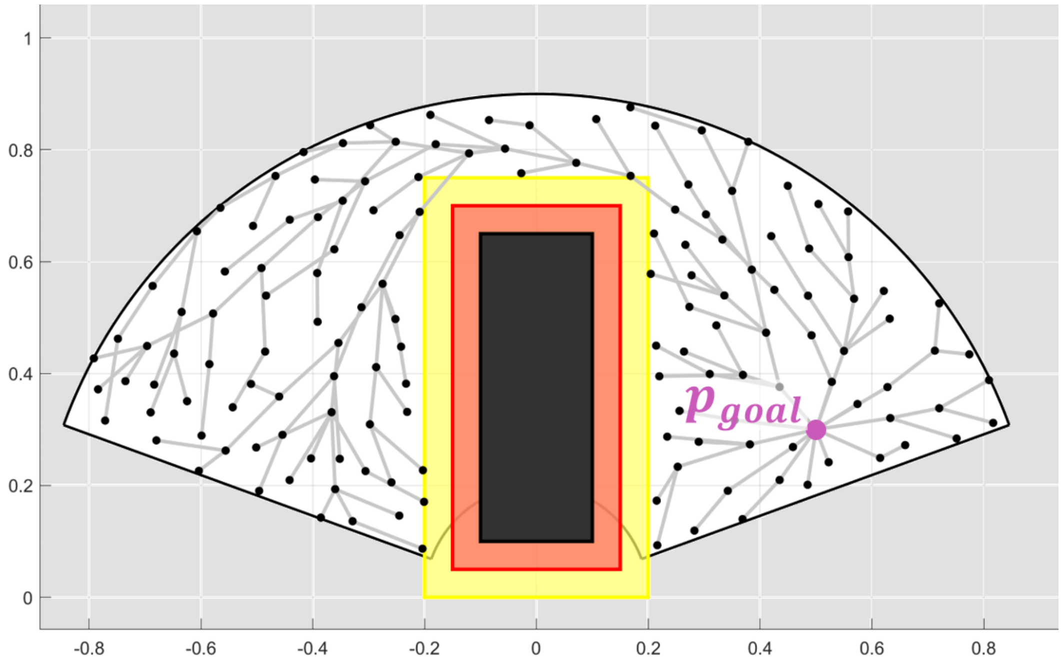


Figure 4.16: Representation of a RRT-based *environment exploring structure* with d_{step} = 0.1 m, created in the same map as the one of figure 4.6.

It is worth mentioning that, since the goal position is the root of the tree, its cost is $c_{goal} = 0$. Moreover, it is impossible to evaluate the angle δ of a node connected to the root. Therefore, its children angular component of the connection cost must be set zero.

Finally, a connection from each node n_i inside the neighborhood to the new node is attempted. These connections are confirmed only if the edge is not in collision and the total cost of the i-th node is reduced. An example of a optimized tree structure is portrayed in figure 4.16.

The computational complexity of the tree creation phase is the same as the classical RRT* algorithm, $O(nlog(n))$, where $n = n_{total}$ is the estimated number of nodes to completely discretize the map free-space.

4.4.3 Online tree analysis for optimal direction selection

The procedure to determine the optimal direction $\boldsymbol{v_1}$ follows the same principles of the grid analysis, presented in section 4.3.2. It consists in finding the best node in the structure considering the optimal path towards the objective, possible collisions and the user intention. The selected node, n_{opt} becomes an intermediate goal position, $\boldsymbol{p_{opt}}$, between the end-effector position $\boldsymbol{p_{ee}}$ and the target position $\boldsymbol{p_{goal}}$. The principal direction for the *damping shape* construction is aligned with the line connecting $\boldsymbol{p_{ee}}$ to the optimal intermediate goal position.

It is worth mentioning that, since the *environment exploring structure* is a tree, it is always possible to reach the final objective by exploiting the *parent-children* relationship between the nodes. Moreover, each node *parent* minimizes the connection cost inside the neighborhood, hence, once a locally optimal node has been chosen, following the direction towards its *parent* is equivalent to follow an optimal path to reach the objective.

Based on these considerations, n_{opt} is selected as the parent of the node n_{best} with the lowest total cost (the sum of the cost of n_{best} and the connection cost with $\boldsymbol{p_{ee}}$) inside a neighborhood of radius r_n around the end-effector position $\boldsymbol{p_{ee}}$.

To include user intention in the selection of the optimal node n_{opt}, only the nodes neighbors to $\boldsymbol{p_{ee}}$ whose parent is sufficiently aligned with the direction of the human force are considered. An angular threshold α_{span} is set: it corresponds to the maximum allowed angle between the force and the optimal direction, i.e. the vector going from $\boldsymbol{p_{ee}}$ to $\boldsymbol{p_{opt}}$.

Moreover, collisions with the obstacles must be avoided: therefore, the segment connecting $\boldsymbol{p_{ee}}$ to $\boldsymbol{p_{opt}}$ must not intersect any of the obstacles *red zone*.

Finally, an additional constraint must be added to avoid proceeding in the wrong direction along the tree: the selected optimal direction must point in the same direction of the vector connecting the best node n_{best} in the neighborhood of $\boldsymbol{p_{ee}}$ and its parent n_{opt}.

A graphical representation of the selection process is provided in figure 4.17.

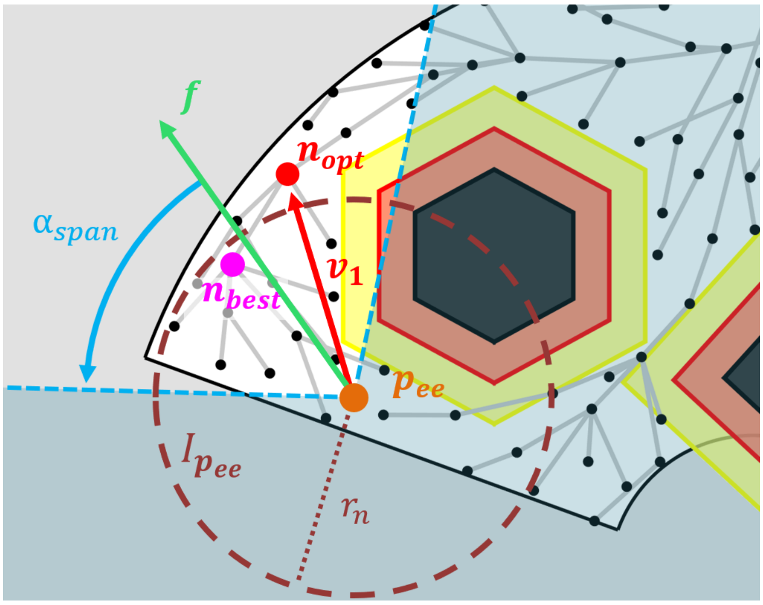


Figure 4.17: Illustration of the optimal direction choice process. n_{best} is the node inside $I_{p_{ee}}$ that minimizes the total cost. Its parent n_{opt} respects all the criteria: the optimal direction $\boldsymbol{v_1}$ is the one towards n_{opt}.

To summarize, a node n_{opt} is selected according to the following criteria:

- **Minimum Cost in the neighborhood**:

$$\begin{cases} I_{\boldsymbol{p_{ee}}} \leftarrow n_i : \|\boldsymbol{p_{ee}} - \boldsymbol{p_i}\| \leq r_n \\ n_{best} \leftarrow n_j : c_j + Cost(\boldsymbol{p_{ee}}, \boldsymbol{p_j}) < c_i + Cost(\boldsymbol{p_{ee}}, \boldsymbol{p_i}), \ \forall n_i \neq n_j \in I_{\boldsymbol{p_{ee}}} \\ n_{opt} \leftarrow Parent(n_{best}) \end{cases}$$

where $Parent(n_k)$ is a function that retrieves the parent of the k-th node of the tree, c_k is the cost of such node, $\boldsymbol{p_k}$ is its position in the map and $Cost(\boldsymbol{p_{ee}}, \boldsymbol{p_k})$ is a function that computes the connection cost between the end-effector position and the k-th node position. $I_{\boldsymbol{p_{ee}}}$ is a subset of the tree nodes that contains all the nodes belonging to the neighborhood of radius r_n around the end-effector position $\boldsymbol{p_{ee}}$.

- **User intention**:
$$\frac{(\boldsymbol{p_{opt}} - \boldsymbol{p_{ee}}) \cdot \boldsymbol{f}}{||\boldsymbol{p_{opt}} - \boldsymbol{p_{ee}}|| \, ||\boldsymbol{f}||} \geq \cos(\alpha_{span})$$
where $\boldsymbol{p_{opt}}$ is the position of the node n_{opt}, selected according to the **Minimum Cost** constraint, $\boldsymbol{f}$ is the vector representing the input force exerted by the human on the end-effector, expressed in global coordinates, and α_{span} is the maximum misalignment threshold for the search.

- **Collision Avoidance**:
$$\nexists P : P \in \overline{\boldsymbol{p_{opt}}\boldsymbol{p_{ee}}} \, \wedge \, P \in red(obst_i), \, \forall i : 1..\#_{obst}$$
where P is a point in the Cartesian space, $\overline{\boldsymbol{p_{opt}}\boldsymbol{p_{ee}}}$ is the segment connecting n_{opt} and the end-effector positions, $red(obst_i)$ represent the red zone around the i-th obstacle of the map.

- **Tree alignment**:
$$\frac{(\boldsymbol{p_{opt}} - \boldsymbol{p_{ee}}) \cdot (\boldsymbol{p_{opt}} - \boldsymbol{p_{best}})}{||\boldsymbol{p_{opt}} - \boldsymbol{p_{ee}}|| \, ||\boldsymbol{p_{opt}} - \boldsymbol{p_{best}}||} \geq 0$$
where $\boldsymbol{p_{best}}$ is the position of n_{best}, which is the node with minimum cost in the neighborhood $I_{\boldsymbol{p_{ee}}}$ (see **Minimum Cost** constraint).

However, few exceptions may occur. When the end-effector is in the objective proximity, the root node may belong to the neighborhood of $\boldsymbol{p_{ee}}$. In this case, user intention should no longer be considered in the choice, since the purpose is to reach the final position and the root has no parent. As a consequence, when $\boldsymbol{p_{goal}}$ is inside the neighborhood and the direction towards it is collision-free, the operator will be redirected directly to the goal position.

Moreover, the **User intention** constraint has to be discarded if no node satisfies it, meaning that no collision-free path is available in the direction that the user has selected. To solve this problem, the selection process is repeated ignoring the intention constraint and analyzing the entirety of the neighborhood, instead of just a portion of it.

Additionally, it is possible to regulate how much the user is forced to follow the optimal path. This is achieved by adjusting r_n, the radius of the neighborhood considered in the constrained minimization. The minimum value admissible as radius is d_{step}, which correspond to the maximum distance between two connected nodes in the tree. Choosing a value inferior than d_{step} may lead to an empty neighborhood, meaning that no node is available for the selection.

Increasing r_n, instead, is equivalent to including additional nodes in the selection process. Therefore, the probability of finding a feasible path aligned with the input force increases.

If the value of the neighborhood radius is low, a local search is enforced and its more probable to choose the optimal node, ignoring the user intention constraint. As a result the operator would be redirected on a path of optimally connected nodes. If r_n is larger, then it is more probable to accommodate the user intention: the operator will follow a smoother but sub-optimal path. It is also more probable that the tree root belongs to the neighborhood, hence, the user is redirected faster to the goal provided that a linear and collision-free path towards $\boldsymbol{p}_{goal}$ is available.

The main drawback of a larger neighborhood is an increased computational complexity. The more nodes are included in the selection, the more constraint checks are required to find n_{best}.

The computational complexity of the tree analysis phase is $\mathcal{O}(n)$, where $n = n_{total}$ is the number of nodes in the tree.

4.5 Control strategy

The proposed manual guidance control strategy can be subdivided into two parts. The main part correspond to the online control loop: variable admittance control is adopted to translate the joints position and velocity ($\boldsymbol{q}, \dot{\boldsymbol{q}}$) and the force measurement from the sensor ($\boldsymbol{f}_s$) into a new position and speed reference ($\boldsymbol{q}_{ref}, \dot{\boldsymbol{q}}_{ref}$) for the robot low-level controllers to follow. It is fundamental to reduce the computations performed in the online part, since the loop time must be lower than robot controller sampling period.

The second part consists in a set of operations that are executed offline, before the motion inside the map. In this case there is no restriction on the algorithm complexity and execution time. This phase is used to construct an *environment exploring structure*, as described previously in this chapter. The distinction between these two parts allow to perform only a minimal part of path planning and collision avoidance at run-time, where the resources are limited.

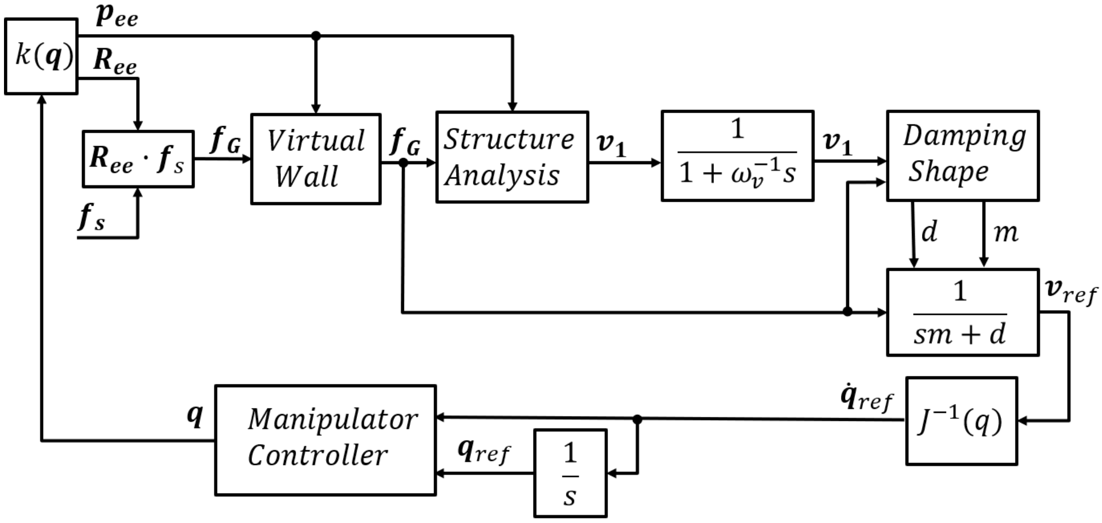


Figure 4.18: Complete block scheme representation of the online control loop.

In figure 4.18 a complete scheme of the online control loop is sketched.

First, the input force measured by the sensor ($\boldsymbol{f}_s$) is translated into the robot base (global) reference frame ($\boldsymbol{f}_G$). Then, the virtual walls effect is taken into consideration and the force input is scaled accordingly. The position of the end-effector is compared with every obstacle and the component of the force which is normal to the closest side of the obstacle polygon is scaled according to a function inversely proportional to the distance between the two. This works as a safety feature: if the operator decides to ignore the feedback provided by the variable admittance strategy, its motion is slowed down and eventually stopped, preventing the collision with the obstacle.

Then, an *environment exploring structure* is analyzed to determine the locally optimal direction to follow $\boldsymbol{v}_1$, taking into account the position of the end-effector inside the cluttered environment and the intention of motion of the operator. This output is filtered with a low-pass filter to avoid discontinuities in the direction value that can result in a discontinuous selection of the damping parameter. Based on the filtered optimal direction, the damping shape is constructed using the procedures described in Chapter 3.

Finally, the shape is compared with the direction of the input force: the damping coefficient d of the admittance filter is computed according to Chapter 3, while the mass value m is chosen as $\frac{1}{10}$ of the damping in order to keep the filter bandwidth constant. The output of this filter is a Cartesian speed reference for the end-effector, and Differential kinematic inversion is applied to obtain the final joint speed and position reference for the robot.

The selection of the optimal direction is based on the analysis of an *environment exploring structure*. The purpose of this structure is to discretize the available space, providing a number of key-points (nodes) distributed in the free-space. When the robot moves, the structure is analyzed in order to detect the best node to be reached: the optimal direction is selected as the unit vector pointing towards that key-point in the structure. The structure generation is based only on the map, starting position and objective, hence it is performed in the secondary offline phase, executed before the beginning of the motion. Moreover, it is possible to reuse the computed structure for different movements, as long as the objective position and the layout do not change.

The distinction between creation and exploration allows to drastically reduce the computations performed online. In fact, since the heavier map exploration is executed offline, it is only necessary to perform collision avoidance procedures in a neighborhood of the end-effector position, reducing the loop time of the controller and improving the quality of the human-machine interaction. Another advantage is that it is possible to perform more complex optimizations during the structure creation phase, improving the quality of the planned path.

Chapter 5

Experimental results

5.1 Experimental setup

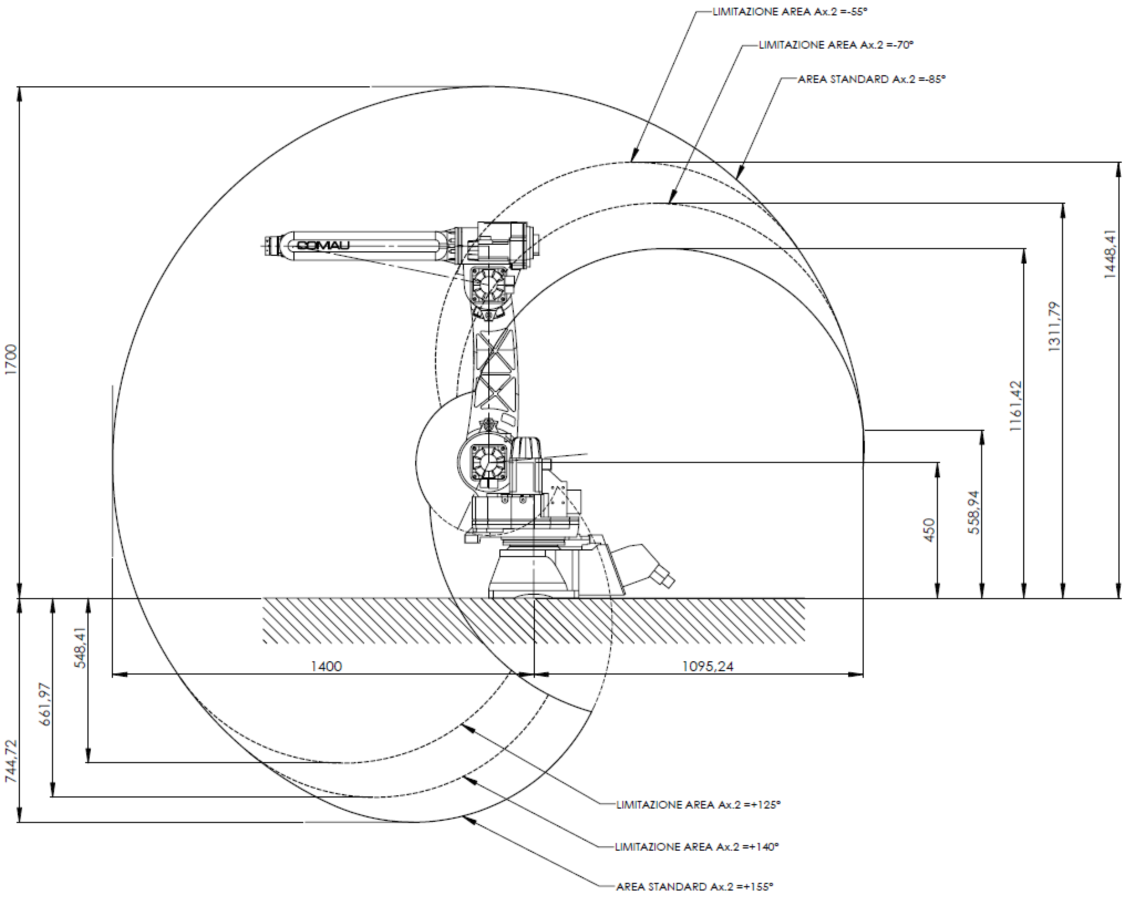

LIMITAZIONE AREA Ax.2 =-55°
LIMITAZIONE AREA Ax.2 =-70°
AREA STANDARD Ax.2 =-85°
COMAU
1700
1448,41
1311,79
1161,42
558,94
450
1400
1095,24
548,41
661,97
744,72
LIMITAZIONE AREA Ax.2 =+125°
LIMITAZIONE AREA Ax.2 =+140°
AREA STANDARD Ax.2 =+155°

Figure 5.1: Representation of the Comau Smart Six workspace.

The experiments are conducted on a Comau Smart Six manipulator. This industrial manipulator, commonly adopted for welding, assembly and packaging operations, has a wrist payload of 6 kg and a maximum arm extension of 1.4 meters: its working area, depicted in figure 5.1, is fully exploited to simulate a realistic manual guidance experience, where the user has to drive the robot end-effector inside the working cell not only with his/her arm, but also by following it with the entire body.

The robot is endowed with a Comau C4G controller which, in its Open configuration, is linked through an Ethernet connection to a real-time external PC running RTAI-Linux extension. In Open modality, the trajectory planning algorithms of the C4G are bypassed and the low level joint speed and position control loops are preserved, in order to realize the references computed by the external pc. The two devices communicate at frequency of 500 Hz: each 2 ms the joint measurements are transferred from the C4G, acting as a client, to the PC, which is the server.

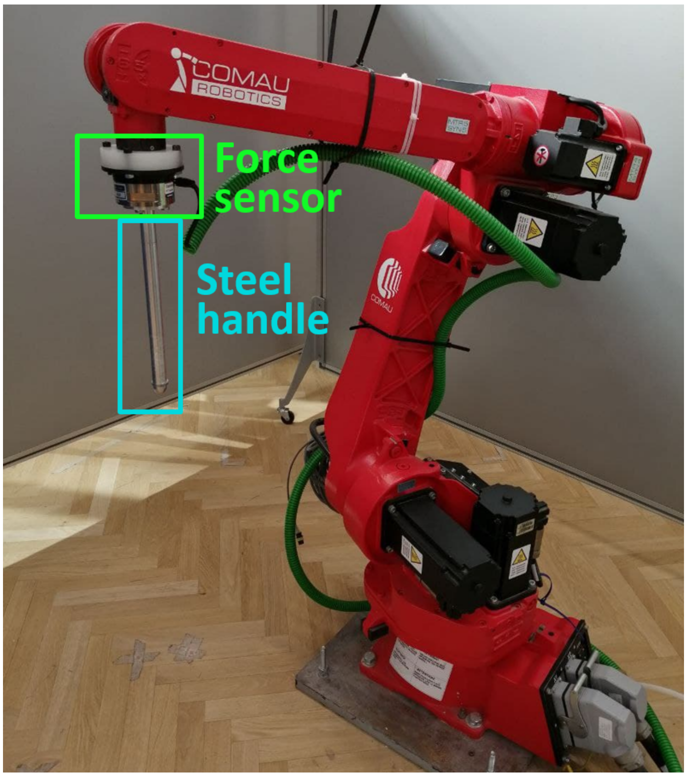


Figure 5.2: The Comau Smart Six manipulator equipped with the F/T sensor and the steel handle.

An ATI 6-axis Force-Torque sensor is attached to the manipulator end-effector and a steel handle is firmly connected to it, as it can be seen in figure 5.2. The sensor is linked to the PC through a DAQ board, managed by the RTAI system, and has a sample frequency of 10 kHz.
The PC enacts the strategies described in Chapter 3 and 4: the joint data is combined with the force measurements coming from the sensor to compute new position and speed references. These references are sent back to the C4G controller, which closes the inner control loop on the manipulator axes. The program runs in hard real-time on the PC, meaning that it is perfectly in sync with the robot controller and its execution time is kept under the 2 ms connection limit.
For each experiment, the user is asked to grab the steel handle and conduct the robot end-effector towards a specific point inside the workspace. Since the developed approaches concern only translations of the end effector, the wrist is always kept perpendicular to the ground and the rotations of the handle are blocked. As mentioned in Chapter 3, obstacle avoidance is implemented considering a bi-dimensional simplification of the environment. Hence, only the X and Y directions are affected by the variable admittance control strategy. Along the Z axis, an invariable admittance filter is implemented. Constraining obstacle avoidance to a horizontal plane provides more meaningful results, preventing the height of the operator from influencing the outcomes. The shape of the map is heavily constrained by the dimensions of the robot operative space. The task must be executed at a comfortable height for the operator, hence the end-effector travels at an average of 1.25 meters on the Z-axis, where the manipulator reach is reduced. Moreover, configuration singularities should be considered in the map limits: moving the end-effector too close or too far from the robot base may lead to problems during the kinematic inversion, causing abrupt changes in the robot speed that can harm the operator. As a result, the motion must be constrained between a specified minimum and maximum distance from the base. Consequently, if a rectangular map were to be used, its dimensions would be reduced to account for the operator safety. On the contrary, circular maps better exploit the shape of the robot operative space, hence they are adopted for every experiment.

Simple polygons represent the obstacles in every bi-dimensional map, such as rectangles or hexagons. These kind of shapes approximates the impediments that are commonly found in real working environments, like pallets, machines or pillars. Moreover, a 5 cm thick *red zone* surrounds each obstacle: it has the safety purpose described in Chapter 4 and it cannot be invaded. A new collision will be counted each time the end-effector enter such zone. Additionally, a 5 cm thick *yellow zone* is generated around each *red zone*. Its purpose is to prevent the creation of trajectories that would lead the operator too close to the *red zone*.

5.2 Experimental campaign

Three sets of experiments involving fourteen volunteers were performed to validate the variable admittance control techniques presented in Chapter 3 and 4.

In the first set, grid-based (**VAF-GRID**) and tree-based (**VAF-RRT**) approaches are compared to invariable admittance filter (**IAF**) in a standard manual guidance task, executed by an operator which is fully aware of the surrounding working environment. The volunteer is asked to reach a specific point in the workspace as precisely as possible and in a reasonable amount of time, avoiding the obstacles that are present in the environment. The objective position is indicated with a mark on the ground and the obstacles are realized by means of cardboard boxes. The volunteers are requested to complete three round trips from the starting position to the target one. The obtained results are examined in terms of *Execution time*, *total path length*, *required energy* and *positioning error*. A detailed comparison between the performance achieved by the two developed techniques and the classical invariable admittance filter is carried out.

The second set of experiments aims at demonstrating the advantages of the developed methods in more difficult transportation tasks. Like in the first experiment, the volunteers are asked to reach three times a specific point in space, marked on the ground, in presence of three different cluttered maps. However, the obstacles that are present in the environment are invisible and unknown to the human. This set-up allows to simulate a situation where the operator field of view is obstructed by the dimensions of the transported object. **IAF**, **VAF-GRID** and **VAF-RRT** are compared in terms of *number of collisions* occurred during task execution and *energy required* to complete it.

Finally, the third set of experiments aims at exhibiting the full capabilities of the **VAF-RRT** method for the transportation of bulky objects in a complex workspace, where **VAF-GRID** and **IAF** would inevitably fail. The volunteers have to overcome three different maps containing convex shaped obstacles and dead-ends. Both obstacle layout and target position are unknown, hence the human can only rely on the haptic feedback provided by the robot via **VAF-RRT** to successfully accomplish the tasks. The *number of collisions* occurred and the final *positioning error* will be considered as performance indicators. Additionally, in this third set of experiments, tree based manual guidance is tested with two different degrees of user autonomy. On one hand, high autonomy RRT (**RRT**), which is the same algorithm used in the previous experiments, has a large search radius to better adapt to the user intention of motion. On the other hand, optimal RRT (**OPT-RRT**) adopts a minimal neighborhood radius for the selection process, so it restricts the operator freedom of motion and constrain him/her on an optimized path. After each set of experiments, the volunteers are asked to evaluate their experience in a survey.

5.3 First experiment

For the first set of experiments, the volunteer is asked to guide the end-effector from a given starting position to a specific destination. A realistic work cell layout is simulated by placing two cardboard boxes between the start and goal position, as it can be seen in figure 5.3. The two boxes emulate obstacles laying inside the workspace, that the operator has to avoid to prevent damages to the transported cargo. A small flag on the ground marks the objective position that the human has to reach as accurately as possible. When he/she is satisfied with the final positioning, then he/she has to return to the starting point. This process is repeated for a total of three round trips to gather sufficient data for the analysis.

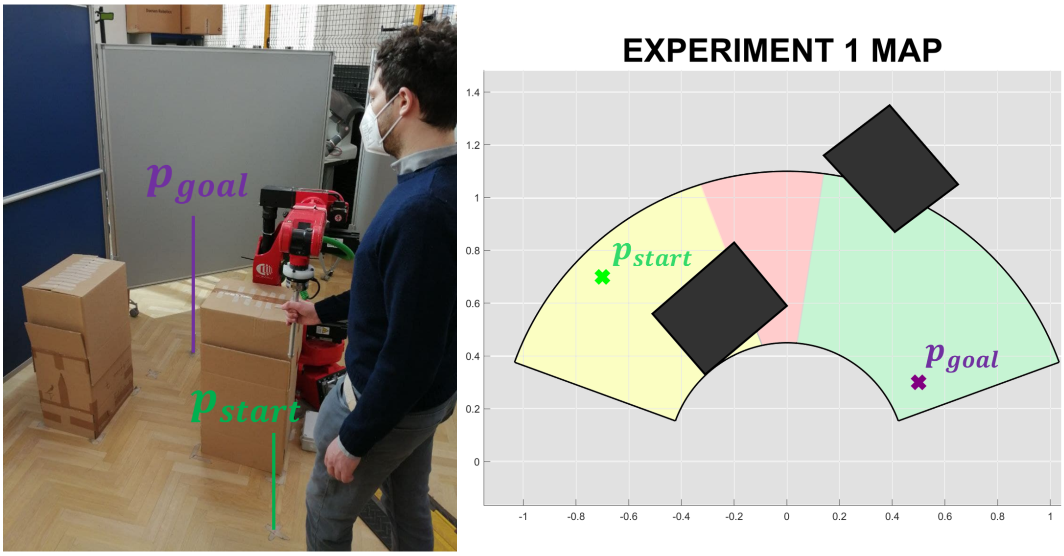


Figure 5.3: To the left, a volunteer performing the first experiment in the realistic obstacle layout. To the right, the 2D virtual representation of the workspace.

At first, the user adopts invariable admittance filter (**IAF**). This translates the human force in a displacement of the end-effector aligned with it, independently from the obstacle layout. This case corresponds to the classic control strategy adopted in manual guidance, where the trajectory planning task is left entirely to the human and the robot reduces uniformly his/her effort. An average damping value of $d_{cost} = 100Nm/s$ is adopted for the constant admittance filter: it represents a good compromise between the ease of motion obtained with a low damping coefficient ($d_{min} = 50Nm/s$) and the precision in final positioning, improved with high damping ($d_{max} = 1000Nm/s$).
Then, the experiment is repeated on the same map with the two variable admittance control strategies conceived in this thesis:

1. **Grid-based Variable Admittance Filter (GRID-VAF)** with:
   - Construction parameters: $n_b = 9$, $n_{lateral} = 2$, $n_{over} = 2$, $n_{sg} = 3$, $l_{max} = 2$
   - A $10cm$ thick area zone around the *yellow zone* of each obstacle, for subgrid creation.
   - Exploration parameters: $s_f = 2$

2. **RRT-based Variable Admittance Filter (RRT-VAF)** with:

   - Construction parameters: $d_{step} = 0.1\,m$, $weights = [10 \quad 5 \quad 1]$ (respectively for parent distance, alignment, obstacle distance)

   - Exploration parameters: $r_n = \infty$ (all nodes are considered in the search), $\alpha_{span} = 20°$.

These variable approaches provide a directional haptic feedback to the user during the navigation towards the objective. They allow the human to discriminate between the directions that are available and the ones that lead to collisions by perceiving different resistances to the end-effector motion. The damping coefficient is computed through the variable admittance algorithms, while the mass parameter is always evaluated as $m = d/10$. In this way, a $10\,rad/s$ bandwidth is enforced for the admittance filter while the operator is maneuvering the robot, resulting in a more intuitive collaboration. The purpose of this test is to compare the developed approaches to the invariable admittance control strategy when applied to the classical manual guidance problem, which assumes that the operator is fully aware of the surroundings.
Execution time, *total path length*, *required energy* and *final positioning error* of the results are analyzed to determine if the variable admittance strategies improve the quality of the collaboration task. At the end of the experiment the users are asked to fill-in a survey concerning their impressions on the assisted guidance. For each of the three proposed control strategies, the user has to answer a set of five questions which topics are described in the following:

- **Q1**: The amount of physical effort perceived during task execution.

- **Q2**: The ease in reaching the objective and avoiding all the obstacles.

- **Q3**: The robot motion coherence with respect to human intention.

- **Q4**: The perception of the robot assistance in avoiding all the obstacles.

- **Q5**: The perception of the robot assistance in the final positioning.

5.3.1 Outcomes

In this sub-section the performances of **IAF**, **VAF-GRID** and **VAF-RRT** are compared in therms of execution time, required energy, total path length and final positioning error. Additionally, since the trajectories followed by the different volunteers are very similar, the evolution of the damping coefficient along the path will be analyzed, highlighting the peculiarities of each control technique.

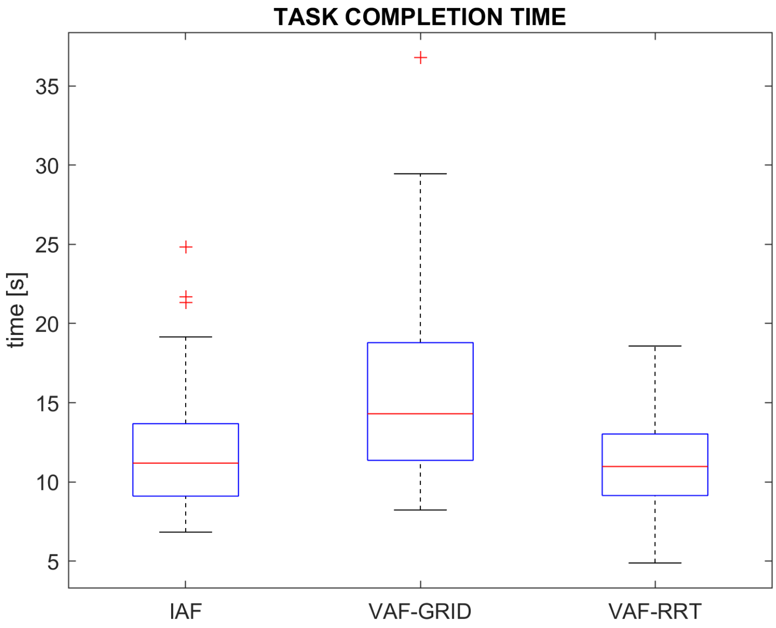


Figure 5.4: Statistics of the execution time for the first experiment.

As it can be seen in figure 5.4, there is not a significant difference between **IAF** and **VAF-RRT** in the amount of time required to complete the task. However, the adoption of **VAF-GRID** causes a 25% increase in average completion time. These results highlight the behavioral difference of the two variable admittance approaches. **VAF-GRID** is a simpler approach, its purpose is to guide the operator away from the obstacles while driving him/her along the grid principal direction, hence it tends to redirect the user in a not always straightforward and intuitive way. As a consequence, additional time is spent understanding the haptic feedback. On the other hand, **VAF-RRT** is a more accommodating approach: there is not a principal motion direction like in the grid case, and it indulges the user intention of motion, reducing the execution time. Similar considerations can be made concerning the average path length followed by the operator.

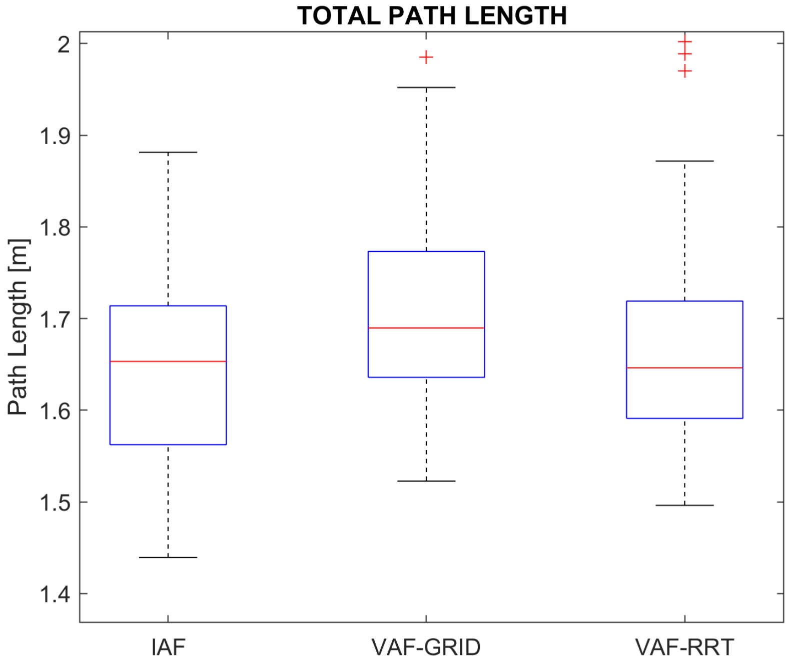


Figure 5.5: Statistics of the total distance traveled in the first experiment.

Inspecting figure 5.5, it can be noticed that the distance traveled by the end-effector from the start to the goal position is equivalent in case of **IAF** or **VAF-RRT**. With **VAF-GRID**, instead, the average path length slightly increases. The grid-based variable admittance control tends to drive the end effector further from the obstacles than the RRT-based approach, increasing the traveled distance and, consequently, the time required to complete the task.

The advantages of the variable admittance control techniques with respect to the invariable damping coefficient become apparent when the human energy required to complete the task is considered.
From figure 5.6 it can be observed that the human effort required to execute the manual guidance task under the **IAF** approach is higher than the one needed with the variable admittance strategies. A 25% reduction of the human energy is obtained with **VAF-GRID** and a 35% decrease with **VAF-RRT**.

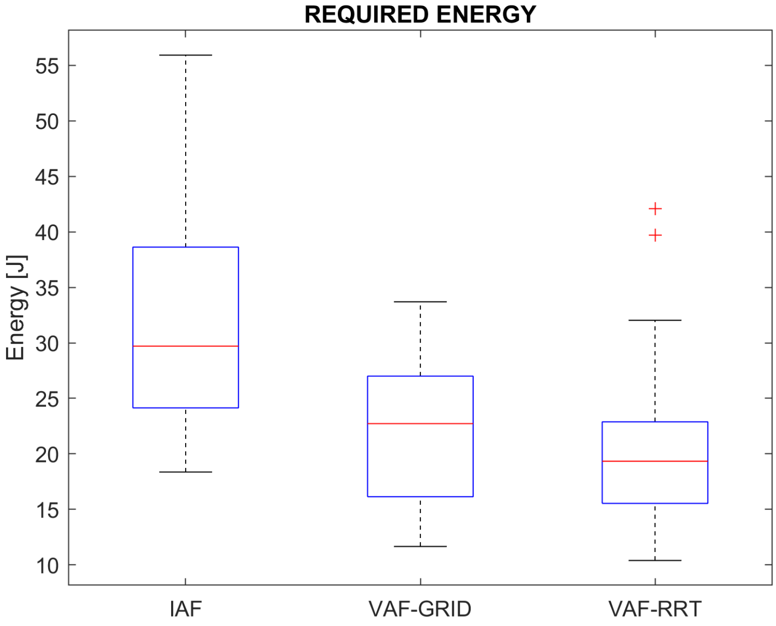


Figure 5.6: Statistics of the energy consumed by the operator during the execution of the first experiment.

Although the **VAF-GRID** results seem in contrast with the execution time and path length outcomes, it is possible to explain the energy reduction by analyzing the interaction between the human and the robot. In fact, when the operator perceives a negative directional feedback, he slows down the motion and explores other directions. When he discovers the minimum effort direction he adapts to the robot suggestion and he is awarded with a lower resistance to the motion. As a consequence, the total energy is reduced, while the path length and execution time increase.

An even greater reduction in human effort is obtained with the **VAF-RRT** approach, due to the interaction mechanism mentioned before. In RRT-based manual guidance, however, the human intention of motion has a greater impact on the selection of the minimum effort direction, as described in Chapter 4. The result is a mutual adaptation between the human and the robot that ensures minimal resistance for the majority of the path described by the end-effector, reducing the total required energy accordingly.

To better understand the differences between the two variable admittance techniques, it is worth examining the behavior of the damping coefficient during the execution of the task. This is possible since the trajectories followed by distinct users in a completely visible environment are very similar.

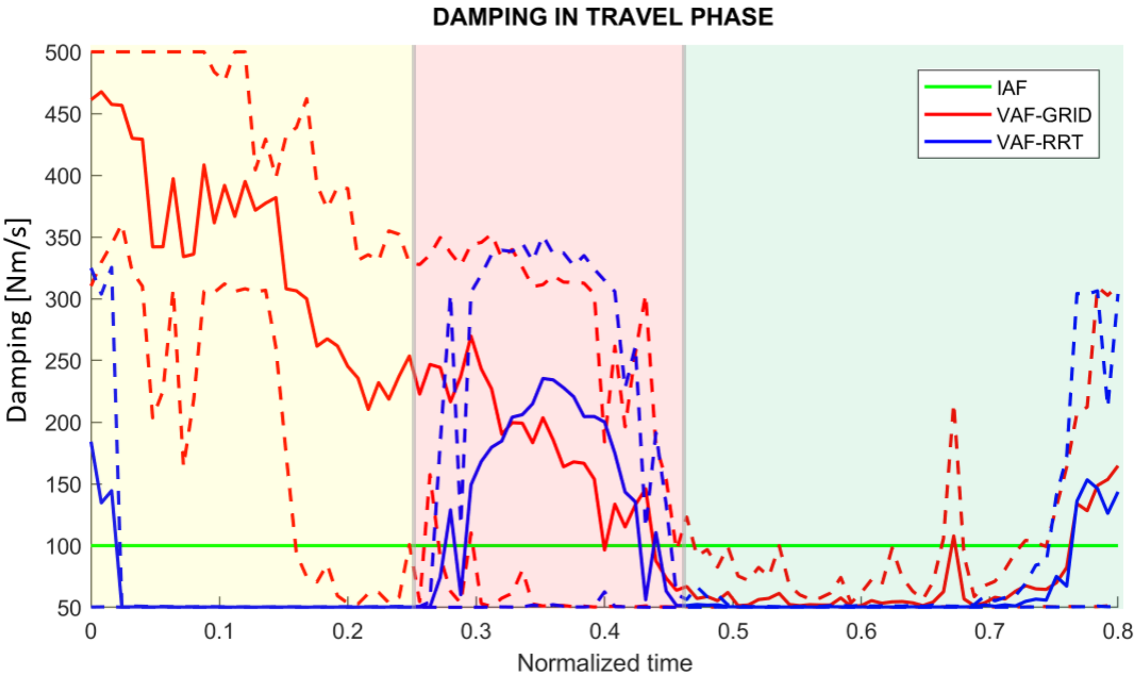


Figure 5.7: Evolution of the mean values (continuous line) of the damping parameter, along with its 25th and 75th percentiles (dashed lines), in a normalized time scale during the travel phase of the first experiment. The background colors refer to the different sectors of the map portrayed in figure 5.3 on the right.

Such comparison is depicted in figure 5.7, where the average damping coefficients computed by the three different control strategies are reported with continuous line on a normalized time scale. The 25th and 75th percentiles are represented by means of dashed lines with the same colors. In this graph, only the travel phase is shown, namely it spans from the instant when the operator grabs the end-effector at the starting position to the moment when he/she reaches, for the first time, a point within 7 cm from the goal position. The plot is subdivided into three zones, identified by different background colors, which correspond to the ones used in figure 5.3 to highlight the specific sector of the map associated with the damping trends of figure 5.7.

In the first sector, painted in yellow, the user has to move slightly on the left side in order to drive the robot end-effector towards the corridor formed by the two box obstacles, avoiding the first one on the right. As it can be seen in figure 5.7, **VAF-GRID** and **VAF-RRT** behave very differently in this situation. Remember that grid-based obstacle avoidance relies on a structure aligned with the line connecting the starting and goal position, and works by guiding the operator forward along that direction. In this layout, an obstacle is placed sideways with respect to the grid principal direction, hence the user is first brought forward and then redirected along the obstacle side. The result is a non-intuitive interaction with the user, since he/she would prefer a smoother, curved trajectory that starts sideways and overcomes the obstacle by crossing closer to its angle. This situation highlights one of the major disadvantages of **VAF-GRID**: motion is not facilitated in the direction which is normal to the grid principal direction, and, in general, a higher damping factor is chosen. Moreover, this section of the graph is characterized by high dispersion, caused by the different reactions of the volunteers. Some of the users explore the different directions and adapt faster to the robot, decreasing the damping, while others try to force the movement in their preferred direction, increasing it. On the opposite, **VAF-RRT** is based on a mutual adaptation with the human: the selection of the minimum effort direction is strongly based on the intention of the operator, estimated from the force exerted on the end-effector. The user is accommodated as long as he/she drives the robot along a feasible path towards the goal. Hence, in the first part of the movement, the values assumed by the damping coefficient are very close to the minimum.

The second section of the graph, highlighted in red, corresponds to the sector of the map where a change in direction must occur. The user has moved to the side of the obstacle and the goal position is in view, hence the variable admittance control strategies start redirecting him/her towards the destination by increasing the damping factor. This behavior is particularly evident in the **VAF-RRT** trend, where the damping coefficient is increased to make the operator aware of the new optimal direction and then decreased once he/she exits the red sector and starts moving towards the goal position.

In the final part of the motion (green background color), the user travels along an almost straight path towards the objective. The damping is minimized in both cases, reducing the human effort and the wasted energy. This trend remains constant for the last part of the travel phase, until the goal position is in proximity: at that point the damping value increases again to improve the final positioning precision as it can be noticed in figure 5.8.

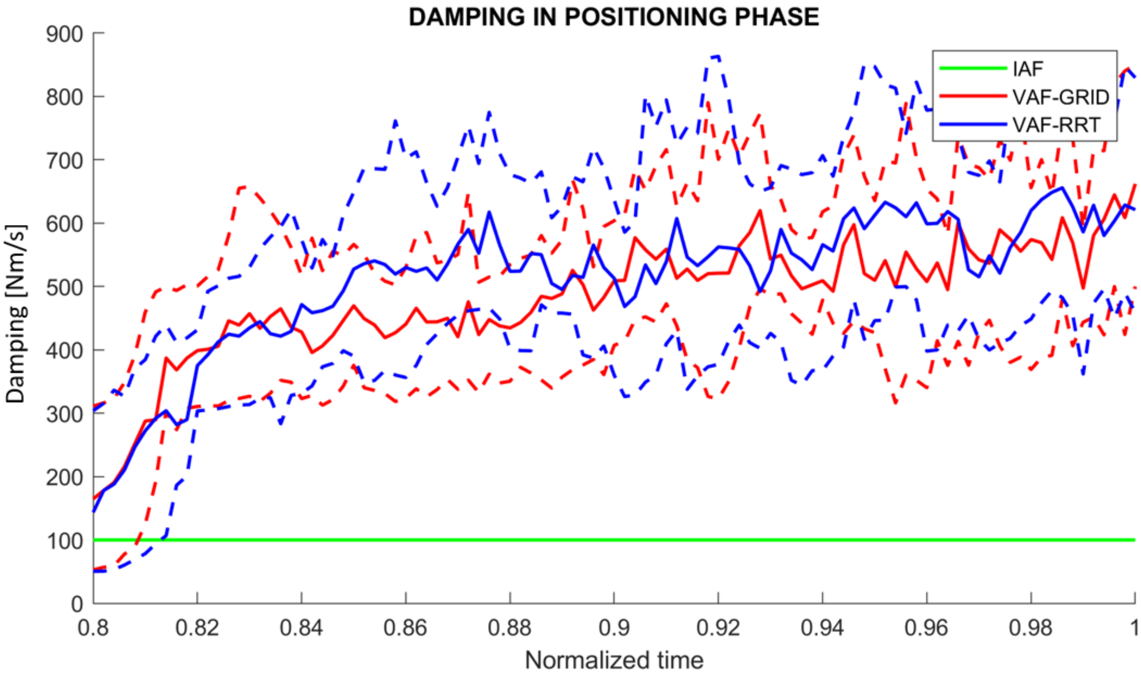


Figure 5.8: Behavior of the mean values (continuous line) of the damping parameter, along with its 25th and 75th percentiles (dashed lines), in a normalized time scale during the positioning phase.

When the user has overcome the obstacles and reached the destination zone, variable admittance control strategies increase the damping coefficient, while **IAF** maintains a constant value. Such increase is justified by the vicinity of the objective: in fact, the purpose of both **VAF-GRID** and **VAF-RRT** is to reduce the final positioning error by exploiting the directional feedback provided to the user. Since the interaction is passive, the operator achieves the final positioning through small circular explorations, while he is constantly redirected towards the target position. The more he/she reduces the error, the higher are the damping factors perceived when drifting away from the goal: this is represented in the graph as a wave trend which tends to increase in time. Despite the fact that the *positioning* phase lasts about 20% of the total execution time, its impact on the total energy is negligible, since the exploration movements performed in this phase are characterized by a reduced input force and velocity.

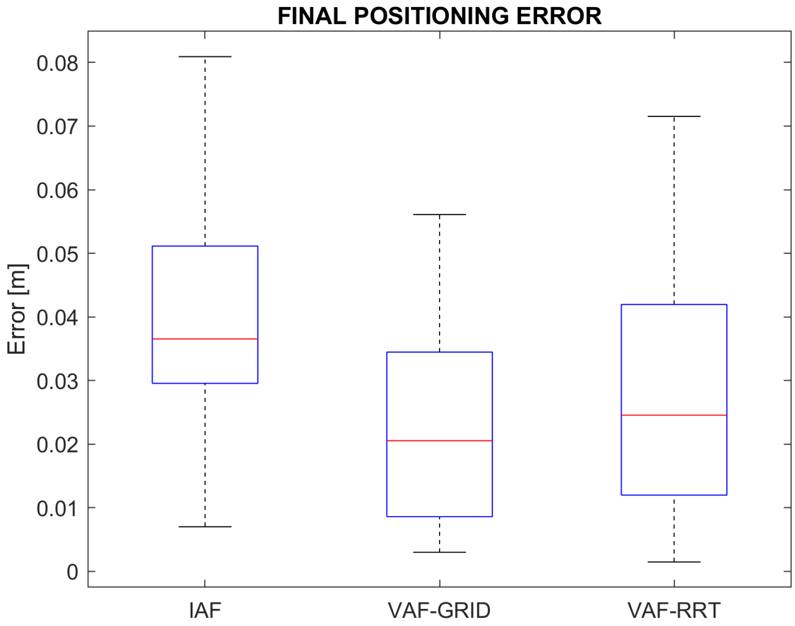


Figure 5.9: Statistics of the final positioning error in the first experiment.

The advantages of the variable admittance control feedback can be also observed when the average positioning precision is considered. Figure 5.9 shows that a 40% reduction in average positioning error is obtained by choosing **VAF-GRID** and **VAF-RRT** over **IAF**. The precision of the operator is increased considerably, since he no longer relies only on his/her visual feedback, which is sensible to perspective distortions, but he/she is helped also through the haptic feedback provided by the robot.

At the end of this set of experiments, the volunteers were asked to fill-in a small survey concerning their impressions about the three proposed control strategies. For each of the five questions, reported in section 5.3, they were asked to rate (in a 1 to 10 scale) their collaboration experience.

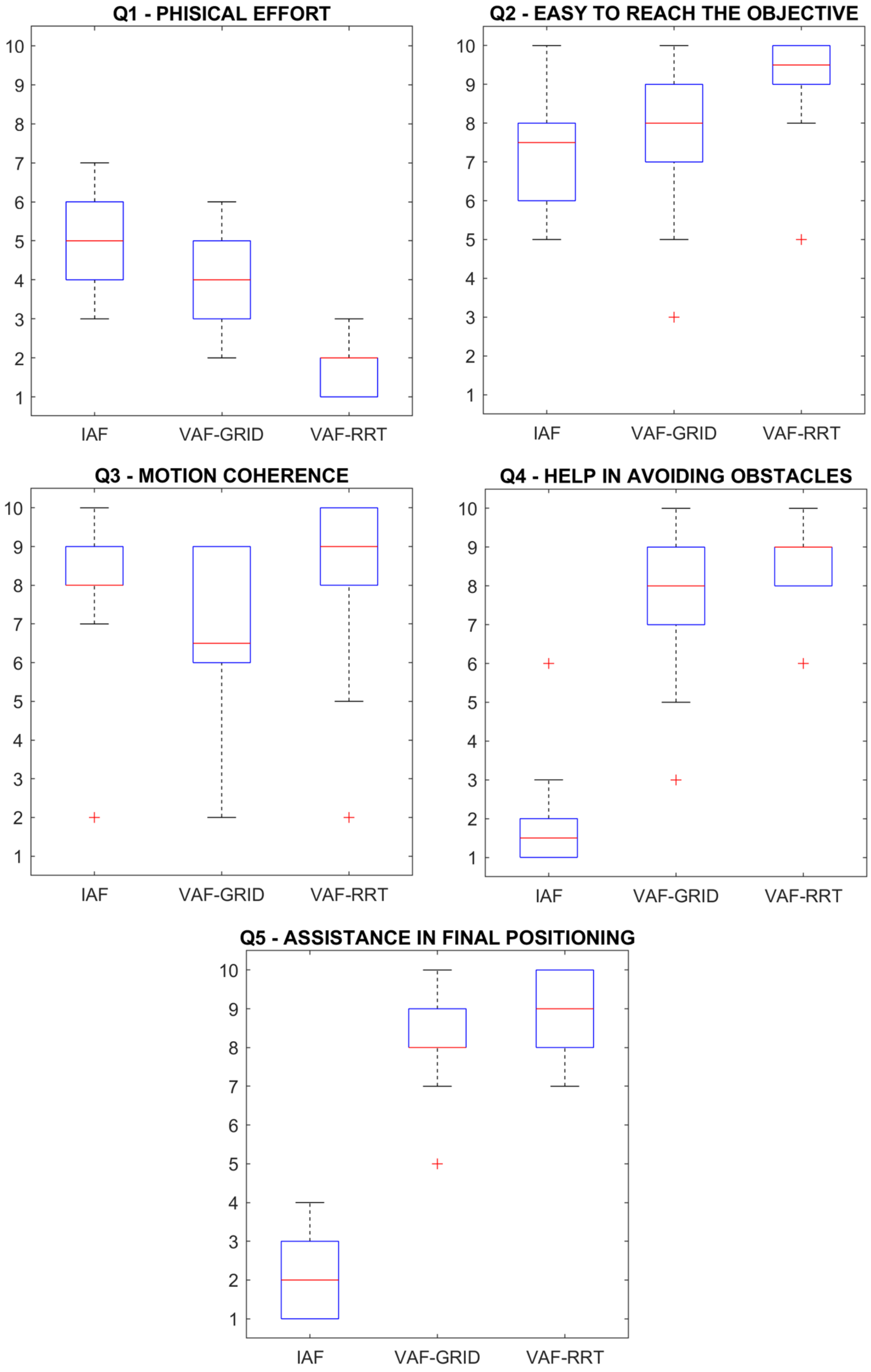


Figure 5.10: Statistics of the answers provided by the volunteers at the end of the first experiment.

The results of the survey are reported in figure 5.10, subdivided into five graphs comparing the three different approaches. The answers provided by the users confirm all the observations expressed in this section that, so far, were based only on the analysis of the experimental results. From **Q1** it is evident that the effort perceived by the operator is lower when one of the variable admittance control technique is in use. **Q3** highlights the negative impact of the **VAF-GRID** limitations on the quality of the interaction: the intention of the operator in the first part of the movement result incoherent with the feedback provided by the robot. From **Q4** and **Q5** it possible to conclude that the user clearly perceives the benefits of the assisted guidance in keeping the end-effector away from the obstacles and in helping him/her with the final positioning.
In general, from the analysis of the opinion of the volunteers, it is apparent that **VAF-RRT** is superior than the other two approaches, concerning the reduction of the human fatigue, the coherence with user intention and the assistance provided during the task execution.

5.4 Second experiment

The outcomes of the first set of experiments highlighted the features of the obstacle avoidance approaches applied to the classical manual guidance problem. The user was fully aware of the surroundings, hence he could plan a trajectory even in the absence of a feedback from the robot. However, the purpose of this thesis is to allow the operator to reach the goal position even in case the bulkiness of the transported object obstructs his/her field of view. For this reason, the second set of experiments aims at validating the developed strategies in this harder guidance task. To achieve this objective, an unknown and invisible working environment is implemented, and the volunteer is asked to reach a specific point in the workspace, marked on the ground like in the first experiment, minimizing the collisions with the invisible obstacles in the map. Nevertheless, it is not recommended to execute the task while blindfolded for obvious safety reasons. Therefore, the obstruction are generated only in the 2D map and not recreated in

reality: collisions are reproduced by surrounding all obstacles with virtual walls, hence if the end-effector is driven too close to the obstacle perimeter, then it is slowed down and stopped, simulating an impact.
Three different scenarios were created, with increasing level of difficulty. The first one includes a simple *gate*, i.e. a narrow passage between two obstacles (see figure 5.11).

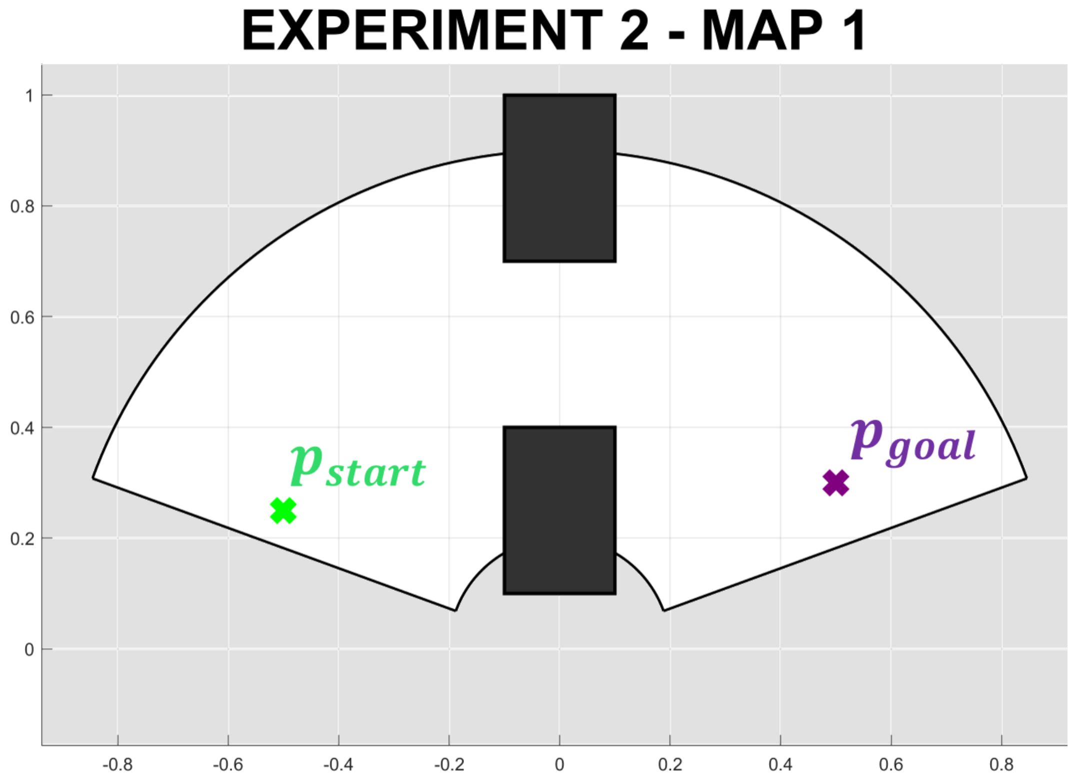


Figure 5.11: Example of an imaginary scenario for the second experiment, belonging to the first level of complexity.

The second presents two obstacles that force the user to change the direction twice (shown in figure 5.12).

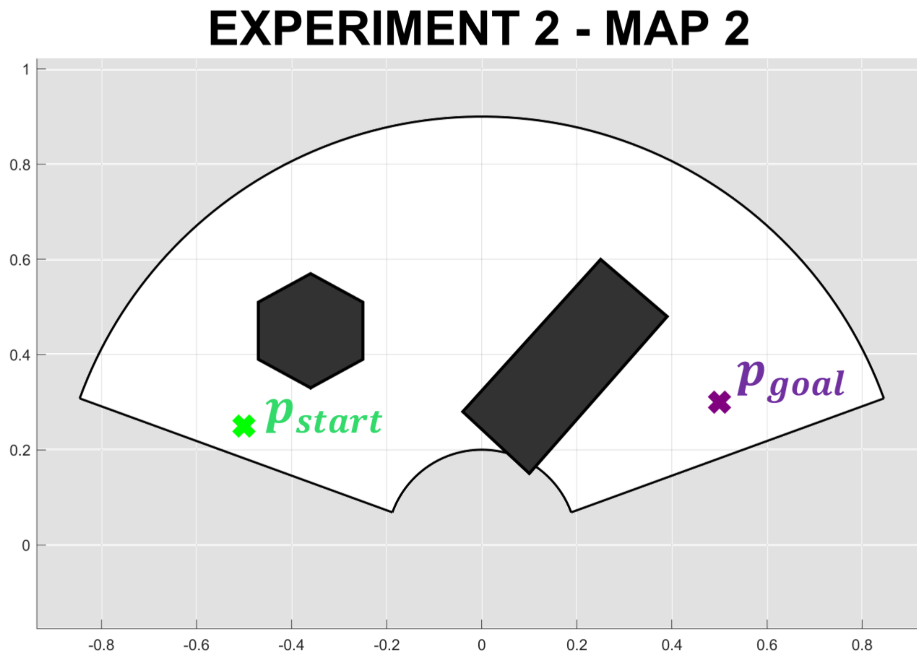


Figure 5.12: Example of an imaginary scenario for the second experiment, belonging to the second level of complexity.

The third, portrayed in figure 5.13, contains both a *gate* and another obstacle positioned sideways, creating a dead end where the operator might get stuck.

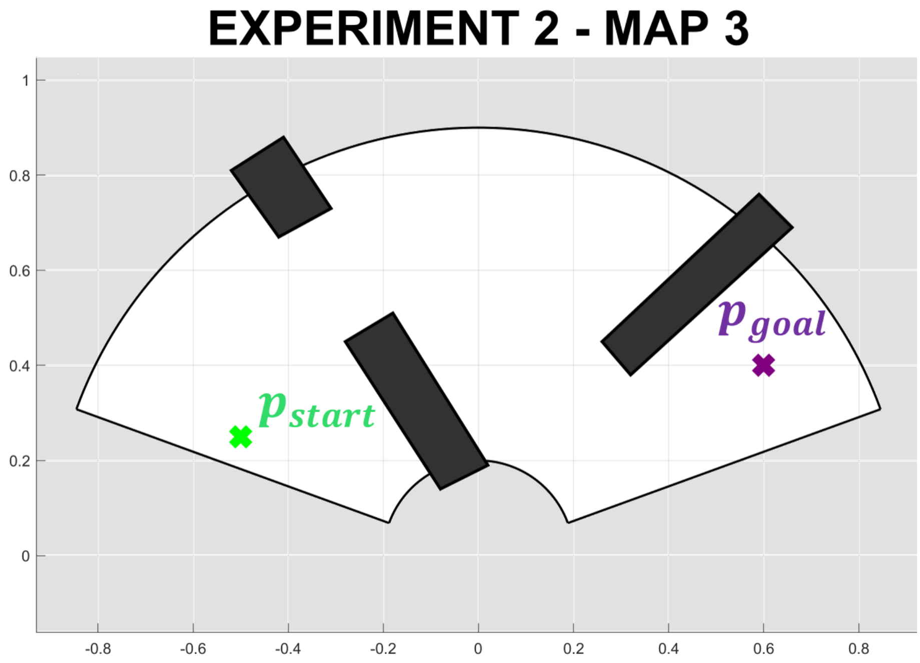


Figure 5.13: Example of an imaginary scenario for the second experiment, belonging to the third level of complexity.

The user is asked to reach the objective as accurately as possible and with the least number of collision. To minimize the impacts the operator must rely on the feedback provided by the robot.
The experiment is repeated one time for each of the three complexity level considered, testing the three control strategies: **IAF**, **VAF-GRID** and **VAF-RRT**. The order of execution and the map selection is randomized, hence the users cannot alter the result by orienting themselves based on their memories. At the end of this set of manual guidance tasks in an unknown environment, the volunteers are asked to fill-in a second survey involving three questions. The topics are hereby described:

- **Q1**: The ease of reaching the objective, avoiding all the obstacles.
- **Q2**: The confidence in having completed the task without any collision.
- **Q3**: The perception of the robot assistance in the final positioning.

5.4.1 Outcomes

Although the first and second set of experiments may seem very similar, the differences in their execution are significant. If the obstacles are visible, then the operator is confident that all collisions will be avoided, and guides the end-effector faster. Instead, in an invisible workspace layout he/she moves slowly, exploring the available space and relying on haptic perceptions to reach the objective safely if one of the variable admittance control techniques is adopted. Moreover, in the first experiment the human can plan an optimized path, while in the second one it is impossible. If **IAF** is adopted, the user proceeds blindly in the environment, mapping the workspace by trial and error. If **VAF-GRID** or **VAF-RRT** are in use, the path that he/she follows depends on the quality of the haptic feedback provided by the robot and on the user sensitivity.
Since the path varies between different maps, comparing the execution times, path length or damping behavior, like in the first experiment, would be meaningless. Instead, the three approaches will be compared in terms of number of

collisions occurred, *total energy required* to reach the objective, *final positioning error* and *average distance* from the obstacles.

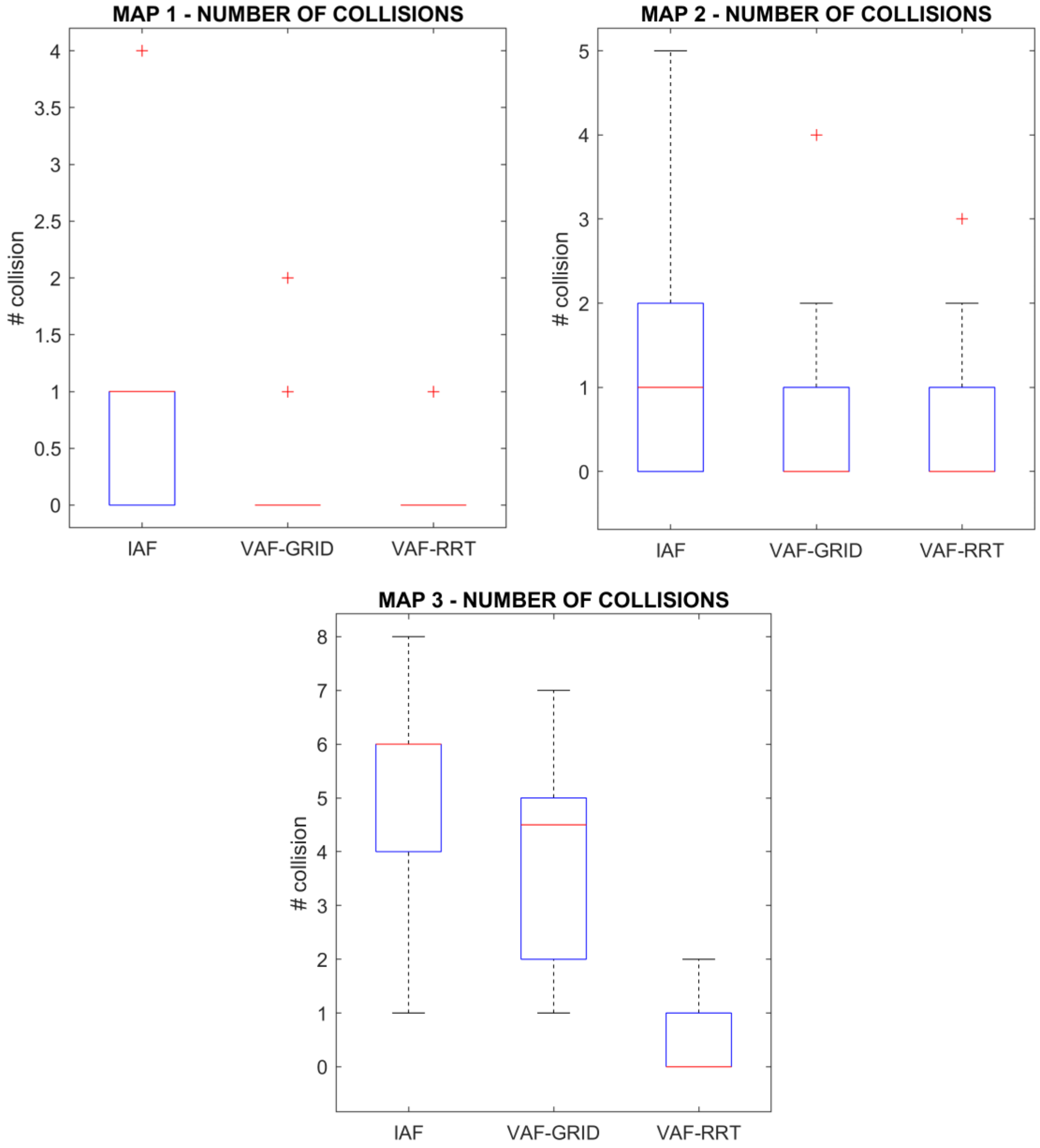


Figure 5.14: Statistics of the number of collisions occurred during the execution of the second experiment. Three graphs are reported representing the three different maps with increasing level of difficulty.

Figure 5.14 shows three box-plots comparing the number of collisions occurred during task execution. Each graph corresponds to a different virtual workspace, constructed in order to simulate three realistic environments with increasing levels of complexity. In the first map, the statistics show that if **IAF** is in use, meaning that no directional haptic feedback is generated by the robot, then an average of 1 collision is registered. Nevertheless it is possible for the operator to cross the central gate by chance, without hitting any obstacle. In the same layout **VAF-GRID** and **VAF-RRT** reduce the average number of collisions to 0. In the second map, the positive effects of **VAF-GRID** and **VAF-RRT** are noticeable: the two variable admittance strategies manage to nullify the average number of collisions and reduce its variability with respect to **IAF**, ensuring a smoother execution of the task. In the third map, the human faces the most complex obstacle layout, composed of a gate and a wall oriented diagonally, which creates a dead end (refer to figure 5.13). From the graphical comparison it is apparent that **IAF** is not suitable to solve the task: out of the 14 volunteers, no one was able to avoid collisions. An average of 6 collisions per experiment has been measured and the variability of the results is elevated. Furthermore, similar outcomes are obtained when **VAF-GRID** is in use. The average number of collision has decreased to 4.5 and the variability is lower, but no operator was able to complete the task without collisions. In fact, this experiment highlight another limit of the grid-based approach: the lack of a global planning strategy. As mentioned before **VAF-GRID** is a simple and reactive approach, which purpose is to guide the operator forward in a direction parallel to the line connecting starting and goal position. The grid structure generation and analysis is based only on a local exploration of the available space: the minimum effort direction never leads to an immediate collision, but there is no certainty that the objective will eventually be reached by following it. Consequently, grid-based obstacle avoidance treat dead ends like the rest of the environment. The user is guided forward until the frontal obstacle is detected, and then, since it is not possible to proceed forward, the user will be redirected backwards. The global path planning task is left to the operator, he/she must drive the end-effector back and try other paths. However, these experiment demonstrate that this process confuses the human and, in general, increases the number of collisions. On the other hand, **VAF-RRT** is based on the analysis of a tree structure, hence it is always possible to proceed towards the objective independently of the current position. The limitation of **VAF-GRID** is automatically overcome, since the tree branches are always pointing out of dead-ends by construction. The statistics show that when RRT-based obstacle avoidance is in use, the average number of collision is brought to zero and the variance between different users is reduced.

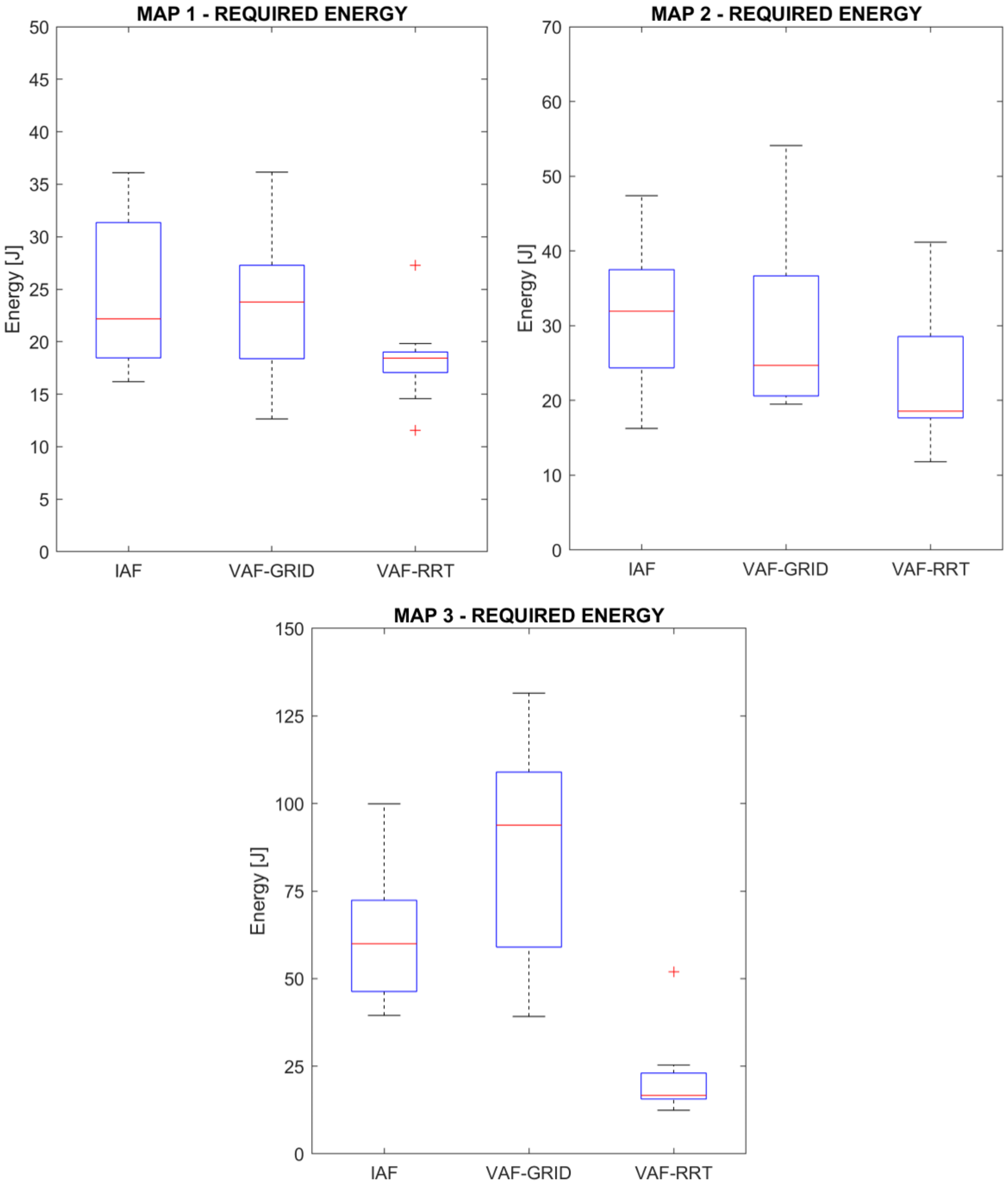


Figure 5.15: Statistics of the human energy consumed in the second experiment.

The observations on the number of collisions are confirmed when the energy required to complete the task is taken into consideration, as it can be seen in figure 5.15.
The first map represent a very simple scenario, in fact, there is no significant difference in energy consumption between the three manual guidance strategies. In the second map, the benefits of the proposed approaches are underlined by a 20% and 40% energy decrease, respectively for **VAF-GRID** and **VAF-RRT**, compared to **IAF**. As expected, the third map results show that **VAF-GRID** is ineffective, due to the limitations described before: 50% more energy is wasted with this strategy. **VAF-RRT**, instead, is proven to be effective even in this complex scenario, as demonstrated by the 70% reduction in total energy consumed.

In figure 5.16 the average distance from the end effector to the closest obstacle is reported on a normalized time scale, along with its 25th and 75th percentiles. From this representation it is possible to notice the positive effects of the navigation assistance on the trajectory described by the user. When **IAF** is adopted, the operator tends to travel closer to the obstacles with respect to **VAF-GRID** and **VAF-RRT** case. In fact, this behavior is observable in all three maps of the second experiment. While the two variable admittance techniques try to guide the user away from the obstacles, on a safer trajectory, **IAF** does not provide any feedback, hence there are instants when the end-effector come too close to an obstacle, followed by instants when it drifts away from them more than necessary.
Moreover, the variance of the distance is lower in **VAF-GRID** and **VAF-RRT** cases with respect to the **IAF** case: this is due to the fact that the passive feedback provided by the two variable admittance approaches constrain the operators on a smoother trajectory, that minimizes the risk of collisions.

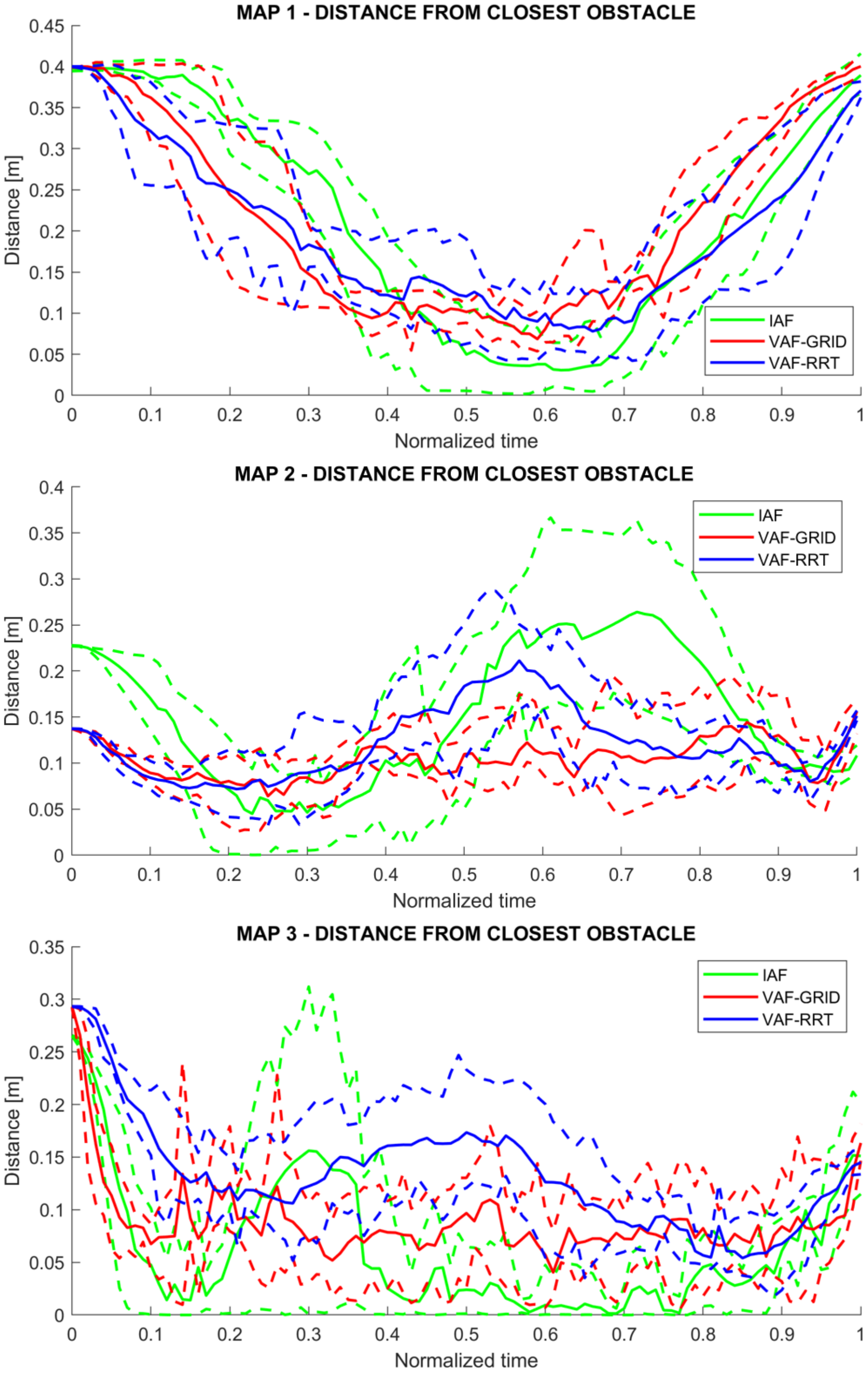


Figure 5.16: Graph representation of the average distance from the closest obstacle (continuous line) in a normalized time scale, with its 25th and 75th percentiles (dashed lines).

Concluded this set of experiments, a second survey is proposed to the volunteers. They are asked to grade (from 1 to 10) their collaborative experience during the manual guidance operation, the impact of the feedback on the task execution and their confidence in having reached the goal without collisions.

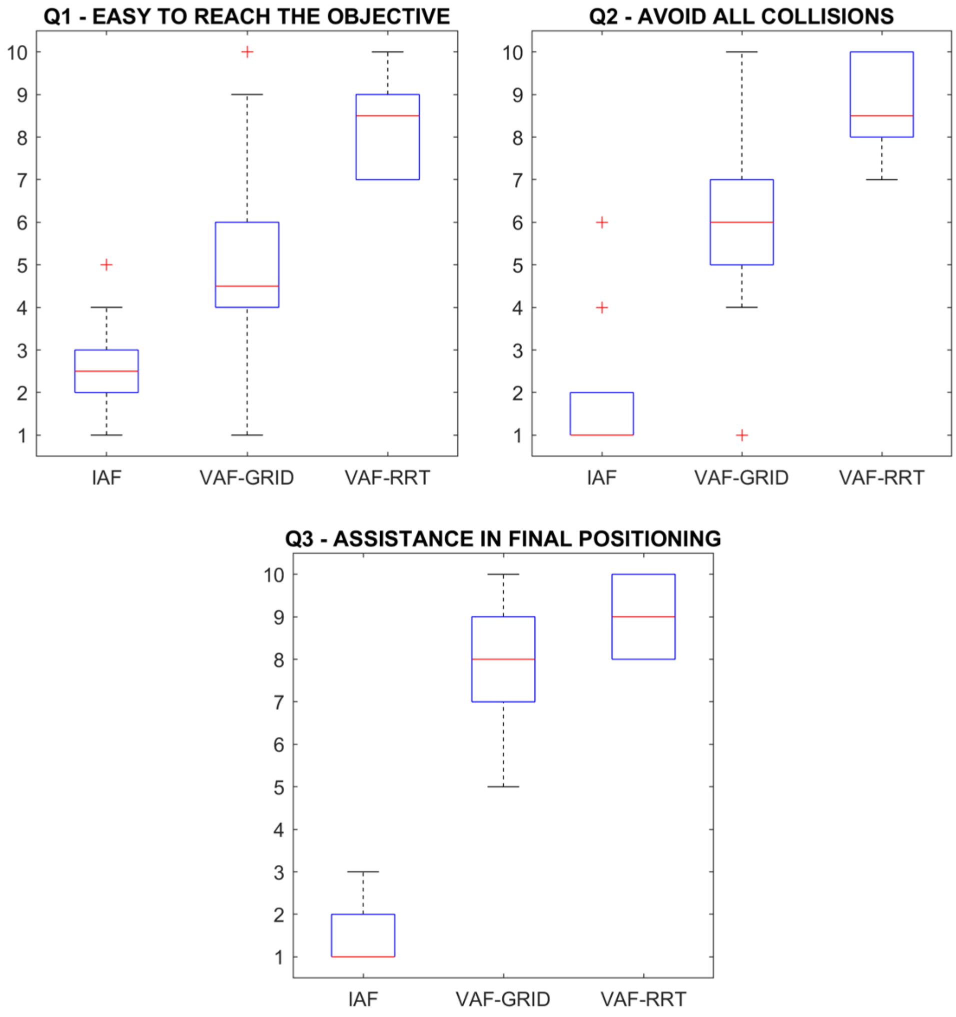


Figure 5.17: Questionnaire statistics about the second experiment.

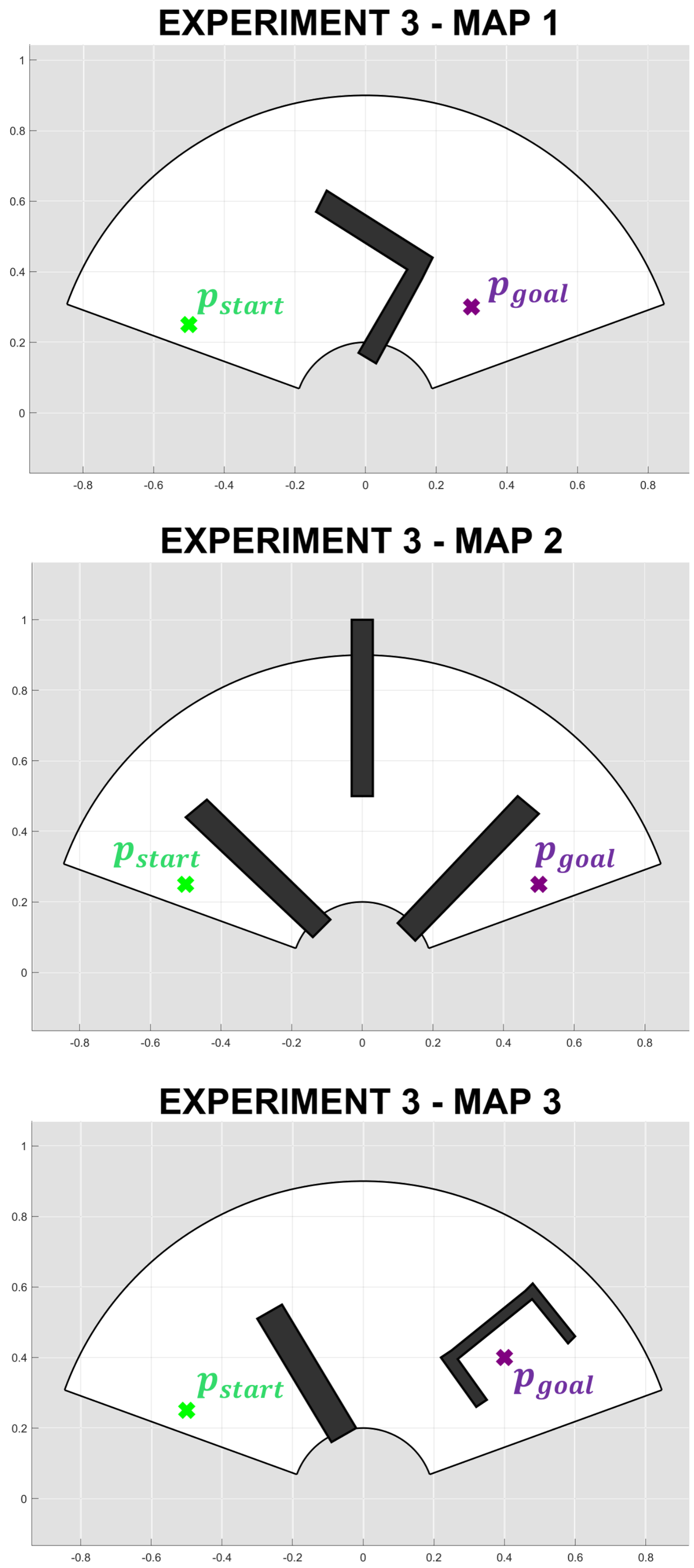
The statistics of their answers, depicted in figure 5.17, are in line with the consideration reported in this section. As expected, **IAF** fails to solve the navigation problem in an unknown environment. It is hard for the users to find a safe path towards the goal position: to do so they have to proceed blindly, colliding with the invisible obstacles until, by trial and error, they reach the objective.

Furthermore, **Q1** and **Q2** highlight the fact that, despite providing some form assistance in avoiding impacts, **VAF-GRID** is not able to satisfy the operators due to its strong limitations. Eventually, it is possible to conclude that **VAF-RRT** provides a valuable haptic feedback to the operator during the interaction, allowing him/her to reach the target position quickly, even in case he/she is not aware of the surrounding obstacles.

5.5 Third experiment

In this set of experiments the volunteer is asked to reach an unknown goal position inside an unknown complex workspace. As a result, his/her guidance is entirely based on the feedback provided by the robot. A set of three possible layouts is proposed to the operator: each of them contain multiple obstacles creating elaborated patterns. The three unknown maps are depicted in figure 5.18. Due to the high complexity level, only tree-based manual guidance (**VAF-RRT**), is applied to solve these navigation problems. In fact, both **IAF** and **VAF-GRID** approaches would fail. Indeed, **IAF** is equivalent to walking blindly in the environment, while **VAF-GRID** is discarded because of its multiple limitations demonstrated in the first and second experiments.

The objective of this third set of experiments is twofold: the main purpose is to show the full capabilities of **VAF-RRT** as an effective obstacle avoidance approach for manual guidance also in highly structured and complex scenarios, while the secondary goal is to compare the effects of different levels of autonomy allowed to the user. This variation is obtained by modifying the neighborhood radius r_n, including different amounts of nodes in the search for the optimal direction (see Chapter 4).


EXPERIMENT 3 - MAP 1

EXPERIMENT 3 - MAP 2

EXPERIMENT 3 - MAP 3

Figure 5.18: The three maps for the third experiment, with increasing level of complexity.

The volunteer has to repeat the experiment two times for each of the three maps, with two different variants of tree-based manual guidance:

1. **RRT-based Variable Admittance Filter (VAF-RRT)**: this is the same algorithm used in the other two sets of experiments, and it has a search radius $r_n = \infty$. All the nodes belonging to the tree structure are included in the selection of the minimum effort direction. In this case user autonomy is maximized because of the higher probability of finding an available node aligned with the direction of the input force.

2. **Optimal RRT-based Variable Admittance Filter (OPT-RRT)**: this version acts with a reduced search radius $r_n = \frac{3}{2}d_{step}$. Only the nodes close to the end-effector position are included in the selection of the optimal direction to follow and the chances that one of them is aligned with the input force are low. Consequently, the probability of redirecting the user along an optimal trajectory, ignoring his/her intentions, is high.

The two variants are compared in therms of *execution time*, *required energy* and *number of collisions* to determine if the quality of the human-machine collaboration is influenced by the user autonomy. Afterwards, a third survey is proposed to the volunteers, regarding the following five topics:

- **Q1**: The ease in perceiving the correct direction to follow.

- **Q2**: The impact of practice on the perception of the feedback.

- **Q3**: The ease in perceiving that the target has been reached.

- **Q4**: The confidence in final positioning accuracy.

- **Q5**: The confidence in having completed the task without any collision.

5.5.1 Outcomes

Concerning the analysis of the *required energy*, *path length*, *execution time*, *number of collisions* and *positioning error*, no relevant differences between **VAF-RRT** and **OPT-RRT** were observed. For this reason, only the results obtained by adopting **VAF-RRT** are presented in this section. The results show that a high final positioning precision and a low number of collisions are achievable with **VAF-RRT**, even if the volunteers had to rely only on their tactile perception in order to complete the tasks.

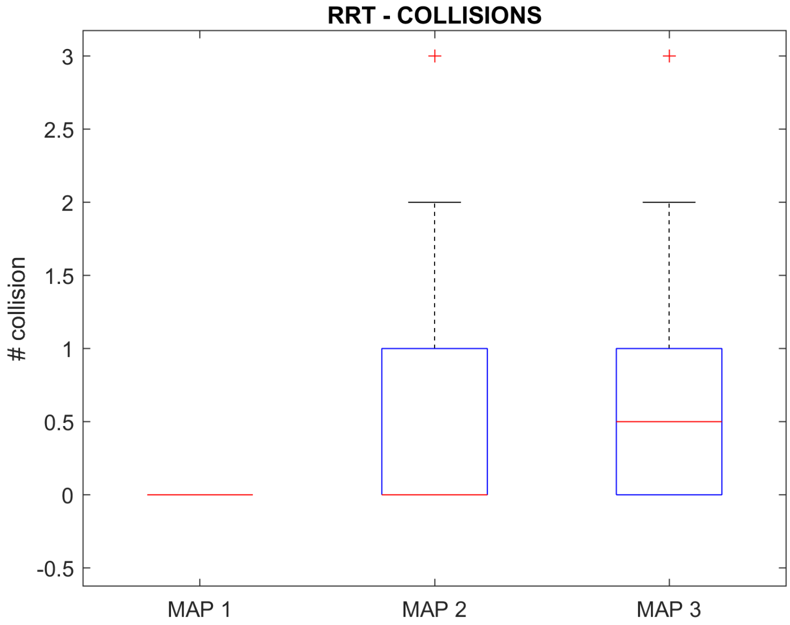


Figure 5.19: Statistics of the number of collisions occurred during the execution of the third experiment, navigating in three complex scenarios.

Figure 5.19 reports the number of collisions occurred in each of the three complex scenarios. No collision occurred in the first scenario for all the volunteers. In the second map the user is required to guide the end-effector in a narrow corridor, following a zig-zag pattern. The graph shows that, although in some cases a limited number of collision took place, on average **VAF-RRT** is able to prevent them all. The third map presents a positioning challenge: the volunteer has to avoid a wall and then reach the unloading position which is located inside a C-shaped obstacle, as represented in figure 5.18. Even in this third scenario, the analysis of the results indicates that, on average, only 0.5 collision happen under **VAF-RRT** assistance.

Furthermore, accuracy on the final positioning must be examined to verify that RRT-based manual guidance is effective in reaching the unknown goal.

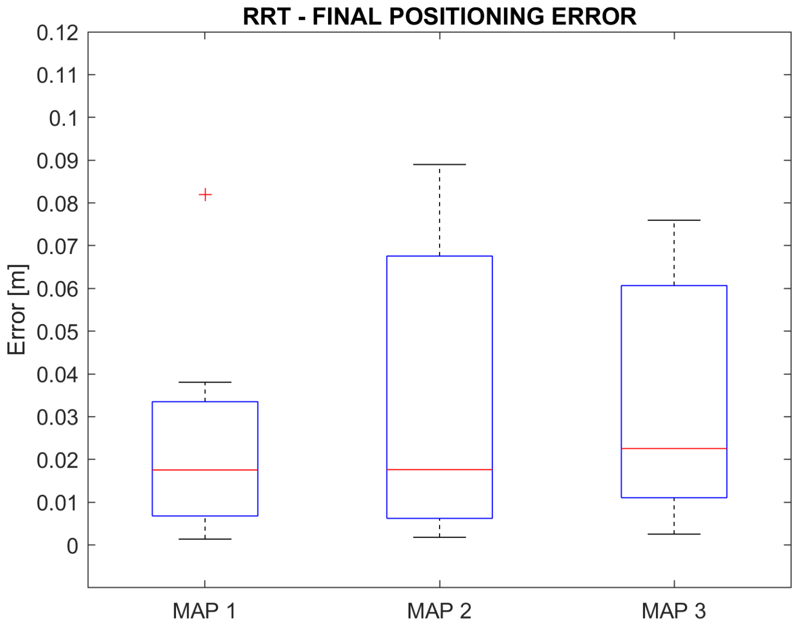


Figure 5.20: Statistics of the final positioning error with respect to an unknown goal, subdivided for the three complex maps.

In figure 5.20 the positioning error performances are shown. Even though the user has to navigate inside a unknown complex environment to reach an invisible destination, the variable haptic feedback provided by **VAF-RRT** allows him/her to position the end-effector at an average of 2 cm from the real objective. Moreover, the variance of these results highlight the fact that, even in the worst case scenario, the operator reached a final position within 10 cm from the target.

After this last set of experiments, the volunteers completed a third survey, where they were asked to compare the high autonomy technique, **VAF-RRT**, to the second more restrictive approach, **OPT-RRT**.

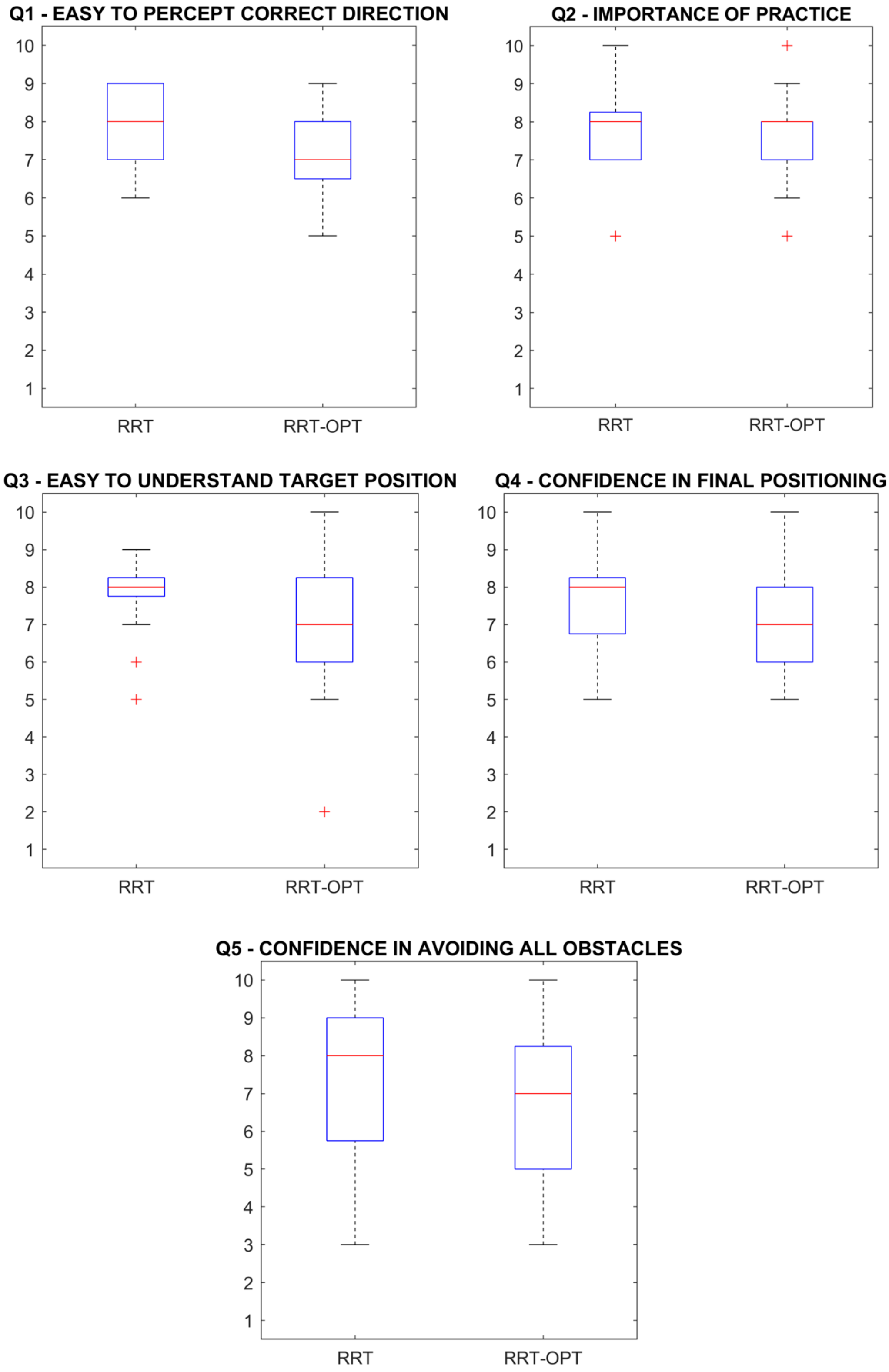


Figure 5.21: Questionnaire statistics about the third experiment, comparing the RRT-based approach with two different levels of autonomy.

From the analysis of the operator responses, the results depicted in figure 5.21 are obtained. The user, in general, perceives the two approaches in a similar way. However, upon closer inspection, **Q1** and **Q3** highlight that with **VAF-RRT**, which adopts a higher level of autonomy, both directional perception and target realization are enhanced. Additionally, **Q2** reveals that the volunteers perceived improvements in their sensibility while executing the three sets of experiments: they realized that, with additional practice, they would obtain better results in final positioning and task execution time.
The outcomes of the third set of experiments confirm the results obtained in the other two. **VAF-RRT** is a valid control strategy that allows to reach a specific point inside the workspace with high accuracy. Despite working only as a passive feedback, it is able to guide the operator along complex trajectories, overcoming multiple invisible obstacles. It allows to reduce the physical effort required to complete the task and minimize the total number of collisions.

Chapter 6

Conclusions

In this thesis, a new variable admittance control strategy for manual guidance operations in a cluttered workspace is proposed. A recent result in this field is the goal-driven variable admittance control, reported in [19]: it presents a new geometrical interpretation of the admittance filter, which is able to provide a directional feedback to the user and lead him/her towards an unknown objective. However, it is limited by the assumption that the movement happens in a completely free environment.

The main contribution of this work is an improved version of that technique, which is capable of guiding the human along complex trajectories towards the objective position, minimizing collisions with the obstacles that are present inside the workspace, even if the operator field of view is obstructed by the bulky size of the transported cargo.

The developed algorithm can be partitioned into two macro-phases. In the first phase, which is executed offline, the workspace map is analyzed to create an *environment exploring structure*, constituted by a set of organized samples describing the available space. In the second phase, instead, the position of the end-effector is compared to such structure in order to determine an optimal direction to follow, where the resistance perceived by the human will be minimized. In this case, also the user intention of motion is taken into consideration. Hence, a mutual adaptation between the human and the robot is established, and unnecessary movement restrictions that may increase the user fatigue are avoided.

In this work, two types of *environment exploring structure* are developed. The first one is based on a multiple layer grid. It enacts a reactive approach to obstacle avoidance: the trajectories are only locally optimized in order to avoid nearby obstacles. The second proposed structure, instead, is inspired by the Rapidly exploring Random Tree algorithm, a classical path generation procedure commonly used in robotics. Not only it allows to avoid collisions, but it is also able to suggest to the user an optimized feasible path in a proactive and farsighted way.

Once an optimal direction to follow has been determined, the admittance coefficients are modulated according to the principles of GDVAC [19]. The damping coefficient for the filter is evaluated as the distance from the end-effector position to the intersection point between the direction of the force applied by the user and the associated *damping shape*, centered in the end-effector position and oriented according to the optimal direction. The mass coefficient is selected as a scaled version of the damping value, so that the bandwidth of the filter is kept constant and the interaction results more intuitive. The operator perceives less effort if the force applied on the end-effector is aligned to the optimal direction, and a gradually increasing fatigue the more it is misaligned. Moreover, the shape is opportunely modified in order to enhance the difference between the correct direction to follow and the ones that would lead to collisions.

The effectiveness of the proposed control strategies is evaluated through three sets of experiments on a Comau Smart Six manipulator, involving 14 volunteers. Grid-based manual guidance (**VAF-GRID**) and RRT-based manual guidance (**VAF-RRT**) are compared to an invariable admittance filter (**IAF**) in terms of execution time, required human energy, path length, number of collisions and final positioning error.

The experimental results showed that, when applied to simple scenarios, both **VAF-GRID** and **VAF-RRT** are able to reduce the operator effort and improve the final positioning precision with respect to **IAF**. However, in more complex working environments, the limitations of **VAF-GRID** emerges, preventing an intuitive and effective interaction with the human. On the other hand, **VAF-RRT** is able to guide the operator on an optimized path, minimizing the impacts and lowering human fatigue by up to a 70% factor. In fact, the outcomes successfully validate **VAF-RRT** as a solution for complex navigation tasks. The operators are able to reach an invisible target in an unknown cluttered environment with high accuracy, with a maximum final error of 10 cm and an average one under 2 cm. The collisions against invisible obstacles are minimized, with an average number of impacts close to zero in all the proposed maps and no collisions registered in 70% of the experiments.

6.1 Future developments

RRT-based variable admittance control has been proven a successful approach to help the human operator in the execution of manual guidance tasks, both in a completely visible setup and in an invisible workspace. For this reason, some ideas describing possible future developments are provided in the following:

- As described in chapter 3, extending the developed algorithms to the third dimension is not a hard task. Nevertheless, it would be interesting to study the performance of the RRT-based approach in a three-dimensional workspace, where the obstacles can be also passed over. Not only the end-effector but also the manipulator arm should be considered for the collision avoidance. Moreover, the proposed solutions were developed for a six degree of freedom manipulator, but if a redundant robot were to be used, then it is possible to exploit the extended Jacobian method to control the entire robot configuration.

- Only translational admittance control is analyzed in this thesis and implemented in the experiments. The end-effector is free to move in any Cartesian direction, but the handle is always constrained in a vertical orientation. Additionally, the transported object is represented as a particle without dimensions, placed exactly in the hand of the manipulator. An interesting development would be to simulate the transportation of a rigid body. In this case, the different orientations of the end-effector would definitely have an impact on the path planning. A target orientation could be imposed and then achieved through rotational admittance control methods.

- In this work, the proposed algorithms solve the problem of reaching a unique goal position by navigating inside an unknown workspace. A possible improvement, however, could be to consider more than one destination: a set of work stations evenly distributed inside the environment can be specified, that the operator has to reach in a specific order or at random. A possible solution would be to develop new *environment exploring structures*, that include multiple objectives. These structures can then be used, in combination with the estimated user intention of motion, in order to detect which objective the human wants to reach and help him/her reaching it.

- Another interesting hint could be to introduce a vision system or a Lidar scanner to map the environment in real time. In fact, a known and static workspace layout is assumed in this work. However, in a realistic work cell, the environment changes in time. New obstacle avoidance strategies can be developed in order to redirect the operator in real time, guiding the user to the target position and modifying the optimal path in order to avoid any unforeseen or moving obstacle.

Bibliography

[1] R. Bloss. “Collaborative robots are rapidly providingmajor improvements in productivity, safety, programing ease, portability and cost while addressing many new applications”. In: *Industrial Robot: An International Journal* 43.5 (2016), pp. 463–468.

[2] M. Knudsen and J. Kaivo-Oja. “Collaborative Robots: Frontiers of Current Literature”. In: *Journal of Intelligent Systems: Theory and Applications* 3.2 (2020), pp. 13–20.

[3] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib. “Progress and prospects of the human–robot collaboration”. In: *Auton Robot* 42 (2018), pp. 957–975.

[4] V. Villani, F. Pini, F. Leali, and C. Secchi. “Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications”. In: *Mechatronics* 55 (2018), pp. 248–266.

[5] N. Hogan. “Impedance Control: An Approach to Manipulation”. In: *American Control Conference* (1984), pp. 304–313.

[6] A. Keemink, H. van der Kooij, and A. Stienen. “Admittance control for physical human–robot interaction”. In: *The International Journal of Robotics Research* 37.11 (2018), pp. 1421–1444.

[7] R. Ikeura, T. Moriguchi, and K. Mizutani. “Optimal variable impedance control for a robot and its application to lifting an object with a human”. In: *Robot and Human Interactive Communication, 11th IEEE International Workshop on* (2002), pp. 500–505.

[8] R. Ikeura and H. Inooka. “Variable impedance control of a robot for cooperation with a human”. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation* 3 (1995), pp. 3097–3102.

[9] Luca Bascetta, Gianni Ferretti, Gianantonio Magnani, and Paolo Rocco. “Walk-through programming for robotic manipulators based on admittance control”. In: *Robotica* 31.7 (2013), 1143–1153.

[10] F. Ficuciello, L. Villani, and B. Siciliano. “Variable Impedance Control of Redundant Manipulators for Intuitive Human–Robot Physical Interaction”. In: *IEEE Transactions on Robotics* 31.4 (2015), pp. 850–863.

[11] V. Duchaine, B. Mayer St-Onge, D. Gao, and C. Gosselin. “Stable and Intuitive Control of an Intelligent Assist Device”. In: *IEEE Transactions on Haptics* 5.2 (2012), pp. 148–159.

[12] T. Tsumugiwa, R. Yokogawa, and K. Hara. “Variable impedance control with regard to working process for man-machine cooperation-work system”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* 3 (2001), pp. 1564–1569.

[13] Z. Li, B. Huang, Z. Ye, M. M. Deng, and C. C. Yang. “Physical Human–Robot Interaction of a Robotic Exoskeleton By Admittance Control”. In: *IEEE Transactions on Industrial Electronics* 65.12 (2018), pp. 9614–9624.

[14] C. T. Landi, F. Ferraguti, L. Sabattini, C. Secchi, and C. Fantuzzi. “Admittance control parameter adaptation for physical human-robot interaction”. In: *IEEE International Conference on Robotics and Automation (ICRA)* (2017), pp. 2911–2916.

[15] F. Dimeas and N. Aspragathos. “Online Stability in Human-Robot Cooperation with Admittance Control”. In: *IEEE Transactions on Haptics* 9.2 (2016), pp. 267–278.

[16] F. Dimeas and N. Aspragathos. “Fuzzy learning variable admittance control for human-robot cooperation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014), pp. 4770–4775.

[17] F. Dimeas and N. Aspragathos. “Reinforcement learning of variable admittance control for human-robot co-manipulation”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), pp. 1011–1016.

[18] P. N. Sharkawy A. Koustournpardis and N. Aspragathos. “Variable Admittance Control for Human-Robot Collaboration based on Online Neural Network Training”. In: (2018), pp. 1334–1339.

[19] D. Bazzi, M. Lapertosa, A. M. Zanchettin, and P. Rocco. “Goal-driven variable admittance control for robot manual guidance”. In: *Intelligent Robots and Systems (IROS), International Conference on* (2020), pp. 9759–9766.

[20] G. Van De Logt and F. Odekerken. “A numerical approach for planning robot trajectories considering collision avoidance”. In: *Systems Man and Cybernetics 1996, IEEE International Conference on* 2 (1996), pp. 1501–1507.

[21] L. Blackmore and B. Williams. “Optimal manipulator path planning with obstacles using disjunctive programming”. In: *American Control Conference* (2006).

[22] O. Khatib. “Real-time obstacle avoidance for manipulators and mobile robots”. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation* 2 (1985), pp. 500–505.

[23] W. Xiang, X. Ying, G. Chao, and L. Mengru. “An Improved Potential Field Method of the Manipulator Obstacle Avoidance Planning Based on Polishing Path”. In: (2019), pp. 143–147.

[24] J. Park, J. Kim, and J. Song. “Path Planning for a Robot Manipulator based on Probabilistic Roadmap and Reinforcement Learning”. In: *International Journal of Control, Automationm and Systems* 5.6 (2007), pp. 674–680.

[25] Steven M. Lavalle. “Rapidly-Exploring Random Trees: A New Tool for Path Planning”. In: (1998).

[26] C. Yuan, W. Zhang, G. Liu, X. Pan, and X. Liu. "A Heuristic Rapidly-Exploring Random Trees Method for Manipulator Motion Planning". In: *IEEE Access* 8 (2020), pp. 900–910.

[27] B. Lacevic and P. Rocco. "Sampling-based Safe Path Planning for Robotic Manipulators". In: *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)* (2018), pp. 1–7.

[28] X. Lei-ping, C. C. Zi-li, and S. Shao-jie. "Obstacle Avoiding Research on the Manipulator Based on Genetic Algorithm". In: *2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control* (2011), pp. 865–867.

[29] X. Yang and M. Meng. "Dynamical trajectory generation with collision free using neural networks". In: *1998 Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on* 3 (1998), pp. 1634–1639.