**National Technical University of Athens**
**School of Mechanical Engineering**
**Section of Mechanical Design and Automatic Control**
**Control Systems Lab**

# Optimal Motion Planning in 3D Workspaces: Integrating a Panel-Method-Based Motion Planner with Continuous Deep Reinforcement Learning

Diploma Thesis

**Marios Malliaropoulos Katsimis**

Supervisor: K. J. Kyriakopoulos, Professor NTUA
Advisor: Ch. Bechlioulis, Associate Professor University of Patras

Athens, 2023

# Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Kostas J. Kyriakopoulos, for his invaluable guidance and academic mentorship throughout my studies. I am truly fortunate to have had the opportunity to be a part of his team in the Control Systems Lab of NTUA.

I am also deeply thankful to Professor Charalampos Bechlioulis for his advice and expertise in the subject matter. His guidance has been pivotal in the successful completion of this study. I also extend my appreciation to the PhD candidate Panagiotis Rousseas for his feedback and thoughtful discussions.

I want to thank my family for their encouragement and motivation throughout all these years. Last but certainly not least, I am grateful to Theodora, Kostis and all my friends who have been by my side, making this journey truly remarkable.

# Abstract

This diploma thesis proposes a novel and proven correct reactive method for planning three-dimensional optimal motion in complex environments. By combining fluid flow equations, optimal control theory, and deep reinforcement learning techniques, this study offers an interdisciplinary and unique approach, effectively merging positive attributes from different scientific fields. The method models the 3D motion planning problem by solving streamlines of the potential fluid flow, enabling the proper handling of various terrain types. This is achieved through the discretization of the geometry into surface panels, while the safety criteria are ensured via a set of von-Neumann boundary conditions. The proposed fluid-based planner guarantees a continuous-time, natural-looking, stable and safe solution for the motion planning problem with Artificial Harmonic Potential Fields (AHPFs). Furthermore, this thesis presents a model-based reinforcement learning algorithm for learning the optimal non-linear control in continuous time and action space with respect to an infinite horizon cost function. The algorithm utilizes an actor-critic scheme based on policy iteration, to successively approximate the optimal solution of the Hamilton-Jacobi-Bellman equation. This way, the optimal robot motion is obtained by iteratively updating the fluid flow parameters (i.e., the controller parameters) in a deterministic manner. The proposed method demonstrates fast convergence and outperforms widely used methods such as the RRT$^*$, highlighting its contribution to the field of 3D optimal motion planning.

# Περίληψη

Η παρούσα διπλωματική εργασία προτείνει μια νέα και αποδεδειγμένα ορθή μέθοδο για τον σχεδιασμό της τρισδιάστατης βέλτιστης κίνησης ενός ρομπότ. Η μέθοδος προσφέρει μια διεπιστημονική και καινοτόμα προσέγγιση, συνδυάζοντας τις εξισώσεις ροής των ρευστών, την θεωρία βέλτιστου ελέγχου και τεχνικές βαθιάς ενισχυτικής μάθησης. Το πρόβλημα της βέλτιστης πλοήγησης μοντελοποιείται μέσω της επίλυσης της ροής ασυμπίεστου, μη-συνεκτικού και αστρόβιλου ρευστού. Αυτό επιτυγχάνεται διακριτοποιώντας την γεωμετρία του χώρου σε επιφανειακά πάνελς και ικανοποιώντας στην συνέχεια τα κριτήρια ασφάλειας μέσω ενός σύνολο οριακών συνθηκών von-Neumann. Η προτεινόμενη μέθοδος ανήκει στην κατηγορία των Τεχνικών Αρμονικών Δυναμικών Πεδίων για τον σχεδιασμό πορείας και εγγυάται την γένεση συνεχούς-χρόνου και ομαλών τροχιών, ενώ παράλληλα σταθεροποιεί το σύστημα με ασφάλεια στον τελικό στόχο. Επιπρόσθετα, η παρούσα εργασία παρουσιάζει έναν αλγόριθμο ενισχυτικής μάθησης, βασισμένο σε μοντέλο του περιβάλλοντος, για την εκμάθηση του βέλτιστου μη-γραμμικού ελέγχου σε συνεχή χρόνο. Οι παράμετροι της ροής του ρευστού ανανεώνονται επαναληπτικά μέχρι να συγκλίνουν στην βέλτιστη λύση, η οποία δίνεται από την εξίσωση Hamilton-Jacobi-Bellman (HJB). Η προτεινόμενη μέθοδος επιδεικνύει ταχεία σύγκλιση και υπερτερεί έναντι ευρέως δημοφιλών μεθόδων όπως είναι η RRT*, αναδεικνύοντας με τον τρόπο αυτό την συνεισφορά της στον κλάδο του σχεδιασμού κίνησης των ρομπότ σε τρισδιάστατους χώρους.

# Contents

**Contents** **i**

**1 Introduction** **1**

1.1 The Motion Planning Problem . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.3 Thesis Contribution . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.4 Thesis Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

**2 The Optimal Motion Planning Problem** **6**

2.1 Problem Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.2 Preliminaries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2.2.1 Optimal Nonlinear Control . . . . . . . . . . . . . . . . . . . . . 7

2.2.2 Harmonic Artificial Potential Fields . . . . . . . . . . . . . . . . 8

2.2.3 General Solution of the Laplace Equation . . . . . . . . . . . . 8

**3 Control Design Methodology** **12**

3.1 Proposed Control Policy . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.2 Proposed Fluid Flow Model . . . . . . . . . . . . . . . . . . . . . . . 13

3.2.1 Type of Singularity Elements . . . . . . . . . . . . . . . . . . . 13

3.2.2 The Panel Method . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.3 Control Policy Structure . . . . . . . . . . . . . . . . . . . . . . . . . 16

3.4 Safety Conditions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

3.5 Stability Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

3.6 Proposed Motion Planning Algorithm . . . . . . . . . . . . . . . . . . . . 23

**4 Optimal Motion Planning Solution 24**

4.1 Optimal Control Policy . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

4.2 Model-Based Reinforcement Learning . . . . . . . . . . . . . . . . . . . 25

4.2.1 Reinforcement Learning Framework . . . . . . . . . . . . . . . 25

4.2.2 Value Function Approximation . . . . . . . . . . . . . . . . . . 26

4.2.3 Policy Improvement . . . . . . . . . . . . . . . . . . . . . . . . 27

4.3 Proposed Optimal Motion Planning Algorithm . . . . . . . . . . . . . 29

4.4 Proofs of Optimality . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

**5 Simulation Results 32**

5.1 Urban Environment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

5.1.1 Optimal Trajectories - Comparison with RRT* . . . . . . . . . 34

5.1.2 Critic Network . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

5.1.3 Actor: AHPF Policy and Convergence Criteria . . . . . . . . . 42

5.2 Forest Environment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

**6 Conclusions 53**

6.1 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

6.2 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

6.3 Limitations and Future work . . . . . . . . . . . . . . . . . . . . . . . 54

**Bibliography 58**

# Abbreviations

**AHPFs** Artificial Harmonic Potential Fields.

**APFs** Artificial Potential Fields.

**CSL** Control Systems Lab.

**DNN** Deep Neural Networks.

**GBMs** Graph-Based Methods.

**HJB** Hamilton-Jacobi-Bellman.

**MAE** Mean Absolute Error.

**NF** Navigation Functions.

**NTUA** National Technical University of Athens.

**ODE** Ordinary Differential Equation.

**RBFs** Radial Basis Functions.

**RRT** Rapidly Exploring Random Trees.

**SBMs** Sampling-Based Methods.

**w.r.t.** with respect to.

# Chapter 1

# Introduction

## 1.1 The Motion Planning Problem

Motion planning has garnered significant attention from researchers throughout the years and is pivotal for the future of robotics [1]. Motion planners are involved in a wide range of applications, spanning from computer-aided design to robotic surgeries and the coordination of autonomous vehicles in disaster scenarios. The primary objective of motion planning is to find a control sequence that generates feasible trajectories, enabling the system to navigate towards a desired position. Addressing the motion planning problem entails challenges related to ensuring a collision-free navigation, implementing computationally efficient algorithms in real time, and providing optimal control policies. While solutions on the motion planning problem have long been established, pursuing optimality remains more challenging. So far, the optimal motion planning problem has been mainly treated with open-loop sampling-based methods. However, in complex and high-dimensional environments it becomes evident that naive discretization of the robot's state and action space can lead to high computational times, discretization errors and sub-optimal results [2]. Hence, addressing optimality in motion planning necessitates continuous-time solutions that can effectively tackle the infinite-horizon optimal control problem. Continuous-time reactive approaches not only mitigate the limitations of discrete-time methods but also allow to study the problem with higher fidelity. This includes generating smooth trajectories that result in a natural movement of the system, providing proven safety by incorporating the constraints within the system dynamics and overall delivering robust solutions.

## 1.2 Related Work

The motion planning problem has been heavily studied in the two dimensions. However, extending motion planning to three-dimensional environments introduces new challenges. One approach is to decouple planar maneuvers and altitude changes [3], while other researchers focus on directly expanding traditional methods in higher-dimensional spaces. The latter requires careful treatment and entails challenges related to establishing the existence of a solution, meeting the safety requirements and compensating for computational feasibility. In the literature, motion planning algorithms are mainly divided into two categories: discrete methods and reactive methods.

Path planning in discrete environments is generally classified into graph-based methods (GBMs) and sampling-based methods (SBMs). The former includes many popular algorithms, e.g. A$^*$ [4] and Dijkstra's algorithm [5], that construct a graph representation of the discretized space and then search for a feasible path. SBMs utilize an additional obstacle detection module to explore the environment by generating and connecting nodes in a tree-like structure. The most popular sampling-based approaches are Probabilistic Roadmaps [6] and Rapidly Exploring Random Trees (RRT) [7]. While these methods are proven probabilistic complete, they lack proven optimality. Another drawback of discrete-time methods is the significant increase in time complexity as the discretization resolution, the map complexity and the space dimensions grow. A noteworthy contribution towards achieving optimality in sampling-based methods is the introduction of RRT* by Karaman and Frazzoli [8]. This algorithm offers asymptotic optimality guarantees by dynamically updating the tree connections through a rewiring technique. Numerous extensions of RRT$^*$ have been developed over the years and mainly focus on generating smoother trajectories [9], addressing kino-dynamic constraints [10] and dealing with the slow convergence and the high memory consumption [11, 12].

On the other hand, Artificial Potential Fields (APFs) employ continuous potential functions over the entire workspace, enabling the robot to navigate through closed-loop velocity commands that are derived by the gradient of the potential field. The main limitation of APFs is the presence of local minima inside the workspace, which can lead to non-convergent paths. To address this issue in two-dimensional spaces Rimon and Koditschek introduced Navigation Functions (NF) [13], using a family of APFs that is applied in a spherical transformation of the workspace. However, the transformation of the workspace into a spherical world relies on complex numbers and is therefore very difficult to be extended to 3D space. In more recent works, researchers have explored the use of Artifical Harmonic Potential Fields (AHPFs) [14–18] in two dimensions, as they are, by construction, free of local minima. Optimal path planning solutions for AHPFs have shown promising results in two-dimensional workspaces by using integral reinforcement learning [19–21]. The non-optimal motion planning problem has also been addressed in

3D environments [22, 23]. Yet, optimal motion planning methods using APFs in three dimensions remains an unexplored area.

## 1.3 Thesis Contribution

In this thesis, a novel approach for the optimal motion planning problem in continuous three-dimensional and constrained spaces is presented. More specifically, the differential equations of fluid flow are utilized to construct the AHPF motion planner. The motivation here stems from identifying the robot as being similar to a fluid particle, which obeys the fundamental laws of mass and momentum conservation in fluid mechanics. By adopting the assumptions of potential flow theory that render the flow incompressible, inviscid and irrotational, it is ensured that all fluid particles will move towards a sink element. Consequently, the robot, guided by the natural-looking trajectories of the fluid flow motion, is proven to eventually converge to the desired goal position. To ensure that the controller can be implemented in real time, the boundary of the workspace is discretized into surface panels and a solution for the potential flow equations is derived analytically. Moreover, in order to meet the safety criteria of the motion planning problem, a set of von-Neumann boundary conditions on the discretized geometry is employed.

In contrast to most existing optimization algorithms in motion planning, which are either stochastic in nature or heavily influenced by sampling techniques (e.g., evolutionary algorithms and model-free reinforcement learning approaches), this work introduces a robust deterministic optimal solution. The proposed optimal path planning algorithm merges rigid principles of optimal control theory with a model-based deep reinforcement learning approach. In particular, based on the successive approximation theory [24, 25] and dynamic programming [26, 27], an actor-critic reinforcement learning architecture is employed. Since a model of the environment can be derived using the proposed AHPF motion planner, the critic neural network leverages on-trajectory sampling data to approximate the Hamilton-Jacobi-Bellman equation. Subsequently, the actor scheme is responsible for improving the control parameters while preserving the harmonic properties and safety guarantees of the motion planner.

The contribution of this work lies not only on the provable safe and convergent nature of the motion planner, but also on its ability to provide deterministic optimal solutions. The main innovative aspects of this research are summarized as follows:

- A proved stable and safe control policy for the motion planning problem in obstacle-cluttered and constrained 3D workspaces is established.

- In contrast to discrete open-loop methods, the proposed reactive controller generates closed-loop, continuous and smooth velocity commands, which can be seamlessly integrated in most control systems.

- A deterministic optimization procedure is accomplished. Moreover, incorporating traditional optimal control principles into the reinforcement learning framework, guarantees the convergence and robustness of the proposed optimization algorithm.
- The proposed approach is shown to outperform state-of-the-art optimal path planning algorithms such as the RRT$^*$.

## 1.4 Thesis Outline

In **Chapter 2** the optimal motion planning problem is formulated along with preliminary control theory and fluid mechanics tools. Next, in **Chapter 3** the control policy for a safe and stable 3D motion planner is proposed. **Chapter 4** presents the optimal solution for the motion planning problem by introducing a continuous model-based reinforcement learning algorithm. The effectiveness of the proposed methodology is demonstrated in **Chapter 5** through extensive simulations. This thesis is concluded in **Chapter 6** along with its limitations and the proposed future directions.

# Chapter 2

# The Optimal Motion Planning Problem

## 2.1 Problem Formulation

Consider a point robot operating within a three-dimensional bounded and connected workspace $\mathcal{H} \subset \mathbb{R}^3$ with inner distinct obstacles $O_i, i = 1, \ldots, M$, along with a desired goal position $\mathbf{p_d} \in \mathcal{W} - \partial\mathcal{W}$. Also, assume that the robot has fully knowledge of the workspace and is subjected to single integrator dynamics:

$$\dot{\mathbf{p}}(t) = \mathbf{u}(t), \qquad \mathbf{p}(0) = \mathbf{p_0} \tag{2.1}$$

where $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\top \in \mathcal{W} \triangleq \mathcal{H} - \bigcup_{i=1}^{M} \mathcal{O}_i$ is the fully observed state of the robot, $\mathbf{p_0}$ is the initial position and $\mathbf{u}(t)$ is the control policy, i.e., a velocity command. Next, the infinite horizon cost function is defined, consisting of a state-related term $Q$ and an input-related term $R$:

$$V(\mathbf{p_0}; \mathbf{p_d}) = \int_0^\infty Q\left(\mathbf{p}(\tau); \mathbf{p_d}\right) + R\left(\mathbf{u}(\tau)\right)\, d\tau. \tag{2.2}$$

The goal of this thesis is to derive the optimal control policy $\mathbf{u}(t)$ for the dynamics of equation (2.1) that safely drives the robot to the desired position $\mathbf{p_d}$ while at the same minimizes the cost function (2.2).

## 2.2 Preliminaries

### 2.2.1 Optimal Nonlinear Control

Optimal control plays a fundamental role in the field of control systems. Of particular interest is the Hamilton-Jacobi-Bellman (HJB) equation, which serves as a means to ensure optimality across various applications. In this section, the derivation of the HJB equation is studied, establishing the theoretical foundations necessary to minimize the cost function of a continuous-time non-linear system. Consider the dynamics of the non-linear system:

$$\dot{x} = f(x(t); u(t), t), \tag{2.3}$$

with state $x(t) \in \mathbb{R}^n$ and control input $u(t) \in \mathbb{R}^m$, subject to the usual assumptions to stabilize the system in its equilibrium point. Also, the value function of the optimal problem is defined in a generic form as follows:

$$V(x(t)) = \int_0^\infty L(x(\tau); u(\tau))\, d\tau. \tag{2.4}$$

The objective here is to find a continuous optimal control $u^*(t)$ that drives the system from an initial state $x(t_0)$ to a final state $x(t_f)$ with a minimum cost $V^*(x(t_0), x(t_f))$. Note that once the optimal policy $u^*(t)$ and consequently the system dynamics are specified, the value function depends only on the initial and final state of the system. According to Bellman's principle of optimality the optimal control problem over the total time interval $[t_0, t_f]$ can be decomposed in determining the optimal cost-to-go in smaller consecutive time steps $t + \Delta t$. By deriving the total time derivative of the value function:

$$\frac{d}{dt} V(x(t)) = \frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial x}\right)^\top \frac{dx}{dt},$$

Bellman's optimality condition for continuous-time systems results in a partial differential equation for the optimal cost:

$$-\frac{\partial V^*}{\partial t} = \arg\min_{u(t)} \left( \left(\frac{\partial V^*}{\partial x}\right)^\top f(x(t); u(t)) + L(x(t); u(t)) \right). \tag{2.5}$$

This expression is known as the Hamilton Jacobi Bellman (HJB) equation with the corresponding Hamiltonian function:

$$H(x(t); u(t); \nabla V) = L(x(t); u(t)) + \nabla V^\top f(x(t); u(t)). \tag{2.6}$$

The two-point boundary value problem of the HJB equation provides an optimal solution for general non-linear control systems. However, solving the HJB analytically proves to be challenging in most cases. As a result, this thesis aims to integrate suitable machine learning techniques with the principles of classical control theory in order to approximate the optimal solution for the 3D motion planning problem.

### 2.2.2 Harmonic Artificial Potential Fields

Artificial Potential Fields (APFs) have been widely used in path planning applications. APFs provide a suitable navigation policy by modeling the robot's environment through a potential field, which exhibits high values in proximity to the obstacles and a single local minimum in the goal position. To achieve this, a repulsive term responsible for preventing robot collisions, along with an attractive term that ensures convergence to the desired position, should be properly designed. The general formulation of the potential field, given the boundary geometry $\mathbf{s}$ and the desired goal position $\mathbf{p_d}$ can be expressed as follows:

$$\Phi(\mathbf{p};\mathbf{s}) = \Phi_{Repulsive}(\mathbf{p};\mathbf{s}) + \Phi_{Attractive}(\mathbf{p};\mathbf{p_d}). \tag{2.7}$$

By following velocity commands of the negated gradient of the potential field $\mathbf{u} = -\nabla\Phi(\mathbf{p};\mathbf{s})$, the robot can navigate safely towards the target position.

The main limitation of APFs is the presence of local minima inside the workspace. As the complexity of the environment rises it is highly likely that the repulsive forces from the boundary will counterbalance the attractive term, causing the robot to get stuck in a local minimum. To overcome this drawback, a set of harmonic functions is introduced in this work, which owing to the Maximum Principle cannot attain a maximum at any interior point of the domain [28]. Thus, the harmonic potential field eliminates the robot's risk of becoming trapped anywhere else besides in the goal position.

In the following section the analytical solution of a harmonic potential field that relies solely on boundary information will be presented.

### 2.2.3 General Solution of the Laplace Equation

The objective of the present section is to introduce the theoretical framework of the fundamental solutions method for solving the Laplace equation. This approach has a significant contribution in Fluid Mechanic applications [29–32], where it has been utilized to numerically solve the incompressible potential flow both around and inside bodies of complex geometry. The main advantage is, that solving a fluid dynamic problem in an entire workspace, reduces to defining the appropriate conditions only on the boundary, which also eliminates the necessity of constructing a grid in the entire space.

Consider a simply connected workspace $\mathcal{W}$ with volume $V$ enclosed by the outer boundary $S$ as shown in Figure 2.1. Note that the normal vector to the boundary $\mathbf{n}$ is defined such that it always points outwards. The desired potential field $\Phi$ is obtained by solving the Laplace equation

$$\nabla^2\Phi = 0, \tag{2.8}$$

using the Fundamental Solutions Method.

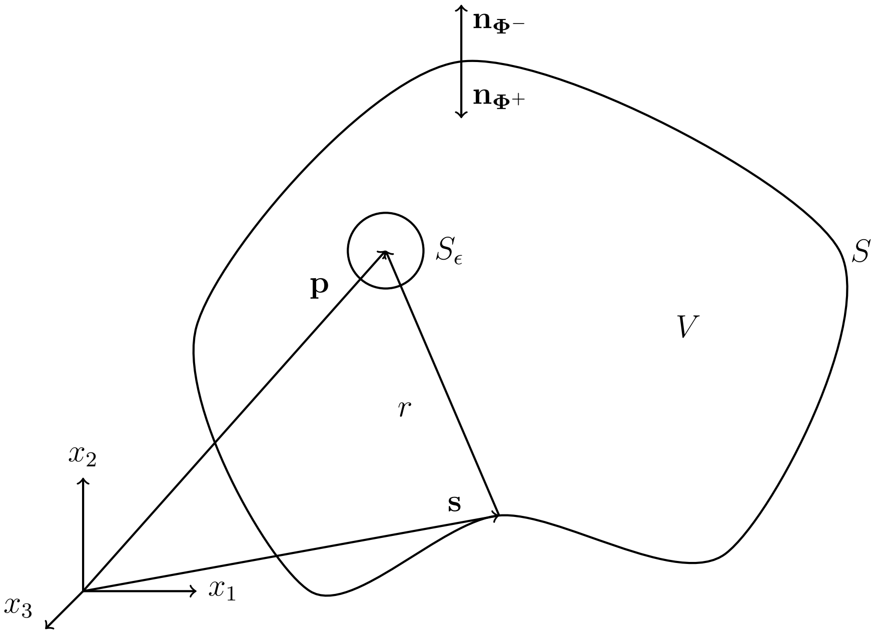

$\mathbf{n}_{\mathbf{\Phi}^-}$
$\mathbf{n}_{\mathbf{\Phi}^+}$
$S_\epsilon$
$S$
$\mathbf{p}$
$V$
$r$
$x_2$
$\mathbf{s}$
$x_1$
$x_3$

**Figure 2.1:** *The thee-dimensional workspace illustrating the enclosed volume to satisfy the Laplace equation.*

In this context, Gauss's theorem is used to relate the divergence of a vector field $\mathbf{F}$ inside the volume with the flux through the corresponding closed surface, thus setting the first step in the development of a boundary element method:

$$\iiint_V (\nabla \cdot \mathbf{F})\ dV = \oiint_S (\mathbf{F} \cdot \mathbf{n})\ dS. \tag{2.9}$$

By selecting $\mathbf{F} = \Phi_1\nabla\Phi_2 - \Phi_2\nabla\Phi_1$, one can derive Green's second identity, which is expressed as follows:

$$\iiint_V \left(\Phi_1\nabla^2\Phi_2 - \Phi_2\nabla^2\Phi_1\right)\ dV = \oiint_S (\Phi_1\nabla\Phi_2 - \Phi_2\nabla\Phi_1\cdot)\,\mathbf{n}\,dS. \tag{2.10}$$

Green's second identity is a powerful tool for transforming the partial differential form of the Laplace equation to an integral equation. The next step is to select two scalar functions of position:

$$\begin{aligned}\Phi_1 &= \frac{1}{r}\\ \Phi_2 &= \Phi,\end{aligned}$$

where $\Phi$ is the desired potential field of (2.8) and $r = \|\mathbf{p} - \mathbf{s}\|$ is the distance of the position $\mathbf{p}$ from the boundary surface, as shown in Figure 2.1.

Since the goal is to determine a harmonic potential field, satisfying $\nabla^2\Phi = 0$, the first term of the volume integral is eliminated. However, as point $\mathbf{p}$ gets closer to the surface, a singularity is observed owing to $\lim_{r\to 0}\frac{1}{r} = \infty$, thus $\nabla^2\Phi_1 = 0, \forall r \in \mathbb{R}\setminus\{0\}$. To completely eliminate the volume integral, a small area $\epsilon$ around the point $\mathbf{p}$ is excluded from the integration and (2.10) takes the form:

$$0 = \oiint_S (\Phi_1\nabla\Phi_2\cdot\mathbf{n} - \Phi_2\nabla\Phi_1\cdot\mathbf{n})\, dS - \oiint_{S_\epsilon} (\Phi_1\nabla\Phi_2\cdot\mathbf{n}_\epsilon - \Phi_2\nabla\Phi_1\cdot\mathbf{n}_\epsilon)\, dS. \quad (2.11)$$

By integrating over the surface of the sphere, (2.11) reduces to:

$$\begin{aligned}0 &= \oiint_S \left(\frac{1}{r}\nabla\phi\cdot\mathbf{n} - \Phi\nabla\frac{1}{r}\cdot\mathbf{n}\right) dS - 4\pi\Phi \Rightarrow\\ \Phi &= \frac{1}{4\pi}\oiint_S \left(\frac{1}{r}\nabla\phi\cdot\mathbf{n} - \Phi\nabla\frac{1}{r}\cdot\mathbf{n}\right) dS. \qquad (2.12)\end{aligned}$$

Next, the expression of the partial derivatives with respect to the surface is obtained as:

$$\nabla_S\left(\frac{1}{r}\right)\cdot\mathbf{n} = \nabla_S\left(\frac{1}{|\mathbf{p}-\mathbf{s}|}\right)\cdot\mathbf{n} = \frac{\mathbf{r}\cdot\mathbf{n}}{r^3} \quad (2.13)$$

$$\nabla_S\Phi\cdot\mathbf{n} = \frac{\partial\Phi}{\partial n}, \quad (2.14)$$

which results in the final expression of the potential field $\Phi$ inside the volume $V$:

$$\Phi(\mathbf{p}) = \frac{1}{4\pi}\oiint_S \left(\frac{\partial\Phi}{\partial n}(\mathbf{s})\frac{1}{|\mathbf{p}-\mathbf{s}|} - \Phi(\mathbf{s})\frac{(\mathbf{p}-\mathbf{s})\cdot\mathbf{n}(\mathbf{s})}{|\mathbf{p}-\mathbf{s}|^3}\right) dS(\mathbf{s}). \quad (2.15)$$

A closer look in equation (2.15) reveals that the potential field inside the volume $V$ relies solely on information on the boundary. Therefore, the solution of the Laplace equation (2.8) is not unique and based on the selection of the boundary conditions $\frac{\partial \Phi}{\partial n}$ and $\Phi(\mathbf{s})$ the resulting potential field will be represented by the composition of different fundamental solutions.

So far, only the inner potential field enclosed by the boundary surface $S$ has been considered. However, based on the convention that the normal vector on the boundary is pointing away from the volume being considered, the boundary $S$ separates the problem into an inner and an outer potential field.

$$\Phi^{+}(\mathbf{p}) = \frac{1}{4\pi} \oiint_{S} \left( \frac{\partial \Phi}{\partial n}(\mathbf{s}) \frac{1}{|\mathbf{p} - \mathbf{s}|} - \Phi(\mathbf{s}) \frac{(\mathbf{p} - \mathbf{s}) \cdot \mathbf{n}(\mathbf{s})}{|\mathbf{p} - \mathbf{s}|^{3}} \right) dS(\mathbf{s}) \tag{2.16}$$

$$\Phi^{-}(\mathbf{p}) = -\frac{1}{4\pi} \oiint_{S} \left( \frac{\partial \Phi}{\partial n}(\mathbf{s}) \frac{1}{|\mathbf{p} - \mathbf{s}|} - \Phi(\mathbf{s}) \frac{(\mathbf{p} - \mathbf{s}) \cdot \mathbf{n}(\mathbf{s})}{|\mathbf{p} - \mathbf{s}|^{3}} \right) dS(\mathbf{s}). \tag{2.17}$$

This theory can be extended to a workspace that contains any arbitrary number of non-intersecting sub-spaces. To do this, the direction of the normal vectors should be adjusted based on the volume of interest. Then, owing to the principle of superposition, the resulting potential field is obtained by adding the respective contributions from the boundary surfaces. In the context of robot navigation, inner obstacles of the environment can be modeled as different sub-spaces, effectively addressing the challenges posed by multi-connected workspaces.

# Chapter 3

# Control Design Methodology

The primary focus of this chapter is to present a parameterized control policy for the motion planning problem in three-dimensional workspaces. The controller generates continuous-time trajectories that aim to ensure convergence towards the goal position while satisfying all safety criteria. Additionally, the proposed methodology provides a robust navigation solution for any initial position inside the workspace, guaranteeing both reactivity and smoothness.

## 3.1 Proposed Control Policy

In the context of AHPF theory, the goal is to find a free of local minima control policy that attracts the robot towards the desired position and repulses it away from obstacles. The proposed controller utilizes the properties of the solution of a Laplace problem to construct such a vector field that provides safe and stable navigation.

In this work a distribution of singularity elements on the boundary of the workspace is employed, which as presented in section 2.2.3 results in a potential field that satisfies the Laplace equation. By following the negated gradient of the potential, the proposed parameterized controller is formulated as:

$$\mathbf{u}(\mathbf{p}) = -\nabla\Phi(\mathbf{p};\mathbf{s}) = -\mathbf{w}(\mathbf{s})\mathbf{v}(\mathbf{p};\mathbf{s}), \tag{3.1}$$

where $\mathbf{v}(\mathbf{p};\mathbf{s})$ denotes the basis function of the velocity in position $\mathbf{p} \in \mathbb{R}^3$ that is induced by the distribution of the singularity elements on the boundary $S$, and $\mathbf{w}(\mathbf{s})$ are the respective weights, in this case also the tuning parameters of the control policy.

## 3.2 Proposed Fluid Flow Model

### 3.2.1 Type of Singularity Elements

Having presented the fundamental theory of the boundary element method, the purpose of this section is to discuss the choice of an appropriate basis function, capable of providing a vector field for the navigation of the robot that meets all the control requirements.

The potential field that was derived using Green's second identity does not pose a unique representation. Based on the chosen boundary conditions one can derive many combinations of singularity elements to best fit the nature of the problem. By taking a closer look in (2.15), it can be noticed that the first term corresponds to a distribution of point sources-sinks and the second term to a distribution of point doublets with weights $\sigma = \frac{\partial \Phi}{\partial n}$ and $\mu = \Phi(\mathbf{s})$ respectively:

$$\Phi_{Repulsive}(\mathbf{p};\mathbf{s}) = \frac{1}{4\pi} \oiint_S \left( \sigma(\mathbf{s}) \frac{1}{|\mathbf{p}-\mathbf{s}|} - \mu(\mathbf{s}) \frac{(\mathbf{p}-\mathbf{s}) \cdot \mathbf{n}(\mathbf{s})}{|\mathbf{p}-\mathbf{s}|^3} \right) dS. \tag{3.2}$$

The vector fields of the singularity elements, illustrated in Figure 3.1 and in Figure 3.2, are capable of modeling various potential flow problems. In the case of motion planning the interest is to construct a repulsive from the boundary potential field, thus we set $\mu(s) = 0$ to obtain a representation of sources distribution on the boundary of the workspace. Moreover, to ensure the convergence of the control policy to the desired position, an additional attractive term in the goal position is employed. The final potential field $\Phi(\mathbf{p};\mathbf{s}) = \Phi_{Repulsive}(\mathbf{p};\mathbf{s}) + \Phi_{Attractive}(\mathbf{p_d})$ inside the workspace is given as:

$$\Phi(\mathbf{p};\mathbf{p_d};\mathbf{s}) = \frac{1}{4\pi} \left( \oiint_S \left( \sigma(\mathbf{s}) \frac{1}{|\mathbf{p}-\mathbf{s}|} \right) dS - \sigma_d \frac{1}{|\mathbf{p}-\mathbf{p_d}|} \right). \tag{3.3}$$

Here the integral is calculated over the boundary surface $S(\mathbf{s}) \triangleq \partial W$, $\mathbf{p_d} \in \mathcal{W} - \partial\mathcal{W}$ is the goal position and $\sigma(\mathbf{s}), \sigma_d$ denote the weights of the source elements on the boundary and in the goal position respectively.

It should be noticed that the attractive term of the potential field exhibits a singularity in the goal position which does not have a physical meaning. For that reason, the position of the point source is excluded from the workspace and the corresponding flux can be calculated by integration of a spherical surface around $\mathbf{p}_d$ as follows:

$$\oiint_{\mathcal{G}} \left( -\sigma_d \nabla_p \left( \frac{1}{|\mathbf{p}-\mathbf{p_d}|} \right) \cdot \mathbf{n}_{\mathcal{G}} \right) = w_0. \tag{3.4}$$

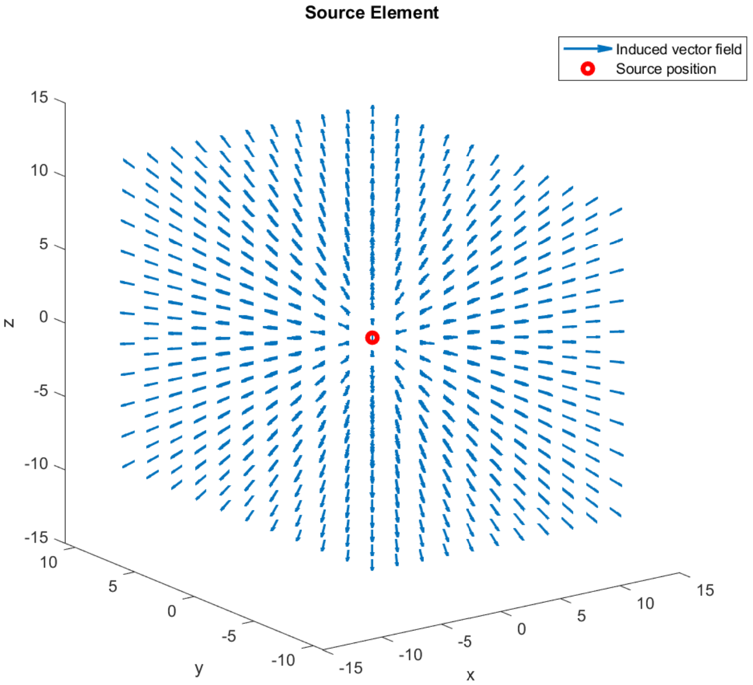


**Figure 3.1:** *Vector field induced by a source singular point.*

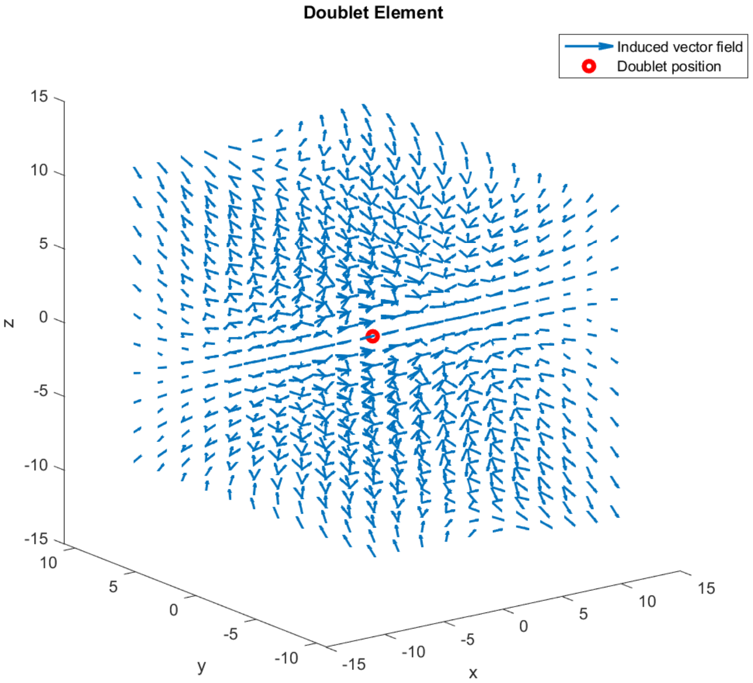


**Figure 3.2:** *Vector field induced by a doublet singular point.*

### 3.2.2 The Panel Method

At this point, a selection of the type of boundary conditions has been made, which specifies the type of singularity elements that will be placed on the boundaries. To further simplify the formulation of the potential field, the boundary geometry is discretized into flat two-dimensional panels, reducing this way the complexity of the surface integral.

Regarding the weights distribution of the source terms, we choose that the weights (i.e, control parameters) within each panel are constant. By discretizing the surface in $K$ panels, the potential field (3.3) is now formulated as:

$$\Phi(\mathbf{p};\mathbf{p_d};\mathbf{s}) = \sum_{i=1}^{K}\left(\frac{1}{4\pi}\oint_{\Delta S_i}\sigma_i\frac{1}{|\mathbf{p}-\mathbf{s}|}\,dS\right) - \sigma_d\frac{1}{4\pi|\mathbf{p}-\mathbf{p_d}|}. \tag{3.5}$$

The corresponding velocity induced by the boundary elements in the three-dimensional space is given by:

$$\begin{aligned}\nabla_p\Phi(\mathbf{p};\mathbf{p_d};\mathbf{s}) &= \sum_{i=1}^{K}\left(\frac{1}{4\pi}\sigma_i\oint_{\Delta S_i}\frac{\mathbf{p}-\mathbf{s}}{|\mathbf{p}-\mathbf{s}|^3}\,dS\right) - \sigma_d\frac{\mathbf{p}-\mathbf{p_d}}{4\pi|\mathbf{p}-\mathbf{p_d}|^3}\\ &= \sum_{i=1}^{K}\left(\sigma_i\mathbf{v}_i(\mathbf{p};\mathbf{s})\right) - \sigma_d\frac{\mathbf{p}-\mathbf{p_d}}{4\pi|\mathbf{p}-\mathbf{p_d}|^3}.\end{aligned} \tag{3.6}$$

After the geometry discretization and the selection of the weight's distribution order in each panel are defined, the total velocity in position $\mathbf{p}\in\mathbb{R}^3$ induced by the repulsive potential field is limited to finding the induced velocity by each single panel and specifying its respective weight. The total vector field is essentially decomposed into the basis function $\mathbf{v}(\mathbf{p};\mathbf{s})$, where each term corresponds to the influence of a single panel on position $\mathbf{p}$. Defining each panel $i=1,\dots,K$ by the three vertices $\mathbf{p}_{e_{i,j}} = \left[x_j, y_j, z_j\right]^\top \in\mathbb{R}^3$, $j=1,\dots,3$, the computation of each basis function term is represented schematically as follows:

$$\begin{pmatrix} x,y,z\\ x_1,y_1,z_1\\ x_2,y_2,z_2\\ x_3,y_3,z_3\\ \sigma \end{pmatrix}_i \overset{{}^0\mathbf{T}_i}{\Rightarrow} \begin{pmatrix} \xi,\eta,\zeta\\ \xi_1,\eta_1,\zeta_1\\ \xi_2,\eta_2,\zeta_2\\ \xi_3,\eta_3,\zeta_3 \end{pmatrix}_i \Rightarrow \begin{pmatrix} \text{Velocity influence coefficient of panel } i\\ \text{in the local coordinate system:}\\ \mathbf{v}'_i \end{pmatrix}$$

$$\overset{{}^0\mathbf{T}_i^\top}{\Rightarrow} \begin{pmatrix} \text{Velocity influence coefficient of panel } i\\ \text{in the global coordinate system:}\\ \mathbf{v}_i \end{pmatrix}$$

It is important to observe that each coefficient depends only on the panel's geometric characteristics and can be calculated analytically. More precisely, in order to calculate the velocity coefficient $\mathbf{v}_i$ in position $\mathbf{p} \in \mathbb{R}^3$, the local coordinates of each panel are being used as shown in Figure 3.3. This way, the computation of each term is simplified since the integral runs over the two-dimensional surface of the panel and the resulting velocity is then transformed to the global 3D coordinate system.

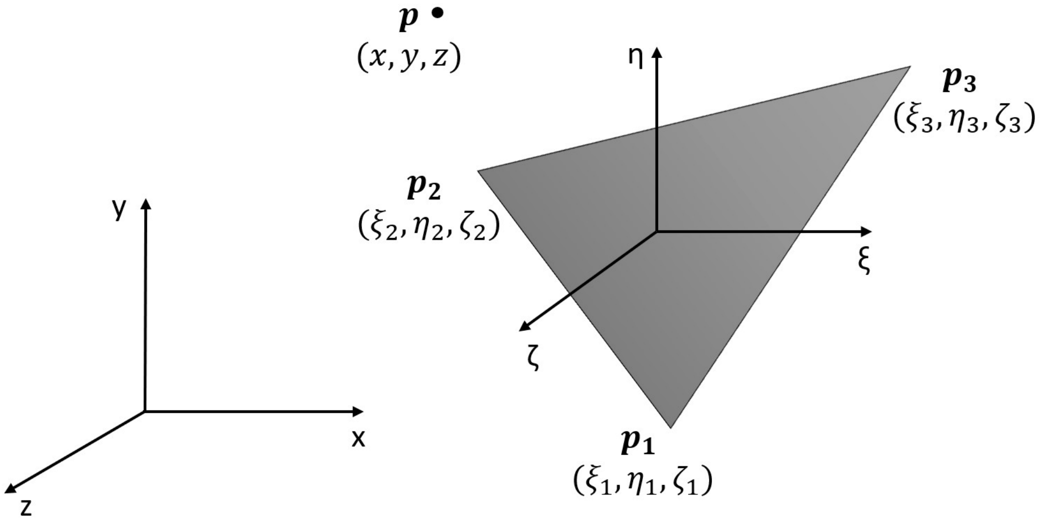


**Figure 3.3:** *Local coordinate system of triangular panels.*

## 3.3 Control Policy Structure

Having presented the formulation of the Fundamental Solutions Method (FSM) to obtain a solution of the Laplace problem, it will be used in the context of path planning as the basis of the Artificial Harmonic Potential Field (AHP). Additionally, in order to analytically calculate the velocity potential within the workspace $\mathcal{W}$ that is induced by a sources distribution on the boundary $\partial\mathcal{W}$, the panel method is employed. The control policy is defined by the following structure:

$$\mathbf{u}(\mathbf{p}) = -\nabla\Phi(\mathbf{p}; \mathbf{p_d}; \mathbf{w}; \mathbf{M}) = -\mathbf{w}^\top \mathbf{v}(\mathbf{p}; \mathbf{p_d}; \mathbf{M}), \tag{3.7}$$

where $\mathbf{M} \triangleq \left[\mathbf{M}_1, \ldots, \mathbf{M}_K\right] \in \mathbb{R}^{3K \times 3}$ is a combined matrix of the connectivity list and the triangular mesh vertices, while $\mathbf{p}$, $\mathbf{p_d} \in \mathbb{R}^3$ denote the current and the goal position of the robot respectively. Each surface panel element $\mathbf{M}_i = \left[\mathbf{p}_{e_{i,1}}, \mathbf{p}_{e_{i,2}}, \mathbf{p}_{e_{i,3}}\right]^\top$, $i = 1, \ldots, K$ is represented by the position of the corresponding edge points $\mathbf{p}_{e_{i,j}} \in \mathbb{R}^3$, $j = 1, 2, 3$. Also, $\mathbf{w} = \left[w_0, w_1, \ldots w_K\right]^\top \in \mathbb{R}^{K+1}$ are the tunable parameters of the control policy, properly adjusting the potential field that is formed by the function basis:

$$\mathbf{v}(\mathbf{p}; \mathbf{p_d}; \mathbf{M}) \triangleq \left[\mathbf{v}_0(\mathbf{p}; \mathbf{p_d}), \mathbf{v}_1(\mathbf{p}; \mathbf{M}), \ldots, \mathbf{v}_K(\mathbf{p}; \mathbf{M})\right]^\top : \mathbb{R}^3 \to \mathbb{R}^{K+1}. \tag{3.8}$$

The sink $\mathbf{v}_0(\mathbf{p}; \mathbf{p_d})$ creates an attractive towards the goal position vector field:

$$\mathbf{v}_0(\mathbf{p}; \mathbf{p_d})^\top = \frac{1}{4\pi} \frac{\mathbf{p} - \mathbf{p_d}}{|\mathbf{p} - \mathbf{p_d}|^3}, \tag{3.9}$$

while the rest of the basis constitutes a repulsive potential field from the boundary as follows:

$$\begin{aligned} \mathbf{v}_i(\mathbf{p}; \mathbf{M}_i) &= \left[v_{i_x}(\mathbf{p}; \mathbf{M}_i), v_{i_y}(\mathbf{p}; \mathbf{M}_i), v_{i_z}(\mathbf{p}; \mathbf{M}_i)\right] \\ &= {}^{M_i}\mathbf{T}_0 \cdot \left[v_{i_\xi}(\mathbf{p}'; \mathbf{M}'_i), v_{i_\eta}(\mathbf{p}'; \mathbf{M}'_i), v_{i_\zeta}(\mathbf{p}'; \mathbf{M}'_i)\right]^\top, \end{aligned} \tag{3.10}$$

where ${}^{M_i}\mathbf{T}_0$ is the transformation matrix from the local coordinate system of the panel $O_{M_i}(\xi, \eta, \zeta)$ to the global coordinate system $O_0(x, y, z)$ with $\mathbf{p}' = \left[\mathbf{p}, 1\right] \cdot {}^0\mathbf{T}_{M_i}$, $\mathbf{M}'_i = \left[\mathbf{M}_i, \mathbf{1}_{3\times1}\right] \cdot {}^0\mathbf{T}_{M_i}$ expressing the position of the robot and the position of the panel vertices w.r.t. the panel's local coordinate system respectively. The velocity in position $\mathbf{p}' = \left[\xi, \eta, \zeta\right] \in \mathcal{W}$, induced by the $i^{th}$ panel, is calculated in the local coordinate system of the panel as follows:

$$v_{i_\xi} = \frac{1}{4\pi}\left(S_{12}^{(i)}Q_{12}^{(i)} + S_{23}^{(i)}Q_{23}^{(i)} + S_{31}^{(i)}Q_{31}^{(i)}\right) \tag{3.11}$$

$$v_{i_\eta} = \frac{1}{4\pi}\left(-C_{12}^{(i)}Q_{12}^{(i)} - C_{23}^{(i)}Q_{23}^{(i)} - C_{31}^{(i)}Q_{31}^{(i)}\right) \tag{3.12}$$

$$v_{i_\zeta} = \frac{1}{4\pi}\left(J_{12}^{(i)} + J_{23}^{(i)} + J_{31}^{(i)}\right) \tag{3.13}$$

where:

$$
\begin{aligned}
\mathbf{M}'_i &= \left[\mathbf{p}'_{e_{i,1}}, \mathbf{p}'_{e_{i,2}}, \mathbf{p}'_{e_{i,3}}\right]^\top \\
\mathbf{p}'_{e_{i,j}} &= \left[\xi_{i,j}, \eta_{i,j}, \zeta_{i,j}\right] \\
S_{ab}^{(i)} &= \frac{\eta_{i,b} - \eta_{i,a}}{d_{ab}^{(i)}} \\
C_{ab}^{(i)} &= \frac{\xi_{i,b} - \xi_{i,a}}{d_{ab}^{(i)}} \\
Q_{ab}^{(i)} &= \ln \frac{r_a^{(i)} + r_b^{(i)} - d_{ab}^{(i)}}{r_a^{(i)} + r_b^{(i)} + d_{ab}^{(i)}} \\
J_{ab}^{(i)} &= \tan^{-1}\left(\frac{m_{ab} e_a^{(i)} - h_a^{(i)}}{z r_a^{(i)}}\right) - \tan^{-1}\left(\frac{m_{ab} e_b^{(i)} - h_b^{(i)}}{z r_b^{(i)}}\right) \\
d_{ab}^{(i)} &= \sqrt{(\xi_{i,b} - \xi_{i,a})^2 + (\eta_{i,b} - \eta_{i,a})^2} \\
m_{ab}^{(i)} &= \frac{\eta_{i,b} - \eta_{i,a}}{\xi_{i,b} - \xi_{i,a}} \\
r_a^{(i)} &= \sqrt{(\xi - \xi_{i,a})^2 + \zeta^2} \\
e_a^{(i)} &= (\xi - \xi_{i,a})^2 + \zeta^2 \\
h_a^{(i)} &= (\xi - \xi_{i,a})(\eta - \eta_{i,a})
\end{aligned}
$$

In this AHPF-structure the harmonic panels are placed outside of the boundary to avoid any singularities in the vertices. Additionally, convergence towards the goal position is ensured by the formulation of the control policy since the source term $\mathbf{v}_0(\mathbf{p}; \mathbf{p_d})$ renders the vector field attractive. Owing to the minimum-maximum principle of harmonic functions, the resulting potential field will exhibit a single minimum point in the robot's goal position.

## 3.4 Safety Conditions

To guarantee collision avoidance in the proposed navigation policy, this section introduces a set of safety criteria. Safety is ensured by imposing the following conditions as hard constraints in the motion planning problem:

$$\mathbf{n}(\mathbf{p})^\top \mathbf{u}(\mathbf{p}) \geq 0, \forall \mathbf{p} \in \partial \mathcal{W}, \tag{3.14}$$

where $\mathbf{n}(\mathbf{p})$ is the normal vector on a boundary point and $\mathbf{u}(\mathbf{p})$ is the control input, dictating a velocity command at the boundary that always points inwards. As has been proved in **Lemma 1**, this property can be relaxed if the safety condition holds for a finite set of boundary points.

**Lemma 1** (Boundary Safety [20])**.** *Consider the boundary $\partial\mathcal{W}$ of the workspace as well as a finite number of uniformly distributed points $\mathbf{p}_j \in \partial\mathcal{W}, j = 1, \ldots, N$ along with their respective normal vectors $\mathbf{v} = \mathbf{v}(\mathbf{p}_j), j = 1, \ldots, N$ pointing inwards the workspace. There exists a number $N_0 \in \mathbb{N}$ such that*

$$\mathbf{v}^\top \mathbf{v} > 0, \quad \forall j = 1, \ldots, N \quad \textit{with} \quad N > N_0, \tag{3.15}$$

*guarantees safety over the whole boundary $\partial\mathcal{W}$ as described by 3.14.*

*Proof.* The proof is extensively presented in [20]. □

Hence, the safety condition (3.14) can be expressed as a set of $N$ linear inequalities with respect to the weights of the basis function:

$$\mathbf{A} \cdot \mathbf{w} \leq \mathbf{0}_{N\times 1} \Rightarrow$$
$$\begin{bmatrix} A_1^1 & \ldots & A_1^{K+1} \\ \vdots & \ddots & \vdots \\ A_N^1 & \cdots & A_N^{K+1} \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ \vdots \\ w_{K+1} \end{bmatrix} \leq \mathbf{0}_{N\times 1}, \tag{3.16}$$

where $A_b^i = \mathbf{n}^\top(\mathbf{p}_b)\mathbf{v}_i(\mathbf{p_b}; \mathbf{M}_i)$, $i = 1, \ldots, K+1$ corresponds to $i^{\text{th}}$ panel term of the induced normal velocity on the boundary point $\mathbf{p}_b \in \partial\mathcal{W}$, $b = 1, \ldots, N$.

Finally, matrix $\mathbf{A}$ is augmented by an additional condition $A_s = \big[1, 0, \ldots, 0)\big] \in \mathbb{R}^{1\times(K+1)}$ in order to address the convergence of the robot towards the goal position by guaranteeing a positive weight for the point source term.

Consequently, the control policy that satisfies both the safety and the convergence requirements of the motion planning problem is formulated as follows:

$$\begin{aligned} \mathbf{u}(\mathbf{p}) = -\nabla\Phi(\mathbf{p}; \mathbf{p_d}; \mathbf{M}) &= -\mathbf{w}^\top \mathbf{v}(\mathbf{p}; \mathbf{p_d}; \mathbf{M}) \\ \text{s.t} \quad \mathbf{A}\mathbf{w} &\leq -\epsilon \mathbf{I}_{(N+1)\times 1} \end{aligned} \tag{3.17}$$

To acquire an initial policy that satisfies the harmonic properties and the safety constraints, the following dummy quadratic optimization problem can be solved:

$$\begin{aligned} \mathbf{w}^0 &= \arg\min_{\mathbf{w}}\{\mathbf{w}^\top \mathbf{w}\} \\ \text{s.t} \quad \mathbf{A}\mathbf{w} &\leq -\epsilon \mathbf{I}_{(N+1)\times 1}, \end{aligned} \tag{3.18}$$

where $\epsilon$ is a small positive number to set a lower limit for the magnitude of the attractive potential field. Additionally, as $\epsilon$ approaches zero, the velocity field becomes tangent to the boundary surface. In that manner, $\epsilon$ can also compensate for a safe distance threshold from the boundary. Important to note is that when the number of control parameters is equal to the number of points at which the safety condition is satisfied, the problem simplifies into a linear set of equations.

Besides delivering a robust and stable robot navigation solution, the control parameters $\mathbf{w}^0$ also serve the purpose of initializing the optimization algorithm, as will be discussed in Chapter 4.

## 3.5 Stability Analysis

In this section the proofs of safety and convergence for the proposed structure are presented. First, the existence of a solution for the control system (3.7) is established. Subsequently, a stability analysis is carried out to validate that the non-linear controller stabilizes the system at the goal position.

**Lemma 2** (Existence of Solution)**.** *There exists a set of control parameters* $\mathbf{w}$ *for the proposed control policy* $\mathbf{u} = -\nabla\Phi$ *(3.7), such that:* ***1)*** *the potential field satisfies the Laplace equation* $\nabla^2\Phi = 0$ *and* ***2)*** *the problem constraints (3.18) are satisfied.*

*Proof.* In order to establish the existence of a solution for the formulation of the motion planning problem, we begin from a set of parameters $\mathbf{w}$ that provide a Laplacian potential field $\Phi$. Then it will be demonstrated that this set also satisfies the constraints $\mathbf{A}\mathbf{w} \leq -\epsilon\mathbf{I}_{(N+1)\times 1}$. As presented in section 2.2.3, the controller is parameterized by a basis of harmonic functions, which guarantees that both the curl and divergence of the vector field are zero, satisfying by definition the Laplace equation. Owing to Gauss's theorem the velocity field inside the workspace is related to the safety conditions on the boundary as follows:

$$\oiint_{\partial\mathcal{W}} (\mathbf{u} \cdot \mathbf{n})\ d(\partial\mathcal{W}) = \iiint_{\mathcal{W}} (\nabla \cdot \mathbf{u})\ d\mathcal{W}, \tag{3.19}$$

and substituting the control policy (3.7) in (3.19) results in:

$$\begin{aligned}
&\oiint_{\partial\mathcal{W}} (-\nabla\Phi \cdot \mathbf{n})\ d(\partial\mathcal{W}) = \iiint_{\mathcal{W}} -\nabla^2\Phi\, d\mathcal{W} \Rightarrow \\
&\oiint_{\partial\mathcal{W}} (-\nabla\Phi \cdot \mathbf{n})\ d(\partial\mathcal{W}) = 0.
\end{aligned} \tag{3.20}$$

Equation (3.20) represents the compatibility principle and indicates that the flux of the vector field that traverses a loop enclosed by the workspace boundary is zero. This implies that if the vector field, due to the safety constraints, points inwards toward the workspace, a sink term within the workspace is essential to compensate for the fulfillment of (3.20). As presented in section 3.1, a closed contour around the singularity of the goal position is excluded from the workspace, thus (3.20) becomes:

$$\begin{aligned}
&\oiint_{\partial\mathcal{W}'} (-\nabla\Phi \cdot \mathbf{n})\ d(\partial\mathcal{W}') - \oiint_{\mathcal{G}} (-\nabla(\Phi \cdot \mathbf{n})\ d\mathcal{G} = 0 \Rightarrow \\
&\oiint_{\partial\mathcal{W}'} (-\nabla\Phi \cdot \mathbf{n})\ d(\partial\mathcal{W}') - \oiint_{\mathcal{G}} (-\nabla(\Phi_{Repulsive} \cdot \mathbf{n})\ d\mathcal{G} - w_0 = 0
\end{aligned} \tag{3.21}$$

Additionally, based on the safety conditions (3.18) the following inequalities must be satisfied:

$$\oiint_{\partial\mathcal{W}'} (-\nabla\Phi \cdot \mathbf{n})\ d(\partial\mathcal{W}') > 0 \tag{3.22}$$

$$w_0 > 0 \tag{3.23}$$

By substituting the compatibility condition (3.21) in inequalities (3.22), (3.23) we conclude that a solution for the path planning problem exists if:

$$w_0 > \oiint_{\mathcal{G}} (\mathbf{u}_{Repulsive} \cdot \mathbf{n})\ d\mathcal{G} \tag{3.24}$$

This completes the proof and indicates that a solution exists when the attractive term generates a greater inward flux at the goal position than the flux induced by the repulsive term. □

**Lemma 3** (Stability Analysis)**.** *The controller* $\mathbf{u}(\mathbf{p})$ *(3.7) stabilizes the system (2.1) in the desired position* $\mathbf{p}_d$*.*

*Proof.* Consider the exponential transformation $\exp(\cdot) : \mathbb{R} \to \mathbb{R}_+$ of the potential field as a Lyapunov candidate $L(\mathbf{p}) = \exp(\Phi(\mathbf{p}))$ to prove the stability of the system. It is evident that $L(\mathbf{p}) > 0 \; \forall \mathbf{p} \in \mathcal{W} \setminus \{p_d\}$. As the goal position $\mathbf{p}_d$ is approached, the potential field becomes:

$$\lim_{p \to p_d} \Phi(\mathbf{p}) = \lim_{p \to p_d} \left( -\frac{w_0}{4\pi} \frac{1}{\|\mathbf{p} - \mathbf{p_d}\|} \right) = -\infty, \tag{3.25}$$

which results in:

$$\lim_{p \to p_d} \exp(\Phi(\mathbf{p})) = \lim_{\Phi \to -\infty} \exp(\Phi) = 0. \tag{3.26}$$

Thus, the Lyapunov function exhibits a global minimum in the goal position and is positive everywhere else within the workspace. To prove stability through Lyapunov arguments, the time derivative has to be negative $\forall \mathbf{p} \in \mathcal{W} \setminus \{p_d\}$:

$$\begin{aligned} \frac{dL}{dt} = \frac{dL}{d\Phi}\frac{d\Phi}{dp}\frac{dp}{dt} = \exp(\Phi(\mathbf{p}))\nabla\Phi^\top \mathbf{u}(\mathbf{p}) \overset{(3.7)}{=} \\ -\exp(\Phi(\mathbf{p}))\nabla\Phi^\top \nabla\Phi \end{aligned} \tag{3.27}$$

This proves that $\dot{L} < 0 \; \forall \mathbf{p} \in \mathcal{W} \setminus \{p_d\}$ and concludes the proof. □

## 3.6 Proposed Motion Planning Algorithm

Having proposed a control architecture for the motion planning problem that provably avoids obstacle and boundary collisions and is guaranteed to converge to the goal position, the corresponding algorithm is summarized as follows:

**Algorithm 1** Motion Planning Algorithm

```
Given a Workspace $\mathcal{W}$.
Discretize the boundary geometry $\partial\mathcal{W}$ into $K$ triangular panels.
Select $N$ control points $\mathbf{p}_b \in \partial\mathcal{W}$, $b = 1, \ldots, N$ on the boundary surface.
Construct the safety set of linear inequalities:
    $\mathbf{A} = \begin{bmatrix} A_1^1 & \ldots & A_1^{K+1} \\ \vdots & \ddots & \vdots \\ A_N^1 & \cdots & A_N^{K+1} \end{bmatrix},$
    where $A_b^i = \mathbf{n}^\top(\mathbf{p}_b)\mathbf{v}_i(\mathbf{p_b}; \mathbf{M}_i)$, $i = 1, \ldots, K+1$.
Find the policy parameters by solving the constrained quadratic problem:
    $\mathbf{w} = \arg\min_{\mathbf{w}}\{\mathbf{w}^\top\mathbf{w}\}$
    $\text{s.t} \quad \mathbf{A}\mathbf{w} \leq -\epsilon\mathbf{I}_{(N+1)\times 1},$
    where $\epsilon$ is a small positive constant.
The navigation policy that stabilizes the system in the goal position $\mathbf{p}_d \in \mathcal{W}$ is
    established by following the negated gradient of the harmonic potential field:
    $\mathbf{u}(\mathbf{p}) = -\mathbf{w}^\top\mathbf{v}(\mathbf{p}; \mathbf{p_d}; \mathbf{M}).$
```

# Chapter 4

# Optimal Motion Planning Solution

In this chapter an optimal robot navigation scheme for three-dimensional workspaces is presented. The proposed method is based on an off-policy continuous model-based deep reinforcement learning algorithm originated from the policy iteration approach. The algorithm is structured in an actor-critic fashion and the robot learns the optimal control parameters by successively approximating the solution of the HJB equation, effectively merging the advantages of optimal control theory and machine learning techniques.

## 4.1 Optimal Control Policy

To tackle the optimal motion planning problem in continuous time and action space, the infinite horizon integral cost is defined as follows:

$$V(\mathbf{p_0}; \mathbf{p_d}) = \int_0^\infty r\left(\mathbf{p}(\tau); \mathbf{u}(\tau); \mathbf{p_d}\right)\, d\tau, \tag{4.1}$$

where $r\left(\mathbf{p}; \mathbf{u}; \mathbf{p_d}\right) = Q(\mathbf{p}; \mathbf{p_d}) + R(\mathbf{u})$ consists of a state-related term $Q(\mathbf{p}; \mathbf{p_d}) = \alpha\|\mathbf{p}-\mathbf{p_d}\|^2$ and an input-related term $R(\mathbf{u}) = \beta\|\mathbf{u}\|^2$ with $\|\cdot\|$ denoting the Euclidean 2-norm. Here, notice that $R$ represents the energy of the system, while $Q$, analogous to ”Rise Time”, penalizes the robot for deviating from the target position. Next, define the Hamiltonian of the optimal non-linear control problem:

$$H(\mathbf{p}, \mathbf{u}, \nabla_p V) = \nabla_p V^\top \mathbf{u} + Q(\mathbf{p}; \mathbf{p_d}) + R(\mathbf{u}), \tag{4.2}$$

which, as discussed in Chapter 2, constructs the Hamilton-Jacobi-Bellman (HJB) equation for the optimal value function $V^*$:

$$H(\mathbf{p}, \mathbf{u}^*, \nabla_p V^*) = 0. \tag{4.3}$$

Finally, the optimal path planning policy is given by the stationary condition, $\frac{\partial H(\mathbf{p},\mathbf{u},\nabla_p V^*)}{\partial u}|_{\mathbf{u}=\mathbf{u}^*} = 0$, as follows:

$$\mathbf{u}^* = -\frac{1}{2\beta}\nabla_p V^*. \tag{4.4}$$

In order to obtain the optimal control policy $\mathbf{u}^*$ it is clear that an analytical expression for the value function $V^*$ is essential. By substituting the optimal policy (4.4) in the HJB optimality condition (4.3) one can find such an expression for the value function. However, this procedure entails challenges associated with solving a hard non-linear partial differential equation, including high computation times and numerical instabilities, further complicated by the difficulties of incorporating the safety constraints of the control problem.

## 4.2 Model-Based Reinforcement Learning

### 4.2.1 Reinforcement Learning Framework

To address these challenges, a reinforcement learning approach is employed. This procedure involves approximating the value function with a Deep Neural Network (DNN) and iteratively updating the control policy along with the corresponding value function. By leveraging various machine learning techniques, this method aims to derive an optimal path planning policy through successive approximations [24], mitigating the challenges of solving the HJB directly.

The structure of the proposed optimal motion planning method is illustrated in Figure 4.1. In particular, the learning algorithm is initialized with an admissible control possible that can be derived as presented in Chapter 3 and establishes the model-based nature of this approach. Since the system (2.1) is fully observed and the robot is capable to sense the surrounding environment in each state, on-trajectory samples of the value function (4.1) under the control policy (3.7) are collected and stored in a memory buffer. Subsequently a Deep Neural Network is employed to approximate the value function and derive the optimal policy (4.4) for the current iteration step. Finally, the control parameters are updated through a quadratic optimization problem that minimizes the value function, subject to the navigation constraints.

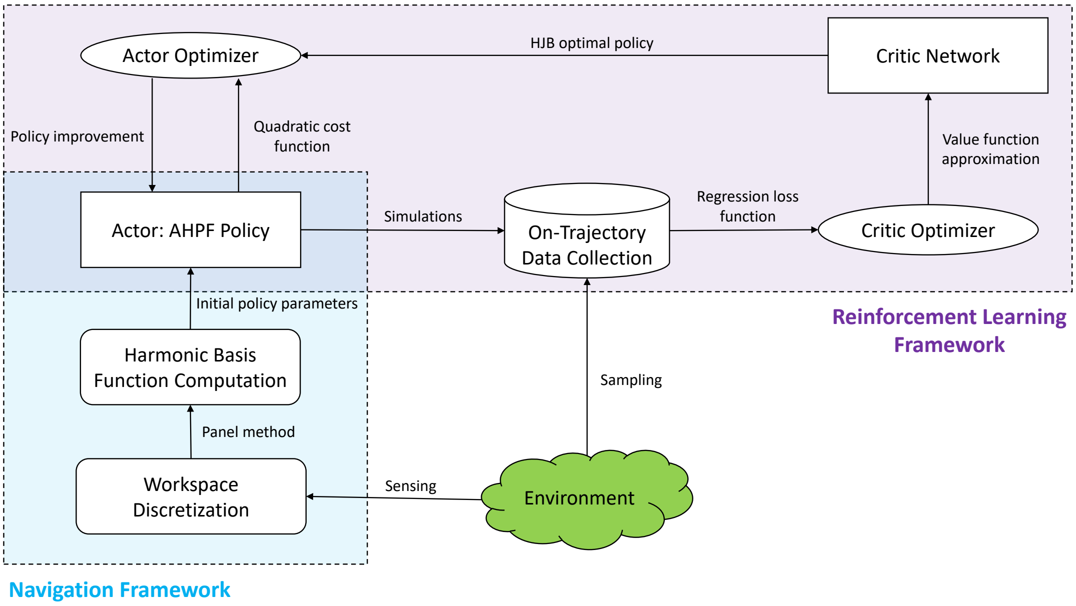


**Figure 4.1:** *Proposed optimal motion planning framework using an actor-critic reinforcement learning architecture.*

In the upcoming sections, the essential components that constitute the reinforcement learning algorithm will be presented, namely the value function approximation step (critic optimizer) and the policy weight update step (actor optimizer).

### 4.2.2 Value Function Approximation

As stated in section 4.1 an analytical expression for the value function is essential to derive the optimal path planning policy. To this end, a multi-layer fully connected Deep Neural Network (DNN) is utilized as a universal approximator of the value function:

$$\hat{V} = h^{(L)} = \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}, \tag{4.5}$$

where the DNN is represented by the following structure:

- $\mathbf{h}^{(r)} = f^{(r)}\left(\mathbf{W}^{(r)}\mathbf{h}^{(r-1)} + \mathbf{b}^{(r)}\right) \in \mathbb{R}^{n_r \times 1}$ denotes the output vector at each layer $r = 1, \ldots, L$, which consists of $n_r$ neurons.

- $f^{(r)}$ is the activation function in each layer.

- $\mathbf{W}^{(r)} = \begin{bmatrix} W_1^1 & \cdots & W_1^{n_{r-1}} \\ \vdots & \ddots & \vdots \\ W_{n_r}^1 & \cdots & W_{n_r}^{n_{r-1}} \end{bmatrix} \in \mathbb{R}^{n_r \times n_{r-1}}$ represents the connections $W_m^k$, of the neurons $m = 1, \ldots, n_r$ and $k = 1, \ldots, n_{r-1}$ among two consecutive layers.

The proposed reinforcement learning algorithm is classified as model-based since an initial admissible policy for the motion planning problem can be obtained according to Algorithm 2. By conducting multiple simulations, the value function associated with an initial safe and stable control policy can be evaluated within the workspace $\mathcal{W}$, thereby constructing the training dataset of the network. More precisely, an uniform distribution of initial points $\mathbf{p_h} \in \mathcal{W}$, $h = 1, \ldots, H$ inside the workspace is chosen and the accumulated cost $\overline{V}(\mathbf{p}; \mathbf{p_d})$ from an initial position $\mathbf{p_h}$ to a position $\mathbf{p}$ along the robot's trajectory is computed by solving the following system of differential equations:

$$\begin{aligned} \frac{d\mathbf{p}(t)}{dt} &= \mathbf{u}(\mathbf{p}(t)), & \mathbf{p}(0) &= \mathbf{p_h} \\ \frac{d\overline{V}(\mathbf{p}; \mathbf{p_d})}{dt} &= r(\mathbf{p}(t); \mathbf{u}(t); \mathbf{p_d}), & V(\mathbf{p_h}; \mathbf{p_d}) &= 0 \end{aligned} \tag{4.6}$$

According to Bellman's principle of optimality the cost-to-go for any on-trajectory point $\mathbf{p}$ can be calculated as follows:

$$V(\mathbf{p}; \mathbf{p_d}) = \overline{V}(\mathbf{p_d}; \mathbf{p_d}) - \overline{V}(\mathbf{p}; \mathbf{p_d}), \tag{4.7}$$

which constructs the training dataset of the critic DNN.

In order to get an expression for the optimal solution (4.4) of the HJB equation, the grad of the value function with respect to the robot's position is calculated analytically in a back-propagating fashion. If the partial derivative $\frac{\partial \hat{V}}{\partial \mathbf{h}^{(r)}}$ of the value function in layer $r$ is known, then the partial derivative $\frac{\partial \hat{V}}{\partial \mathbf{h}^{(r-1)}}$ of the value function w.r.t. the preceding layer is expressed as:

$$\begin{aligned} \frac{\partial \hat{V}}{\partial \mathbf{h}^{(r-1)}} &= \frac{\partial \hat{V}}{\partial \mathbf{h}^{(r)}} \cdot \frac{\partial \mathbf{h}^{(r)}}{\partial \mathbf{h}^{(r-1)}} = \\ &= \mathbf{W}^{\top(r)} \cdot \left( \frac{\partial \hat{V}}{\partial \mathbf{h}^{(r)}} \odot f'^{(r)} \left( \mathbf{W}^{(r)} \mathbf{h}^{(r-1)} + \mathbf{b}^{(r)} \right) \right). \end{aligned} \tag{4.8}$$

### 4.2.3 Policy Improvement

At this point, a closed-form expression for the optimal vector field (4.4) w.r.t. the cost function (4.1) has been obtained. However, directly following the optimal policy of the HJB equation would result in a navigation policy that violates both the harmonic properties and the safety constraints. To this end, an improved policy is derived by minimizing the following expression:

$$\begin{aligned} \mathbf{w}^{(i+1)} &= \arg\min_{\mathbf{w}}\{\frac{1}{2}\|\mathbf{u}^{(i)} - \mathbf{u}^{*(i)}\|^2\} \\ &\text{s.t} \quad \mathbf{A}\mathbf{w} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}, \end{aligned} \tag{4.9}$$

Note that by solving this optimization problem the improved policy strives to reach as close to the optimal policy as possible while at the same time maintaining the harmonic basis function of the motion planning policy. A free of local minima vector field is therefore guaranteed, while incorporating the safety conditions ensures that the improved policy continues to provide collision-free and convergent paths towards the target position. The objective function of (4.9) can be expanded to a quadratic form as follows:

$$\begin{aligned} \frac{1}{2}\|\mathbf{u}^{(i)} - \mathbf{u}^{*(i)}\|^2 = \frac{1}{2}\mathbf{w}^\top(-\mathbf{v}(\mathbf{p}))(-\mathbf{v}(\mathbf{p})^\top)\mathbf{w} - \left(\frac{1}{2\beta}\nabla_p \hat{V}(\mathbf{p})^{(i)}\right)^\top \mathbf{v}(\mathbf{p})\mathbf{w} \\ + \left(-\frac{1}{2\beta}\nabla_p \hat{V}(\mathbf{p})^{(i)}\right)^\top \left(-\frac{1}{2\beta}\nabla_p \hat{V}(\mathbf{p})^{(i)}\right), \end{aligned} \tag{4.10}$$

In this expression $\mathbf{v}(\mathbf{p}) \in \mathbb{R}^{K+1}$ is the basis function for the robot's velocity command in position $\mathbf{p} \in \mathbb{R}^3$, $\mathbf{w} \in \mathbb{R}^{K+1}$ are the control parameters of the AHPF formulation (3.7), and $\hat{V}(\mathbf{p})^{(i)}$ is the DNN approximation of the value function under the control policy $\mathbf{u}^{(i)}$. The policy in each iteration of the reinforcing learning algorithm is updated by solving the following constrained quadratic minimization problem over a number of sample $\mathbf{p_m} \in \mathbb{R}^3, m = 1, \ldots, L$ within the workspace:

$$\begin{aligned} \mathbf{w}^{(i+1)} &= \arg\min_{\mathbf{w}}\{\frac{1}{2}\mathbf{w}^\top \overline{\mathbf{H}}(\mathbf{p})\mathbf{w} + \overline{\mathbf{B}}(\mathbf{p})^{(i)}\mathbf{w}\} \\ &\text{s.t} \quad \mathbf{A}\mathbf{w} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}, \end{aligned} \tag{4.11}$$

where:

$$\begin{aligned} \overline{\mathbf{H}}(\mathbf{p}) &= \sum_{m=1}^{L} \left(\mathbf{H}(\mathbf{p_m})^\top \mathbf{H}(\mathbf{p_m})\right) \in \mathbb{R}^{(K+1)\times(K+1)} \\ \overline{\mathbf{B}}(\mathbf{p})^{(i)} &= \sum_{m=1}^{L} \left(-\mathbf{H}(\mathbf{p_m})^\top \mathbf{B}(\mathbf{p_m})^{(i)}\right)^\top \in \mathbb{R}^{1\times(K+1)} \\ \mathbf{H}(\mathbf{p}) &= -\mathbf{v}(\mathbf{p}) \\ \mathbf{B}(\mathbf{p})^{(i)} &= -\frac{1}{2\beta}\nabla_p \hat{V}(\mathbf{p})^{(i)} \end{aligned}$$

## 4.3 Proposed Optimal Motion Planning Algorithm

Having thoroughly examined each individual component of the reinforcement learning framework, the optimal motion planning algorithm for 3D workspaces is summarized in this section.

**Algorithm 2** Optimal Motion Planning Algorithm

```
Given a Workspace $\mathcal{W}$.
Discretize the boundary geometry $\partial\mathcal{W}$ into $K$ triangular panels.
Select $N$ control points $\mathbf{p}_b \in \partial\mathcal{W}$, $b = 1, \dots, N$ on the boundary surface.
Construct the safety set of linear inequalities: $\mathbf{A} = \begin{bmatrix} A_1^1 & \dots & A_1^{K+1} \\ \vdots & \ddots & \vdots \\ A_N^1 & \dots & A_N^{K+1} \end{bmatrix}$, where $A_b^i = \mathbf{n}^\top(\mathbf{p}_b)\mathbf{v}_i(\mathbf{p_b}; \mathbf{M}_i)$, $i = 1, \dots, K+1$.
Find an initial admissible policy $\mathbf{u}^{(i)}(\mathbf{p}) = -\mathbf{w}^{(i)\top}\mathbf{v}(\mathbf{p}; \mathbf{p_d}; \mathbf{M})$ by solving the following quadratic problem:

    $\mathbf{w}^{(i)} = \arg\min_{\mathbf{w}} \{\mathbf{w}^\top \mathbf{w}\}$
    $\text{s.t} \quad \mathbf{A}\mathbf{w} \leq -\epsilon \mathbf{I}_{(N+1)\times 1},$

where $\epsilon$ is a small positive constant.
while the control parameters $\mathbf{w}$ have not converged do
    • Model-based sampling: Sample a set of initial points $p_h \in \mathcal{W}$, $h = 1, \dots, H$ and compute simulations to collect $H'$ on-trajectory samples of the cost function (4.1) by solving the system of ODEs (4.6).
    • Critic Approximation: Approximate the critic network (4.5) by obtaining the DNN parameters $\phi$ that minimize the loss function:

        $\phi = \arg\min_{\phi} \frac{1}{H'} \sum_{h=1}^{H'} \Big( V(\mathbf{p_h}) - \hat{V}(\mathbf{p_h}; \phi) \Big)^2 \tag{4.12}$

    • Actor optimizer: Update the control policy parameters by solving the following constrained quadratic optimization problem over a sample of $L$ points within the workspace:

        $\mathbf{w}^{(i+1)} = \arg\min_{\mathbf{w}} \{ \frac{1}{2} \mathbf{w}^\top \overline{\mathbf{H}}(\mathbf{p}) \mathbf{w} + \overline{\mathbf{B}}(\mathbf{p})^{(i)} \mathbf{w} \}$
        $\text{s.t} \quad \mathbf{A}\mathbf{w} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}, \tag{4.13}$

    ,
    where:

        $\overline{\mathbf{H}}(\mathbf{p}) = \sum_{m=1}^{L} \Big( \mathbf{H}(\mathbf{p_m})^\top \mathbf{H}(\mathbf{p_m}) \Big) \in \mathbb{R}^{(K+1)\times(K+1)}$
        $\overline{\mathbf{B}}(\mathbf{p})^{(i)} = \sum_{m=1}^{L} \Big( -\mathbf{H}(\mathbf{p_m})^\top \mathbf{B}(\mathbf{p_m})^{(i)} \Big)^\top \in \mathbb{R}^{1\times(K+1)}$
        $\mathbf{H}(\mathbf{p}) = -\mathbf{v}(\mathbf{p})$
        $\mathbf{B}(\mathbf{p})^{(i)} = -\frac{1}{2\beta} \nabla_p \hat{V}(\mathbf{p})^{(i)}$

end while
```

## 4.4 Proofs of Optimality

This section presents the essential technical validation of the optimal control methodology. The main goal is to prove that by starting from an initial admissible policy, every subsequent control policy derived by the proposed reinforcement learning algorithm will be admissible and converge to the optimal solution of the HJB equation.

**Definition 1** (Admissible Policy)**.** *A control policy u(p) is defined as admissible with respect to the cost function (4.1) on $\mathcal{W}$ if $u(p)$ is continuous on $\mathcal{W}$, $u(p_d) = 0$, $u(p)$ stabilizes (2.1) on $\mathcal{W}$, $V(p)$ is finite $\forall p \in \mathcal{W}$ and the trajectories under the control $u(p)$ are safe.*

**Lemma 4** (Policy Update Admissibility)**.** *Consider the admissible control policy $u^{(i)}$ (3.7) at the i-th iteration step along with the corresponding value function $V^{(i)}$ (4.1). The improved policy $u^{(i+1)}$ that is obtained following the proposed methodology will be admissible.*

*Proof.* The updated policy $u^{(i+1)}$ is proven to be admissible on $\mathcal{W}$ by satisfying the criteria of Definition 1 as follows:

- Since the path planning policy strictly preserves a set of harmonic basis functions in every iteration step, $u^{(i+1)}$ is continuous.

- The requirement $u^{(i+1)}(p_d) = 0$ is trivial, since the robot finishes the navigation task upon reaching the goal position $p_d$.

- To prove that $u^{(i+1)}$ stabilizes (2.1) on the goal position $p_d$ we rely on **Lemma** 3, which provides a comprehensive stability analysis for the proposed control policy structure (3.7). It is important to note that in each step of the reinforcement learning process only the controller parameters are updated, while the harmonic basis function remains intact. This ensures the fulfilment of the stability requirements for every policy throughout the optimization process.

- The value function, which is approximated through a critic fully connected Deep Neural Network, is finite over the entire workspace.

- It is evident that both in every policy improvement step (4.11) and in the derivation of the initial policy (3.18) the set of linear inequalities $\mathbf{A}\mathbf{w} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}$ is integrated as hard constraints in the problem, thus rendering the navigation safe with respect to boundary collisions.

□

# Chapter 5

# Simulation Results

In this chapter the simulation results of the proposed optimal motion planning algorithm are presented in two different environments. To showcase the advancement of the proposed algorithm, a comparative study is conducted with RRT$^*$, one of the most widely used algorithms in the field of optimal motion planning. Additional results regarding the critic network approximation and the convergence of the reinforcement learning algorithm are provided to bolster the effectiveness and validity of the proposed methodology. All simulations were implemented in MATLAB, supplemented with C++ Mex files to solve the fluid flow equations, on a Windows 10 PC with an eight-core Ryzen 7 CPU and 32GB of RAM.

## 5.1 Urban Environment

The robot navigation in an urban environment is examined, inspired by the application of delivery drones in densely populated areas, where minimizing both the delivery distance and the control effort (i.e., battery consumption) is essential. In Figure 5.1 and Figure 5.2 the reference map of a city district is depicted along with the corresponding simulation environment. In this application the drone covers an area of $400 \cdot 10^6 m^3$, which is bounded in all directions. In particular, the lower boundary along the z-axis prevents collisions with the ground and preserves a safe distance from pedestrians. The remaining axis are bounded to maintain the drone's connectivity. Regarding the parameters of the proposed motion planning algorithm, the workspace is discretized into 5,000 triangular panels, while the safety constraints (3.14) are satisfied in 10,000 boundary points.

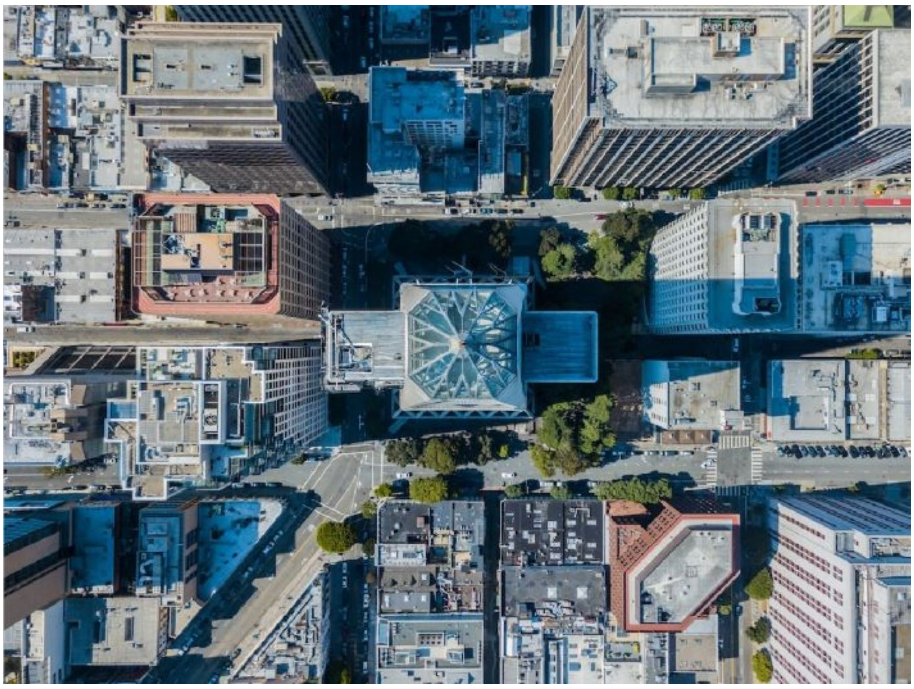


**Figure 5.1:** *The reference map of the urban environment.*

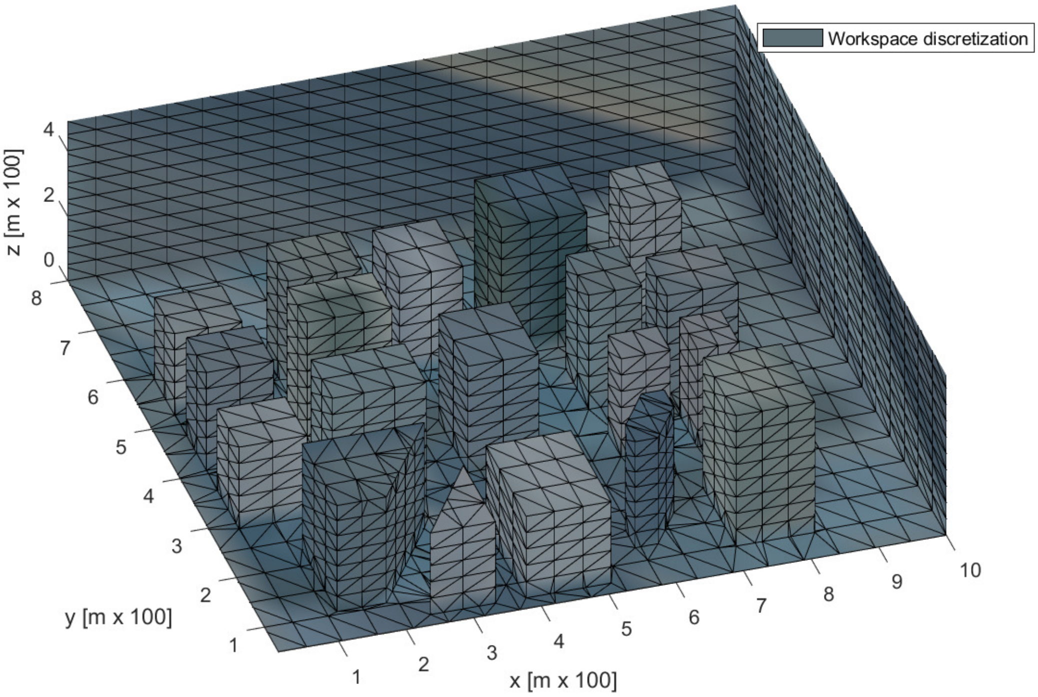


**Figure 5.2:** *The discretized simulation environment.*

### 5.1.1 Optimal Trajectories - Comparison with RRT*

The algorithm presented in this thesis is compared to RRT*, a widely used path planning algorithm, proven to converge to the optimal solution. Given that the vanilla RRT* algorithm produces minimum trajectory lengths, two different implementations for the velocity term of the quadratic cost function (4.1) are explored. The baseline RRT* approach [33] samples both the position and control space, making it closely related to the method used in this thesis and allowing a direct comparison between the two approaches. Additionally, an enhanced implementation of the RRT* method is considered, where the optimal distance is obtained through the traditional RRT* algorithm [8] and is afterwards combined with the closed-form solution of the optimal on-trajectory velocity $u^* = \frac{\alpha}{\beta}\|p - p_d\|$. This approach should place our method at a disadvantage, as the cost is minimized on the resulting trajectories of the RRT* using the provably optimal velocity norm. In total, the RRT* method was executed 25 times for each initial position, as its probabilistic nature requires a statistical sample.

| **Start Position** | **Cost Proposed Method** | | **Cost RRT*** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Initial | Optimal | Mean | | Median | | Min | | Max | |
| | | | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline |
| [0.5;3.9;1.0] | 137 | 72 | 86 | 164 | 86 | 150 | **69** | 119 | 108 | 292 |
| [1.4;7.5;0.8] | 138 | **90** | 120 | 275 | 124 | 178 | 91 | 150 | 141 | 650 |

**Table 5.1:** *Cost comparison among the proposed method, the baseline RRT* method [33] and the enhanced RRT* method with respect to the quadratic function* (4.1) *in the urban environment.*

| **Start Position** | **Length Proposed Method** | | **Length RRT*** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Initial | Optimal | Mean | | Median | | Min | | Max | |
| | | | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline |
| [0.5;3.9;1.0] | 10.2 | **5.6** | 8.7 | 9 | 8.7 | 8.9 | 7.3 | 7 | 10.5 | 11.2 |
| [1.4;7.5;0.8] | 8.5 | **4.9** | 10.3 | 10.4 | 10.5 | 9.3 | 7.6 | 7.6 | 12.4 | 15.7 |

**Table 5.2:** *Trajectory lengths comparison (in* $[m \times 100]$*) among the proposed method, the baseline RRT* method [33] and the enhanced RRT* method in the urban environment.*

The drone trajectories from two initial positions $p_{0_1} = [0.5, 3.9, 1]^\top$, $p_{0_2} = [1.4, 7.5, 0.8]^\top$ towards the delivery destination $p_d = [5.6, 3.4, 2]^\top$ are illustrated in Figure 5.3, Figure 5.4, Figure 5.5, and Figure 5.6 from various viewpoints. The paths of the initial and the optimal policies are represented by solid blue and red lines respectively, while the optimal path derived by the RRT* method is depicted with a solid yellow line. The corresponding cost and path length of each trajectory are summarized in Table 5.1 and Table 5.2.

It is evident that the proposed motion planning algorithm outperforms the RRT* method both in terms of trajectory lengths and in terms of cost with respect to a quadratic form infinite horizon value function. The optimal policy of the proposed algorithm manages to consistently achieve lower travel distances, while only 1 out of 50 trajectories, that RRT* generates, reaches the goal position with a lower cost. Additionally, the method presented in this thesis generates smooth and natural-looking trajectories, compared to RRT*. Another advantage of the proposed algorithm is the fact that it converges to the optimal solution in a deterministic manner, enhancing the robustness of the navigation problem. Regarding the computational time, a direct comparison is not possible since the presented potential field approach generates an optimal path from every initial point within the workspace. In contrast, each run of the RRT* method depends on a specific initial position, further compounded by the need for multiple simulations due to its stochastic nature.

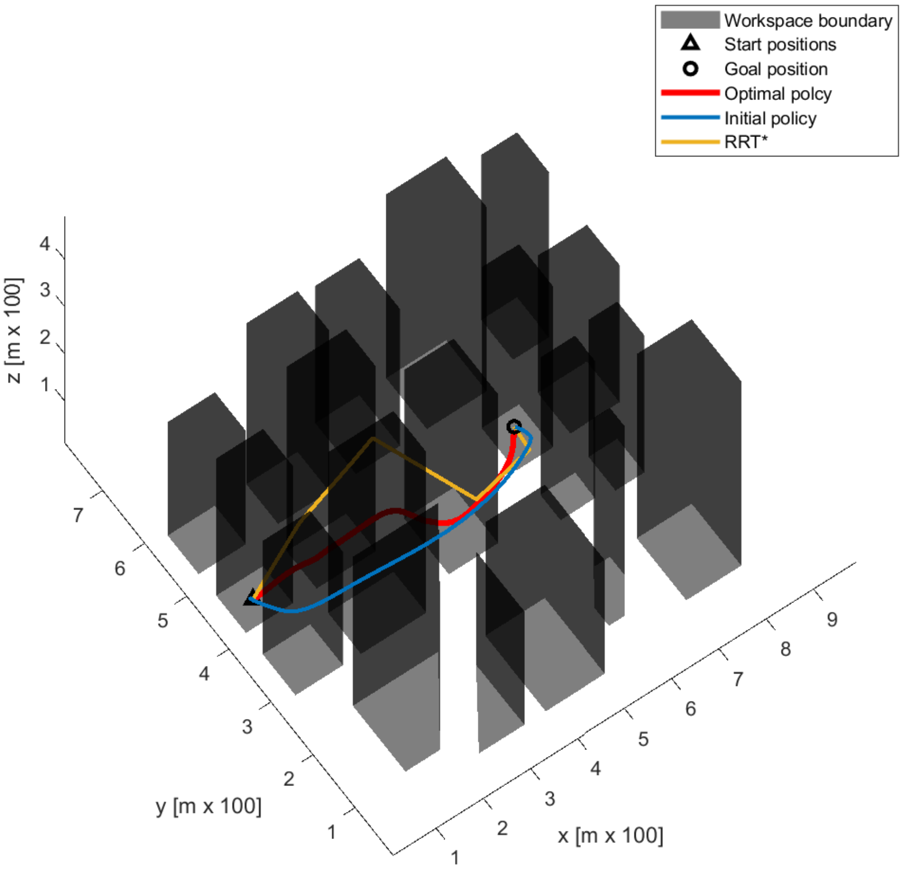


**Figure 5.3:** *Comparative simulation results of the urban environment from the starting position* $p_0 = [0.5, 3.9, 9.1]^\top$ *in isometric view. The figure depicts the initial trajectory (blue line), the optimal trajectory of the proposed method (red line) and the optimal trajectory of the RRT* method (yellow line).*

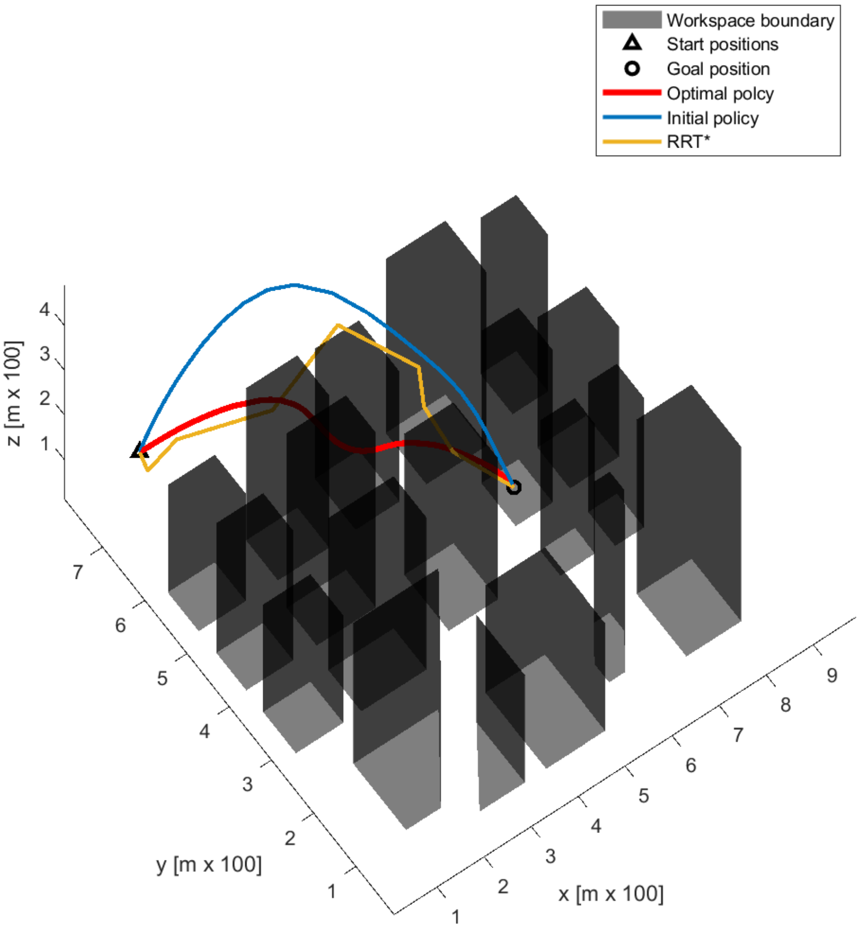


**Figure 5.4:** *Comparative simulation results of the urban environment from the starting position $p_0 = [1.4, 7.5, 0.8]^\top$ in isometric view. The figure depicts the initial trajectory (blue line), the optimal trajectory of the proposed method (red line) and the optimal trajectory of the RRT* method (yellow line).*

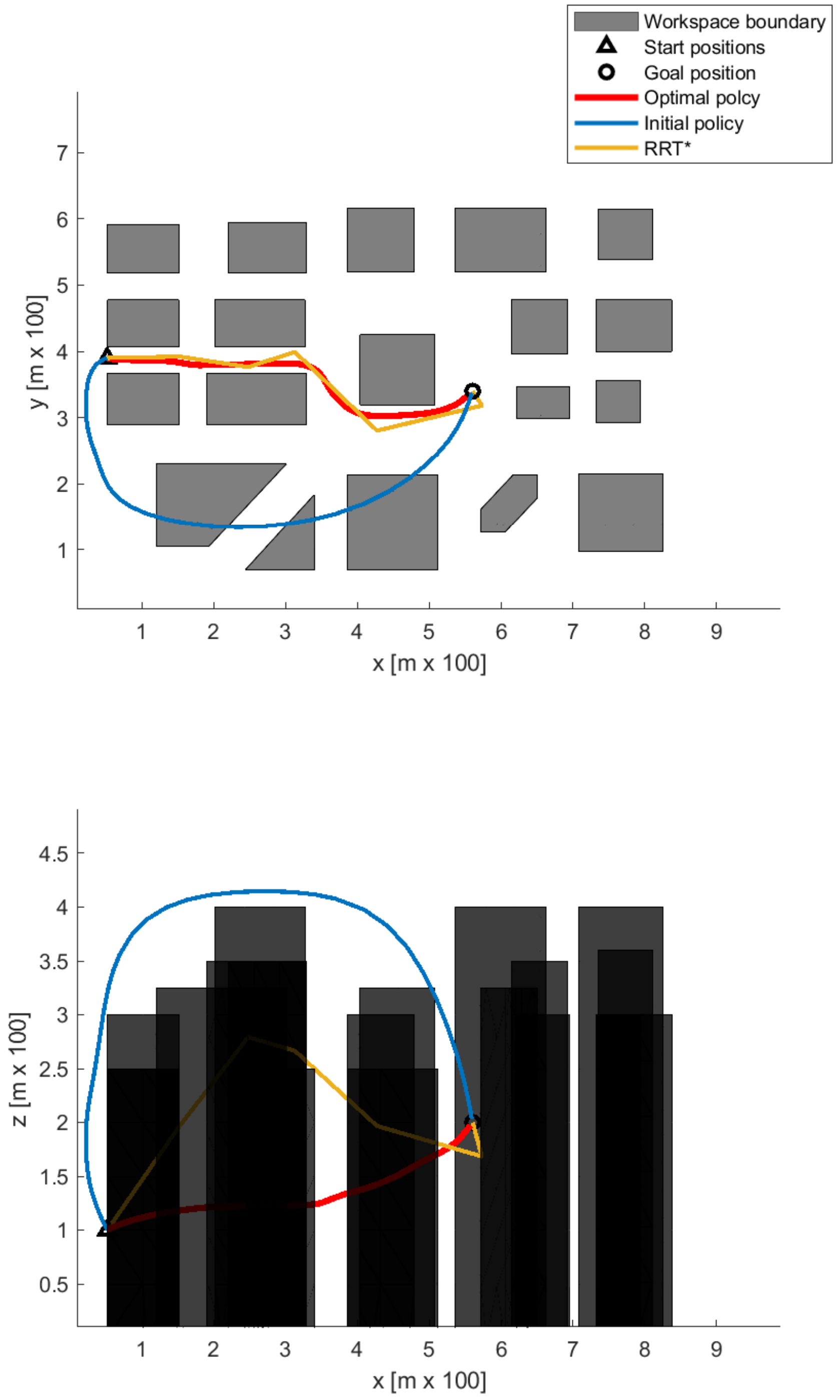


**Figure 5.5:** *Comparative simulation results of the urban environment from the starting position* $p_0 = [0.5, 3.9, 9.1]^\top$ *in front and top views. The figure depicts the initial trajectory (blue line), the optimal trajectory of the proposed method (red line) and the optimal trajectory of the RRT* method (yellow line).*

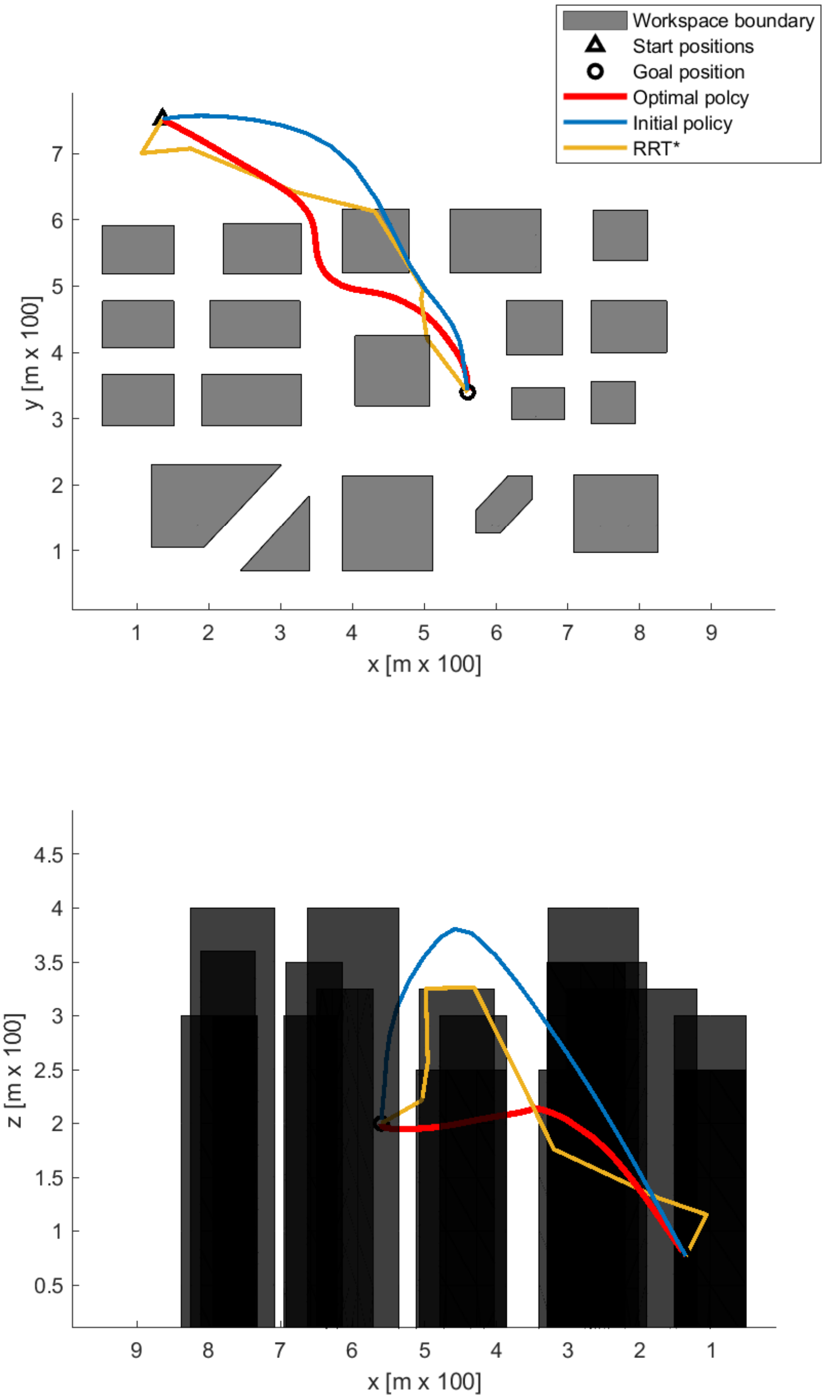


**Figure 5.6:** *Comparative simulation results of the urban environment from the starting position* $p_0 = [1.4, 7.5, 0.8]^\top$ *in front and top views. The figure depicts the initial trajectory (blue line), the optimal trajectory of the proposed method (red line) and the optimal trajectory of the RRT* method (yellow line).*

### 5.1.2 Critic Network

At this point, the approximation capability of the critic Deep Neural Network is showcased. By looking at Figure 5.7, where the regression of the DNN is demonstrated, it becomes clear that the critic network is highly capable to provide an accurate approximation of the cost function. More specifically, the trained critic network at the final iteration of the reinforcement learning algorithm has an absolute mean squared error $mse = 0.1$ over the entire data set of 20,000 trajectory points.

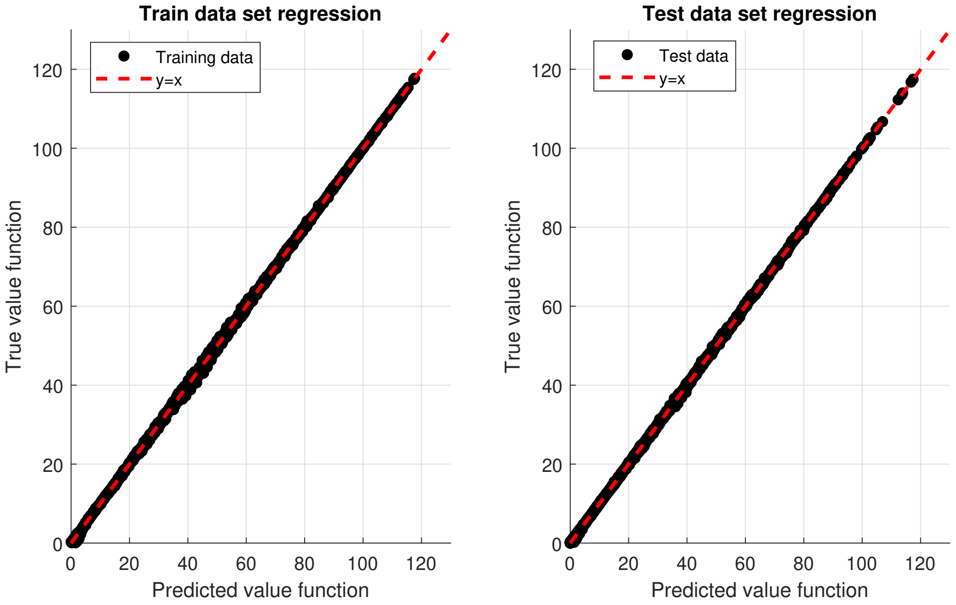


**Figure 5.7:** *Regression plots of the final critic Deep Neural Network (DNN) on the training and testing data sets in the urban environment.*

The value function approximation in two different height coordinates along with the corresponding normalized vector fields are visualized in Figure 5.8. Clearly the value function is continuous over the entire workspace as discussed in **Lemma 4** and has the desired form with a global minimum in the goal position. However, a closer look at the grad of the cost function reveals that the both the safety criteria and the absence of local minima are not guaranteed. This justifies the fact that the policy update step in each iteration of the reinforcement learning algorithm is driven by a constrained quadratic optimization problem (4.11) instead of directly following the HJB optimal policy (4.4).

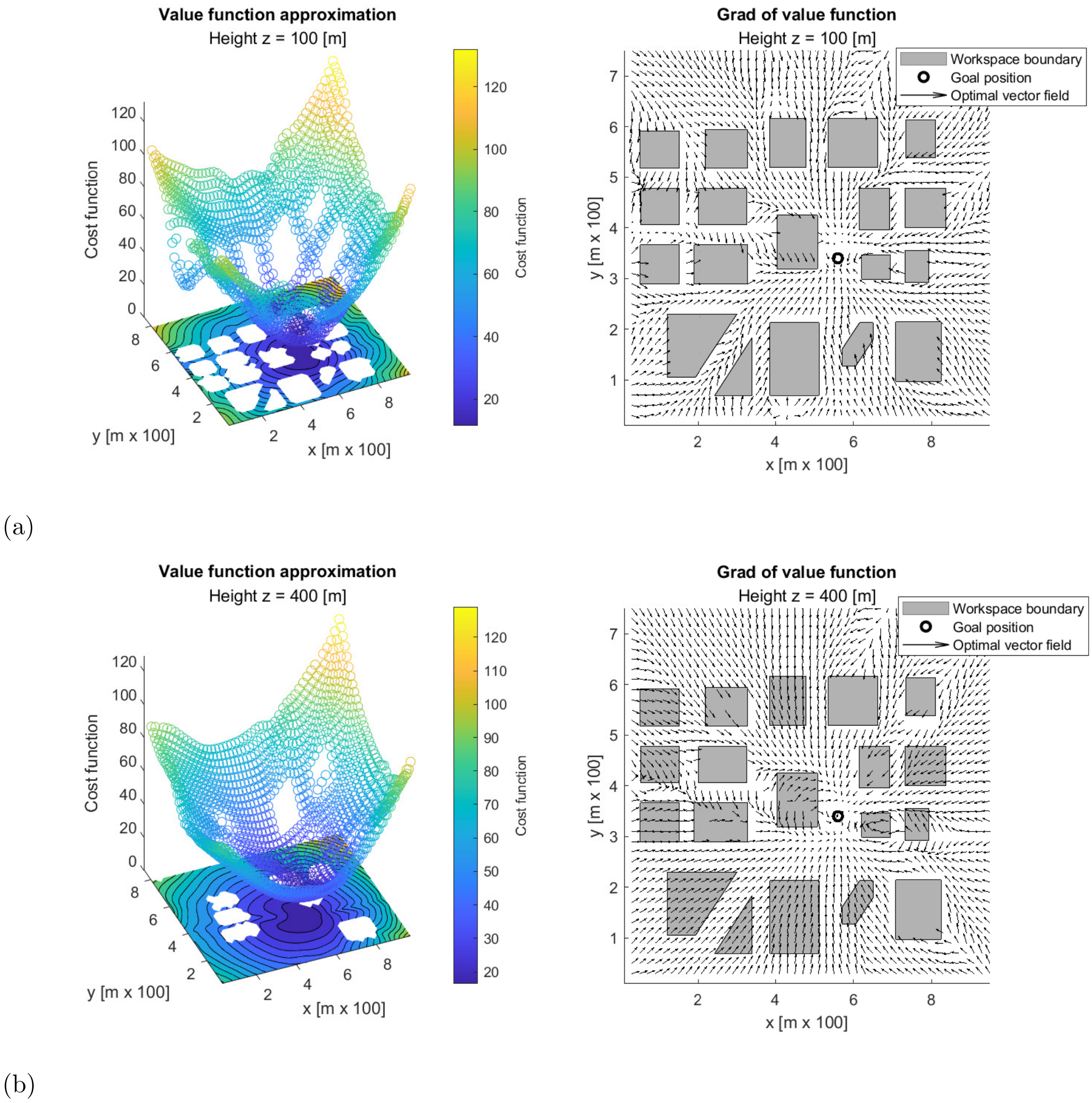


**Figure 5.8:** *The final cost function approximation in the urban environment and the corresponding optimal vector field at two different heights (a)* $z = 100[m]$ *and (b)* $z = 400[m]$.

### 5.1.3 Actor: AHPF Policy and Convergence Criteria

In order to validate the performance and convergence of the proposed reinforcement learning algorithm consider the following metrics at each consecutive iterations $(i), (i+1)$:

$$MAE_{weights} = \sum_{k=1}^{K} \frac{1}{K} |w_k^{(i)} - w_k^{(i+1)}|$$
$$MAE_{cost} = \sum_{l=1}^{L} \frac{1}{L} V(\mathbf{p}_l)^{(i)} - V(\mathbf{p}_l)^{(i+1)},$$

where $\mathbf{w}^{(i)} \in \mathbb{R}^{K+1}$ denotes the control parameters of the motion planning policy (i.e., panel weights) and $V(\mathbf{p}_l)$ denotes the value function (4.1) evaluated over several initial position samples within the workspace $p_l \in \mathbb{R}^3$. The evolution of the weights mean difference and the cost function mean difference at each iteration step is illustrated in Figure 5.9. It is evident that the reinforcement learning algorithm smoothly converges to a final set of control parameters. At the same time the algorithm minimizes the value function across various initial positions, highlighting the advantageous properties of global optimality in the proposed optimization approach.

Next, the initial and the optimal normalized vector fields are illustrated in Figure 5.10, with a color-map representing the z-coordinate of each vector's endpoint. Once again the improvement of the initial policy is clear, as after the convergence of the proposed algorithm all velocity vectors point towards the minimum-distance path. Also notice that compared to the policy, derived using the HJB optimality condition, the proposed method provides collision-free trajectories and preserves the absence of local minima inside the workspace. Finally, Figure 5.11 represents the normalized vectors fields in the entire three-dimensional workspace with each color-map denoting the magnitude of the velocity commands.

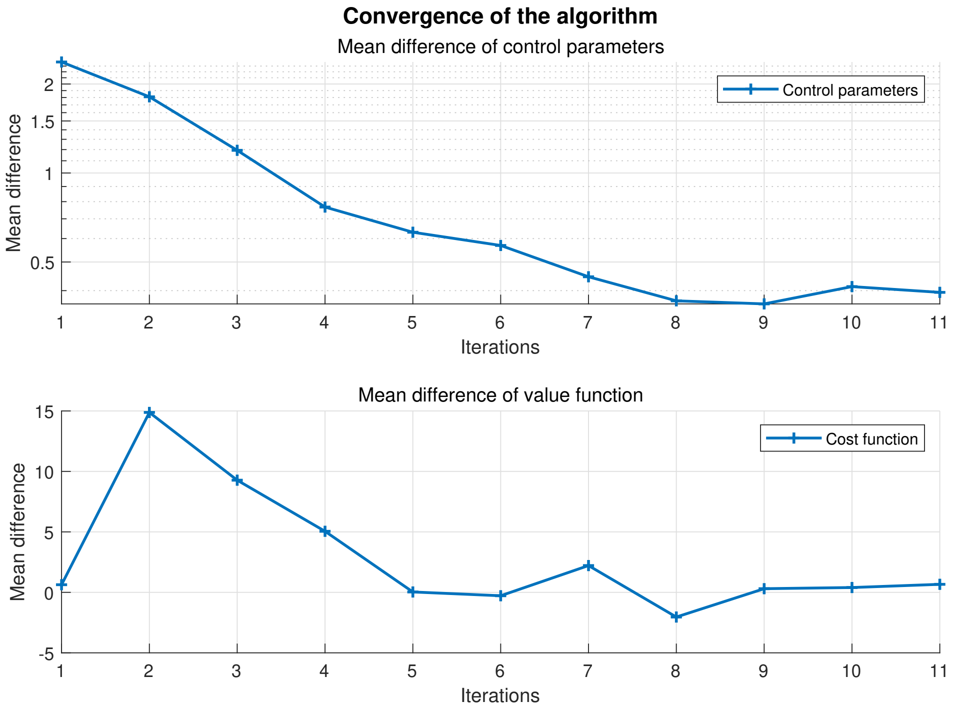
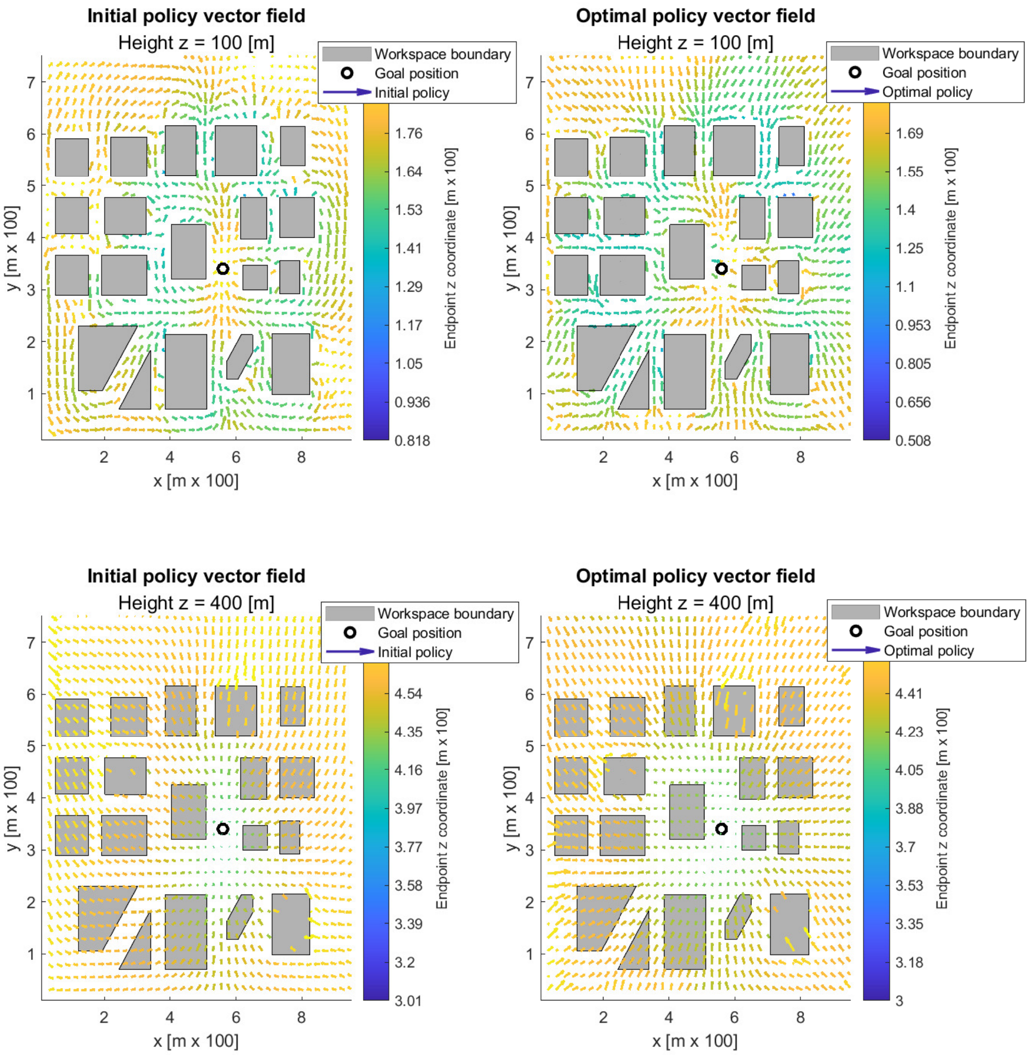


**Figure 5.9:** *Mean absolute difference of the control parameters and mean difference of the cost function at each iterations step of the proposed reinforcement learning algorithm.*



(a)

(b)

**Figure 5.10:** *The normalized vector fields illustrating the initial and the optimal control policy of the urban environment at two different heights (a) $z = 100[m]$ and (b) $z = 400[m]$. The color-map represents the height of each vector's endpoint.*

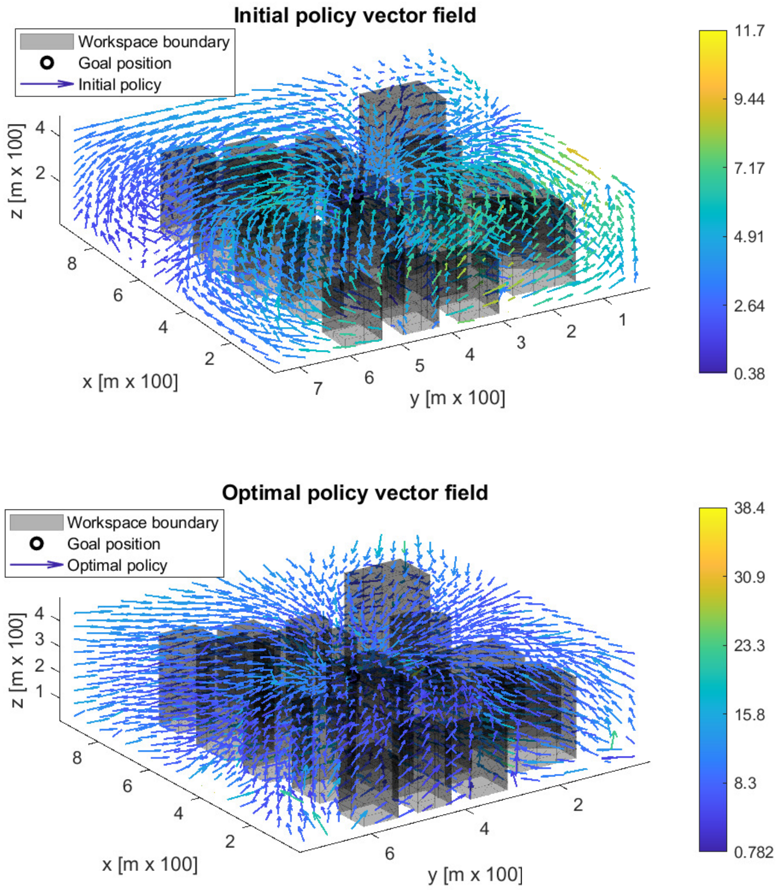


**Figure 5.11:** *The vector fields illustrating the initial and the optimal control policy in isometric view of the urban environment. The color-map depicts the magnitude of the vector field.*

## 5.2 Forest Environment

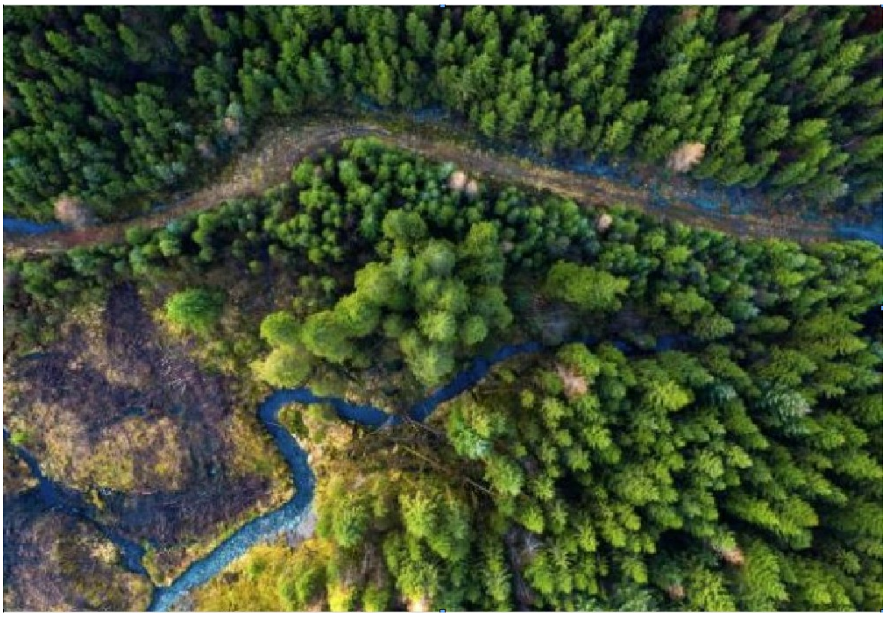


**Figure 5.12:** *The reference map of the forest environment.*

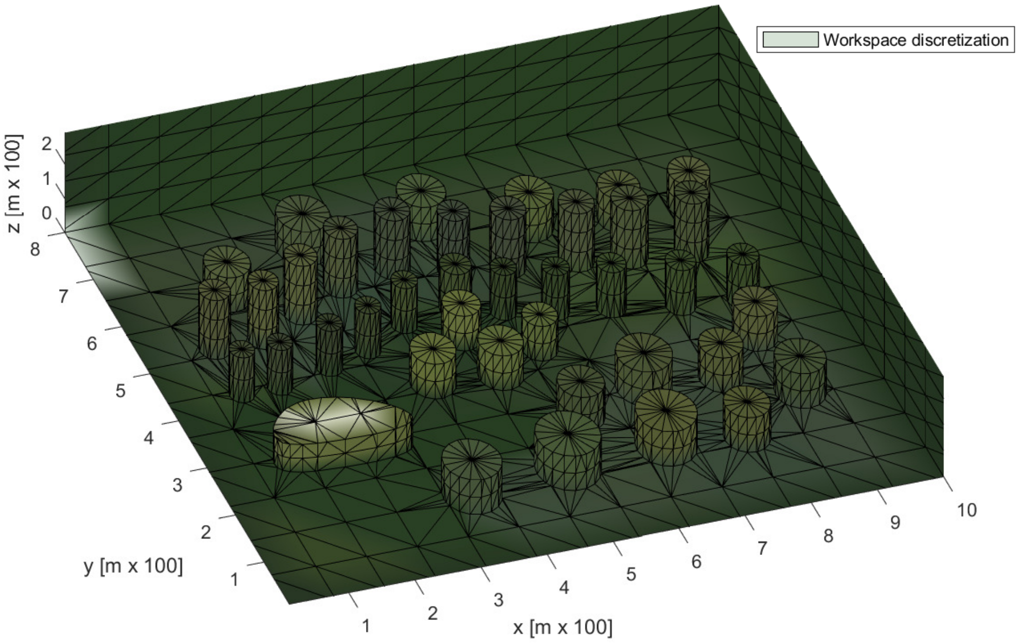


**Figure 5.13:** *The corresponding discretized simulation environment.*

The proposed optimal motion planning framework is applied in a forest workspace, which is characterized by a plethora of obstacles and very narrow passages. The reference map of a forest environment along with the corresponding simplified simulation environment are depicted in Figure 5.12 and Figure 5.13 respectively. It is important to note that the entire workspace is bounded. More specifically at the lower end of the z-axis the boundary prevents collisions with the ground while in the rest of the axis limits the boundary is utilized to ensure connectivity maintenance. Furthermore the source terms of the control policy are distributed into 5,000 triangular panels, while a point source indicating the goal position is placed in $p_d = [9.1, 1.4, 0.2]^\top$. In order to ensure the avoidance of collision with the boundary, the safety condition (3.14) is met in 10,000 sampling points on the boundary.

The trajectories from the starting position $p_0 = [4.1, 6.8, 0.5]^\top$ towards the goal position $p_d = [9.1, 1.4, 0.2]^\top$ are illustrated in Figure 5.13 and Figure 5.14. Table 5.3 and Table 5.4 summarize the cost and path length compared to RRT$^*$. The results demonstrate a superior performance of the proposed algorithm compared to two different implementations of RRT$^*$ in an obstacle-cluttered environment. Additionally, Figure 5.16 showcases the approximation of the critic network, while in Figure 5.17 and Figure 5.18 the initial control policy is compared with the optimal solution in order to illustrate the superiority of the latter.

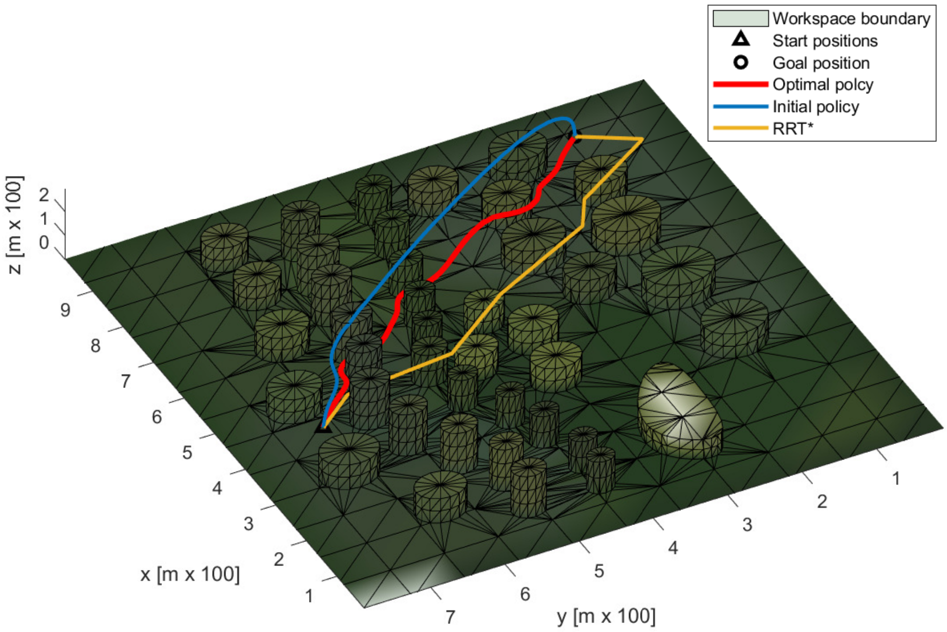


**Figure 5.14:** *Comparative simulation results of the forest environment from the starting position* $p_0 = [4.1, 6.8, 0.5]^\top$ *in isometric view. The figure depicts the initial trajectory (blue line), the optimal trajectory of the proposed method (red line) and the optimal trajectory of the RRT* method (yellow line).*

| **Start Position** | **Cost Proposed Method** | | **Cost RRT*** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Initial | Optimal | Mean | | Median | | Min | | Max | |
| | | | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline |
| [4.1;6.8;0.5] | 375 | **193** | 219 | 860 | 216 | 875 | **193** | 505 | 262 | 1320 |

**Table 5.3:** *Cost comparison among the proposed method, the baseline RRT* method [33] and the enhanced RRT* method with respect to the quadratic function* (4.1) *in the forest environment.*

| **Start Position** | **Length Proposed Method** | | **Length RRT*** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Initial | Optimal | Mean | | Median | | Min | | Max | |
| | | | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline | Enhanced | Baseline |
| [4.1;6.8;0.5] | 9.1 | **7.74** | 9.9 | 10.2 | 9.8 | 10.3 | 9 | 8.9 | 11.3 | 12.1 |

**Table 5.4:** *Trajectory lengths comparison (in* $[m \times 100]$*) among the proposed method, the baseline RRT* method [33] and the enhanced RRT* method in the forest environment.*

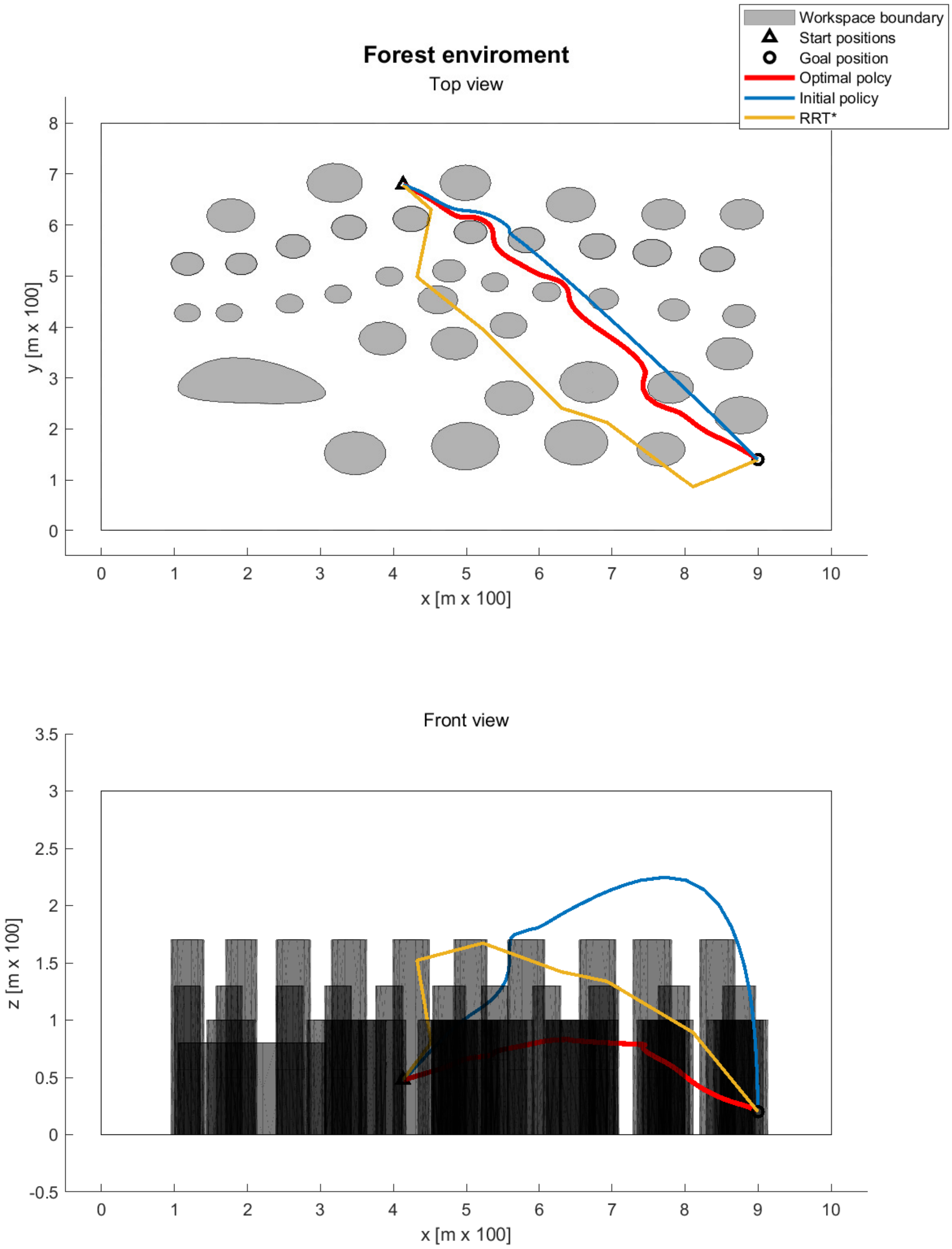


**Figure 5.15:** *Comparative simulation results of the forest environment from the starting position* $p_0 = [4.1, 6.8, 0.5]^\top$ *in front and top views. The figure depicts the initial trajectory (blue line), the optimal trajectory of the proposed method (red line) and the optimal trajectory of the RRT* method (yellow line).*

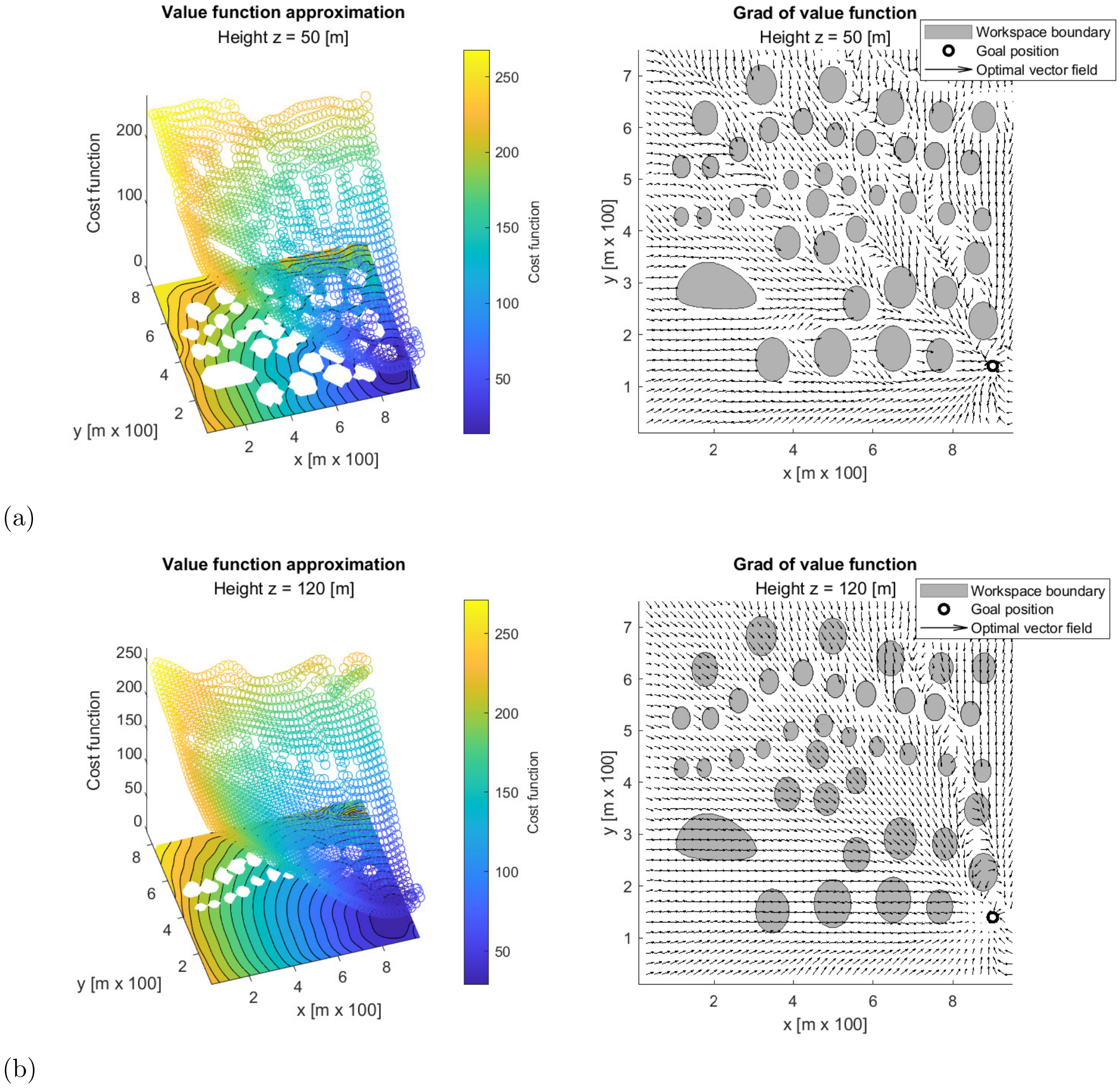
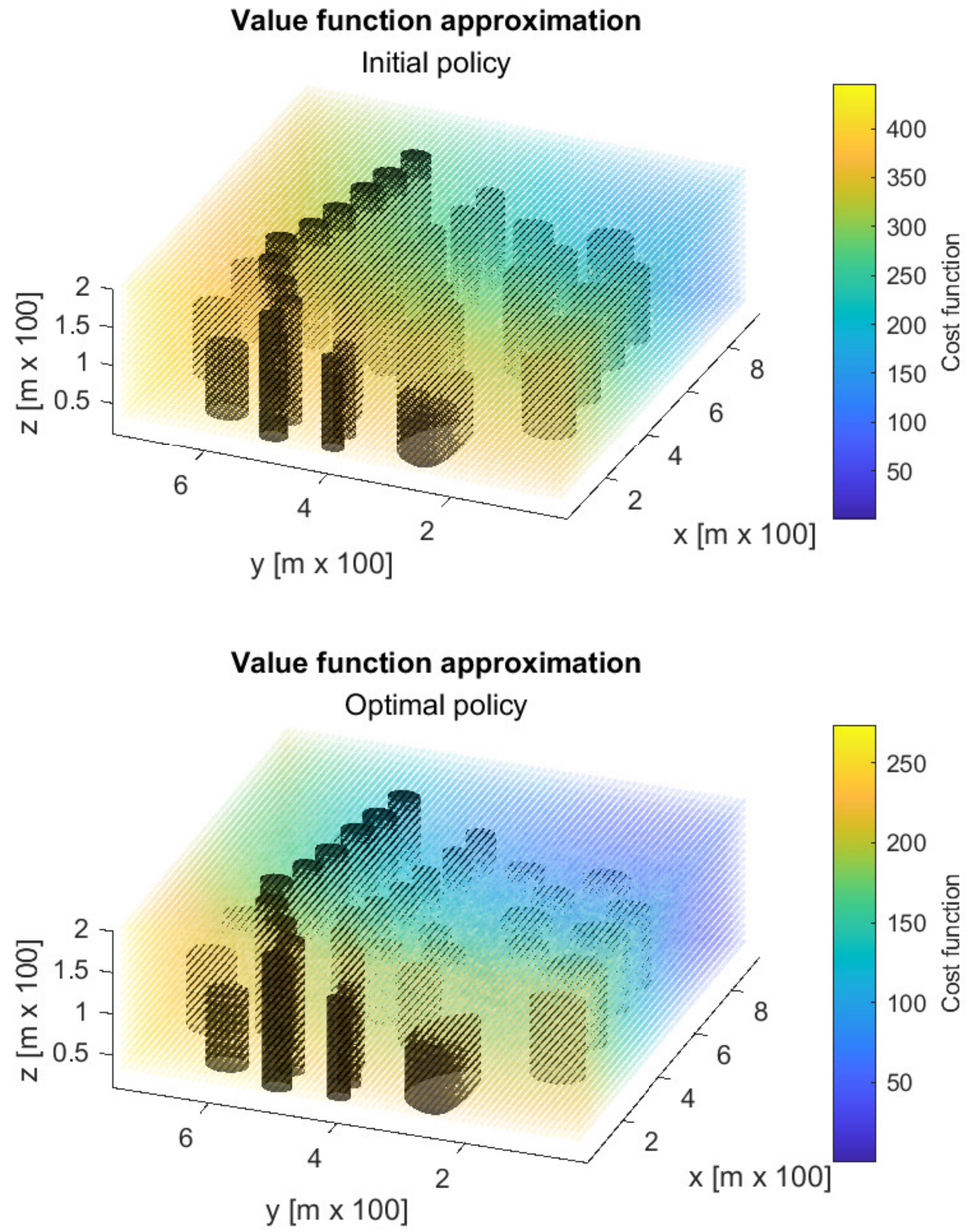


**Figure 5.16:** *The final cost function approximation in the forest environment and the corresponding optimal vector field at two different heights (a)* $z = 50[m]$ *and (b)* $z = 120[m]$.



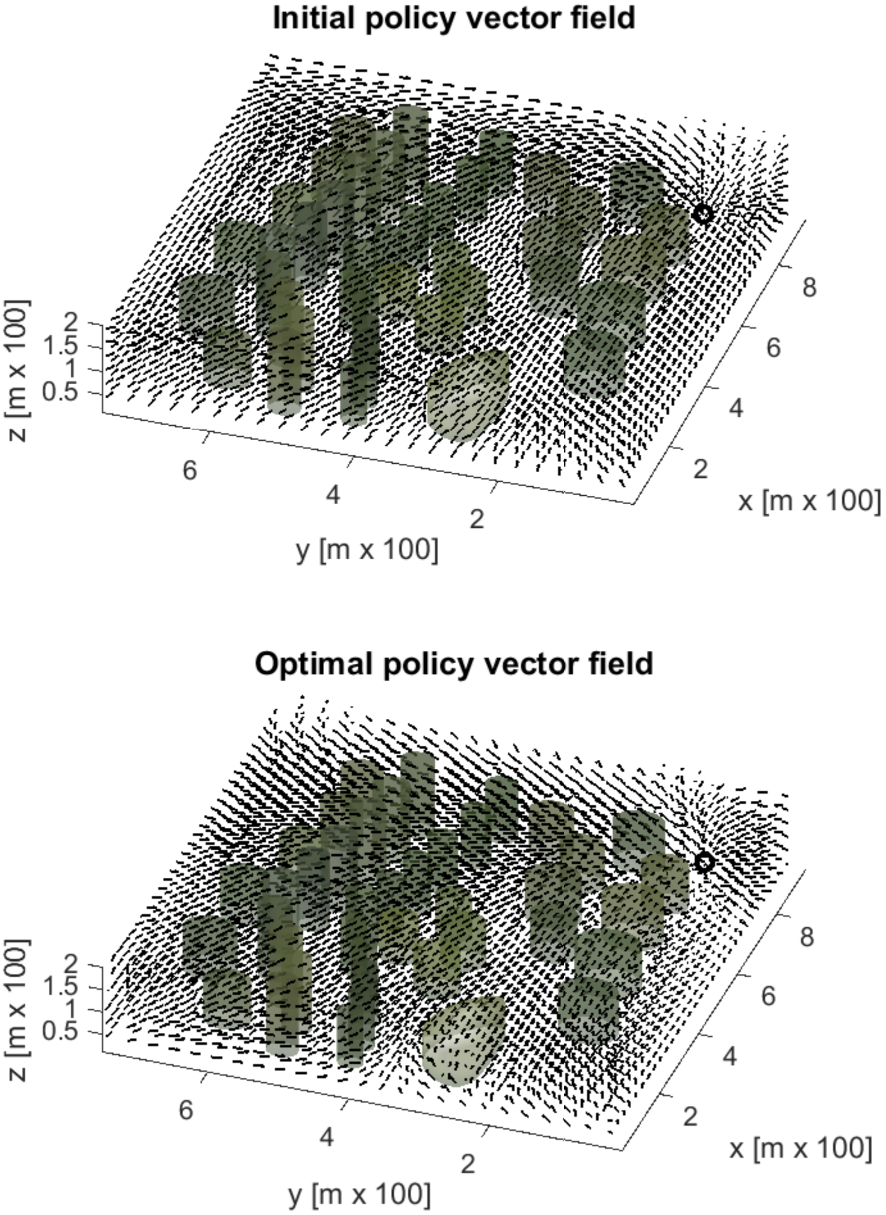
**Value function approximation**

Initial policy

**Value function approximation**

Optimal policy

**Figure 5.17:** *The cost function associated with the initial and the optimal control policy in isometric view of the forest environment.*


Initial policy vector field

Optimal policy vector field

**Figure 5.18:** *The vector fields illustrating the initial and the optimal control policy in isometric view of the forest environment.*

# Chapter 6

# Conclusions

## 6.1 Overview

The approach proposed in this thesis has shown significant advantages over existing methodologies, marking a notable stride towards the future directions of optimal path planning in continuous three-dimensional spaces. The control planning policy is structured in a parameterized form of a potential fluid flow model, which consists of a distribution of repelling sources on the boundary and an attractive sink at the goal position. Notably, this work presents a compelling proof of the existence of a stable solution to the motion planning problem in bounded and obstacle-cluttered 3D workspaces. Safety is also ensured by determining the weight factors of the potential flow components based on a set of linear von-Neumann boundary constraints. To address optimality, the control parameters are updated without affecting the harmonic properties of the potential flow, such as incompressibility, and based on the principles of dynamic programming successively converge towards the optimal solution of the Hamilton-Jacobi-Bellman equation. This is implemented by the proposed model-based actor-critic reinforcement learning scheme, which determines the appropriate strength of the source/sink elements on the boundary to optimally drive the entire flow towards the desired target.

## 6.2 Conclusions

The results of this study lead to the following conclusions:

- The motion planning policy is proved to produce smooth continuous trajectories that stabilize the system on the target position while at the same time avoiding collisions. This reactive solution clearly outperforms sampling-based methods, which rely on spatial discretization and in complex environments either do not provide smooth enough trajectories for autonomous vehicles or may fail to find a solution [34].

- The proposed approach generates a feasible path for every initial position within the workspace, unlike sampling-based methods that depend on the starting position. This could also be utilized for the coordination of multiple robotic vehicles, since a single expression for the path planning of every agent in the environment is a priori known.
- This motion planning method can be applied to complex boundary and obstacle geometries directly, eliminating the need for a map transformation, which was a significant limitation of extending previous approaches to three-dimensional spaces. This is further supported by applications in Fluid Mechanics, where the potential flow equations have been well-established for simulating challenging environments (e.g., the flow around helicopter rotors [35]).
- A deterministic optimal solution for the motion planning problem is achieved by leveraging the compelling theory of non-linear optimal control. The proposed deep reinforcement learning successfully overcomes the challenge of solving the difficult non-linear differential equations involved in the Hamilton-Jacobi-Bellman (HJB) optimality condition.
- The model-based nature of the proposed reinforcement learning algorithm confronts the challenges faced by model-free approaches, such as high sample complexity of the action and state space, the need for memory buffer storage techniques, and the exploration-exploitation trade off. The results demonstrate that the proposed algorithm converges within a few iterations.

## 6.3 Limitations and Future work

The motion planning approach of this thesis entails some limitations. Firstly, compensating for dynamic environments and noisy systems can be challenging, requiring the use of an additional low-level planner. In this work, the control policy's harmonic fluid-based basis function is strictly maintained throughout the entire optimization process. As a result, the improvement of the policy is limited by the optimal performance that can be achieved through this specific parameterized potential flow formulation. Additionally, the necessity to solve a hard constrained optimization problem during the training process presents a challenge in terms of computational complexity. In other words, the primary focus of this study lies on prioritizing the proven stability and safety of the control policy, which results in some limitations for attaining globally optimal solutions.

The future proposals to address the aforementioned challenges are summarized as follows:

**Improved motion planning policy**:

- Implementing a more complex formulation of the harmonic potential field should be examined. This involves incorporating either a higher order of weights distribution in each panel element or introducing additional terms to achieve a more sophisticated representation of the fluid flow. During this thesis, the investigation of including doublet and vortex elements in the potential field formulation has exhibited improved trajectory shapes, particularly in highly constrained non-convex workspaces. This enhancement is justified by the fact that vortex panels can provide improved control of the tangential velocity component, as depicted in Figure 6.1.

- The exploration of parameterizing the control weights with Radial Basis Functions (RBFs) has also yielded promising results during this research. Specifically, the weights of the harmonic policy can be parameterized through the following policy structure:

  $$\mathbf{u}(\mathbf{p}) = \nabla\Phi(\mathbf{p}) \cdot \overline{\mathbf{w}}(\mathbf{p}) = \nabla\Phi(\mathbf{p}) \cdot \mathbf{M}(\mathbf{p}) \cdot \mathbf{w}, \tag{6.1}$$

  where:

  - $\mathbf{M}(p) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \mathbf{z} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{z} \end{bmatrix} \in \mathbb{R}^{K\times(KL)+1}$

  - $\mathbf{z}(\mathbf{p}) = [e^{-c\|\mathbf{p}-\mathbf{p}_1\|}, e^{-c\|\mathbf{p}-\mathbf{p}_2\|}, \dots, e^{-c\|\mathbf{p}-\mathbf{p}_L\|}]$, denotes the vector of RBFs with $\mathbf{p}_i \in \mathbb{R}^3, i = 1, 3, \dots L$ the corresponding center of each term.

  - $\begin{bmatrix} w_0, w_1, \dots w_K \end{bmatrix}^\top \in \mathbb{R}^{K+1}$ are the tunable parameters of the control policy. It is of high importance to note that this approach allows the control parameters to be influenced by the robot's position within the workspace, thus enhancing optimality through a comprehensive representation of the potential field. However, a vital trade-off should be considered, as this approach compromises the harmonic properties of the potential field and consequently the stability of the control policy, while also significantly increases the computational time.

    Indicative results of the proposed extensions are showcased in Figure 6.2 to demonstrate the benefits of exploring these alternative approaches.

**Improved reinforcement learning approaches**:

- The investigation of alternative approaches for the policy improvement step is also recommended. Valid proposals include employing a gradient decent method or integrating the safety constraints into the cost function in order relax the quadratic optimization problem.

- Model-free reinforcement learning techniques offer promising future prospects. This provides parameter-free solutions to the motion planning problem, with the potential to enhance global optimality. By utilizing an initial admissible policy, obtained through the AHPF solution for the motion planning problem, improved convergence and accuracy of model-free approaches can be achieved. For instance, incorporating the Deep Deterministic Policy Gradient (DDPG) algorithm in the proposed reinforcement learning framework has shown some benefits throughout this study. In particular, this approach eliminates the need for a hard constrained optimization problem since the safety criteria are included in the critic network formulation. Furthermore, limitations of the DDPG algorithm [2] such as slow convergence and insufficient exploration are improved by initializing the critic network with on-trajectory samples, as presented in Chapter 4.

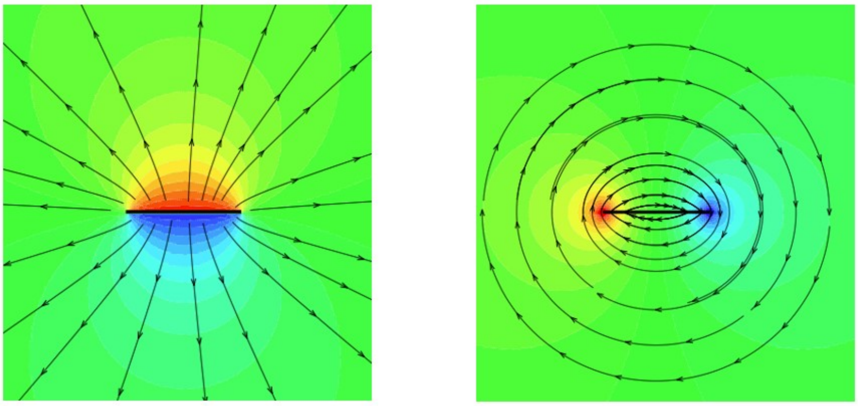


**Figure 6.1:** *The vector field of a source panel (left) and the vector field of vortex panel (right). From: [29].*

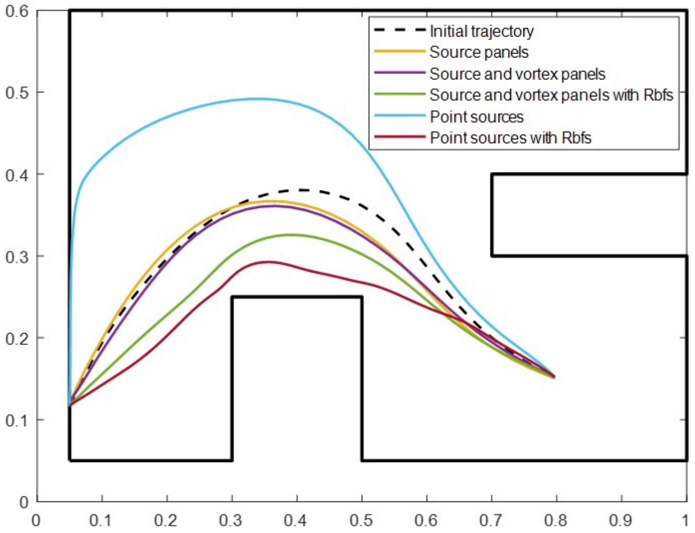


**Figure 6.2:** *Comparative trajectories of alternative proposed control policies.*

# Bibliography

[1] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood, "The grand challenges of science robotics," *Sci. Robot.*, vol. 3, no. 14, p. eaar7650, Jan. 2018.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," Sep. 2015.

[3] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *J. Intell. Robot. Syst.*, vol. 57, no. 1-4, pp. 65–100, Jan. 2010.

[4] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A systematic literature review of a* pathfinding," *Procedia Computer Science*, vol. 179, pp. 507–514, 2021, 5th International Conference on Computer Science and Computational Intelligence 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921000399

[5] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Rob. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[7] S. M. LaValle, "Rapidly-exploring random trees : a new tool for path planning," *The annual research report*, 1998.

[8] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[9] D. Lee, H. Song, and D. H. Shim, "Optimal path planning based on spline-RRT* for fixed-wing UAVs operating in three-dimensional environments," in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*. IEEE, Oct. 2014.

[10] J. Wang, B. Li, and M. Q. Meng, "Kinematic constrained bi-directional rrt with efficient branch pruning for robot path planning," *Expert Syst. Appl.*, vol. 170, p. 114541, 2021.

[11] O. Arslan and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal motion planning," in *2013 IEEE International Conference on Robotics and Automation.* IEEE, May 2013.

[12] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, and M. S. Muhammad, "Rrt*-smart: A rapid convergence implementation of rrt*," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[13] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Rob. Autom.*, vol. 8, no. 5, pp. 501–518, 1992.

[14] J.-O. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," 1992.

[15] S. G. Loizou, "Closed form navigation functions based on harmonic potentials," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 6361–6366.

[16] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robot navigation in complex workspaces using harmonic maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1726–1731.

[17] P. D. Grontas, P. Vlantis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Computationally efficient harmonic-based reactive exploration," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2280–2285, 2020.

[18] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Trajectory planning in unknown 2d workspaces: A smooth, reactive, harmonics-based approach," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1992–1999, 2022.

[19] ——, "Optimal robot motion planning in constrained workspaces using reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6917–6922.

[20] ——, "Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2005–2011, 2021.

[21] ——, "Optimal motion planning in unknown workspaces using integral reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6926–6933, 2022.

[22] Y. Zhang and K. P. Valavanis, "A 3-d potential panel method for robot motion planning," *Robotica*, vol. 15, pp. 421 – 434, 1997.

[23] D. Lau, J. Eden, and D. Oetomo, "Fluid motion planner for nonholonomic 3-d mobile robots with kinematic constraints," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1537–1547, 2015.

[24] G. N. Saridis and C. S. G. Lee, "An approximation theory of optimal control for trainable manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 152–159, 1979.

[25] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109805000105

[26] F. L. Lewis, D. Vrabie, and V. L. Syrmosn, *Optimal Control.* Hoboken, NJ,USA: Wiley, 2012.

[27] D. Vrabie and F. L. Lewis, "Adaptive optimal control algorithm for continuous-time nonlinear systems based on policy iteration," *2008 47th IEEE Conference on Decision and Control*, pp. 73–79, 2008.

[28] L. Evans, *Partial Differential Equations*, ser. Graduate studies in mathematics. American Mathematical Society, 2010. [Online]. Available: https://books.google.gr/books?id=Xnu0o_EJrCQC

[29] V. Riziotis, *Aerodynamics lecture notes*, Athens, NTUA, 2021.

[30] J. Katz and A. Plotkin, *Low-Speed Aerodynamics, Second Edition.* NY,USA: Cambridge University Press, 2012.

[31] F. T. Johnson, *A general panel method for the analysis and design of arbitrary configurations in incompressible flows.* Seattle,WA,USA: Boeing Commercial Airplane Company, 1980.

[32] J. Hess and A. Smith, "Calculation of potential flow about arbitrary bodies," *Progress in Aerospace Sciences*, vol. 8, pp. 1–138, 1967. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0376042167900036

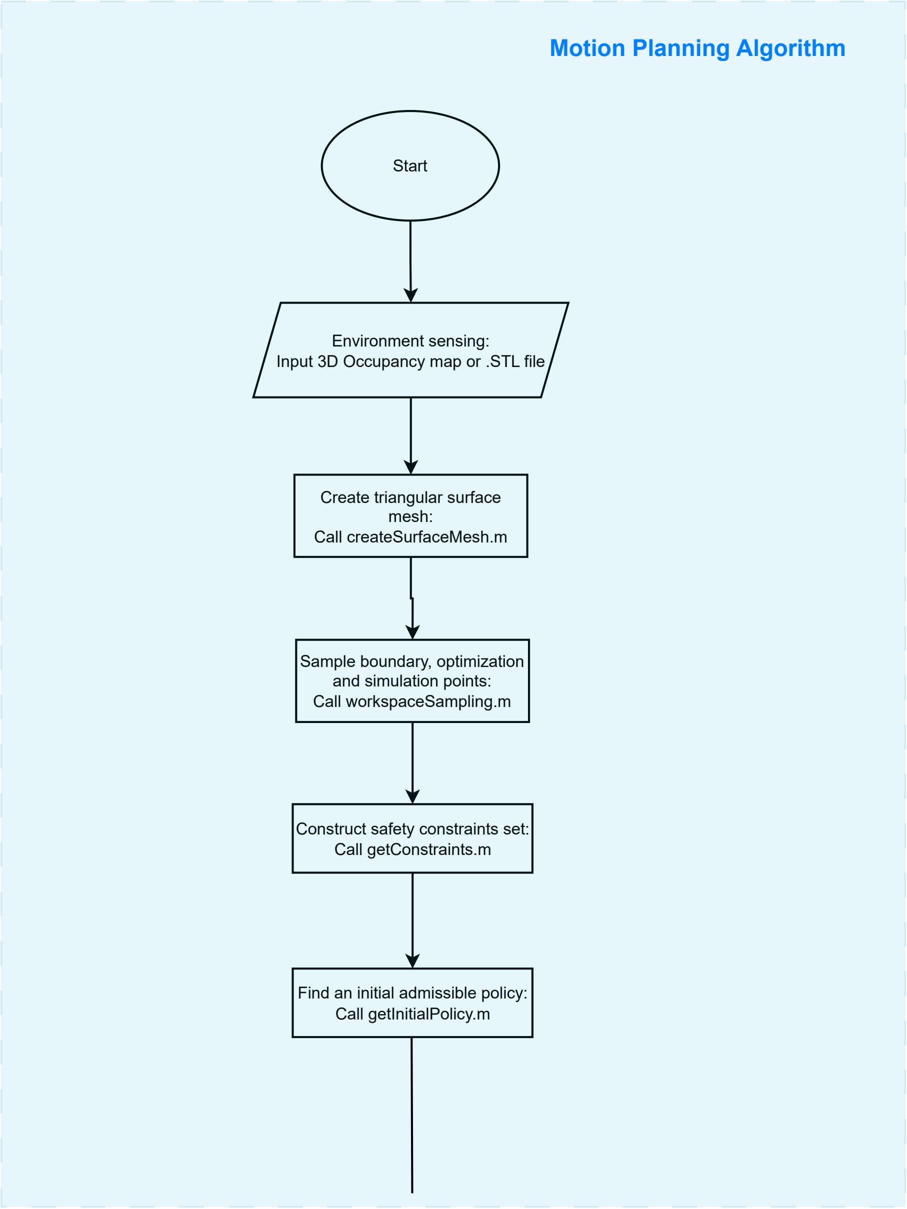
[33] I. Noreen, A. Khan, and Z. Habib, "Optimal path planning using RRT* based approaches: A survey and future directions," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, 2016.

[34] A. Mujumdar and R. Padhi, "Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles," *J. Aerosp. Comput. Inf. Commun.*, vol. 8, no. 2, pp. 17–41, Feb. 2011.

[35] E. A. Theologos, G. Papadakis, and V. Riziotis, *AERODYNAMIC ANALYSIS OF HELICOPTER IN INTERACTION WITH WIND TURBINE'S WAKE.* Delft, The Netherlands: 44th European Rotorcraft Forum, 2018.

# Appendix A

# Software Structure


Motion Planning Algorithm
Start
Environment sensing:
Input 3D Occupancy map or .STL file
Create triangular surface mesh:
Call createSurfaceMesh.m
Sample boundary, optimization and simulation points:
Call workspaceSampling.m
Construct safety constraints set:
Call getConstraints.m
Find an initial admissible policy:
Call getInitialPolicy.m

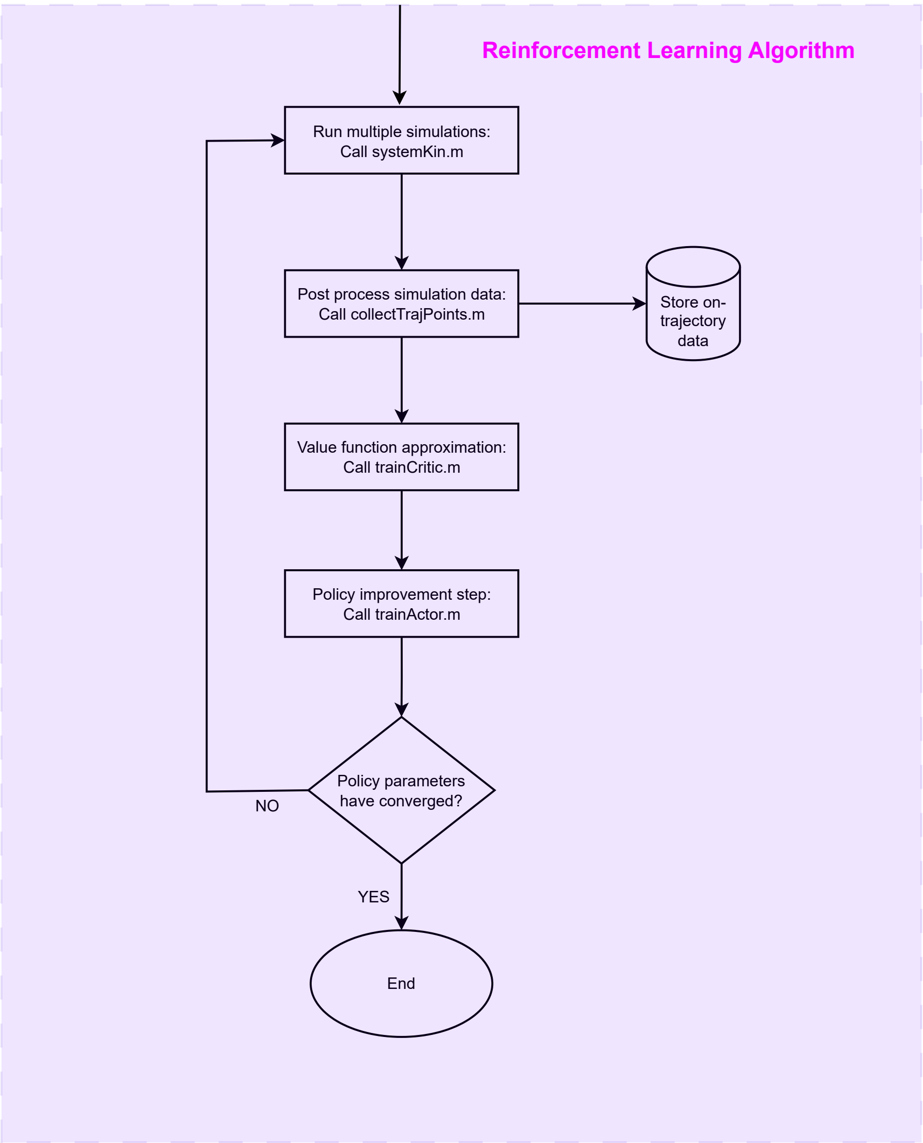

Reinforcement Learning Algorithm
Run multiple simulations: Call systemKin.m
Post process simulation data: Call collectTrajPoints.m
Store on-trajectory data
Value function approximation: Call trainCritic.m
Policy improvement step: Call trainActor.m
Policy parameters have converged?
NO
YES
End

**Figure 3:** *Software structure of the optimal motion planning algorithm.*

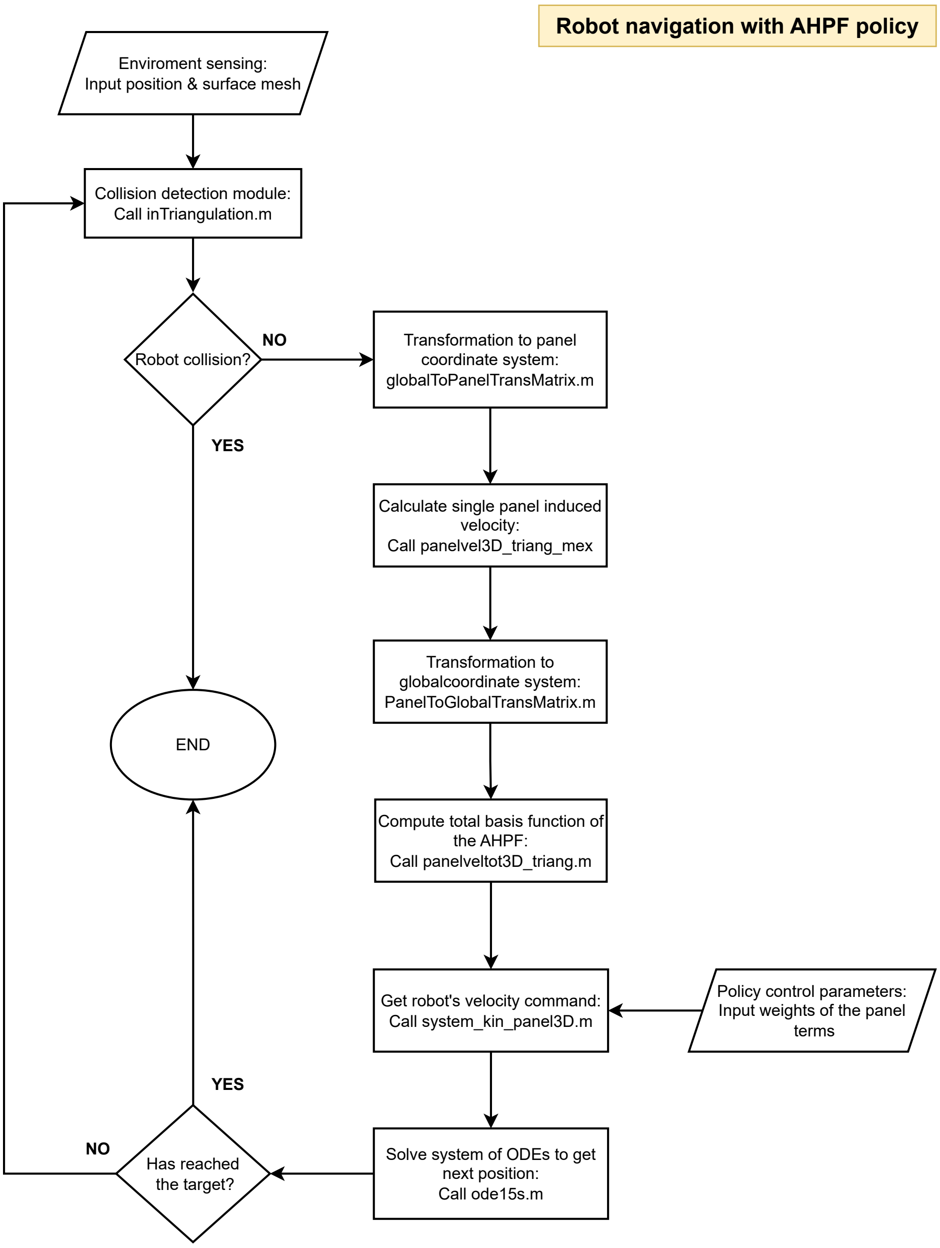

Robot navigation with AHPF policy
Enviroment sensing:
Input position & surface mesh
Collision detection module:
Call inTriangulation.m
Robot collision?
NO
YES
Transformation to panel coordinate system:
globalToPanelTransMatrix.m
Calculate single panel induced velocity:
Call panelvel3D_triang_mex
Transformation to globalcoordinate system:
PanelToGlobalTransMatrix.m
END
Compute total basis function of the AHPF:
Call panelveltot3D_triang.m
Get robot's velocity command:
Call system_kin_panel3D.m
Policy control parameters:
Input weights of the panel terms
YES
NO
Has reached the target?
Solve system of ODEs to get next position:
Call ode15s.m

**Figure 4:** *Structure of the robot navigation algorithm.*

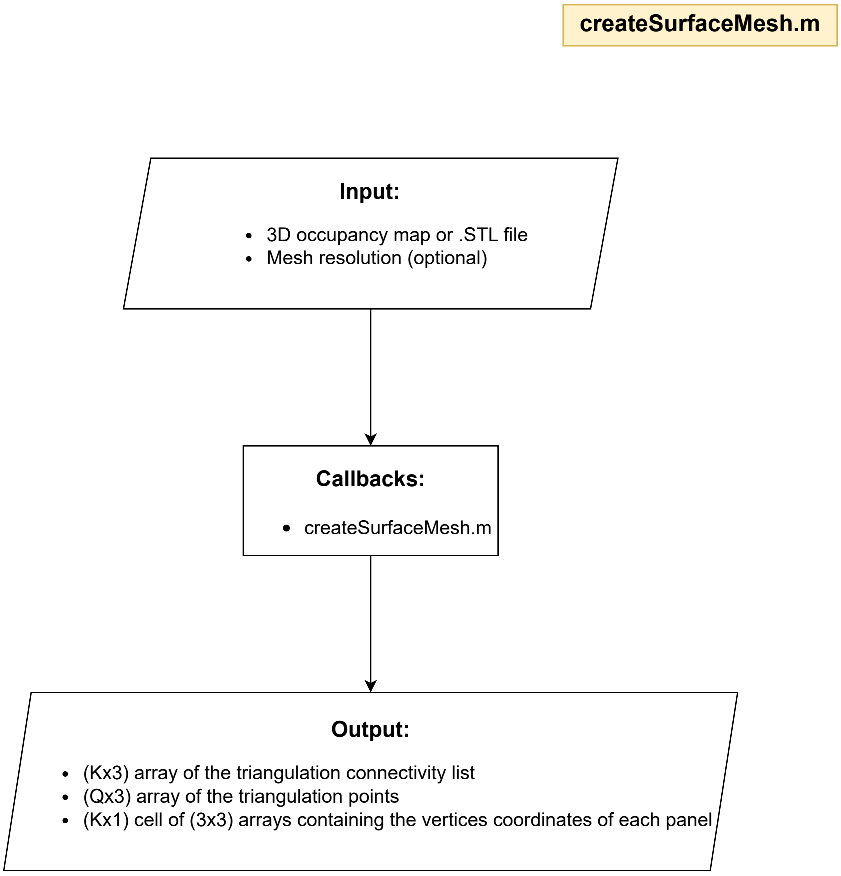

createSurfaceMesh.m
Input:
3D occupancy map or .STL file
Mesh resolution (optional)
Callbacks:
createSurfaceMesh.m
Output:
(Kx3) array of the triangulation connectivity list
(Qx3) array of the triangulation points
(Kx1) cell of (3x3) arrays containing the vertices coordinates of each panel

**Figure 5:** *Structure of the surface mesh algorithm.*

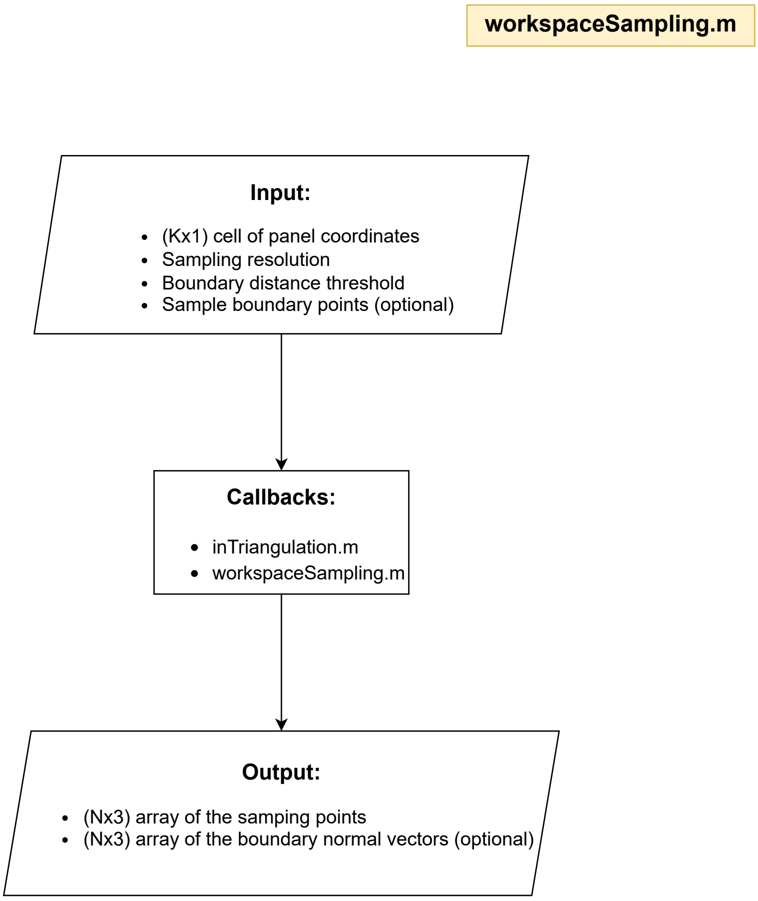

workspaceSampling.m
Input:
(Kx1) cell of panel coordinates
Sampling resolution
Boundary distance threshold
Sample boundary points (optional)
Callbacks:
inTriangulation.m
workspaceSampling.m
Output:
(Nx3) array of the samping points
(Nx3) array of the boundary normal vectors (optional)

**Figure 6:** *Structure of the 3D sampling algorithm.*

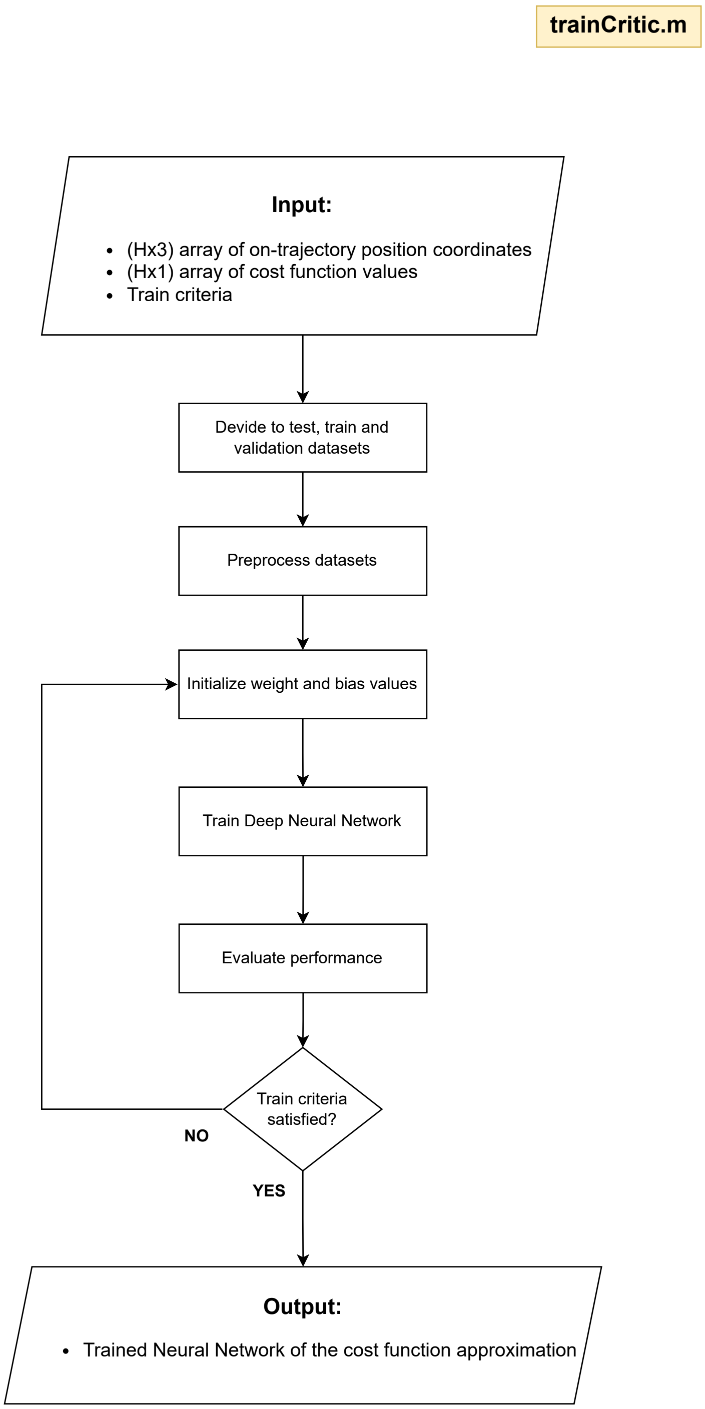

trainCritic.m
Input:
(Hx3) array of on-trajectory position coordinates
(Hx1) array of cost function values
Train criteria
Devide to test, train and validation datasets
Preprocess datasets
Initialize weight and bias values
Train Deep Neural Network
Evaluate performance
Train criteria satisfied?
NO
YES
Output:
Trained Neural Network of the cost function approximation

**Figure 7:** *Structure of the critic training algorithm.*

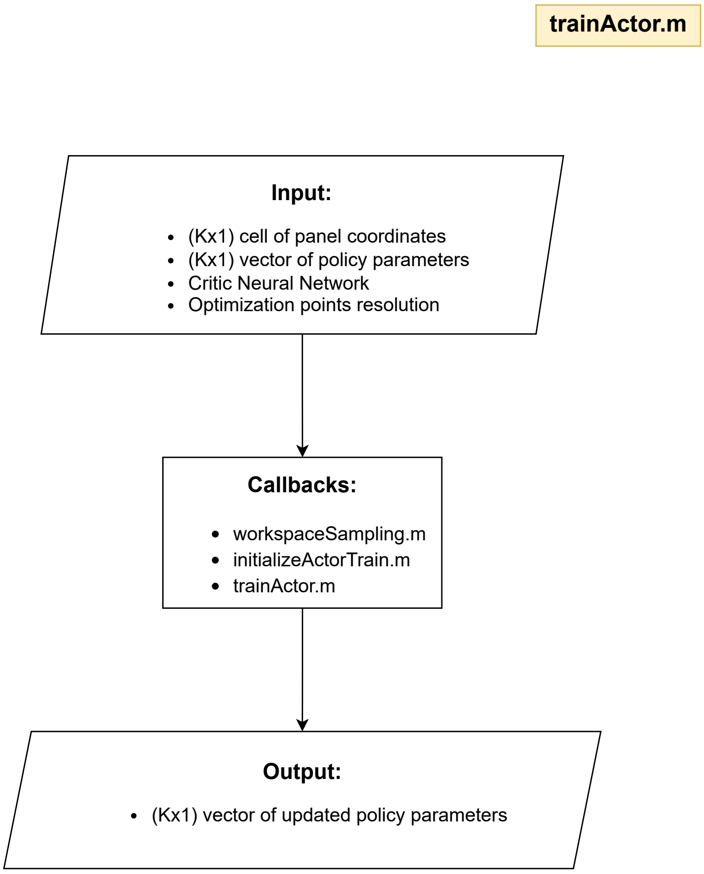

trainActor.m
Input:
(Kx1) cell of panel coordinates
(Kx1) vector of policy parameters
Critic Neural Network
Optimization points resolution
Callbacks:
workspaceSampling.m
initializeActorTrain.m
trainActor.m
Output:
(Kx1) vector of updated policy parameters

**Figure 8:** *Structure of the policy improvement training algorithm.*