Optimal Motion Planning in Constrained Workspaces Using Reinforcement Learning

Diploma Thesis
National Technical University of Athens

Panagiotis Rousseas

Supervisor: Prof. K. Kyriakopoulos
Advisor: Dr. Ch. Bechlioulis

July 2020

Acknowledgements

I would like to first of all thank my supervisor, Prof. Kostas Kyriakopoulos, for both his invaluable advice and directions and for passing on his philosophy of rigor and methodical work. This work would not be possible without the persistent advice and guidance of Dr. Charalampos Bechlioulis and the staff of the Control Systems Laboratory. Finally, I would like to thank all my mentors throughout these last five years and last but not least all the people that were by my side through this period.

Abstract

In this work, a novel solution to the optimal motion planning problem is proposed, through a continuous, deterministic and provably correct approach, with guaranteed safety and which is based on a parametrized Artificial Potential Field (APF). In particular, Reinforcement Learning (RL) is applied to adjust appropriately the parameters of the underlying potential field towards minimizing the Hamilton-Jacobi-Bellman (HJB) error. The proposed method, outperforms consistently a Rapidly-exploring Random Trees (RRT*) method and consists a fertile advancement in the optimal motion planning problem. Finally this work gives rise to a new outlook on solutions for the aforementioned problem.

Περίληψη

Στην παρούσα εργασία, μια καινοτόμος λύση στο πρόβλημα του βελτίστου σχεδιασμού πορείας προτείνεται, μέσω μιας συνεχούς αιτιοκρατικής και αποδεδειγμένα σωστής προσέγγισης, με εξασφαλισμένη ασφάλεια πλοήγησης, και η οποία είναι βασισμένη σε παραμετροποιημένα Τεχνητά Αρμονικά Πεδία. Συγκεκριμένα, εφαρμόζεται Ενισχυτική Μάθηση προκειμένου να προσαρμοστούν καταλλήλως οι παράμετροι του αντίστοιχου δυναμικού πεδίου προκειμένου να ελαχιστοποιηθεί το σφάλμα της εξίσωσης Hamilton-Jacobi-Bellman. Η προτεινόμενη μέθοδος υπερισχύει συστηματικά έναντι μιας μεθόδου Rapidly-exploring Random Trees και συνιστά σημείο γόνιμης προόδου στο πρόβλημα βέλτιστου σχεδιασμού πορείας. Τέλος, η παρούσα εργασία παρέχει νέες προοπτικές σε λύσεις για το ανωτέρω πρόβλημα.

Contents

Chapter 1

Introduction

1.1 Optimal Motion Planning

Motion planning problems have always been a main focus point of control system theory and robotics. While they might appear to be a classic control theory problem, where traditional methods can be used to control the motion of a robot, certain peculiarities give this type of problems a different flavor. Such peculiarities might be specific restrictions pertaining to the motion of a robot (e.g. non-holonomic constraints) or possible obstacles in the workspace of a robot, calling for the establishment of robust control techniques that will ensure safety during the navigation and convergence to a desired goal position. The aforementioned issues have been tackled in various ways, and safe techniques for navigation have long been established. However, the same cannot be put forward when considering optimality in such problems. While efforts have been made towards the goal of optimizing the motion of actors in a workspace, the problem is in no way considered trivial yet, and we believe that there is room for exploring novel solutions and ameliorating the existing results in the related literature.

In this work, we intend to explore the application of Reinforcement Learning methods in optimizing the motion of a robot moving in a two-dimensional constrained, but fully known workspace with internal fixed obstacles. In particular, an offline solution to the underlying optimization problem is formulated in such a way that ensures safety and convergence with mathematical rigor using robust principles and tools from the successive approximation theory [22]. Subsequently, we establish an on-line reinforcement learning approach for optimizing the motion of a moving robot with respect to a specific utility function, with great emphasis on the rigorous proof of safety and convergence. The motivation behind the online approach stems from the fact that in real-world problems, not all trajectories of a workspace are of use, but rather specific starting-ending point combinations are needed. Therefore, an online approach is not only sufficient, but also advantageous with respect to computational complexity. Finally, the prospect of implementing the online scheme in unknown workspaces further motivates the latter's formulation.

1.2 Significance

This work presents a novel approach whose significance lies not only in its novelty, but more so in its provable nature. Most existing methods for addressing the optimal motion planning problem are probabilistic in nature and can not provide deterministically optimal solutions. While the method itself consist a new tool for engineering disciplines, it moreover opens up new pathways and provides with fresh insight on possible approaches in solving the optimal motion planning problem, further adding to its value.

1.3 Related Work

Since the early days of robotics, many research efforts have been devoted to the motion planning problem and thus many approaches have been formulated. Such approaches can be generally classified as discrete methods, e.g., Configuration Space Decomposition methodologies [24]-[4], Probabilistic Sampling methods, e.g., Rapidly Exploring Random Trees [7]-[27] or Probabilistic Roadmaps [11]-[3] and others such as Manifold Samples [16]. On the other hand, the Optimal motion planning problem has been approached via Receding Horizon control [17]-[21] and Path Homotopy Invariants [2]-[6].

A specific class of solutions to the motion planning problem, and one that aims at addressing both safety and convergence aspects are the APFs, as introduced in [8]. This class of solutions encompasses both information for safety and convergence in the form of the gradient of a potential field. However, APFs entail problems of unwanted local equilibria due to their inherent construction and the topology of the workspace [10]. Rimon and Koditschek managed to produce a family of APFs, namely Navigation Functions (NF) that are applied to a transformed version of the physical workspace in the form of a sphere world[1]. Along with providing a constructive transformation for mapping workspaces with star-shaped obstacles (sets with a point from which any ray crosses the boundary once) to the aforementioned sphere worlds, Rimon and Koditschek aleviated some of the issues of the APFs as well. However, extensive tuning is required to get rid of local minima and in practice these functions prove difficult to be implemented (see [20]).

Aiming at tackling the shortcomings of APFs, a specific sub-category of the latter was introduced, namely the Artificial Harmonic Potential Fields (AHPF) [14]-[9]. The AHPFs are free of local minima by construction, and negate many of the issues of previous NFs. In the present work, the natural progress of previous research efforts [18], leads to inheriting all the strong points of AHPFs and introducing a robust solution to the optimal motion planning problem. In order to accomplish this, a novel approach will be introduced, encompassing past work on Reinforcement Learning Optimization [25], re-framed and adapted for a specific family of AHPF-inspired motion controllers. Our work provides a deterministic and mathematically rigorous approach that exceeds the capabilities of previous probabilistic approaches. The implementation of Reinforcement Learning is pivotal in the current approach, as it overcomes the need for solving a very hard non-linear partial differential equation for calculating the cost

[1] A Euclidean sphere world of dimension N is formed by removing from the interior of a large N-dimensional ball a finite number of non-overlapping smaller balls.

function. Additionally, the latter is rigorously proven to converge under mild assumptions.

1.4 Chapter Overview

In the following section we will first present the optimal motion planning problem, along with preliminary mathematical and control theory tools. Afterwards we will formulate an off-line solution to the above problem using a set of parametrized controllers, after which, a subsequent on-line solution will be presented. Subsequently, the main results of this work will be presented. Finally we will discuss the scope of the above work and future research efforts.

Chapter 2

The Optimal Motion Planning Problem

2.1 Problem Formulation

Consider a point robot operating within a bounded and connected workspace $\mathcal{G} \subset \mathbb{R}^2$ with M inner distinct obstacles $\mathcal{O}_i \subset \mathcal{G}, i = 1, ..., M$ and a desired position $p_0 \in \mathcal{W} \triangleq \mathcal{G} - \cup_{i=1}^{M} \mathcal{O}_i$. Let $p = [x, y]^T \in \mathcal{W}$ denote the robot's position. The robot is considered to have full knowledge of the aforementioned workspace characteristics, as well as of its position. The robot's motion is described by the single integrator model:

$$\dot{p} = u, p(0) = \bar{p} \in \mathcal{W} \tag{2.1}$$

where $p \in \mathcal{W}$ is the state-vector, $u \in \mathbb{R}^2$ is a control input (i.e., velocity command) and $\bar{p} \in \mathcal{W}$ denotes the initial position. Now, consider the optimal motion planning problem of minimizing a cost function that consists of a state-related term, namely $Q(p; p_0)$ and a control input-related term, namely $R(u)$. Hence, the following value function should be subject to minimization:

$$V(\bar{p}; p_0) = \int_0^{\infty} \left[Q(p(\tau; \bar{p}); p_0) + R(u(\tau)) \right] d\tau, \quad \forall \bar{p} \in \mathcal{W} \tag{2.2}$$

where $\bar{p}$ is the initial state of the system $\bar{p} = p(0)$ and p_0 denotes the goal position. The goal of this Thesis is to present a novel solution, i.e. controller forms for the dynamics of Eq. (2.1), that consist both optimal -w.r.t. the cost function of Eq. (2.2)- and safe policies that converge to the desired position. Great emphasis is given to the rigorous proof that the proposed controllers will satisfy the above criteria. Finally we aim at presenting both off-line and on-line Reinforcement Learning (RL) policies for solving the aforementioned problem.

2.2 Preliminaries

In this section the main theoretical tools and the background that are used in this Thesis will be briefly presented.

2.2.1 Optimal Control

Optimization problems lie at the heart of systems' control theory and has thus been a main focal point for many researchers since the dawn of the field. Consider the continuous time non-linear system in the general form:

$$\dot{x} = f(x) + g(x)u \tag{2.3}$$

with state $x(t) \in \mathbb{R}^n$ control input $u(t) \in \mathbb{R}^m$ and the usual assumptions required for the existence of unique solutions and an equilibrium point at $x = 0$. The equilibrium at 0 is considered without loss of generality as can be easily showed with a simple transformation. We assume that system is stabilizable on a set $\Omega \subseteq \mathbb{R}^n$; that is, there exists a continuous control function u(t) such that the closed loop system (2.3) is asymptotically stable on Ω. Then consider that such a control policy that satisfies the asymptotic stability of the system (2.3) has a specific form – is a known function in terms of the state vector x –, and can be expressed in terms of any parameters $k \in \mathbb{R}^{N_0-1}$. Under this assumption the second term of the right-hand side of Eq. (2.3) can be written as:

$$g(x)u = g(x)u(x,k) = h(x,k(x)) \in \mathbb{R}^n \tag{2.4}$$

So, Eq. (2.3) becomes:

$$\dot{x} = f(x) + h(x,k) \tag{2.5}$$

where from now on $k \in \mathbb{R}^{N_0-1}$ are considered the tunable parameters of the control policy. Now define a cost function with its value associated with the feedback control policy $u = \mu(x,k)$ given by:

$$V^{\mu}(x(t)) = \int_t^{\infty} r(x(\tau), u(x(\tau), k(x(\tau))))d\tau \tag{2.6}$$

where $r(x,u)$ a positive definite, real-valued utility function. A policy is called admissible if it is continuous, stabilizes the system, and has finite associated cost. If the cost is smooth, then an infinitesimal equivalent to (2.6) can be found by differentiation to be the equation:

$$r(x,\mu(x,k)) + (\nabla V^{\mu})^T(f(x) + g(x)\mu(x,k(x(\tau)))), V(0) = 0 \tag{2.7}$$

where ∇V^{μ} denotes the gradient of the cost function ∇V^{μ} with respect to x. This is the continuous time Bellman equation. It can be defined based on the continuous time Hamiltonian function:

$$H(x,\mu(x),\nabla V^{\mu}) = r(x,\mu(x,k(x(\tau)))) + (\nabla V^{\mu})^T(f(x) + g(x)\mu(x,k(x(\tau)))) \tag{2.8}$$

The optimal value satisfies the continuous time Hamilton-Jacobi-Bellman equation; therefore, the optimal values of the tuning parameters k will satisfy it as well:

$$k^* = \arg\min_k H(x,\mu(x),\nabla V^*) \tag{2.9}$$

2.2.2 Harmonic Artificial Potential Fields

Several approaches have been implemented in order to solve motion planning problems, one of which with great significance to this Thesis consists a class

of Artificial Potential Fields (APF), namely the Harmonic Artificial Potential Fields (HAPF). The HAPF ψ used here is the same as used in [18]. Therefore, we construct a potential on the extended harmonic space with point sources at the desired configuration $v_d = B(T(p_0))$ and as well as with the points $v_i = B(T(\partial\mathcal{O}_i)), i = 1, ..., M$ with the disjoint Jordan curves that consist the obstacles. The mapping $B(T(\cdot))$ is a two-part transformation that maps the workspace $\mathcal{W}$ to the full plane $\mathbb{R}^2$ and will be discussed thoroughly later. The potential field is as follows:

$$\phi(v, k) = k_d \cdot ln\left(\frac{\|v - v_d\|}{2}\right) - \sum_{i=1}^{M} k_i \cdot ln\left(\frac{\|v - v_i\|}{2}\right) \tag{2.10}$$

where $k_d > 0$ and $k_i \geq 0$ the tuning parameters denoted by the vector $k = [k_d, k_1, ..., k_{N_0}]$. Now we define a reference field ψ based in ϕ as in [18], given by

$$\psi(v, k) = \frac{1 + tanh(w \cdot \phi)}{2} \tag{2.11}$$

where $w \in \mathbb{R}^+$ a non-negative scaling constant. Additionally, the gradient of ψ with respect to v is given by

$$\nabla_v \psi(v, k) = w \cdot \frac{1 + tanh^2(w \cdot \phi)}{2} \cdot \nabla_v \phi(v, k) \tag{2.12}$$

which is bounded and well-defined for all $v \in B(D)$. And the $\nabla_v \phi$:

$$\nabla_v \phi = k_d \frac{v - v_d}{\|v - v_d\|^2} - \sum_{i=1}^{M} k_i \frac{v - v_i}{\|v - v_i\|^2} \tag{2.13}$$

The above field has been proven sufficient for navigation [18].

2.2.3 Neural Networks and Reinforcement Learning Methods

Neural Networks (NN) have long been used to approximate sufficiently well functions within certain compact sets [19]. Consider a function $V : \mathbb{R}^N \to \mathbb{R}$. The latter can be approximated as follows:

$$V(p) = V_L(p) = \sum_{j=1}^{L} w_j \phi_j(p) + \epsilon(p) = w^T \cdot \phi(p) + \epsilon(p), \quad p \in \mathbb{R}^N \tag{2.14}$$

where $\epsilon(p)$ the estimation error of the NN. The above is essentially a single-layer NN with activation functions (basis functions) $\phi(p) = [\phi_1(p), \phi_2(p), \cdots, \phi_L(p)]^T : \mathbb{R}^N \to \mathbb{R}^L$ and respective NN weights $w = [w_1, w_2, \cdots, w_L]^T \in \mathbb{R}^L$. For the scope of this Thesis, the respective theory by Khalaf et al. [1] will hold and more specifically the results of pp.782-786. These NN will be applied in the context of RL methods that will be presented here.

Three main RL processes are examined within the scope of this Thesis, namely a successive approximation theory application [1] for the off-line approach along with linear regression on on-line sampled data [12] and a differential adaptive law [26] for t he on-line approach.

- **Successive Approximation Theory** The main concept of the successive approximation theory is the application of an iterative process through which better approximations for a given function can be obtained by utilising the previous approximation.
- **Linear Regression on samples** This method consists of taking samples of the state variable and the system input along the trajectory and performing linear regression on the integral reinforcement form of the cost function in order to improve the policy of the robot.
- **On-line adaptive law** The last method essentially minimizes the approximation error of the Hamilton-Jacobi-Bellman (HJB) through a steepest-descent tuning law for the weights of the NN.

Chapter 3

Off-line Solution using Parametrized Controllers

Having presented the foundations of this Thesis, we are ready to proceed with presenting solutions to the proposed problem of optimizing the motion of a robot.

3.1 A Set of Parametrized Control Policies Based on HAPFs

3.1.1 A Proposed Parametrized Controller Form

Having presented the basics of HAPFs we will now introduce a family of parametrized control policies $u = h(p, k)$ to the aforementioned problem, where k denotes the control parameter vector. First, assume that we have a diffeomorphic transformation[1] from $\mathcal{W}$ onto the punctured plane denoted by $f : \mathcal{W} \to \mathbb{R}^2 - \{\mathcal{V}_1, ..., \mathcal{V}_M\}$ that satisfies $f(p_0) = \mathcal{V}_0$ and $f(\partial\mathcal{O}_i) = \mathcal{V}_i, i = 1, ..., M$. The proposed parametrized solution is given as:

$$h(p, k) \triangleq P(p) \cdot k = -\mathcal{J}_f^{-1}(p) \cdot g(p) \cdot A \cdot k, \tag{3.1}$$

where the control-parameter vector $k \triangleq [k_0, k_1, \cdots, k_M]^T \in \mathbb{R}^{M+1}$ is analogous to the harmonic potential field weights, A is a square matrix of the following form:

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(M+1)\times(M+1)} \tag{3.2}$$

[1] The adopted transformation is the composition of a diffeomorphism that maps all points inside $\mathcal{W}$ onto the open punctured unit disk [18], with a diffeomorphism that maps the unit disk onto $\mathbb{R}^2$ [14].

and $g(p)$ defines a vector basis:

$$g(p) = \prod_{i=0}^{M} tanh\left(\|f(p) - \mathcal{V}_i\|^2\right) \cdot \cdot \left[\frac{f(p) - \mathcal{V}_0}{\|f(p) - \mathcal{V}_0\|^2}, \frac{-f(p) + \mathcal{V}_1}{\|f(p) - \mathcal{V}_1\|^2}, ..., \frac{-f(p) + \mathcal{V}_M}{\|f(p) - \mathcal{V}_M\|^2}\right] \tag{3.3}$$

with $\mathcal{J}_f(p)$ denoting the Jacobian of the transformation $f(p)$. The above formulation is a direct analog to the gradient of a classic harmonic potential field, enhanced in a way that fits the needs of the optimization process that follows. We will later show that this formulation, besides safety, ensures convergence as well. Furthermore, we have further simplified the problem of stability and safety of the robot incorporating the matrix A in (3.1). As shown in [14], for safe navigation the weight of the attractive term has to be greater than the sum of the weights of all the repulsive ones. It is evident that such formulation can be quite tedious especially when considering an optimization approach. Nevertheless, in our formulation, the equivalent constraint boils down to:

$$k_i > 0, i = 0, .., M \tag{3.4}$$

owing to the adopted form of matrix A in (3.2) (notice that the weight of the attractive term is the sum of all k_i). We will prove analytically how our method will ensure safety and convergence after discussing the optimization problem, as our solution for the vector k will ensure both optimality and (3.4). Hence, we consider the following value function:

$$V(\bar{p}; p_0) = \int_0^{\infty} \left[Q(p(\tau; \bar{p}); p_0) + W(k(\tau))\right] d\tau \tag{3.5}$$

for all $\bar{p} \in \mathcal{W}$, where

$$Q(p; p_0) = \beta \cdot \|p - p_0\|^2, \beta > 0 \tag{3.6}$$

and

$$W(k) = \gamma \cdot \sum_{i=0}^{M} \int_{\alpha(p)}^{k_i} \left(\frac{v_i}{\alpha^2(p)} - \frac{1}{v_i}\right) dv_i, \gamma > 0 \tag{3.7}$$

with $\alpha(p) = \frac{1}{\sqrt{1+M}} \frac{\bar{u}}{\sqrt{\|P(p)\|^2+1}}$ for an upper bound of the velocity $\bar{u}$. Note that the term of Eq. (3.7) can also be written as:

$$W(k) = \gamma \cdot \sum_{i=0}^{M} \left[\frac{1}{2}\left(\left(\frac{k_i}{\alpha(p)}\right)^2 - 1\right) - ln\left(\frac{k_i}{\alpha(p)}\right)\right] \tag{3.8}$$

In classical control problems, the input-related cost is quadratic. However, in our approach, (3.7) is used to ensure safety and convergence. Notice that the selection of W is not heuristic since through its minimization all components of k remain positive. Moreover, the lower bound of the integral in (3.7) ensures that the parameters k_i are such that an upper bound of the velocity control signal is minimized. To see that, consider the control vector form:

$$u = P(p) \cdot k \tag{3.9}$$

The following are true

$$\|u\|^2 = \|P(p) \cdot k\|^2 \leq \|P(p)\|^2 \cdot \|k\|^2 \tag{3.10}$$

Considering now that with the form of the function $W(k)$ - and for reasons that will later become clear, the values k_i of the control will eventually converge to the value of the lower limit of the integral of Eq. (3.7). Therefore, setting this value - let α denote this lower limit of the integral - equal to

$$\alpha(p) = \frac{1}{\sqrt{1+M}} \frac{\|u\|_{max,desired}}{\sqrt{\|P(p)\|^2 + 1}} \tag{3.11}$$

Then

$$\begin{aligned} \|k\|^2 = \sum_{i=0}^{M} {k_i}^2 = \sum_{i=0}^{M} \frac{1}{\sqrt{1+M}} \frac{\|u\|_{max,desired}}{\sqrt{\|P(p)\|^2 + 1}} = \\ = \frac{\|u\|_{max,desired}}{\sqrt{\|P(p)\|^2 + 1}} \end{aligned} \tag{3.12}$$

And consequently

$$\begin{gathered} \|u\|^2 \leq \|P(p)\|^2 \cdot \|k\|^2 \Rightarrow \\ \|u\|^2 \leq \|P(p)\|^2 \frac{\|u\|^2_{max,desired}}{\|P(p)\|^2 + 1} \Rightarrow \\ \|u\|^2 \leq \frac{\|u\|^2_{max,desired}}{\frac{1}{\|P(p)\|^2} + 1} \end{gathered} \tag{3.13}$$

This means that, if the norm $\|P(p)\|$ becomes large, – e.g. on positions where the value of the pseudo-gradient is large – then the control vector will be bounded by the desired value. Additionally $\|P(p)\| = 0$ and the control bound is well defined as in the first equation. Finally, a proper norm for the term $\|P(p)\|$ needs to be chosen. This should be chosen on the basis of low computational cost. We therefore propose the Frobenius norm.

3.1.2 Proposed Transformation

In the present work the transformation f comprises of two parts, namely a first transformation which maps the boundary of the workspace to the unit circle and the boundaries of the obstacles to points inside the latter, while also mapping all points inside the workspace to the open unit disk, excluding the boundary and a second transformation, which maps the unit disk at infinity and all inside points to the two-dimensional plane. Assume these transformations as follows, denoting the unit disk as $\mathcal{D}$.

$$q(p) : \mathcal{G} - \bigcup_{i=1}^{M} \mathcal{O}_i \rightarrow \mathcal{D} - \{\mathcal{Q}_1, ..., \mathcal{Q}_M\} \tag{3.14}$$

while also satisfying:

$$q(p_0) = \mathcal{Q}_0 \text{ and } q(\partial\mathcal{O}_i) = \mathcal{Q}_i, i = 1, ..., M \tag{3.15}$$

And the second transformation:

$$v(q) : \mathcal{D} \rightarrow \mathbb{R}^2 \tag{3.16}$$

while also satisfying:

$$v(\mathcal{Q}_0) = \mathcal{V}_0 \ , \ v(\mathcal{Q}_i) = \mathcal{V}_i, i = 1, ..., M \text{ and } v(\partial\mathcal{D}) \rightarrow \infty \tag{3.17}$$

For the first part, the transformation presented in [18] will be used as is, while we propose for the second part of the transformation the following form

$$v(q) = \frac{q}{1 - \|q\|^2} \tag{3.18}$$

and its inverse Jacobian, with $q = [x, y]^T$:

$$\mathcal{J}_v^{-1}(p) = \begin{bmatrix} \frac{(x^2-y^2-1)(x^2+y^2-1)}{1+\|q\|^2} & \frac{2xy(x^2+y^2-1)}{1+\|q\|^2} \\ \frac{2xy(x^2+y^2-1)}{1+\|q\|^2} & -\frac{(x^2-y^2+1)(x^2+y^2-1)}{1+\|q\|^2} \end{bmatrix} \tag{3.19}$$

If we consider the respective Jacobians of the transformations as $\mathcal{J}_q(p)$ and $\mathcal{J}_v(q)$ it can be easily shown that, for two vectors $v \in \mathcal{R}^2$ and $p \in \mathcal{W}$ that satisfy:

$$v = v(q(p)) \tag{3.20}$$

the following is true

$$\dot{p} = \mathcal{J}_q^{-1}(p) \cdot \mathcal{J}_v^{-1}(q) \cdot \dot{v} \tag{3.21}$$

and therefore

$$\mathcal{J}_f^{-1}(p) = \mathcal{J}_q^{-1}(p) \cdot \mathcal{J}_v^{-1}(q(p)) \tag{3.22}$$

3.1.3 Proposed Parametrized Controller and HAPFs

At this point we will discuss the form of the control function $h(p, k)$ of Eq. (3.1). In order to understand this formulation, consider the following scalar field on the workspace $\mathcal{W}$ with the parameter $k \in \mathbb{R}^{M+1}$:

$$\phi(p, k) = \psi(p, k) \cdot A \cdot k \tag{3.23}$$

where ψ is the following vector field:

$$\psi(v, k) = \psi(f(p), k) = \\ \left[ln\left(\frac{\|f(p) - \mathcal{V}_0\|}{2}\right), -ln\left(\frac{\|f(p) - \mathcal{V}_1\|}{2}\right), ..., -ln\left(\frac{\|f(p) - \mathcal{V}_M\|}{2}\right) \right] \tag{3.24}$$

The gradient of the above field is the following:

$$\nabla_v \psi(p, k) = \left[\frac{f(p) - \mathcal{V}_0}{\|f(p) - \mathcal{V}_0\|^2}, -\frac{f(p) - \mathcal{V}_1}{\|f(p) - \mathcal{V}_1\|^2}, ..., -\frac{f(p) - \mathcal{V}_M}{\|f(p) - \mathcal{V}_M\|^2} \right] \tag{3.25}$$

Note that this resembles heavily the gradient of the Artificial Potential Field as defined in Section 2.2.2 if the gradient $\nabla_p \phi$ of Eq. (2.13) were to be written in the following form:

$$\nabla_v \phi = \nabla_v \psi(p, k) \cdot \begin{bmatrix} k_d \\ k_1 \\ \vdots \\ k_N \end{bmatrix} \tag{3.26}$$

Now we can understand the formulation of Eq. (3.1) or better yet the formulation of Eq. (3.1) as:

$$\dot{p} = -\mathcal{J}_f^{-1}(p) \cdot \left(\prod_{i=0}^{M} tanh\left(\|f(p) - \mathcal{V}_i\|^2 \right) \right) \cdot \nabla_v \psi(p,k) \cdot A \cdot k \tag{3.27}$$

This is evidently a direct analogue to an Gradient of an Artificial Potential Field, being a scaled version of the latter, with the scaling ensuring that the control input remains bounded.

3.2 An Optimal Solution using the Proposed Parametrized Controllers

3.2.1 Optimal Parameter Vector Form

Let us define the Hamiltonian associated with the adopted value function (3.5) as:

$$H(p,k,\nabla V(p)) = \nabla V(p)^T P(p) k + Q(p;p_0) + W(k) \tag{3.28}$$

Hence, the Bellman optimality equation is formed as follows:

$$\nabla V^*(p)^T P(p) k^* + Q(p;p_0) + W(k^*) = 0 \tag{3.28*}$$

from which the optimal control vector k^* is derived by the first optimality condition $\frac{\partial H(p,k,\nabla V^*)}{\partial k}|_{k=k^*} = 0$, as:

$$\frac{k^*}{\alpha^2(p)} - \frac{1}{k^*} = -\frac{1}{\gamma}(\nabla V^*(p))^T P(p) \tag{3.29}$$

that forms a simple quadratic equation. Solving (3.29) for k^* and keeping only the positive roots to establish safe navigation, we obtain the optimal control vector $k^* = [k_0^*, k_1^*, ..., k_M^*]$ as:

$$k_i^* = \frac{\alpha^2(p)\Gamma_i(p) + \sqrt{(\alpha^2(p)\Gamma_i(p))^2 + 4\alpha^2(p)}}{2}, i = 0, 1, .., M \tag{3.30}$$

where $\Gamma_i(p)$ denotes the i-th element of the vector

$$\Gamma(p) = -\frac{1}{\gamma}(\nabla V^*(p))^T P(p)$$

Furthermore, it is evident that with this formulation all elements k_i^* of this vector are strictly positive by construction:

Proposition 1. *The values for the elements of k given by Eq. 3.30 are strictly positive.*

Proof. Consider the i-th element of the vector k:

$$k_i = \frac{\alpha^2(p)\Gamma_i(p) + \sqrt{\alpha^2(p)\Gamma_i(p)^2 + 4\alpha(p)}}{2}$$

We now assume that $k_i \leq 0$, and we get:

$$\frac{\alpha^2(p)\Gamma_i(p) + \sqrt{\left(\alpha^2(p)\Gamma_i(p)\right)^2 + 4\alpha(p)}}{2} \leq 0 \Rightarrow$$
$$\sqrt{\left(\alpha^2(p)\Gamma_i(p)\right)^2 + 4\alpha(p)} \leq -\alpha^2(p)\Gamma_i(p) \Rightarrow$$
$$\left(\alpha^2(p)\Gamma_i(p)\right)^2 + 4 \leq \left(\alpha^2(p)\Gamma_i(p)\right)^2 \Rightarrow$$
$$4 \leq 0$$

We have obviously reached a contradiction, therefore, $k_i > 0$. □

Additionally, notice that if we had an analytical expression for the value function $V^*(p)$ we would be able to directly compute the optimal control vector k^*. However that is not the case, since in our formulation, one should replace the term k^* from (3.30) in (3.28*) and then solve a non-linear partial differential equation, which is rather hard to solve. Nevertheless, we shall remedy this issue employing, first the successive approximation theory [22] in an offline setting, and then RL to provide an online solution.

3.2.2 Primary Stability Analysis

We will now prove the stability of the system. Consider a system as described in the problem formulation, which obeys the dynamics of the single integrator model with the proposed controller form. For reasons of completeness we provide the aforementioned equation here as well

$$\dot{p} = -\mathcal{J}_f^{-1}(p) \cdot \left(\prod_{i=0}^{M} tanh\left(\|f(p) - \mathcal{V}_i\|^2 \right) \right) \cdot \nabla_v \psi(p,k) \cdot A \cdot k \quad \text{(3.27 revisited)}$$

where it is obvious that the gradient $\nabla\psi$ is

$$\nabla_v \psi(v,k) = \left[\frac{v - \mathcal{V}_0}{\|v - \mathcal{V}_0\|^2}, -\frac{v - \mathcal{V}_1}{\|v - \mathcal{V}_1\|^2}, ..., -\frac{v - \mathcal{V}_M}{\|v - \mathcal{V}_M\|^2} \right] \quad (3.25^*)$$

and the field $\psi(v,k)$:

$$\psi(v,k) = \left[ln\left(\frac{\|v - \mathcal{V}_0\|}{2} \right), -ln\left(\frac{\|v - \mathcal{V}_1\|}{2} \right), ..., -ln\left(\frac{\|v - \mathcal{V}_M\|}{2} \right) \right] \quad (3.24^*)$$

Lemma 1. *The controller (3.1) is stabilizing for $p = p_0 \in \mathcal{W}$*

Proof. Consider what happens as p approaches p_0. First consider first the following form:

$$\lim_{x \to 0} \frac{tanh(x)}{x} = \lim_{x \to 0} \frac{e^{2x} - 1}{x\left(e^{2x} + 1\right)} \quad (3.31)$$

This is a 0/0 form and since both the numerator and denominator are continuous, continuously differentiable functions, we can apply de l'Hospital's rule:

$$\lim_{x \to 0} \frac{tanh(x)}{x} = \frac{2e^{2x}}{e^{2x} + 1 + x(2e^{2x} + 1)} = 1 \quad (3.32)$$

Therefore,

$$\lim_{p \to p_0} \frac{tanh(\|f(p) - \mathcal{V}_0\|^2)}{\|f(p) - \mathcal{V}_0\|^2} \cdot (f(p) - \mathcal{V}_0) = \\ = (f(p_0) - \mathcal{V}_0) = 0 \tag{3.33}$$

It is now evident that because of the form of Eq. (3.1), $h(p_0, k) = 0$. □

Lemma 2. *$\forall p_0 \in \mathcal{W}$, $V(p_0)$ is finite.*

Proof. Since Eq. (2.2) is an infinite-time integral, we need to show that for $p = p_0$ the quantity in the integral is - or tends to - zero. For the term $Q(p; p_0)$ it is evident that $Q(p_0; p_0) = 0$. For the function $W(k)$ note that in Lemma (1) we have proven that $g(p_0) = 0$. Therefore, the Eq. (3.30) takes the form:

$$\frac{k_i^*(p_0)}{\alpha^2(p_0)} - \frac{1}{k_i^*(p_0)} = 0 \tag{3.30*}$$

This equation has a single root at $k_i = \alpha(p_0)$, rendering $W(k(p_0)) = 0$. One can also see that Eq. (3.30*) renders immediately $W(k(p_0))$ zero. Therefore, since $W(k(p_0)) + Q(p; p_0) = 0$, the integral of the Value Function of Eq. (2.2) is finite. □

Remark 1. *From Lemma 1 it is evident that the same holds true for $p \to p_i, i = 1, ..., M$.*

We are now ready to tackle the stability of the system.

Proposition 2. *The control method provided herein assures safety w.r.t. the obstacles and the boundary.*

Proof. We have already proven in Proposition 1 that the parameters k are always positive. Furthermore, considering Lemma 1 it is evident that if the robot approaches an obstacle, then due to the form of the chosen controller, the only direction that will be imposed on the robot is the one that drives it radially away from the obstacle. Generally our method directly inherits the properties of the harmonic potential fields - see [15] - . □

Proposition 3. *Assume a workspace as defined in the problem formulation. The system under the control law (3.1) - or (3.27) - with the components of the parameter vector k following the law of Eq. 3.30 converges asymptotically at p_0 for almost every point in $\mathcal{W}$ except for some subset $\Omega \subset \mathcal{W}$, where Ω is the set of critical points of the vector field of Equation (3.25) other than the desired goal and the obstacles.*

Proof. We propose as a candidate for a Lyapunov function the value function of Eq. (2.2)

$$V(p; p_0) = \int_0^\infty r(p, p_0, k) d\tau = \int_0^\infty [Q(p, p_0) + W(k)] \, d\tau \qquad \text{(2.2 revisited)}$$

- This function is positive $\forall p \in \mathcal{W} - \{p_0\}$ as can be easily shown considering that $Q(p) = \|p - p_0\|^2 > 0 \; \forall p \in \mathcal{W} - p_0$ and $W(k) > 0 \; \forall k_i > 0$, $i = 1, ..., M$, considering Proposition (1), where it is proved that the condition for positive elements of k is true under the proposed parameter tuning law.

- Furthermore, all sublevel sets of V are bounded, as $p \to \infty \Rightarrow \mathcal{V}(p) \to \infty$
- Additionally, $V(p) = 0 \iff p = p_0$.

  This is true, as one can easily see that both functions $Q(p; p_0)$ and $W(k)$ are positive. Therefore for the integral of Eq. (2.2) to be zero, these two terms must both be zero. It is obvious that $Q(p; p_0) = 0 \iff p = p_0$. Furthermore, considering the proof of Lemma 1

$$g(p_0) = 0 \tag{3.34}$$

  and thus, substituting in Eq. (3.29) we reach the conclusion that $W(k(p_0)) = 0$. It is evident that these two functions and therefore the Luyapunov Candidate are zero **iff** $p = p_0$. These first three points mean that our Lyapunov candidate is positive definite.

- Lastly we will show that $\dot{V}$ is negative:

$$\dot{V} = \nabla {V_p}^T \dot{p} = \nabla {V_p}^T P k \tag{3.35}$$

where

$$P = -\mathcal{J}_f^{-1}(p) \cdot \left(\prod_{i=0}^{M} tanh\left(\|f(p) - \mathcal{V}_i\|^2 \right) \right) \cdot \nabla_v \psi(p, k) \cdot A$$

Notice that from Eq. (2.2) we have:

$$\dot{V} = -Q(p, p_0) - W(k) < 0 \; \forall p \in \mathcal{W} - \Omega - \{p_0\}$$

And it is obvious that $\dot{V} = 0 \iff p = p_0$ What has been said thus far proves stability for the set $\mathcal{W} - \Omega$ where Ω is the set of critical points of the vector field of Equation (3.25) other than the desired goal and the obstacles. We are not able to prove anything more on that part, however we will tackle this on the implementation of the off-line approach. □

3.2.3 Successive Approximation of the Value Function

In this section, we prove that a method for successively approximating the Value Function of (2.2) is valid. First, we define an admissible control policy.

Definition 1 (**Admissible Control**). *A control vector $k(p)$ is defined to be admissible with respect to (2.2) on $\mathcal{W}$, denoted by $k(p) \in \Psi(\mathcal{W})$, if $k(p)$ is continuous on $\mathcal{W}$, $k(p)$ stabilizes the system on $\mathcal{W}$ and $V(p)$ is finite for all $p \in \mathcal{W}$.*

It is evident that the proposed parametrized control policies are admissible. Moreover, the Hamilton-Jacobi-Bellman equation is linear w.r.t. the value function, which motivates why we adopt successive approximation. The latter was introduced by ([22]) and later expanded by ([1]) for bounded controls. Nevertheless, we shall further prove the validity of this approach in our case, effectively expanding it for a control vector obeying only lower bounds, through the use of the appropriately selected function $W(k)$ in (3.7). Notice that the successive approximation technique is applied to (3.28) and (3.30). Hence, the following lemma proves how (3.30) can be used to improve the tuning policy for the control vector $k(p)$.

Lemma 3 (**Admissibility of Control**). *If at the j-th step* $k^{(j)} \in \Psi(\mathcal{W})$*, and* $V^{(j)} \in C^1(\mathcal{W})$ *satisfies the equation* $H(p, k^{(j)}, \nabla V^{(j)}) = 0$*, then the new control vector* $k^{(j+1)} = \left[k_0^{(j+1)}, k_1^{(j+1)}, \cdots, k_M^{(j+1)}\right] \in \mathbb{R}^{(M+1)}$*, derived by the solution of the equation is:*

$$k_i^{(j+1)} = \frac{\alpha^2(p)\Gamma_i^{(j)}(p) + \sqrt{\left(\alpha^2(p)\Gamma_i^{(j)}(p)\right)^2 + 4\alpha^2(p)}}{2}, \quad i = 0, \cdots, M \tag{3.36}$$

where $\Gamma_i(p)$ *denotes the i-th element of the vector*

$$\Gamma(p) = -\frac{1}{\gamma}(\nabla V^*(p))^T P(p)$$

is an admissible control vector for (3.1) on $\mathcal{W}$*.*

Proof. To show admissibility, notice that $V^{(j)} \in C^1(\mathcal{W})$ and the fact that the transformation f, its Jacobian as well as the field $g(p)$ are continuous for all $p \in \mathcal{W}$ implies the continuity of $k^{(j+1)}$. Since $V^{(j)}$ is positive definite it attains a minimum at $p_0 \in \mathcal{W}$, and thus,$\nabla V^{(j)}$ should vanish there. It is also easy to see that $u^{(j+1)}(p_0) = 0$. Taking the derivative of $V^{(j)}$ along the system trajectory $\dot{p} = P(p)k^{(j+1)}$ we have:

$$\dot{V}^{(j)}(p, k^{(j+1)}) = \left(\nabla V_p^{(j)}\right)^T P(p) k^{(j+1)} \tag{3.37}$$

Writing the HJB equation for this control yields:

$$H(p, \nabla V^{(j)}) = -\left(\nabla V^{(j)}(p)\right)^T \cdot P(p) \cdot k^{(j)} - Q - W(k^{(j)}) = 0 \tag{3.38}$$

Adding the above expression to Eq. (3.37) yields:

$$\begin{gathered}
\dot{V}^{(j)}(p, k^{(j+1)}) = \\
-\left(\nabla V_p^{(j)}\right)^T P(p) \cdot \left[k^{(j)} - k^{(j+1)}\right] - Q - W(k^{(j)}) = \\
= \gamma \sum_{i=0}^{M} \left(\frac{k_i^{(j+1)}}{\alpha(p)^2} - \frac{1}{k_i^{(j+1)}}\right) \left[k_i^{(j)} - k_i^{(j+1)}\right] - Q - W(k^{(j)}) = \\
\gamma \sum_{j=0}^{M} \left[- \int_{\alpha(p)}^{k_i^{(j)}} \left(\frac{v_i}{\alpha(p)^2} - \frac{1}{v_i}\right) dv_i - \left(\frac{k_i^{(j+1)}}{\alpha(p)^2} - \frac{1}{k_i^{(j+1)}}\right) \left(k_i^{(j+1)} - k_i^{(j)}\right) \right] - Q
\end{gathered}$$

The quantity $-Q$ is always negative, and owing to the mean value theorem:

$$- \int_{\alpha(p)}^{k_i^{(j)}} \left(\frac{v_i}{\alpha(p)^2} - \frac{1}{v_i}\right) dv_i - \left(\frac{k_i^{(j+1)}}{\alpha(p)^2} - \frac{1}{k_i^{(j+1)}}\right) \left(k_i^{(j+1)} - k_i^{(j)}\right) \leq 0$$

this term is also negative. Refer to the appendix for a detailed proof. Therefore $\dot{V}^{(j)}(p, k^{(j+1)}) < 0$ and $V^{(j)}(p)$ is a Lyapunov function for $k^{(j+1)}$ on $\mathcal{W}$. From definition 1 $k^{(j+1)}$ is admissible on $\mathcal{W}$. □

We will now prove that with each iteration, the policy $k^{(j+1)}$ is an improving policy.

Lemma 4 (Control Improvement)**.** *Under the current formulation for the cost function, if $V^{(j+1)}$ is the unique positive-definite function that satisfies the equation $H(V^{(j+1)}, k^{(j+1)}) = 0$, then $V^*(p) \leq V^{(j+1)}(p) \leq V^{(j)}(p)$, $\forall p \in \mathcal{W}$.*

Proof. Along the trajectories of $\dot{p} = P(p) \cdot k^{(j+1)}$, $\forall p \in \mathcal{W}$ we have:

$$\begin{aligned} &V^{(j+1)}(p) - V^{(j)}(p) = \\ &= \int_0^\infty \left[Q\left(p(\tau, p, k^{(j+1)}); p_0\right) + W(k^{(j+1)}(p(\tau, p, k^{(j+1)}))) \right] d\tau \\ &- \int_0^\infty \left[Q\left(p(\tau, p, k^{(j+1)}); p_0\right) + W(k^{(j)}(p(\tau, p, k^{(j+1)}))) \right] d\tau = \\ &= -\int_0^\infty \left(\nabla_p \left(V^{(j+1)} - V^{(j)} \right) \right)^T P(p) \cdot k^{(i+1)} d\tau \end{aligned} \tag{3.39}$$

Now consider the two HJB equations expressed for the two successive approximations:

$$H(p, \nabla V^{(j+1)}) = \left(\nabla V^{(j+1)}(p)\right)^T \cdot P(p) \cdot k^{(j+1)} + Q + W(k^{(j+1)}) = 0$$
$$H(p, \nabla V^{(j)}) = -\left(\nabla V^{(j)}(p)\right)^T \cdot P(p) \cdot k^{(j)} - Q - W(k^{(j)}) = 0$$

Adding these two equations yields:

$$\begin{aligned} &-\left(\nabla V^{(j+1)}(p)\right)^T \cdot P(p) \cdot k^{(j+1)} + W(k^{(j+1)}) + \\ &+\left(\nabla V^{(j)}(p)\right)^T \cdot P(p) \cdot k^{(j)} - W(k^{(j)}) = 0 \end{aligned}$$

and solving for the first term:

$$\begin{aligned} &-\left(\nabla V^{(j+1)}(p)\right)^T \cdot P(p) \cdot k^{(j+1)} = \\ &W(k^{(j+1)}) - W(k^{(j)}) - \left(\nabla V^{(j)}(p)\right)^T \cdot P(p) \cdot k^{(j)} \end{aligned}$$

Substituting the above in Eq. (3.39) yields:

$$\begin{aligned} &V^{(j+1)}(p) - V^{(j)}(p) = \\ &= \int_0^\infty \left[W(k^{(j+1)}) - W(k^{(j)}) - \left(\nabla V^{(j)}(p)\right)^T \cdot P(p) \left(k^{(j)} - k^{(j+1)}\right) \right] d\tau \end{aligned}$$

With some more work:

$$W(k^{(j+1)}) - W(k^{(j)}) = \gamma \sum_{i=0}^{M} \int_{k_i^{(j)}}^{k_i^{(j+1)}} \left(\frac{v_i}{\alpha(p)^2} - \frac{1}{v_i} \right) dv$$

and

$$-\left(\nabla V^{(j)}(p)\right)^T \cdot P(p) \left(k^{(j)} - k^{(j+1)}\right) = -\gamma \sum_{i=0}^{M} \left(\frac{k_i^{(j+1)}}{\alpha(p)^2} - \frac{1}{k_i^{(j+1)}} \right) \left(k_i^{(j+1)} - k_i^{(j)}\right)$$

Putting it all together we get:

$$V^{(j+1)}(p) - V^{(j)}(p) =$$
$$= \gamma \sum_{i=0}^{M} \int_{k_i^{(j)}}^{k_i^{(j+1)}} \left(\frac{v_i}{\alpha(p)^2} - \frac{1}{v_i} \right) dv - \sum_{i=0}^{M} \left(\frac{k_i^{(j+1)}}{\alpha(p)^2} - \frac{1}{k_i^{(j+1)}} \right) \left(k_i^{(j+1)} - k_i^{(j)} \right) =$$
$$= \gamma \sum_{i=0}^{M} \left[\int_{k_i^{(j)}}^{k_i^{(j+1)}} \left(\frac{v_i}{\alpha(p)^2} - \frac{1}{v_i} \right) dv - \left(\frac{k_i^{(j+1)}}{\alpha(p)^2} - \frac{1}{k_i^{(j+1)}} \right) \left(k_i^{(j+1)} - k_j^{(j)} \right) \right]$$

Due to the fact that the function $f(x) = \frac{x}{\alpha} - \frac{1}{x}$ is monotonically increasing $\forall x \in \mathbb{R}^+$ and from the geometrical meaning of the last expression we conclude that:

$$V^{(j+1)}(p) - V^{(j)}(p) \leq 0$$

Refer to the appendix for a detailed proof. For the second part of the inequality, lets assume that there exists a value function after some number of iterations that has the property $V^*(p) \geq V^{(j+1)}(p)$. Since by definition $V^*(p)$ is the optimal value function for the optimal control k^*, if there existed a value function field that resulted in reduced cost for the system, then the initial cost function and the related control would not be the optimal. We conclude that this approximation procedure **has** to be bounded by $V^*(p)$ and therefore we conclude that:

$$V^*(p) \leq V^{(j+1)}(p) \leq V^{(j)}(p) \ , \forall p \in \mathcal{W} \tag{3.40}$$

□

3.2.4 Stability Analysis for the whole workspace

Now we are ready to tackle the full proof for stability.

Proposition 4. *Assume a workspace as defined in the problem formulation. The system under the control law (3.1) - or (3.27) - with the components of the parameter vector k starting from a value $k_i^0 > 0, i = 1, ..., M$ and following the adaptive law of (3.30) converges asymptotically at p_0 for almost every point in $\mathcal{W}$ except for some subset $\Omega \subset \mathcal{W}$, where Ω is the set of critical points of the vector field of Equation (3.25) other than the desired goal and the obstacles. Following the successive approximation starting from a set of initial parameters k such that the control has the properties of an harmonic field, the set Ω consists of zero or one-dimensional sets that are saddles, and therefore the only stability point is the desired orientation.*

Proof. Consider the approach described in the present section. In order to complete the proof of Proposition (3) we need to prove that the subset Ω contains at most one-dimensional lines. Starting from a constant set of k parameters, observe that the control scheme consists of a harmonic field suitable for navigation - scaled by the factor of the multiple of $tanh()$ terms - . This means that the set Ω will initially contain only lines of points. Now consider that we have proved that with each iteration of the approximation the value of the value function decreases or stays the same $\forall p \in \mathcal{W}$ - see Lemma (4). We will now assume that the initial lines or points of the set Ω "shift" in $\mathcal{W}$. Noting that the cost function is infinite at those points - since staying at these points would

render the infinite-time integral of Eq. (2.2) infinite - this means that a point that previously had a bounded value now attains an infinite value. However we have reached a contradiction since we have proven that the value function at each point from one iteration to another must decrease or stay constant. We conclude that the zero or one-dimensional subsets of Ω stay this way during our approximation process. □

3.3 Neural Network Approximate HJB Solution

A it has been stated numerous times in the previous chapters, the nonlinear differential equation that was introduced by substituting Eq. (3.30) to Eq. (3.28*) is very hard to solve, and solutions are not guaranteed. In order however to implement what has been discussed so far, we will to propose a method to suitably approximate the value function using a Neural Network and the successive approximation method proposed in ([1]) that was further expanded upon here. This approximation will be in the sense of least squares approximation, and will be based on the Hamilton Jacobi Bellman equation as presented in ([1]). This will provide a fast, computationally practical and efficient approach for finding nearly optimal solutions to the motion planning problem.

3.3.1 Approximation of $V(p)$ using NN

Neural Networks have long been used to approximate smooth functions on prescribed compact sets ([13]). In our case $V^{(i)}(p)$ is approximated as

$$V_L^{(i)}(p) = \sum_{j=1}^{L} w_j^i \phi j(p) = (\boldsymbol{w}_L^{(i)})^T \cdot \boldsymbol{\phi}_L(p) \tag{3.41}$$

which effectively is a single-hidden-layer neural network with L newrons - activation functions - $\sigma_j(p) \in C^1(\mathcal{W})$ and w_j^i the weights of the activation functions. This is also written in vector notation. The weights are tuned in order to minimize the error of the approximation - in a least squares sense - over a number of samples taken on the workspace $\mathcal{W}$. In order now to prove, for this approximation, convergence in the mean, existence of the approximation in the least-squares sense and uniqueness of the approximation as well as admissibility of k_L^{i+1} we refer the reader to ([1]) as the result of Khalaf et. al can directly be applied here as well. The equations for the approximation of the HJB are the following

$$\begin{aligned} w_L = -\Big\langle \nabla\phi_L(p)P(p)k, \nabla\phi_L(p)P(p)k \Big\rangle^{-1} \cdot \\ \Big\langle Q + W, \nabla\phi_L(p)P(p)k \Big\rangle \end{aligned} \tag{3.42}$$

where $\langle f, g \rangle = \int_{\mathcal{W}} fgdp$ a Lebesgue integral, and as stated in ([1]), if we choose the basis functions of the NN to be linearly independent, then

$$\Big\langle \nabla\phi_L(p)P(p)k, \nabla\phi_L(p)P(p)k \Big\rangle$$

is full rank and therefore invertible. Having solved for $\boldsymbol{w}_L$, the improved control vector $k = [k_0, k_1, ..., k_M]$ is given by:

$$k_i = \frac{\alpha^2(p)\Gamma_i(p) + \sqrt{(\alpha^2(p)\Gamma_i(p))^2 + 4\alpha^2(p)}}{2}, i = 0, 1, .., M \tag{3.43}$$

where $\Gamma_i(p)$ denotes the i-th element of the vector

$$\Gamma(p) = -\frac{1}{\gamma} w_L^T \nabla\phi_L(p) P(p)$$

It is evident that a relevant form is implemented for the other formulation of the W term in the optimization function, that is the analogous to Eq. (**??**). Calculating w_L from Eq. (3.42) directly however is expensive computationally. We will resort to introducing sampling on our workspace $\mathcal{W}$ so as to calculate the weights of the approximation of the value function. The terms of (3.42) can be rewritten as

$$\begin{aligned} X = \Big\lfloor \nabla\phi_L(p) P(p) k \Big|_{p_1} \cdots \\ \cdots \nabla\phi_L(p) P(p) k \Big|_{p_N} \Big\rfloor^T \end{aligned} \tag{3.44}$$

$$Y = \Big\lfloor Q + W \Big|_{p_1} \cdots Q + W \Big|_{p_N} \Big\rfloor \tag{3.45}$$

where the N in p_N symbolizes the number of samples in $\mathcal{W}$. It is proved in ([1]) that the above imply that the weights $\boldsymbol{w}_L$ can be calculated by

$$w_{L,N} = -\left(X^T X\right)\left(X^T Y\right) \tag{3.46}$$

In ([1]) Monte Carlo techniques are proposed to calculate the integral of (3.42). We will propose our own sampling techniques in this work. Rather than using Eq. (3.46) directly, more computationally steady methods are used.

3.3.2 Choice of Basis Functions

A descent approximation of the cost function necessitates choosing a suitable for the application set of basis functions. These functions should be linearly independent and, in order to satisfy certain conditions later during the on-line implementation, we choose radial basis functions. More specifically, we choose a two dimensional Gaussian bell function $\sigma(p) : \mathbb{R}^2 \to \mathbb{R}$. Note that the domain of the σ function is not necessarily limited to the workspace $\mathcal{W}$. It is simply imperative that the linear combination of the basis functions should have good approximation capabilities on this set. For example, we may choose - as we will - to place some Gaussian functions **outside** the border of $\mathcal{W}$ and inside the boundaries of the obstacles, in order to improve the neural network's approximation capabilities. The basis functions are given by the formula

$$\begin{aligned} \phi_i(p) = e^{-\left(\frac{x-x_i}{\mu_{x,i}}\right)^2 - \left(\frac{y-y_i}{\mu_{y,i}}\right)^2} \\ i = 1, ..., L \ , \ p = [x, y]^T \in \mathcal{W}, \ p_i = [x_i, y_i]^T \in \mathcal{S} \subset \mathbb{R}^2 \end{aligned} \tag{3.47}$$

where $p_i \ i = 1, ..., L$ the centre of the i_{th} basis function and $\mu_{x,i}, \mu_{y,i}$ are the standard deviations of the Gaussian distribution around the two axis x, y respectively. We have to choose, first of all, the set $\mathcal{S} \subset \mathbb{R}^2$ in which we will place the functions, a proper distribution of the basis functions on $\mathcal{S}$, and the parameters $\mu_{x,i}, \mu_{y,i}$ of the distribution. Herein, we choose the following:

- We choose $\mathcal{S} \subset \mathbb{R}^2$ such that one or two basis functions lie outside the boundary of the workspace $\mathcal{W}$ (heuristically).

- We choose a Cartesian uniform distribution of points $p_i \in \mathcal{S}$.
- We choose constant values $\mu_{x,i} = \mu_x, \mu_{y,i} = \mu_y$, $i = 1, ..., L$ which comes directly from the above uniform distribution of points $p_i \in \mathcal{S}$ according to a desired overlapping percentage $\lambda \in (0, 1)$.

This value is chosen equal to:

$$\mu_x = \mu_y = \sqrt{\frac{-\Delta x^2}{2ln(\lambda)}} \tag{3.48}$$

Where $\Delta x = \Delta y$ is the constant distance between to adjacent points of the uniform distribution of basis function centres in $\mathcal{S}$.

Non Uniform Distribution of BF centres

We can also define the above parameters of the BFs w.r.t. a non-uniform distribution. Mainly, the values $\mu_{x,i}, \mu_{y,i}$ can be chosen as

$$\begin{aligned} \mu_{x,i} &= \sqrt{\frac{-\Delta x_i^2}{2ln(\lambda)}} \\ \mu_{y,i} &= \sqrt{\frac{-\Delta y_i^2}{2ln(\lambda)}} \end{aligned} \tag{3.49}$$

where for the distance Δx_i many methods can be considered, e.g. taking the mean value of $K << L$ distances from the K nearest neighbours of the i_{th} BF. The form of this BF is given in figure 3.1.

3.3.3 Sampling Techniques

In this section the sampling techniques used are presented. Since the navigation theory used here is based on harmonic potentials it would be beneficial if the sampling on the workspace $\mathcal{W}$ resulted in a uniform distribution of points on the harmonic space consisting of the unit disk $\mathcal{D}$.

Probability Density Based Sampling Technique

Let's consider the uniform distribution on the disk, that relates to a constant density function g. The probability that any point q lies within the disk should be equal to 1, therefore

$$\begin{aligned} P(q \in \mathcal{D}) = \iint_{A_\mathcal{D}} g(q) dA = 1 \Rightarrow \\ g(q) A_\mathcal{D} = 1 \Rightarrow \\ g(q) = \frac{1}{A_\mathcal{D}} = \frac{1}{\pi} \end{aligned} \tag{3.50}$$

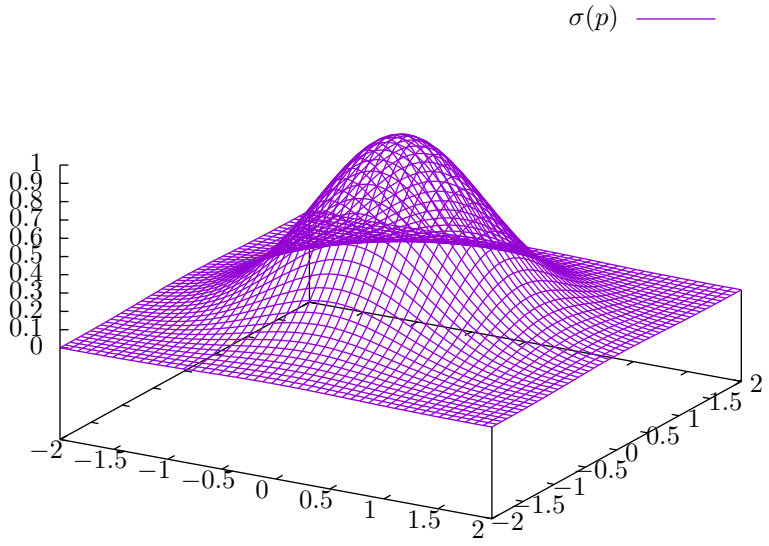


Figure 3.1: Basis Function

Now we need to calculate the accumulative probability function of a random point $q \in \mathcal{D}$. Consider

$$P(q = [x, y]^T) = \frac{1}{\pi} \iint_{A(x,y)} dA = \frac{1}{\pi} \iint_{(0,0)}^{[\rho,\theta]} r dr du \Rightarrow$$
$$P(x, y) = \frac{1}{\pi} \left[\frac{1}{2} r^2 u \right]_{[0,0]}^{[\rho,\theta]} = \frac{1}{2\pi} \rho(x, y)\theta(x, y) \Rightarrow$$
$$P(x, y) = \frac{1}{2\pi} \left(x^2 + y^2\right) Atan2(y, x) =$$
$$= \frac{1}{2\pi} \|q\|^2 Atan2\left(q(p) \cdot e_2, q(p) \cdot e_1\right)$$

where e_1,e_2 the cartesian basis vectors. We can express this accumulative probability function with respect to a point on the workspace $\mathcal{W}$, using the respective transformation.

$$P(x, y) = \frac{1}{2\pi} \|q(p)\|^2 Atan2\left(q(p) \cdot e_2, q(p) \cdot e_1\right) \tag{3.51}$$

Now, to obtain a point that follows the above distribution, we propose the following algorithm. Observe that $P(p) \in [0, 1]$. For N sample points in the

workspace $\mathcal{W}$:

Algorithm 1: Probability Density Based Sampling Technique

initialization;
while $i < N$ **do**
- Produce a uniformly distributed random number $r \in [0, 1]$
- Solve for p the following equation

$$P(x, y) - r = 0 \Rightarrow$$
$$\frac{1}{2\pi}\|q(p)\|^2 Atan2\left(q(p) \cdot e_2, q(p) \cdot e_1\right) - r = 0$$

$i \leftarrow i + 1$
end

Distribution Based Sampling Technique

We would also like to propose a second sampling technique that is based on an already known distribution of points on the disk space. For this purpose, consider the well-known "Sunflower Seed" distribution on a disk of singular radius. The distribution in polar coordinates is the following:

$$r = \sqrt{n}N$$
$$\theta = \frac{2\pi}{\phi^2}n$$

where$N$ the total number of points, $n = 0, 1, \ldots, N$ a sequence that produces the N points and finally ϕ is the Golden ratio. In cartesian coordinates the vector $q \in \mathcal{D}$ is equal to:

$$q = \begin{bmatrix} \sqrt{\frac{n}{N}} cos\left(\frac{2\pi}{\phi^2}n\right) \\ \sqrt{\frac{n}{N}} sin\left(\frac{2\pi}{\phi^2}n\right) \end{bmatrix}$$

In order to find the respective points p, remember that $q = f(p)$.Therefore, we need to solve the equation – given that an explicit or computational form of $p = f^{-1}(q)$ is not available-

$$r(p) = f(p) - q(n) = 0,\ for\ n = 0, 1, ...N$$

This equation can be very hard to find a solution to. However, the problem might be easier to solve if it is considered as an optimization problem

$$p^* = \arg\min_p \|r(p)\|^2$$

It is obvious that the above process can be implemented for any known distribution of points on the disk space $\mathcal{D}$. In the application of the present work, a stochastically enhanced sequential quadratic programming – SQP – method was used to ensure the validity of the sampled points.

3.4 Algorithm for the solution of the Optimal Motion Planning Problem using the Proposed Parametrized Controllers

Having presented all the necessary proofs for the application of on off-line procedure for the solution of the optimal motion planning problem, the algorithmic process to be implemented is the following:

Algorithm 2: Algorithm for the Neural Network Approximation of the Value Function

• **Sampling**;
Select N points $p_i, i = 1, \cdots, N$ within the workspace $\mathcal{W}$.
• **Initialize**;
Select an initial control vector $k(0) = [1, 1, ..., 1]^T \in \mathbb{R}^{(M+1)}$, which is an admissible policy.
while *Weights have not Converged* **do**
• **Weights Improvement Step:** Solve the following linear regression problem:

$$(w^{(j)})^T X = -Y$$

where

$$X = \left[\nabla\phi(p)P(p)k^{(j)}\Big|_{p_1}, \cdots, \cdots, \nabla\phi(p)P(p)k^{(j)}\Big|_{p_N}\right]^T$$

and

$$Y = \left[Q(p;p_0) + W(k^{(j)})\Big|_{p_1}, \cdots, \cdots, Q(p;p_0) + W(k^{(j)})\Big|_{p_N}\right]^T$$

• **Policy Improvement Step:** Update the control vector $k^{(j+1)} = \left[k_0^{(j+1)}, k_1^{(j+1)}, \cdots, k_M^{(j+1)}\right] \in \mathbb{R}^{(M+1)}$:

$$k_i^{(j+1)} = \frac{\alpha^2(p)\Gamma_i^{(j)}(p) + \sqrt{\left(\alpha^2(p)\Gamma_i^{(j)}(p)\right)^2 + 4\alpha^2(p)}}{2}, \quad i = 0, \cdots, M$$

where $\Gamma_i^{(j)}$ the i-th element of the vector

$$\Gamma^{(j)} = -\frac{1}{\gamma}(w^{(j)})^T\nabla\phi(p)P(p)$$

$$j \leftarrow j + 1$$

end
Upon convergence set the control law of the system as follows:

$$u = P(p) \cdot k^{(j)}$$

Chapter 4

On-line Solution using Parametrized Controllers

4.1 On-line Implementation of the Parametrized Controllers

We provide an online approach to tackle the optimal motion planning problem, in order to optimize the path of the robot for a given starting-ending point pair. Reinforcement Learning will be applied in the form of an actor structure in order to minimize the HJB error, thus approximating the value function of the optimization problem. Employing the approximation capabilities of NN, the unknown value function may be modelled as:

$$V(p) = \sum_{i=1}^{L} w_i \phi_i(p) + \epsilon(p) = w^T \cdot \phi(p) + \epsilon(p)$$

where $w \triangleq [w_1, \cdots, w_L]^T \in \mathbb{R}^L$, $\phi(p) \triangleq [\phi_1(p), \cdots, \phi_L(p)] \in \mathbb{R}^L$, and $\epsilon(p)$ denote the optimal weights that minimize the modelling error $\epsilon(p)$ over the workspace $\mathcal{W}$ for a given regressor vector $\phi(p)$. Following the optimality condition, the optimal control vector is given by:

$$\begin{aligned} k(w) = &-\frac{\alpha^2(p)}{2\gamma} P^T(p) \nabla \phi^T(p) w + \\ &+ \sqrt{\left(\frac{\alpha^2(p)}{16\gamma} P(p)^T \nabla \phi^T(p) w\right)^2 + \alpha^2(p)} \end{aligned} \tag{4.1}$$

In the online approach, the estimation $\hat{w}$ of the unknown ideal w will be provided by a gradient scheme that aims at minimizing the error in the HJB equation:

$$e(\hat{w}) = \hat{w}^T \nabla \phi(p) P(p) k(\hat{w}) + Q(p; p_0) + W(k(\hat{w})) \tag{4.2}$$

where $k(\hat{w})$ denotes the estimation of the control vector provided by (4.1) based on the estimation of the NN weights. Hence, we formulate the tuning law for the NN weight estimates to minimize the cost function:

$$E = \frac{1}{2} e^T(\hat{w}) e(\hat{w})$$

In particular, a normalized gradient estimation scheme is adopted as follows:

$$\dot{\hat{w}} = -a\frac{\sigma_2}{m_s}\left[\hat{w}^T\nabla\phi(p)P(p)k(\hat{w}) + Q(p;p_0) + W(k(\hat{w}))\right] \tag{4.3}$$

with $a > 0$, where

$$\begin{aligned}\sigma_2 \triangleq \tfrac{\partial e(\hat{w})}{\partial \hat{w}} &= \nabla\phi(p)P(p)k(\hat{w}) + \\ &+\tfrac{\alpha^2(p)}{2\gamma}\left[\hat{w}^T\nabla\phi(p)P(p) + \gamma\left(\tfrac{k(\hat{w})}{\alpha^2(p)} - \tfrac{1}{k(\hat{w})}\right)\right]\times \\ \times\left[\frac{\left(\hat{w}^T\nabla\phi(p)P(p)\right)}{\sqrt{\left(\frac{\alpha^2(p)}{\gamma}\hat{w}^T\nabla\phi(p)P(p)\right)^2 + 4\alpha^2(p)}} - 1\right]&\left(\nabla\phi(p)P(p)\right)^T\end{aligned}$$

and $m_s = (\sigma_2^T\sigma_2 + 1)^2$.

Theorem 5. *The closed loop system $\dot{p} = P(p)\cdot k(\hat{w})$ with the adaptive law (4.3) guarantees that the trajectory for almost any initial position in the workspace converges safely to the desired position p_0.*

Proof. We adopt the Lyapunov candidate function:

$$L(p,\hat{w}) = V(p) + \frac{1}{2}\tilde{w}^T a^{-1}\tilde{w} \tag{4.4}$$

where $V(p)$ is the unknown value function and $\tilde{w} = w - \hat{w}$ denotes the parametric error. It is easy to see that the above is always positive except for $p = p_0$ and $\tilde{w} = 0$. Now consider the dynamics of the weight estimation (4.3) in the following compact form:

$$\dot{\hat{w}} = -a\frac{\sigma_2}{m_s}e(\hat{w}) \tag{4.5}$$

Notice that the error in (4.2) can be written via a Taylor series expansion around $\hat{w}$ as follows:

$$e = -\sigma_2^T\tilde{w} + e_1 \tag{4.6}$$

where e_1 denotes the effect of the higher order terms. Hence, we may write:

$$\dot{\tilde{w}} = \alpha\frac{\sigma_2}{m_s}e = -\frac{\alpha}{m_s}\sigma_2\sigma_2^T\tilde{w} + \frac{ae_1}{m_s}\sigma_2 \tag{4.7}$$

which leads to:

$$\dot{L} = \nabla V^T(p)P(p)k(\hat{w}) - \left[\frac{1}{m_s}\tilde{w}^T\sigma_2\sigma_2^T\tilde{w}\right] + \left[\frac{e_1}{m_s}\tilde{w}^T\sigma_2\right]$$

Adding and subtracting $w^T\nabla\phi(p)P(p)k(w)$ and invoking the Hamilton-Jacobi-Bellman equation, we obtain:

$$\begin{aligned}\dot{L} \leq &-\left[\frac{1}{m_s}\tilde{w}^T\sigma_2\sigma_2^T\tilde{w}\right] + \left[\frac{e_1}{m_s}\tilde{w}^T\sigma_2\right] \\ &+w^T\nabla\phi(p)P(p)\tilde{K}(\tilde{w})\tilde{w} - Q(p;p_0) - W(k(\hat{w})) + \epsilon\end{aligned}$$

where $\tilde{K}(\tilde{w}) = \frac{dk(w)}{dw}|_{w=\tilde{w}}$ and ϵ involves all modelling error terms. Hence, we conclude:

$$\dot{L} \leq -Q(p;p_0) - W(k(\hat{w})) -$$
$$-\frac{\|\sigma_2\sigma_2^T\|}{m_s}|\tilde{w}|^2 + \left[B + \frac{e_1}{m_s}\left|\sigma_2^T\right|\right]|\tilde{w}| + \epsilon$$

Notice that, assuming persistently excited neurons, the above expression provides essentially a lower bound to $\|\tilde{w}\|$ for which the Lyapunov candidate is negative, which provides convergence as shown in [19]. □

The above controller can be summed up in the following block diagram:

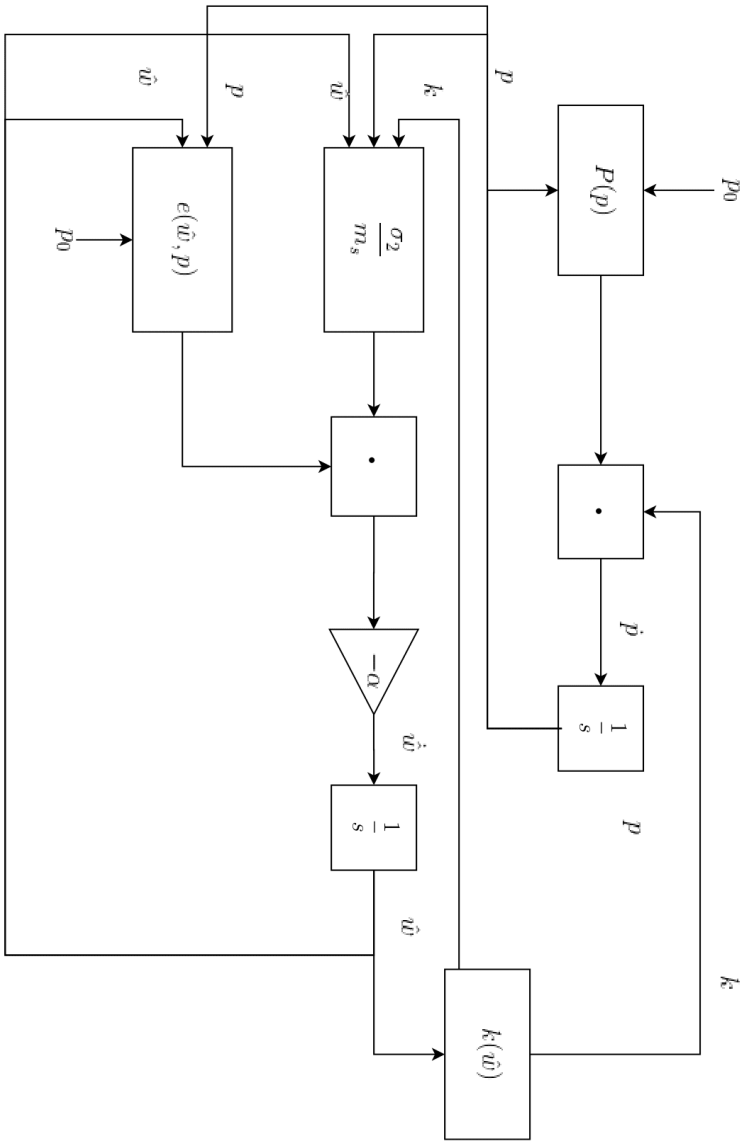


Figure 4.1: Block Diagram of the On-line Controller

Chapter 5

Simulation Results

In this section we will present the results of the offline solution, followed by a comparison between the online approach and an RRT* method. For the proposed algorithm, a grid of 15×15 neurons, consisting of Radial Basis Functions (RBFs), were used. All simulations were implemented with Matlab on a PC running Windows 10, on an intel-i7 quad-core processor. For the RTT* approach, a traditional quadratic form for the input part of the cost function was used. An artificial workspace was designed, with a square outer boundary of side lengths equal to 10[m], and three inner disk obstacles, as presented in Fig. 5.2. The goal position was $p_0 = (1, 1)$ for all runs. In Fig. 5.1, we illustrate the approximation of the value function and the respective vector field that resulted from the successive value function approximation of Algorithm 2. The approximation exhibits the desired behaviour, with large values away from the minimum at the goal position. In Fig. 5.2 we present four trajectories that resulted from various starting points, along with the same trajectories for the RRT* method. In Fig. 5.3, we present the respective tree graphs for each trajectory of the RRT* method. Finally, Table 5.1 includes the results for every run, including the start-end point configurations, the computed cost for each method and the corresponding run time. It is evident that our method consistently outperforms the RRT* optimization method, both in cost function value, and in run time. Additionally, our method produces smooth trajectories. The offline method outperforms the online one as expected, however, the trajectories of the online approach tend to match the offline ones as time progresses and the learning process evolves towards the optimal parameter estimates. Finally, all of the aforementioned trajectories exhibit both safety and convergence, as it has been rigorously proven.

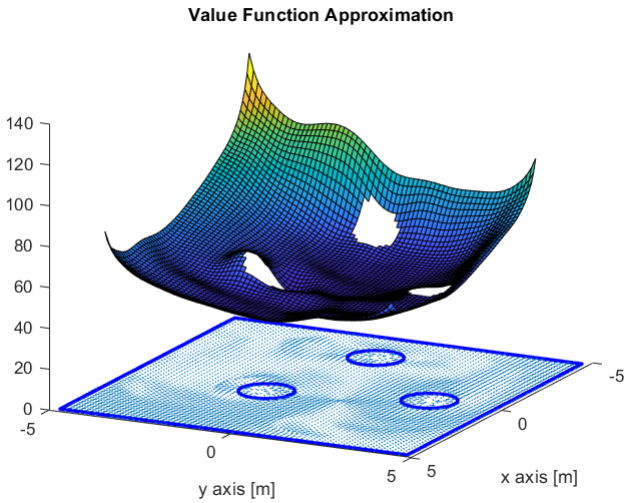


Figure 5.1: The offline vector field and value function approximation.

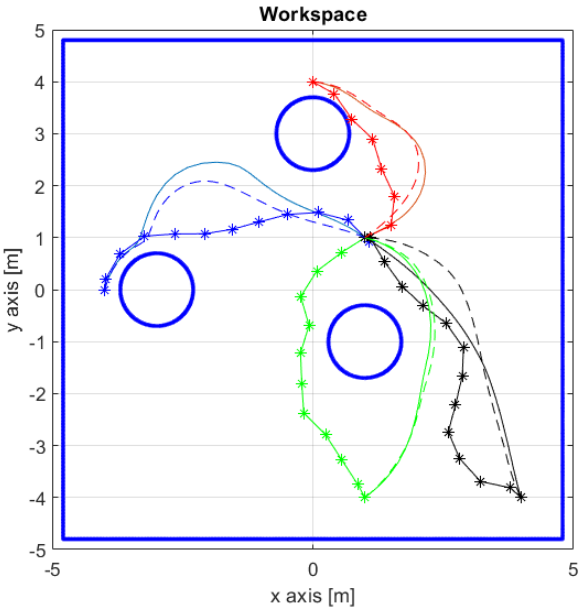


Figure 5.2: The online trajectories (solid lines), the offline trajectories (dashed lines) and the RRT* trajectories (star points).

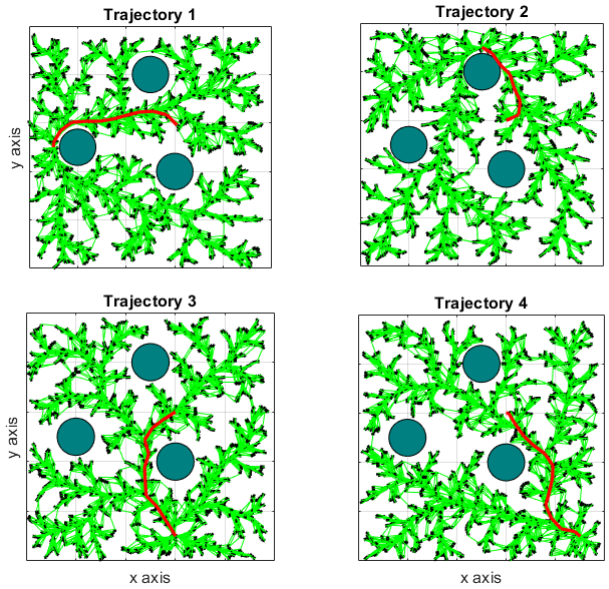


Figure 5.3: The RRT* graphs and trajectories.

Traj. #	Start Pos.[m]	Goal Pos.[m]	Cost Online	Cost RTT	Run T. Online [s]	Run T. RRT [s]
1	(-4,0)	(1,1)	576	830	440	601
2	(0,4)	(1,1)	177	361	192	718
3	(1,-4)	(1,1)	346	827	395	640
4	(4,-4)	(1,1)	770	988	435	612

Table 5.1: Comparative Simulation Results

Chapter 6

Discussion & Future Research

The results of this work are both promising and intriguing. They provide a provably correct solution to the optimal motion planning problem, with both safety and convergence assured by the structure of the parametrized controller. The method both improves the value of a cost function for a given initial policy and consistently outperforms another approach, namely an RRT-Star method. Furthermore, the deterministic nature of the proposed method is an added advantage.

As for future directions, the method will be expanded with more general controller structures, so as to improve not only the computational characteristics, but also the span of the controller basis. The need for an expensive transformation will be negated and a better, with respect to the given value function, control policy can be adopted. Furthermore, concerning the energy term for the value function, a form with physical meaning -e.g. energy input minimization term- can be adopted. Finally, a more general form can be used to extend the present work to unknown workspaces. The above will be implemented using harmonic series approximations for bounded, fully known workspaces, while unknown workspaces will be tackled with point sources in a sequentially discovered unknown workspace.

Chapter 7

Appendix

7.1 Proofs

7.1.1 Proof for Lemma 3

Lemma 6. *The following is true:*

$$-\int_1^{k_j^{(i)}} \left(v_i - \frac{1}{v_i}\right) dv_i - \left(k_j^{(i+1)} - \frac{1}{k_j^{(i+1)}}\right)\left(k_j^{(i+1)} - k_j^{(i)}\right) \leq 0$$

Proof. Consider the function $f(x) = \int_1^x \left(v_i - \frac{1}{v_i}\right) dv_i$ and its derivative $f'(x) = x - \frac{1}{x}$. The expression to be proven becomes, considering the points $a = k_j^{(i)}$, $b = k_j^{(i+1)}$:

$$-f(a) - f'(b)(b-a) \leq 0$$

Further consider that $f(x) \geq 0$, $\forall x \in \mathbb{R}^+$. We now begin the proof. • For $a > b$, we need to prove that $\frac{f(a)}{a-b} \geq f'(b)$.

We have:

$$\frac{f(a)}{a-b} \geq f'(b) \iff$$
$$\frac{f(a)}{a-b} \geq \frac{f(a)-f(b)}{a-b} \geq f'(b)$$

This is easily proven as :

$$\exists\ x_0 \in [b,a] : f'(x_0) = \frac{f(a)-f(b)}{a-b}$$

Therefore we need to prove that $f'(x_0) \geq f'(b)$, which is easily found to be true since it is evident that $f'(x)$ is monotonically increasing $\forall x \in \mathbb{R}^+$. • For $b > a$, we need to prove that $\frac{f(a)}{a-b} \leq f'(b)$. The proof is in the same logic as the previous one and is therefore omitted here. Since $\forall a, b \in \mathbb{R}^+$ we have proven that $-f(a) - f'(b)(b-a) \leq 0$, the proof is complete. □

7.1.2 Proof for Lemma 4

Lemma 7. *The following it true:*

$$\int_{k_j^{(i)}}^{k_j^{(i+1)}} \left(v_j - \frac{1}{v_j}\right) dv - \left(k_j^{(i+1)} - \frac{1}{k_j^{(i+1)}}\right)\left(k_j^{(i+1)} - k_j^{(i)}\right) \leq 0$$

Proof. Consider the function $f(x) = \int_1^x \left(v_i - \frac{1}{v_i}\right) dv_i$ and its derivative $f'(x) = x - \frac{1}{x}$. The expression to be proven becomes, considering the points $a = k_j^{(i)}$, $b = k_j^{(i+1)}$:

$$f(b) - f(a) \leq f'(b)(b-a)$$

• For $a < b$, we need to prove that $\frac{f(b)-f(a)}{b-a} \leq f'(b)$. This is easy as:

$$\exists\ x_0 \in [a,b] : f'(x_0) = \frac{f(b)-f(a)}{b-a}$$

And therefore we need to prove that:

$$f'(x_0) \leq f'(b)$$

Which is easily found to be true since $f'(x)$ is monotonically increasing $\forall x \in \mathbb{R}^+$.
• For $a > b$, we need to prove that $\frac{f(a)-f(b)}{a-b} \geq f'(b)$. The proof is the same as before and is therefore omitted here. Since $\forall a, b \in \mathbb{R}^+$ we have proven that $f(b) - f(a) \leq f'(b)(b-a)$, the proof is complete. □

7.1.3 Proof for Basis Functions

In this section we will prove the form of Eq. 3.48. Consider two Basis Functions of a set of uniformely distributed functions, the distance between the two given by Δx in the x direction and Δy in the y direction. Then the two functions will have the following form - one is considered as centred at $[0, 0]$.

$$\sigma_1(p) = e^{-\left(\frac{x}{\mu_x}\right)^2 - \left(\frac{y}{\mu_y}\right)^2}$$
$$\sigma_2(p) = e^{-\left(\frac{x-\Delta x}{\mu_x}\right)^2 - \left(\frac{y-\Delta y}{\mu_y}\right)^2}$$

Consider that these two have the same form - they come from a simple translation. We will now consider that we would like for the two functions to have a specific point of overlap λ as a specific point of intersection at the diagonal line that passes through the two centres of the functions. Therefore at the point $x = \frac{\Delta x}{2}$ and $y = \frac{\Delta y}{2}$

$$\begin{gathered}\sigma_1 = \sigma_2 = \\ = e^{-\left(\frac{\Delta x}{2\mu_x}\right)^2 - \left(\frac{\Delta y}{2\mu_y}\right)^2} = \lambda\end{gathered} \tag{7.1}$$

If we consider there to be axisymmetric, $\mu_x = \mu_y = \mu$

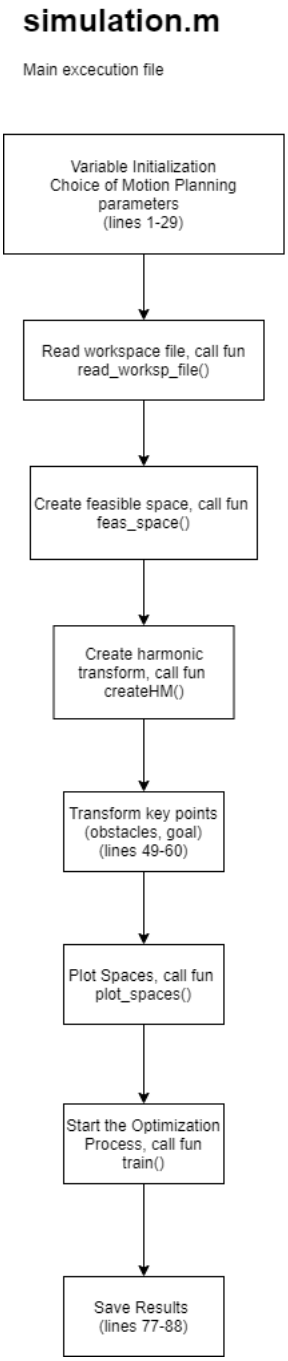
$$\left(\frac{\Delta x}{2\mu_x}\right)^2 + \left(\frac{\Delta y}{2\mu_y}\right)^2 = -ln(\lambda) \Rightarrow$$
$$\frac{1}{2}\left(\frac{\Delta x}{\mu}\right)^2 = -ln(\lambda) \Rightarrow$$
$$\mu = \sqrt{\frac{-\Delta x^2}{2ln(\lambda)}}$$

Note that the above choice of μ values worked better for the online approach, whereas a choice of $\mu = \sqrt{\frac{-\Delta x^2}{4ln(\lambda)}}$ seemed to work better for the offline approach - for the same overlapping parameter λ -. This change relates to choosing a different point of intersection for setting the overlapping equal to the desired value. The first refers to the point being between "diagonal" neurons and the second being between neurons lying on the same vertical/horizontal axis.

7.2 Software Structures

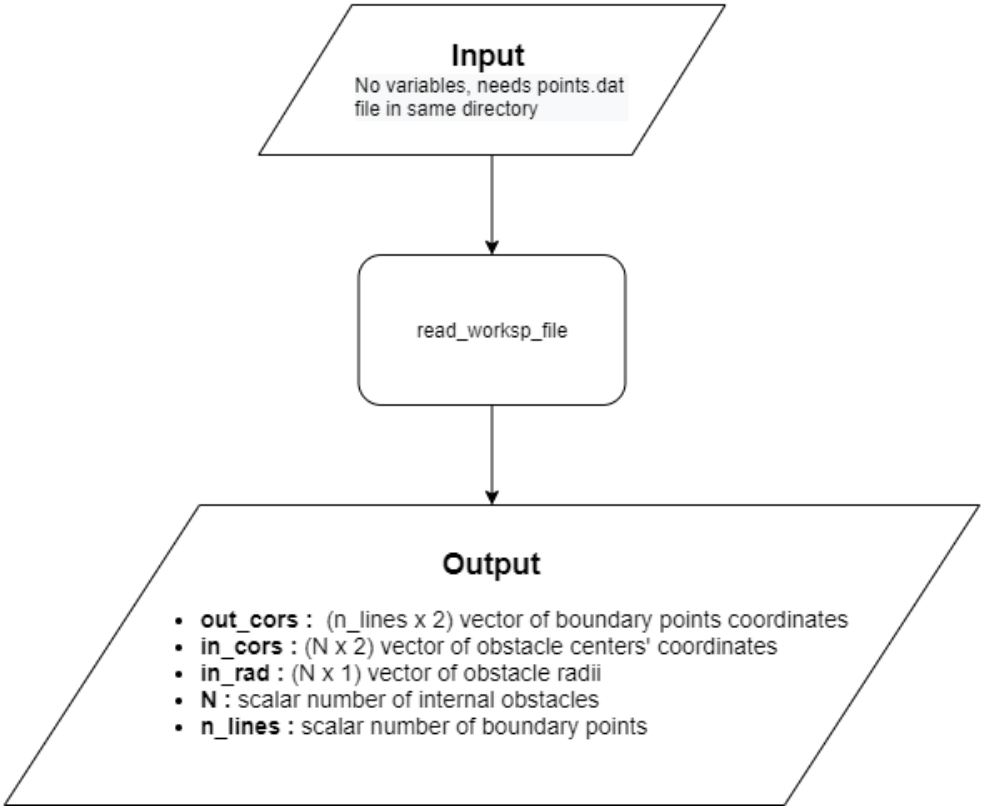
7.2.1 Off-line method Software

Herein the software structures for each method will be presented.



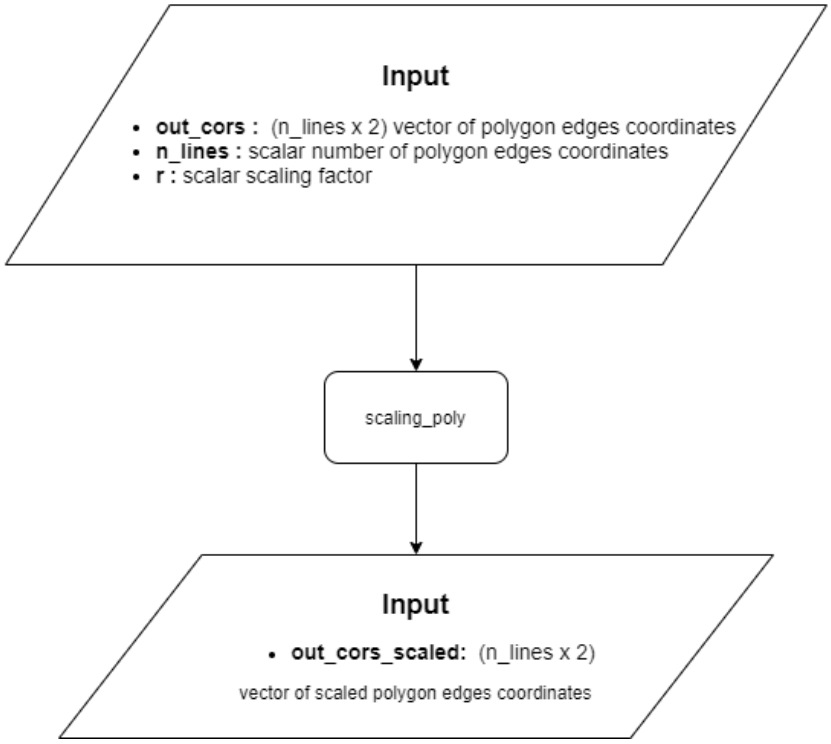
read_worksp_file.m

Function that reads a workspace file named "points.dat" with specific format and outputs space-related variables



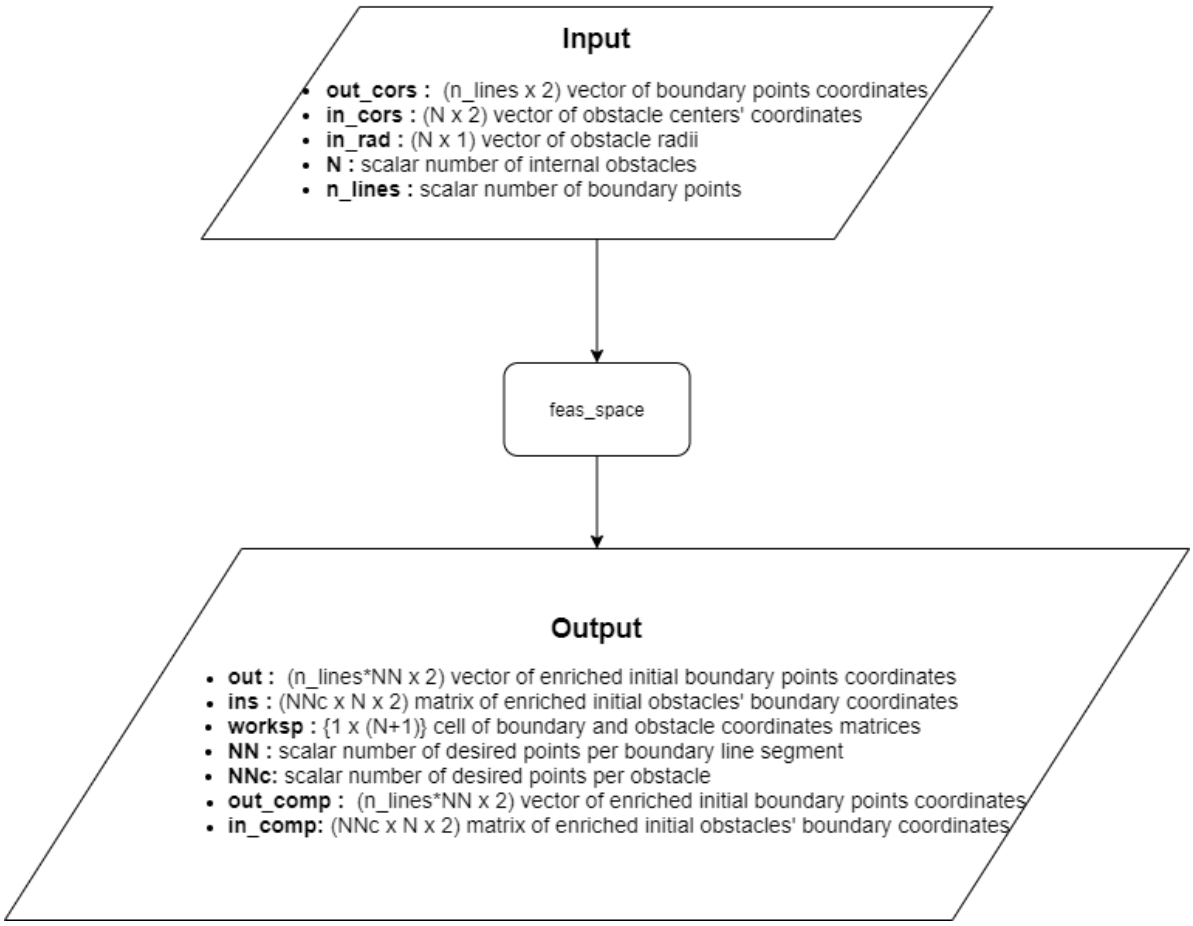
scaling_poly.m

Function that scales a given polygon by a desired scaling factor



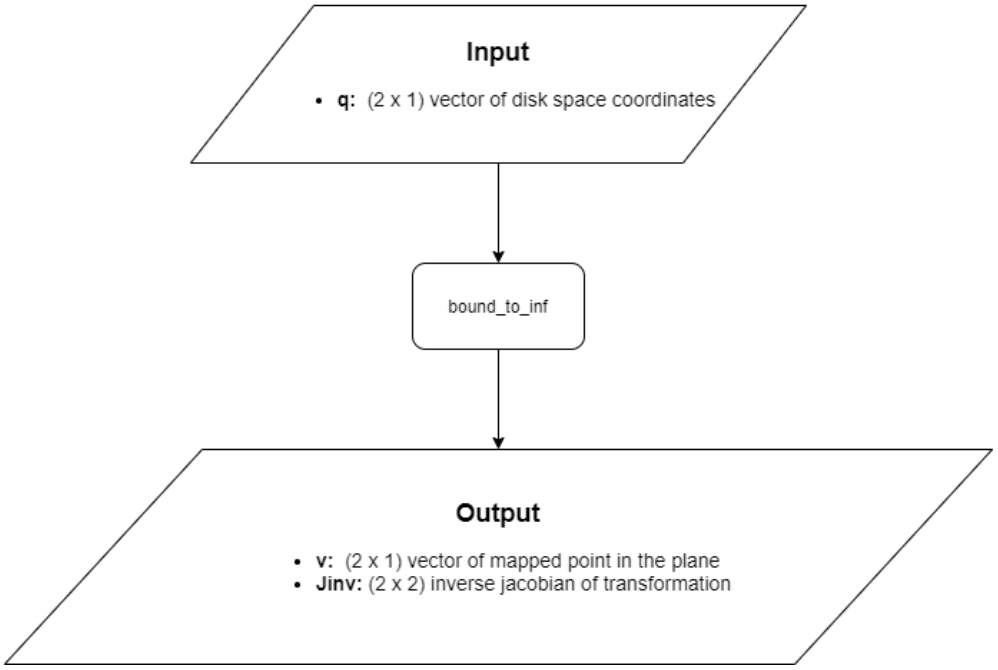
feas_space.m

Function that rescales the workspace according to a robot-related metric -here radius of superscribed circle- and outputs variables for the homeomorphic transformation



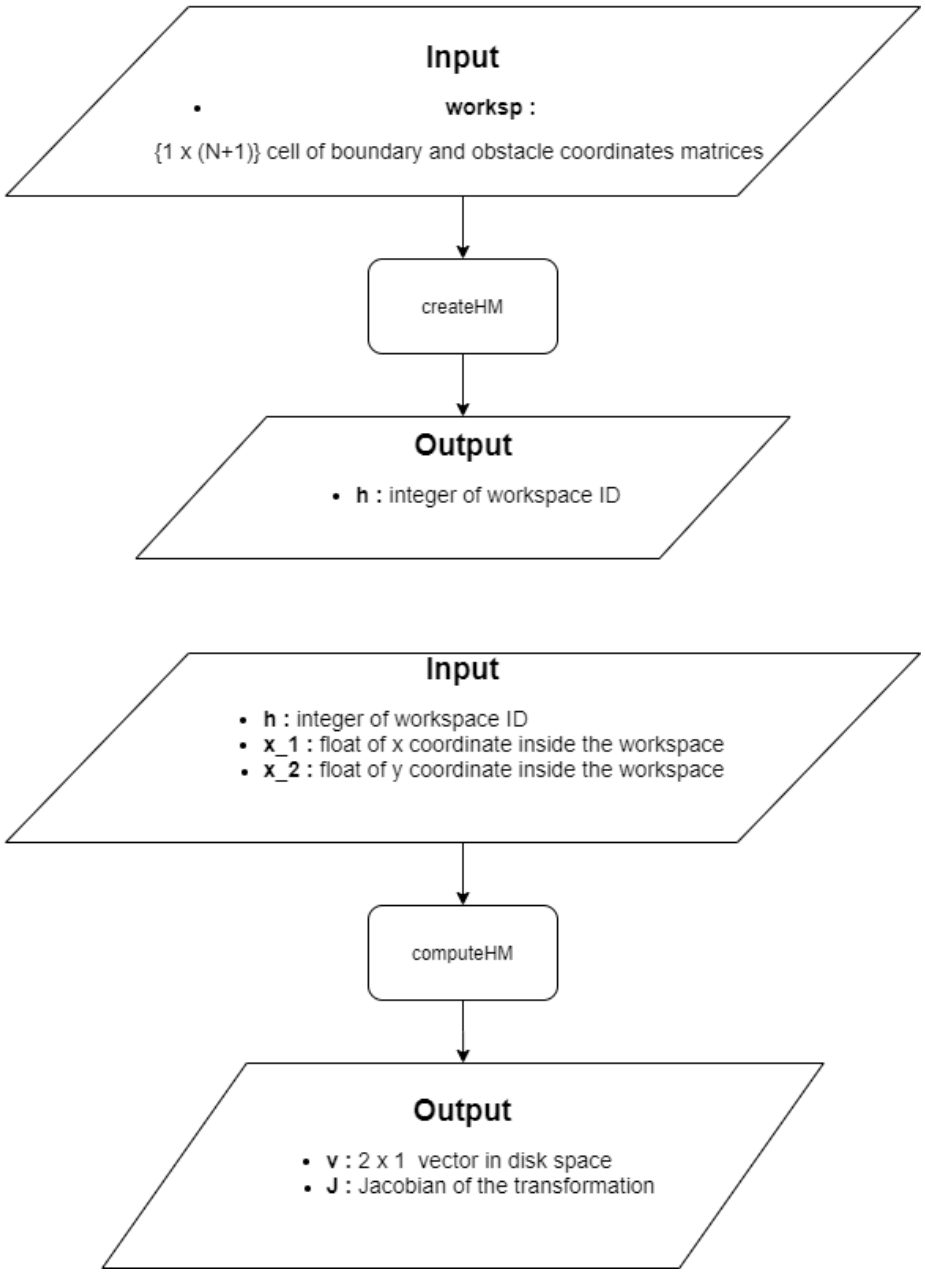
bound_to_inf.m

Function that maps the disk
space to the infinite plane



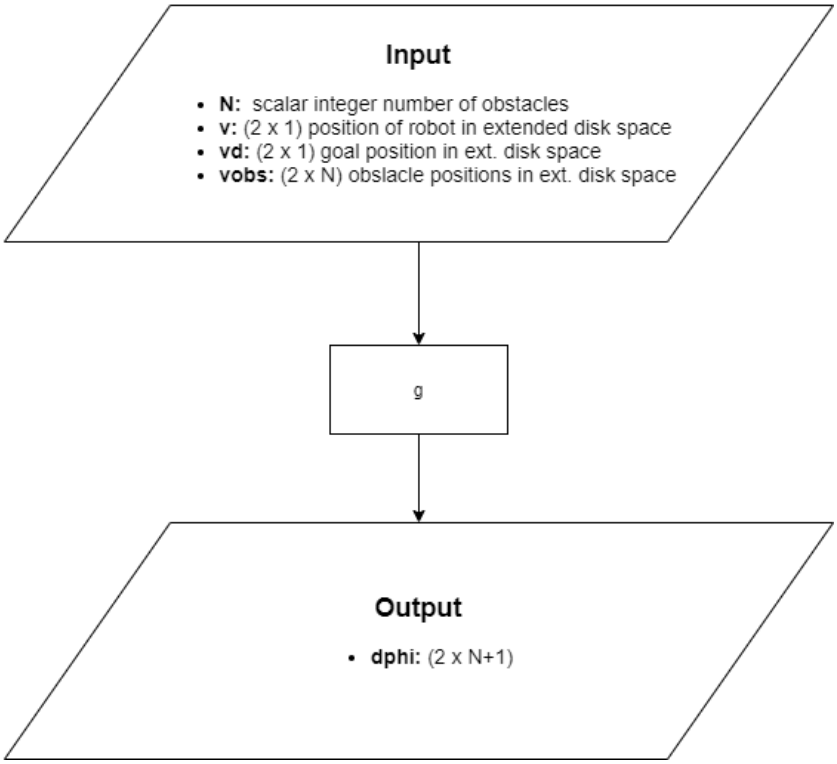
HM functions

Set of functions that constitute the homeomorphic transform from physical to disk space



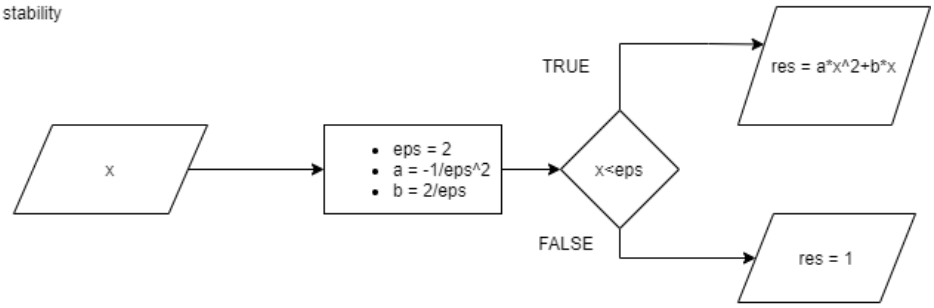
g.m

g(p,pd) function for input to system



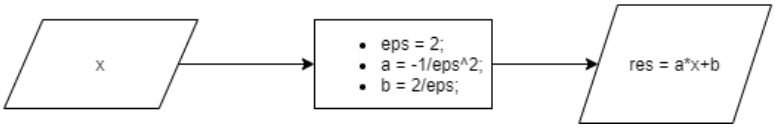
rep_tanh.m

replacement to tanh(x) function to enable simplification of terms for computational stability



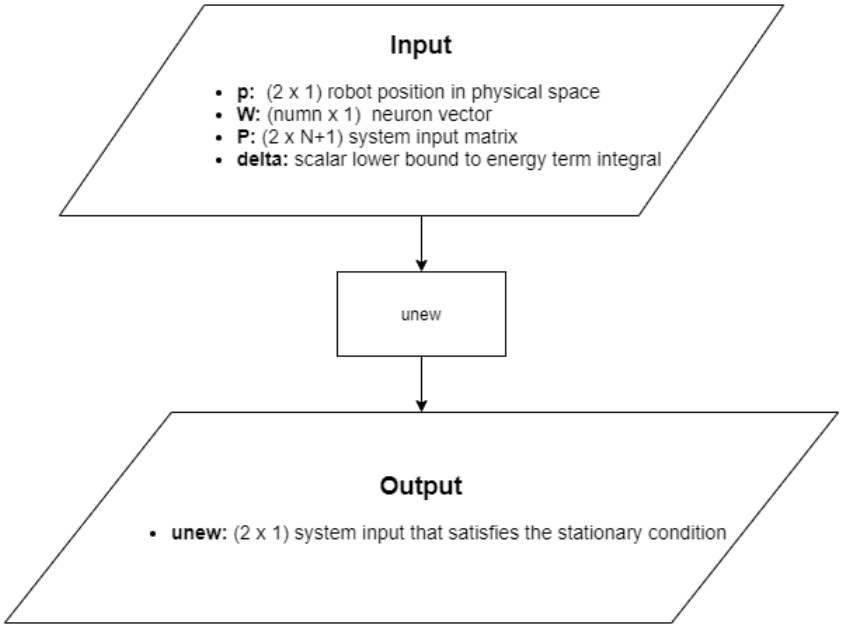
rep_tanh_simplified.m

replacement to tanh(x) function to enable simplification of terms for computational stability



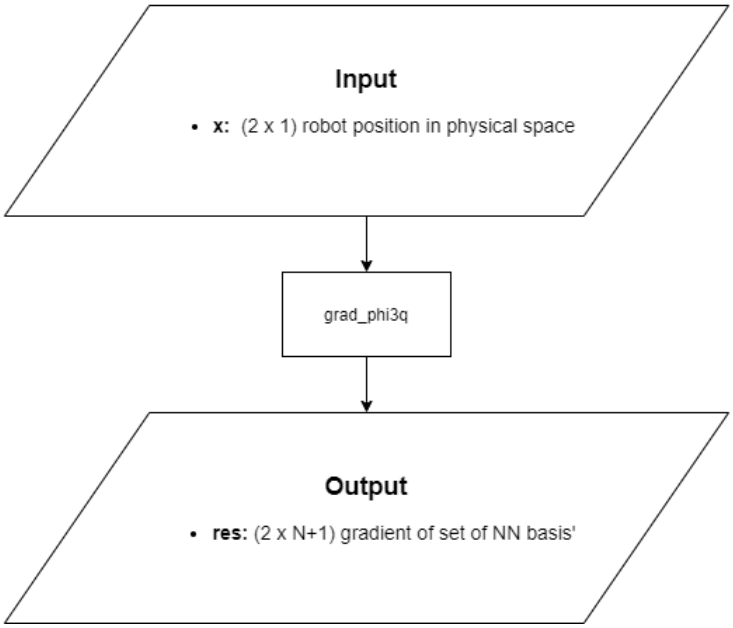
unew.m

input to system function



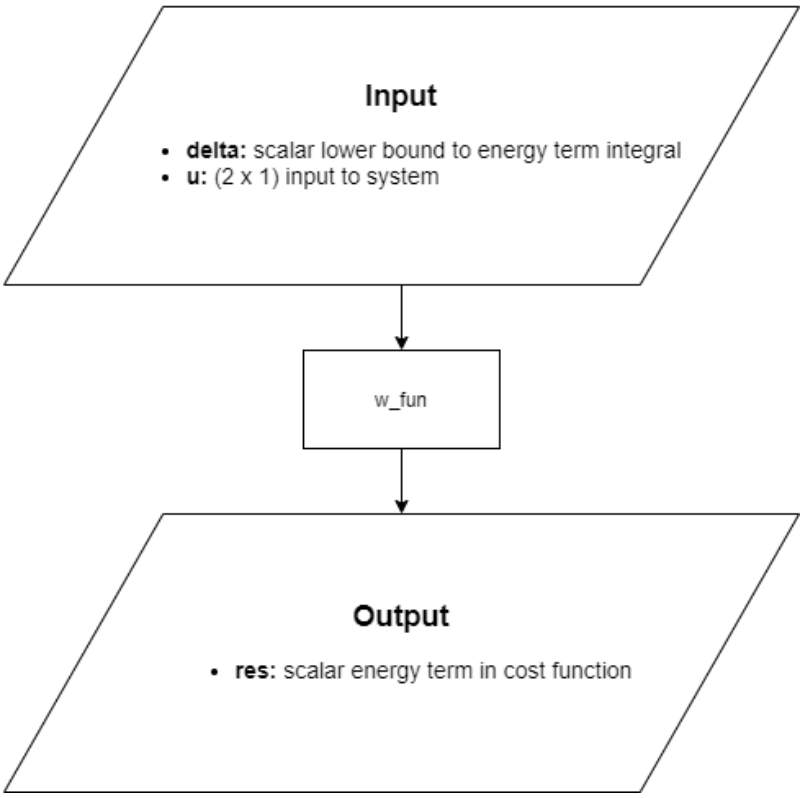
grad_phi3q.m

gradient of vector basis functions
for NN



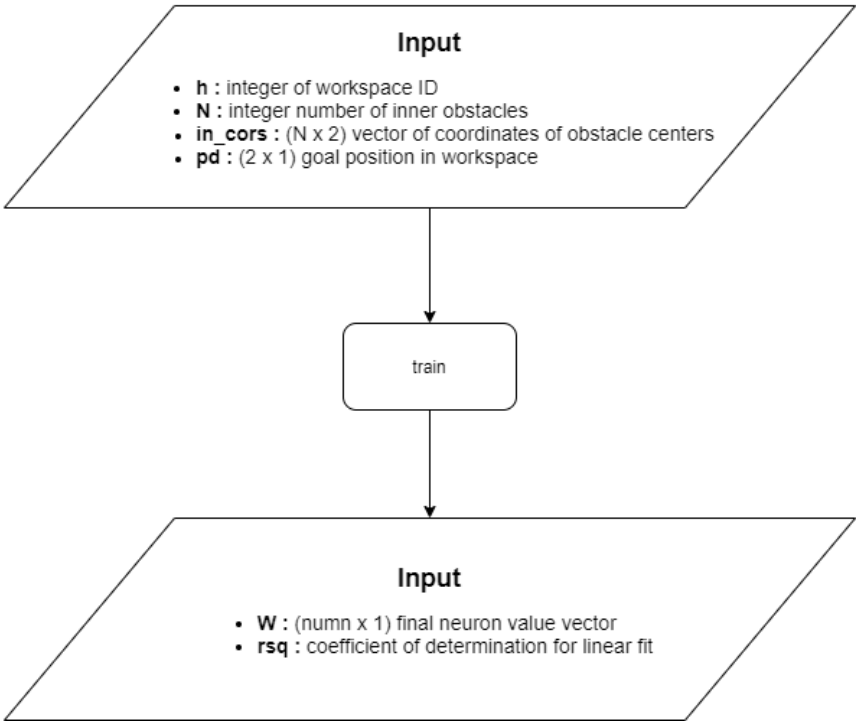
w_fun.m

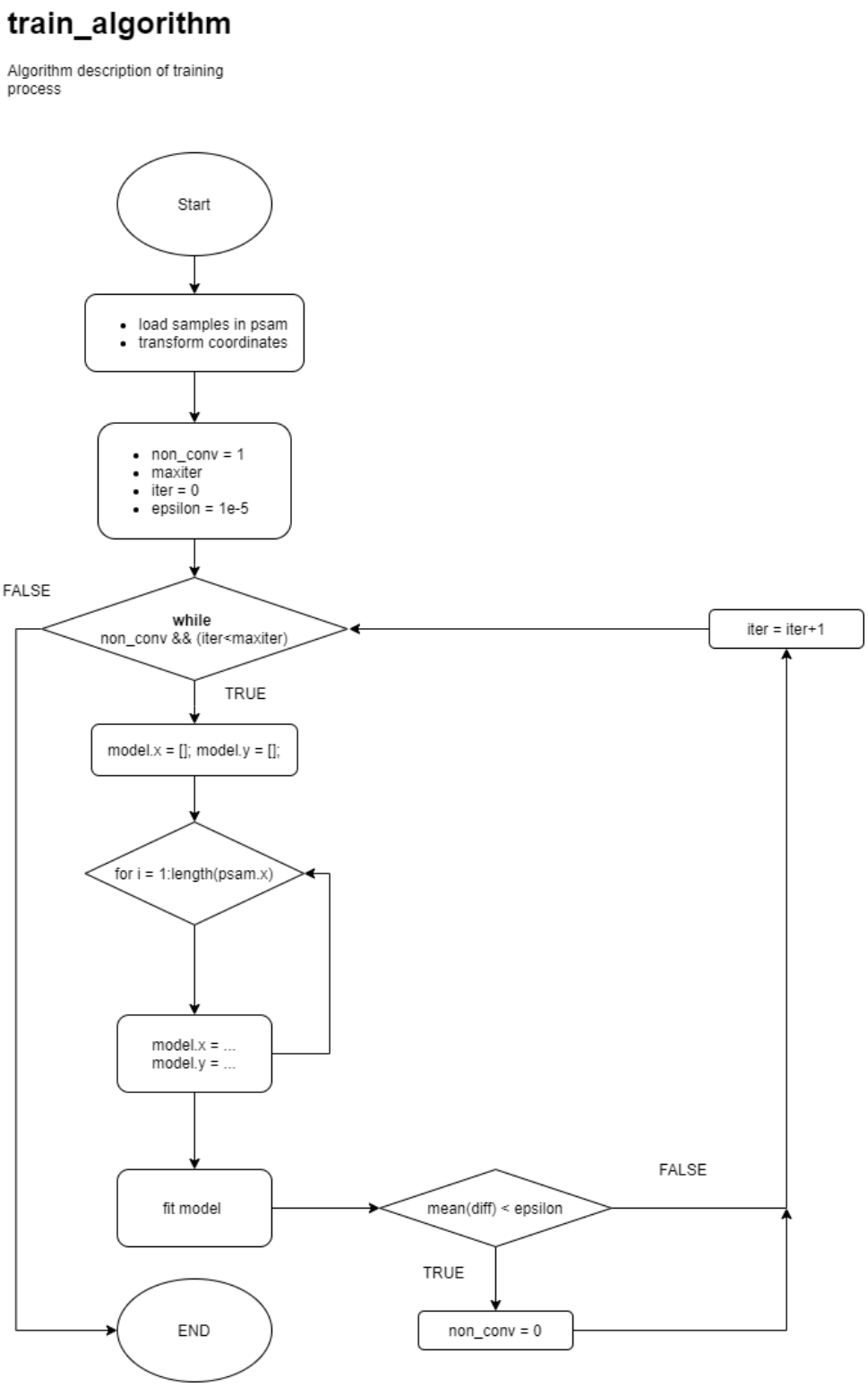
Energy term in cost function



train.m

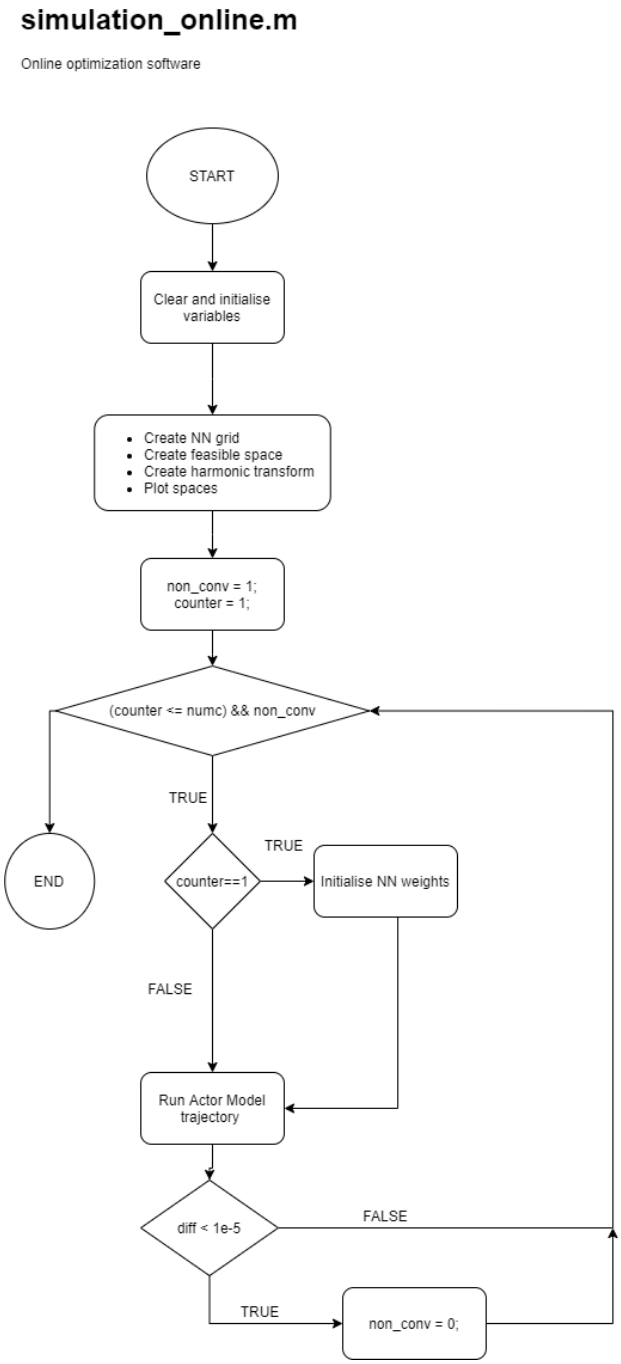
Function that employs the optimization process




train_algorithm
Algorithm description of training process
Start
load samples in psam
transform coordinates
non_conv = 1
maxiter
iter = 0
epsilon = 1e-5
FALSE
while
non_conv && (iter<maxiter)
iter = iter+1
TRUE
model.x = []; model.y = [];
for i = 1:length(psam.x)
model.x = ...
model.y = ...
fit model
mean(diff) < epsilon
FALSE
TRUE
END
non_conv = 0

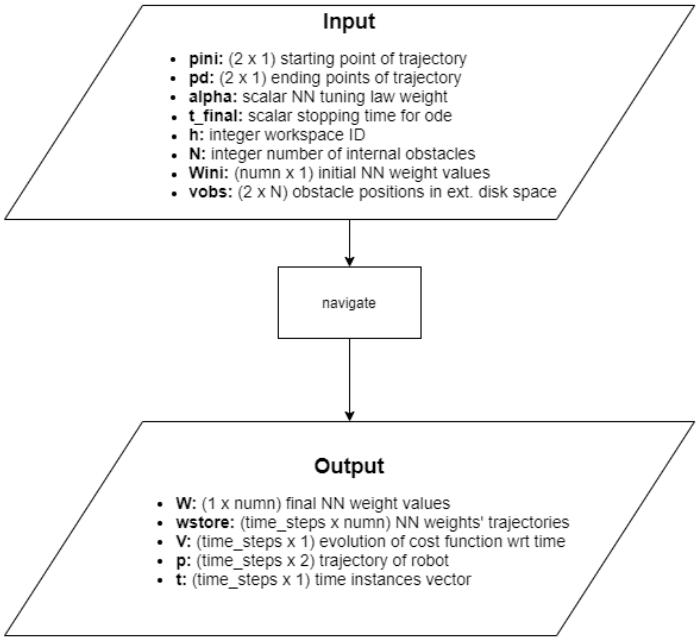
7.2.2 On-line method Software

The on-line method utilizes mostly the same functions as the Off-line one, therefore only the main structure of the algorithm will be presented.



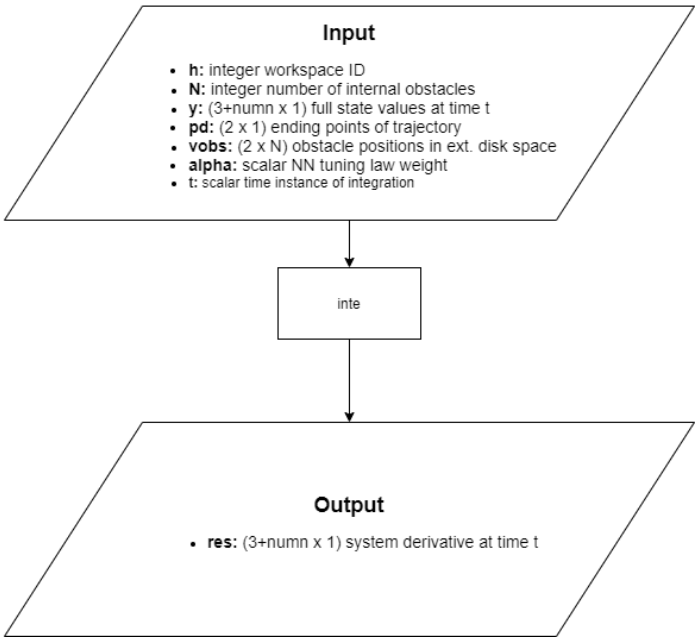
navigate.m

Function that computes a single trajectory along with respective NN weight tuning



inte.m

df/dt = y function for integration



Chapter 8

Bibliography

[1] Murad Abu-Khalaf and Frank L. Lewis. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach. *Automation and Robotics Research Institute*, 1990.

[2] S. Bhattacharya and R. Ghrist. Path homotopy invariants and their application to optimal trajectory planning. *Annals of Mathematics and Artificial Intelligence*, 2018.

[3] R. Bohlin and L. E. Kavraki. Path planning using lazy prm. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 1*, page 521–528, 2000.

[4] J. Canny. The complexity of robot motion planning. *MIT press*, 1988.

[5] Lyshevski S. E. Constrained optimization and control of nonlinear systems: new results in optimal control. *Proceedings of the IEEE conference on decision and control*, page 541–546, 1996.

[6] M. Čáp J. Gregoire and E. Frazzoli. Locally-optimal multi-robot navigation under delaying disturbances using homotopy constraints. *Autonomous Robots, vol. 42, no. 4*, page 895–907, 2018.

[7] S. Karaman and E. Frazzoli. "sampling-based algorithms for optimal motion planning". *The International Journal of Robotics Research, vol. 30, no. 7*, page 846–894, 2011.

[8] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Proceedings. 1985 IEEE International Conference on Robotics and Automation, vol. 2*, page 500–505, 1985.

[9] J. O. Kim and P. K. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation, vol. 8, no. 3*, page 338–349, 1992.

[10] D. Koditschek. Exact robot navigation by means of potential functions:some topological considerations. *Proceedings. 1987 IEEE International Conference on Robotics and Automation, vol. 4*, page 1–6, 1987.

[11] J. Latombe L. E. Kavraki, P. Svestka and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation, vol. 12, no. 4*, page 566–580, 1996.

[12] Frank L. Lewis, Draguna L. Vrabie, and Vassilis L. Syrmos. *Optimal Control, Third Edition*. John Wiley & Sons, Inc, 2012.

[13] Yesildirek A. Lewis F. L., Jagannathan S. *Neural network control of robot manipulators and nonlinear systems*. 1999.

[14] Savvas G. Loizou. Closed form navigation functions based on harmonic potentials. *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[15] Savvas G. Loizou. Closed form navigation functions based on harmonic potentials. *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[16] M. Hemmer O. Salzman and D. Halperin. On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages. *IEEE Transactions on Automation Science and Engineering, vol. 12, no. 2*, pages 529–538,, 2015.

[17] P. Ogren and N. E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Transactions on Robotics, vol. 21, no. 2*, page 188–195, 2005.

[18] Constantinos Vrohidis Panagiotis Vlantis and Kostas J. Kyriakopoulos Charalampos P. Bechlioulis. Robot navigation in complex workspaces using harmonic maps. *2018 IEEE International Conference on Robotics and Automation, Brisbane, Australia*, 2018.

[19] Jing Sun Petros A. Ioannou. *Robust Adaptive Control*. 2012.

[20] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential fields. *IEEE Transactions on Robotics and Automation, vol. 8, no. 5*, page 501–518, 1992.

[21] F. Bu M. Johnson-Roberson S. Kousik, S. Vaskov and R. Vasudevan. Bridging the gap between safety and real-time performance in receding- horizon trajectory design for mobile robots. *CoRR, vol. abs/1809.06746*, 2018.

[22] Lee C. S. Saridis G. An approximation theory of optimal control for trainable manipulators. *IEEE Transactions on Systems, Man, Cybernetics*, page 152–159, 1979.

[23] J. T. Schwartz and M. Sharir. On the piano movers' problem: Iii. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *International Journal of Robotic Research -IJRR, vol. 2*, page 46–75, 1983.

[24] J. T. Schwartza and M. Sharir. On the "piano movers" problem. i: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics, vol. 36*, page 345 – 398, 1983.

[25] K. G. Vamvoudakis and F. L. Lewis. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*.

[26] Kyriakos G. Vamvoudakis and Frank L. Lewis. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automation and Robotics Research Institute*, 1990.

[27] M. Moll Z. Kingston and L. E. Kavraki. Sampling-based methods for motion planning with constraints. *Annual Review of Control, Robotics, and Autonomous Systems, vol. 1, no. 1*, page 159–185, 2018.