2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A Continuous Off-Policy Reinforcement Learning Scheme for Optimal Motion Planning in Simply-Connected Workspaces

†Panagiotis Rousseas, ‡Charalampos P. Bechlioulis, and ††Kostas J. Kyriakopoulos

***Abstract*— In this work, an Integral Reinforcement Learning (RL) framework is employed to provide provably safe, convergent and almost globally optimal policies in a novel Off-Policy Iterative method for simply-connected workspaces. This restriction stems from the impossibility of strictly global navigation in multiply connected manifolds, and is necessary for formulating continuous solutions. The current method generalizes and improves upon previous results, where parametrized controllers hindered the method in scope and results. Through enhancing the traditional reactive paradigm with RL, the proposed scheme is demonstrated to outperform both previous reactive methods as well as an RRT* method in path length, cost function values and execution times, indicating almost global optimality.**

I. Introduction

The kinematic Motion Planning (MP) problem is a fundamental problem in Robotics, hence, a plethora of approaches have risen over the years to address it. While it has been traditionally tackled through both Reactive Approaches (RAs) and open-loop (OL) ones, Optimal Motion Planning (OMP) has only been extensively treated through OL Sampling-Based Methods (SBMs), with only crude approaches in the reactive paradigm. Since RAs exhibit numerous advantages (e.g. robustness, extensions to drift dyhamics, etc.) this work aims at leveraging modern RL in order to enhance the traditional, reactive approaches with optimality.

Particularly, we concentrate on simply-connected workspaces, i.e., workspaces with no internal obstacles. While this might appear as a limiting and unrealistic specification, we posit that it is necessary for formulating a mathematically complete approach; it is well known that strictly global navigation is topologically impossible through reactive fields in multiply connected manifolds [1]. This is traditionally bypassed by neglecting one or more, zero or one-dimensional subsets of the workspace [2]. While this approach is sufficient for navigation, addressing optimality in a formal manner necessitates for the aforementioned topological and geometrical features to be addressed. Therefore, this work is a first step at addressing simply-connected workspaces, where no such considerations are necessary to provide a formal solution to optimal **reactive** MP.

†The author is with the School of Mechanical Engineering, Control Systems Laboratory, National Technical University of Athens, Greece. ‡The author is with the Department of Electrical and Computer Engineering, University of Patras, ††The author is with the Center of AI & Robotics (CAIR), New York University, Abu Dhabi E-mails: prousseas@mail.ntua.gr, chmpechl@upatras.gr, kk4812@nyu.edu

II. Related Work

The MP problem has been tackled mainly through Discrete/SBMs or RAs. SBMs include the A* method [3], Djikstra's algorithm [4], Probabilistic Roadmaps (PRMs) [5] and RRT/RRT* [6], [7]. Notably, SBMs incorporate some form of optimality, more commonly as path-length minimizers with extensions to include more complicated cost formulations and kino-dynamic constraints [8]–[10], while also providing optimality guarantees (in the RRT* case, asymptotically). Additionally, there have been some notable extensions of SBMs through modern, learning-based approaches [11], [12].

On the other hand, RAs, focus on designing a continuous function over a workspace, whose gradient provides collision-free and convergent velocity fields. These include Navigation Functions (NFs) [1], [2] and Artificial Harmonic Potential Fields (AHPFs) [13]–[17]. However, there has been limited work in the context of reactive optimality. Some formal approaches include treating the minimum-time problem [18], or stochastic approaches [19]. However, [18] is limited to unbounded workspaces with circular obstacles, as an approximation of the cost function is constructed based on the obstacle shape, while in [19] the solution of a hard Partial Differential Equation (PDE) is needed. Finally, some less formal methods include tuning the parameters of relevant NFs [20], [21].

More recently, learning methods have also been employed for specific robotic platforms. A comprehensive review is available in [22]. More importantly, in our previous works [23]–[25], we have treated OMP via several approaches, including through a disk transformation as well as parametrized optimal controllers, which resulted in some promising results. However, the parametrized controllers and the constraints imposed for safety significantly limit the space of possible policies, resulting in relatively poor performance wrt path length. As it will become apparent in the sequel this work proposes a *parameter-free* method, thus resulting not only in nearly *globally* optimal cost function, but a conversely nearly optimal path length.

III. Problem Formulation

Consider a point robot[1], operating within a two-dimensional, simply connected, bounded workspace, denoted by $\mathcal{W} \subset \mathbb{R}^2$, with boundary $\partial\mathcal{W}$, along with a desired final

[1]A disk robot can also be considered by applying a workspace transformation that inflates the workspace boundaries: $\partial\mathcal{W}^I = T(\partial\mathcal{W})$ where $T(z) = z + Rn(z)$, $R \in \mathbb{R}_+$ denoting the robot's radius and n denoting the inwards-pointing vector that is normal to the boundary at the point $z \in \partial\mathcal{W}$.

position within the workspace denoted by $p_d \in \mathcal{W}$. In this work, we treat the single integrator dynamics:

$$\dot{p} = u, p(0) = \bar{p}, \tag{1}$$

where $p(t) : \mathbb{R}_+ \mapsto \mathcal{W}$ denotes the robot's position, $\bar{p} \in \mathcal{W}$ denotes the robot's position at time $t = 0$ and $u(t)$ denotes the input policy, i.e., the robot's velocity.

The aim is to design a **reactive velocity input**[2] $u(p) : \mathcal{W} \mapsto \mathbb{R}^2$ that minimizes the following infinite-horizon cost function (analogous to the traditional RL value function):

$$V_u(\bar{p}) = \int_0^\infty Q(p_u(\tau; \bar{p}); p_d) + R(u(\tau))\, \mathrm{d}\tau, \tag{2}$$

where $p_u(t; \bar{p}) : \mathbb{R}_+ \mapsto \mathcal{W}$ denotes the trajectory that stems from integrating (1) under the control input u, starting from the initial position $p(0) = \bar{p} \in \mathcal{W}$. Furthermore, we define the **state-related cost term** Q and the **input-related cost term** R respectively:

$$Q(p; p_d) = \alpha \|p - p_d\|^2, \tag{3a}$$

$$R(u) = \beta \|u\|^2, \tag{3b}$$

where α, β are positive weighting constants and $\|\cdot\|$ denotes the Euclidean norm. The metric (2) along with (3) form a classical cost function from Optimal Regulation theory [26]. The term (3a) reflects the minimization of the *settling time* of the system. The term (3b) penalizes the control input's Euclidean norm, which, when integrated, equals the *energy expenditure* of System (1).

IV. Methodology

A. Reactive Motion Planning

Prior to treating optimality, the acquisition of an initial reactive velocity field that stabilizes (1) safely, is necessary. Herein, we employ the AHPF-based method developed in [17], where harmonic panels are placed outside the boundary $\partial\mathcal{W}$ of the workspace $\mathcal{W}$ to acquire a provably safe and convergent reactive velocity field[3]. We direct the reader to [17] for further details.

B. Preliminaries on Optimality

In order to provide a solution to the optimal Motion Planning problem, we begin be defining a set of admissible policies, which essentially describes safe and convergent velocity vector fields.

Definition 1: (**Admissible Policy**) A policy $u(p) : \mathcal{W} \mapsto \mathcal{A}(\mathcal{W})$, where $\mathcal{A}(\mathcal{W})$ denotes the set of admissible policies, is defined as admissible with respect to the cost function (2) over the workspace $\mathcal{W}$, if: **1)** u is continuous on $\mathcal{W}$, **2)** $u(p_d) = \tilde{0}$, **3)** $u(p)$ stabilizes (1) on $\mathcal{W}$, **4)** $V_u(p)$ is finite $\forall p \in \mathcal{W}$ and **5)** the resulting trajectories of (1) under the control law $u = u(p)$ are safe, i.e., for any $\bar{p} \in (\mathcal{W} - \partial\mathcal{W})$ it holds that $\mathcal{P}_u(\bar{p}) \bigcap \partial\mathcal{W} = \{\emptyset\}$, where $\mathcal{P}_u(\bar{p}) = \bigcup_{t\in[0,+\infty]} p_u(t; \bar{p})$[4].

Therefore, we are only interested in admissible policies that minimize (2). To extract the optimal policy, consider the differential form of (2) [27]:

$$(\nabla V_u)^T u = -\alpha \|p - p_d\|^2 - \beta \|u\|^2, \tag{4}$$

where henceforth, the use of the ∇ symbol implies the gradient wrt the position $p \in \mathcal{W}$ of the robot. Eq. (4) along with the terminal condition $V_u(p_d) = 0$ form a Lyapunov-like PDE [27] which is employed to construct the Hamiltonian

$$H(p, u; \nabla V) = (\nabla V_u)^T u + r(p, u), \tag{5}$$

where $r(p, u) = \alpha \|p - p_d\|^2 + \beta \|u\|^2$. The optimal cost function $V^\star$ satisfies the Hamilton-Jacobi-Bellman equation:

$$\min_{u \in \mathcal{A}(\mathcal{W})} \{H(p, u; \nabla V^\star)\} = 0, \tag{6}$$

In order to ensure that only admissible policies are considered in solving (6), we employ the well-studied Lyapunov-Barrier function (LBF) theory and more specifically, Zeroing Barrier Function (ZBF) theory [28]. In this case we define a ZBF $L(p) : \mathcal{W} \mapsto [0, 1]$

$$L(p) = \begin{cases} 1 - \exp\left(-\left(\frac{d(p)}{d(p)-a}\right)^2\right), & d(p) \le a \\ 1, & d(p) > a \end{cases}, \tag{7}$$

with $a \in \mathbb{R}_+$[5] while the function $d : \mathcal{W} \mapsto \mathbb{R}_+$ computes the distance of the robot to the boundary:

$$d(p) = \min_{z \in \partial\mathcal{W}} \{\|p - z\|\}. \tag{8}$$

Intuitively, the ZBF $L(p)$ is equal to 1 in the interior of the workspace at a distance-to-the-boundary larger than, or equal to a, while for points with a distance less than a, the function varies smoothly (but not analytically) from 1 to 0. System (1) is safe if the time derivative of the ZBF along a trajectory obeys the following:

$$\dot{L} + h(L) \ge 0 \Leftrightarrow (\nabla L)^{\mathrm{T}} u + h(L(p)) \ge 0, \tag{9}$$

where $h(\cdot)$ is a class $\mathcal{K}$ function [28]. Incorporating the above condition, results in the following constrained optimization problem:

$$\begin{aligned} &\min_u \{H(p, u; \nabla V^\star)\} = 0, \\ &\text{s.t.: } C(p; u) \triangleq (\nabla L)^{\mathrm{T}} u + h(L(p)) \ge 0. \end{aligned} \tag{10}$$

The above Hamiltonian is subsequently infused with the ZBF condition (9) which through a Lagrange multiplier $\lambda \in \mathbb{R}_+$ and the Karush–Kuhn–Tucker (KKT) stationary condition becomes:

$$\nabla V_u^\star + \left.\frac{\partial r(p, u)}{\partial u}\right|_{u^\star} - \lambda \left.\frac{\partial C(p; u)}{\partial u}\right|_{u^\star} = 0, \tag{11}$$

[2]To avoid any ambiguity, we note that a reactive field $u(p)$ can be expressed as a function of time, if it is evaluated along trajectories of System (1), i.e, $u(t) \triangleq u(p_u(t; \bar{p}))$, therefore our definitions of time series vs velocity fields are consistent throughout the manuscript.

[3]The method in [17] concerns unknown workspaces, however it can be trivially extended to fully known workspaces.

[4]This definition ensures that under the control law u the robot does not collide with the workspace boundary at any point along all trajectories (for any initial position $\bar{p}$).

[5]The use of the English letter a in (7) is not to be mistaken for the Greek letter α in (3a).

which yields the optimal constrained control as follows:

$$u^\star = -\frac{1}{2\beta}\left(\nabla V_u^\star - \lambda^\star \nabla L\right). \tag{12}$$

The Lagrange multiplier can be extracted by considering the optimal condition $(\lambda^\star)\, C\left(p; u^\star\right) = 0$ (for $\lambda^\star \neq 0$):

$$\lambda^\star = \frac{-2\beta h(L(p)) + (\nabla L)^{\mathrm{T}}(\nabla V^\star)}{\|\nabla L\|^2}, \tag{13}$$

Thus, the optimal Lagrange multiplier is:

$$\lambda^\star(p) = \begin{cases} 0 & \text{if } C^\star \geq 0 \\ \frac{-2\beta h(L(p)) + (\nabla L)^{\mathrm{T}}(\nabla V^\star)}{\|\nabla L\|^2} & \text{if } C^\star < 0 \end{cases}, \tag{14}$$

where $C^\star = C\left(p; -1/2\beta \nabla V^\star\right)$.

C. Policy Iteration scheme

Evidently, extracting the optimal policy (12) and (5) requires the solution of a hard, non-linear PDE. We circumvent this limitation through an Off-Policy (OFFP), Policy Iteration (PI) scheme, introduced in [29], [30] (where a successive approximation framework for nonlinear optimal control is presented for open subsets of $\mathbb{R}^n$). We extend this framework for non-convex workspaces, in the context of OMP. Briefly, this PI scheme begins with an initial, admissible policy $u^{(0)} \in \mathcal{A}\left(\mathcal{W}\right)$ (see Def. 1). This policy admits by definition a continuous cost function $V^{(0)}(\bar{p}) : \mathcal{W} \mapsto \mathbb{R}_+$, which can be employed to yield a sequence of admissible and cost-improving policies, as discussed in [31]. To satisfy the admissibility Def. 1, we propose the following PI scheme:

$$u^{(i+1)} = -\frac{1}{2\beta}\left(\nabla V^{(i)} - \lambda^{(i)}(p)\nabla L\right), \tag{15}$$

where

$$\lambda^{(i)} = \begin{cases} 0 & \text{if } C^{(i)} \geq 0 \\ \frac{-2\beta h(L(p)) + (\nabla L)^{\mathrm{T}}(\nabla V^{(i)})}{\|\nabla L\|^2} & \text{if } C^{(i)} < 0 \end{cases}. \tag{16}$$

where $C^{(i)} = C\left(p; -\frac{\nabla V^{(i)}}{2\beta}\right)$. We prove that the above sequence of policies are admissible and improving wrt (2), in Section V.

D. Off-Policy Cost Function Approximation

It is evident that in the scheme of the preceding subsection, the cost function for a given policy is necessary in (15), (16). In OFFP schemes, the implemented policy and the evaluated policy are different. Let $v(t) : \mathbb{R}_+ \mapsto \mathbb{R}^2$ denote a nominal input policy (the implemented policy), and $u(t) : \mathbb{R}_+ \mapsto \mathbb{R}^2$ denote the evaluated policy. We begin by noting that the reference input can be written as

$$v(t) = v(t) - u(t) + u(t) \triangleq u(t) + \delta(t), \tag{17}$$

while the (unknown) cost function for the input $u(t)$, V_u can be evaluated over trajectories of System (1) **under the control input** $v(t)$ (notice the subscript v for the trajectory $p_v\left(t; \bar{p}\right)$): $V_u(t) = V_u\left(p_v\left(t; \bar{p}\right)\right) : \mathbb{R}_+ \mapsto \mathbb{R}_+$. Taking the time derivative of the above function yields:

$$\begin{aligned} \dot{V}_u &= \left(\nabla V_u\right)^{\mathrm{T}} v = \left(\nabla V_u\right)^{\mathrm{T}} \left(u(t) + \delta(t)\right) \overset{(4)}{=} \\ &= -\alpha \|p - p_d\|^2 - \beta \|u\|^2 + \left(\nabla V_u\right)^{\mathrm{T}} \delta(t). \end{aligned} \tag{18}$$

Integrating both sides on some well-defined interval $[t, t+T]$, $T \in \mathbb{R}_+$ yields:

$$\begin{aligned} &V_u\left(p_v\left(t+T; \bar{p}\right)\right) - V_u\left(p_v\left(t; \bar{p}\right)\right) = \\ &-\int_t^{t+T} r\left(p_v\left(\tau; \bar{p}\right), u\right) d\tau + \int_t^{t+T} \left(\nabla V_u(\tau)\right)^{\mathrm{T}} \delta(\tau) d\tau, \end{aligned} \tag{19}$$

which is essentially the OFFP formulation of IRL [32]. In order to implement the scheme of Subsection IV-C, a sufficiently accurate approximation of the cost function is necessary at each iteration, which is acquired through a linear approximation structure (AS)[6]:

$$V_u = \phi^{\mathrm{T}}(p) w + \epsilon, \tag{20}$$

where $\phi^{\mathrm{T}}(p)w$ denotes the cost function approximation, ϵ denotes the approximation error, $\phi : \mathcal{W} \mapsto \mathbb{R}^n$ denotes a function basis, and $w \in \mathbb{R}^n$ denotes the weights of the AS. An exemplary choice for the basis functions is a set of Gaussian Radial Basis Functions (RBFs) [33], motivated by the local dependence of the related cost function. Thus, Eq. (19) takes the linear (wrt the weights) form:

$$\begin{aligned} &w^{\mathrm{T}}\left[\phi\left(p_v\left(t+T; \bar{p}\right)\right) - \phi\left(p_v\left(t; \bar{p}\right)\right)\right] = \\ &-\int_t^{t+T} r\left(p_v\left(\tau; \bar{p}\right), u\right) d\tau + w^{\mathrm{T}} \int_t^{t+T} \nabla_p^{\mathrm{T}} \phi(\tau) \delta(\tau) d\tau \Rightarrow \\ &w^{\mathrm{T}} X(t; T; \bar{p}; u; v) = Y(t; T; \bar{p}; u; v), \end{aligned}$$

where

$$\begin{aligned} X(t; T; \bar{p}; u; v) = \phi\left(p_v\left(t+T; \bar{p}\right)\right) - \phi\left(p_v\left(t; \bar{p}\right)\right) - \\ \int_t^{t+T} \nabla_p^{\mathrm{T}} \phi\left(p_v\left(\tau + T; \bar{p}\right)\right) \delta(\tau) d\tau \end{aligned} \tag{21a}$$

$$Y(t; T; \bar{p}; u; v) = -\int_t^{t+T} r\left(p_v\left(\tau; \bar{p}\right), u\right) d\tau. \tag{21b}$$

This OFFP framework can be leveraged to acquire a computationally efficient scheme for computing the cost function approximation. In the context of an On-Policy (ONP) method, the trajectories under the policy $u^{(i)}$ need to be computed at each iteration, along with the respective cost, resulting in a significant computational load. In our OFFP framework, an implemented policy is chosen (e.g. the AHPF-based initial policy $u^{(0)}$) and its respective trajectories (see System (1)) are computed. Subsequently, the terms in (21) can be computed relatively inexpensively. The first two terms of the sum in (21a) are computed directly on points sampled over the trajectories. However, it is evident that the integral in (21a) as well as the RHS term of (21b) necessitate the integration of a total of $n+1$ quantities, which should be re-computed at each iteration, as both terms depend on the evaluated policy $u^{(i)}(t)$ (see Eq. (17)). However, in practice, both quantities can be approximated through well-known numerical integration methods (e.g. the trapezoidal approximation). The above process negates the need for the

[6]Note that this is indeed a "parameter-free" policy in the sense that the form of the policy stems from a general function space (the space of cost functions) which is merely approximated through an RBF network.

Algorithm 1: OFFP-PI ALGORITHM

- Given a Workspace $\mathcal{W}$
- Take $J \in \mathbb{N}$ samples $\bar{p}_j \in \partial\mathcal{W}, j \in \{1, \cdots, J\}$
- Starting from an initial policy $u^{(0)} \in \mathcal{A}(\mathcal{W})$
- Compute J trajectories $p_{u^{(0)}}(t; \bar{p}_j)$
- Set $i \leftarrow 0$

while $i == 0$ **or** $w^{(i)}$ have not converged **do**

- Form the matrices $A,\ B$ through (21), (23) over the sampled, on-trajectory points, for the evaluated policy $u = u^{(i)}$ and implemented policy $v = u^{(0)}$,
- Acquire the cost function approximation for $V^{(i)}$ through the vector $w^{(i)}$ through solving (22),
- Acquire the next policy through (15) and (16), where $\nabla V^{(i)} \approx \nabla\phi^{\mathrm{T}}(p)w^{(i)}$,
- $i \leftarrow i + 1$

end while

- The optimal policy is $u^\star \approx u^{(i)}$

re-computation of the trajectories at each step, thus reducing significantly the computational load of the proposed method.

To perform the cost function approximation, given a set of $J \in \mathbb{N}$ pre-computed trajectories (starting from the distinct initial points $\bar{p}_j,\ j \in \{1, \cdots, J\}$), the latter are sampled over intervals $\left[T_k^j, T_{k+1}^j\right],\ k \in \{1, \cdots, K\},\ j \in \{1, \cdots, J\}$ to form the following linear system of equations:

$$\left(A_{(KJ\times n)}\right)^{\mathrm{T}} w_{(n\times 1)} = B_{(KJ\times 1)}, \tag{22}$$

where

$$\begin{aligned} A = \big[&X(T_1^1; T_2^1; \bar{p}_1; u; v), \cdots, X(T_{K-1}^1; T_K^2; \bar{p}_1; u; v), \\ &X(T_1^J; T_2^J; \bar{p}_J; u; v), \cdots, X(T_{K-1}^J; T_K^J; \bar{p}_J; u; v)\big], \end{aligned}$$

$$\begin{aligned} B = \big[&Y(T_1^1; T_2^1; \bar{p}_1; u; v), \cdots, Y(T_{K-1}^1; T_K^2; \bar{p}_1; u; v), \\ &Y(T_1^J; T_2^J; \bar{p}_J; u; v), \cdots, Y(T_{K-1}^J; T_K^J; \bar{p}_J; u; v)\big]^{\mathrm{T}}. \end{aligned} \tag{23}$$

Upon solving (22), given a rich set of basis as well as adequate samples over the workspace (to render the matrix A pseudo-invertible), a sufficiently accurate approximation of the cost function of an admissible policy u can be acquired. The proposed method is finally summarized in Algorithm 1.

V. Technical Results

In this section, we prove the asserted claims of admissibility of the control input, as well as of the control improvement, for the sequence of inputs $u^{(i)}$.

*Lemma 1 (**Control Admissibility**):* Consider System (1), as well as an admissible policy $u^{(i)} \in \mathcal{A}(\mathcal{W})$ along with its respective cost function $V^{(i)}$ (2). Then, the policy $u^{(i+1)}$ (15) is admissible per Def. 1.

Proof: **1)** We begin by proving continuity. Since $u^{(i)}$ is admissible, then $\nabla V^{(i)}$ is continuous by definition. Furthermore, consider the second term, namely

$$\lambda^{(i)}(p)\nabla L = \begin{cases} 0 & \text{if } C^{(i)} \geq 0 \\ \frac{-2\beta C^{(i)}\nabla L}{\|\nabla L\|^2} & \text{if } C^{(i)} < 0 \end{cases}, \tag{24}$$

where $C^{(i)} = C\left(p; -\frac{\nabla V^{(i)}}{2\beta}\right)$. Evidently, we only have to prove continuity for $C^{(i)} = 0$. However,

$$-\frac{2\beta C^{(i)}\nabla L}{\|\nabla L\|^2} = -2\beta\frac{h(L(p))\nabla L}{\|\nabla L\|^2} + \frac{(\nabla L)^{\mathrm{T}}(\nabla V^{(i)})\nabla L}{\|\nabla L\|^2}. \tag{25}$$

The second term's direction is continuous (co-linear with ∇L), while its norm is equal to $\|\nabla V^{(i)}\|\cos(\theta)$, where $\theta = \angle\left(\nabla L, \nabla V^{(i)}\right)$ is also continuous. The first term through some work yields: $\frac{h(L(p))}{\|\nabla L\|^2}\nabla L = -\frac{h(L(p))}{L(p)}\frac{(d(p)-a)^3}{2ad(p)}\nabla d(p)$, which is evidently continuous for the given choice of ZBF.

2) The requirement that $u^{(i)}(p_d) = \tilde{0}$ is trivial as long as $d(p_d) > a$, which can be set as a design specification for choosing $a \in \mathbb{R}_+ - \{0\}$.

3) We prove that $u^{(i)}$ stabilizes System (1) through standard Lyapunov arguments. Consider the cost function $V^{(i)}$ as a Lyapunov candidate. The standard prerequisites for Lyapunov candidates are evidently satisfied (i.e. continuity, single global minimum at p_d, etc.). To prove stability, consider its time derivative along the policy $u^{(i+1)}$:

$$\begin{aligned} \dot{V}^{(i)} &= \left(\nabla V^{(i)}\right)^{\mathrm{T}} u^{(i+1)} \overset{(15),(16)}{=\!=} \\ &-\frac{1}{2\beta}\|\nabla V^{(i)}\|^2 + \frac{\lambda^{(i)}}{2\beta}(\nabla L)^{\mathrm{T}}\nabla V^{(i)}. \end{aligned} \tag{26}$$

Consider the second term $\lambda^{(i)}/2\beta\,(\nabla L)^{\mathrm{T}}\nabla V^{(i)}$ In case $C^{(i)} \geq 0$, then the Lagrange multiplier is "deactivated", i.e., $\lambda^{(i)} = 0$, and $\dot{V}^{(i)} = -\frac{1}{2\beta}\|\nabla V^{(i)}\|^2$. In case $C^{(i)} < 0$:

$$\begin{aligned} &\lambda^{(i)}(\nabla L)^{\mathrm{T}}\nabla V^{(i)} \overset{(16)}{=\!=} \\ &-\frac{2\beta h\left(L(p)\right)(\nabla L)^{\mathrm{T}}\nabla V^{(i)}}{\|\nabla L\|} + \frac{\left|(\nabla L)^{\mathrm{T}}\nabla V^{(i)}\right|^2}{\|\nabla L\|^2}. \end{aligned} \tag{27}$$

However, since $C^{(i)} < 0$, it follows directly that: $(\nabla L)^{\mathrm{T}}\nabla V^{(i)} > 2\beta h\left(L(p)\right) \geq 0$ owing to $h(\cdot)$ being a class $\mathcal{K}$ function and $L : \mathcal{W} \mapsto [0, 1]$. This shows that the first term in (27) is negative. Finally, note that the second term in (27) with the first RHS term in (26) yield:

$$\frac{\left|(\nabla L)^{\mathrm{T}}\nabla V^{(i)}\right|^2}{\|\nabla L\|^2} - \frac{1}{2\beta}\|\nabla V^{(i)}\|^2 = -\frac{1}{2\beta}\|\nabla V^{(i)}\|^2\sin^2(\theta),$$

which is negative, therefore evidently $\dot{V}^{(i)} < 0,\ p \in \mathcal{W} - \{p_d\}$ as a sum of negative terms in (26).

4) The final part of the proof, namely safety, directly follows from considering the value of the velocity field $u^{(i+1)}$ at the boundary of the workspace. For any $z \in \partial\mathcal{W}$, where $h\left(L(d(z))\right) = 0$:

$$u^{(i+1)}(z) = -\frac{1}{2\beta}\left(\nabla V^{(i)} - \frac{(\nabla L)^{\mathrm{T}}(\nabla V^{(i)})}{\|\nabla L\|^2}\nabla L\right). \tag{28}$$

Note that the above sum can be interpreted as subtracting the outwards-pointing (unsafe) component of $-\nabla V^{(i)}$ at the boundary (**if** such an unsafe component exists, else the zero-valued Lagrange multiplier nullifies the subtraction). This evidently renders System (1) **safe**, as all trajectories point inwards at the boundary. A minor detail rests on the existence of an inwards component of $-\nabla V^{(i)}$. This indeed exists, as through (4), it is evident that $u^{(i)}$ and $-\nabla V^{(i)}$ are co-linear. Since $u^{(i)}$ is by definition safe, i.e., inwards-pointing, it is easy to see that there exists indeed such a "feasible" component for $-\nabla V^{(i)}$. This concludes the admissibility proof. ■

*Lemma 2 (**Control Improvement**):* Given an admissible policy $u^{(i)} \in \mathcal{A}(\mathcal{W})$, the policy $u^{(i+1)}$ (15) applied to System (1) successively results in improvement of the cost function (2), i.e., $V^{\star} \leq V^{(i+1)} \leq V^{(i)},\ i \in \mathbb{N}_{+}$. *Proof:* We follow Lemma 1 in [34]. Evaluating two subsequent costs over the trajectory $p_{u^{(i+1)}}(t;\bar{p})$ yields:

$$V^{(i+1)}(\bar{p}) - V^{(i)}(\bar{p}) = -\int_0^{\infty} \frac{d\left(V^{(i+1)} - V^{(i)}\right)^T}{dp} u^{(i+1)} \mathrm{d}\tau. \tag{29}$$

Through evaluating (4) for $V^{(i)}$ and $V^{(i+1)}$, Eq. (29) yields:

$$V^{(i+1)}(\bar{p}) - V^{(i)}(\bar{p}) = -\int_0^{\infty} B \mathrm{d}\tau,$$

where:

$$B = \left(\nabla V^{(i)}\right)^T \left(u^{(i)} - u^{(i+1)}\right) - \beta\left(\|u^{(i+1)}\|^2 - \|u^{(i)}\|^2\right).$$

In order to conclude the proof, it suffices to show that $B \geq 0$. Since from (15) $\nabla V^{(i)} = -2\beta u^{(i+1)} + \lambda^{(i)} \nabla L$, then:

$$B = \underbrace{\lambda^{(i)} (\nabla L)^T \left(u^{(i)} - u^{(i+1)}\right)}_{B''} - \underbrace{2\beta\left[\left(u^{(i+1)}\right)^T \left(u^{(i)} - u^{(i+1)}\right) - \left(\|u^{(i+1)}\|^2 - \|u^{(i)}\|^2\right)\right]}_{B'}.$$

Applying the mean value theorem, $B^{'}$ can be shown to be negative, thus it suffices to prove that $B^{''} \geq 0$. However, note that since $\lambda^{(i)} \geq 0$

$$B^{''} \propto (\nabla L)^T \left(u^{(i)} - u^{(i+1)}\right) \overset{(15)}{=}$$
$$(\nabla L)^T \left(u^{(i)} + \tfrac{1}{2\beta}\nabla V^{(i)} - \tfrac{(\nabla L)^T \nabla V^{(i)}}{2\beta\|\nabla L\|^2}\nabla L + \tfrac{h(L(p))}{\|\nabla L\|^2}\nabla L\right) =$$
$$(\nabla L)^{\mathrm{T}} u^{(i)} + h\left(L(p)\right) \geq 0, \tag{30}$$

owing to $u^{(i)}$ being by definition admissible. To see this, note that since from (15), $u^{(i)} = -\frac{1}{2\beta}\left(\nabla V^{(i-1)} - \lambda^{(i-1)}(p)\nabla L\right)$ (for $i > 1$) :

$$(\nabla L)^{\mathrm{T}} u^{(i)} + h\left(L(p)\right) = C^{(i-1)} + \frac{\lambda^{(i-1)}}{2\beta}\|\nabla L\|^2 \geq 0,$$

where admissibility of $u^{(i)}$ implies $C^{(i-1)} \geq 0$ and $\lambda^{(i-1)} \geq 0$. Therefore, $B^{''} > 0$ which shows that $V^{(i+1)} \leq V^{(i)},\ i \in \mathbb{N}_{+}$. Finally, for $i = 1$, since the initial policy is safe, it renders $\mathcal{W}$ forward invariant. This implies according to ZBF theory (Proposition 3, in [28]) that the proposed function is a ZBF for (1) under the policy $u^{(0)}$ and therefore $C^{(0)} \geq 0$. To complete the proof, it can be shown through contradiction that the above sequence is bounded below by $V^{\star}$. ■

VI. Results

In this section we present synthetic simulations in order to demonstrate the validity of the technical results, as well as the efficacy of our method in providing almost globally optimal policies. All simulations were carried out on a PC running on Ubuntu, with an Intel-i7 processor and 50 Gb RAM (although rarely more than 6 Gb were in use at once during the implementation of the proposed method). Additionally, the parameters $\alpha = 1, \beta = 1$ along with $h(x) \triangleq x$ are chosen. In Figs. 1, 2, 3 we demonstrate the efficacy of our method in providing almost globally optimal trajectories in simply-connected, but highly non-convex workspaces. This is evident through the shape of the presented trajectories, which exhibit almost minimum path length. Additionally, the "almost global optimality" is demonstrated in the right-most figure of Fig. 1, where the negated ratio of the two terms in (5) is depicted. Evidently, this ratio is close to 1 almost everywhere inside the workspace, demonstrating the close-to-global optimality of the final policy.

In Table I, we present comparative results for the workspace of Figs. 1 and 2 between the proposed method and an RRT*, where 50 trials were carried out for statistical significance. Path lengths and cost function (2) values for representative trajectories are presented. In order to produce the cost values for the RRT* method (as it only produces min-length paths), we combine the latter with the closed-form solution for the optimal on-trajectory velocity $v^{\star} = \sqrt{\alpha/\beta}\|p - p_d\|$. This places our method at a disadvantage, as the RRT*'s output of quasi-linear trajectories is optimized separately and enhanced through the provably optimal norm, while our method optimizes both path shape and velocity norm concurrently. Nevertheless, our method outperforms RRT* wrt both metrics, indicating the global optimality of the method. Concerning the execution times, the average time (for the 50 RRT* trials in order to acquire the best results of Table I) was 6 and 8.4mins/traj respectively, while the total times for our method were 5 and 6mins respectively. Finally, we present the cost values of a previous reactive method [23], which is also outperformed by the herein proposed scheme.

VII. Discussion-Future Work

The proposed method is demonstrated to provide a nearly optimal reactive navigation policy, while exhibiting superior computational behavior as well as cost values and path lengths when compared to related methods. Future research efforts will focus on extending the method in multiply connected workspaces, treating non-linear and higher-order systems, as well as addressing higher-dimensional systems.

VIII. Acknowledgements

This work was supported by the European Union's Horizon 2020 Research and Innovation Program PATHOCERT - Pathogen Contamination Emergency Response Technologies under Grant 883484.

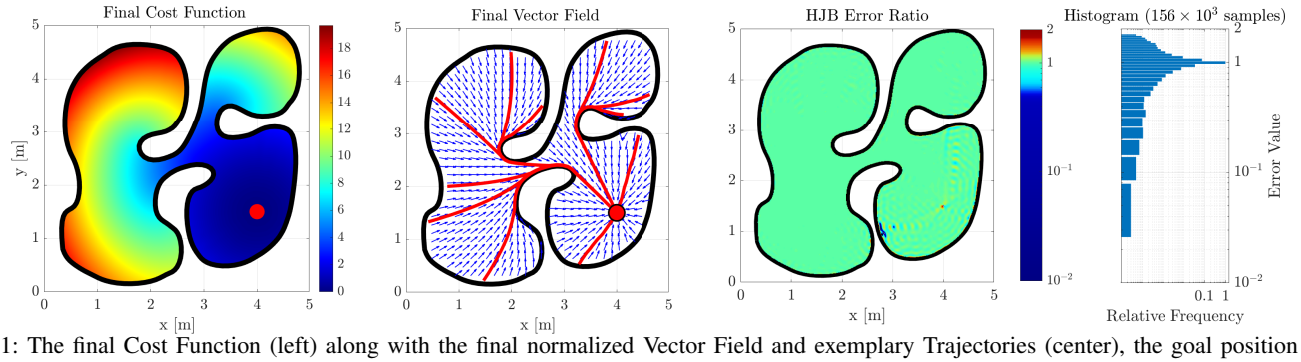

Fig. 1: The final Cost Function (left) along with the final normalized Vector Field and exemplary Trajectories (center), the goal position is depicted with a red disk. The negated HJB $^{\text{LHS}}/_{\text{RHS}}$ error is also depicted along with its corresponding Histogram.

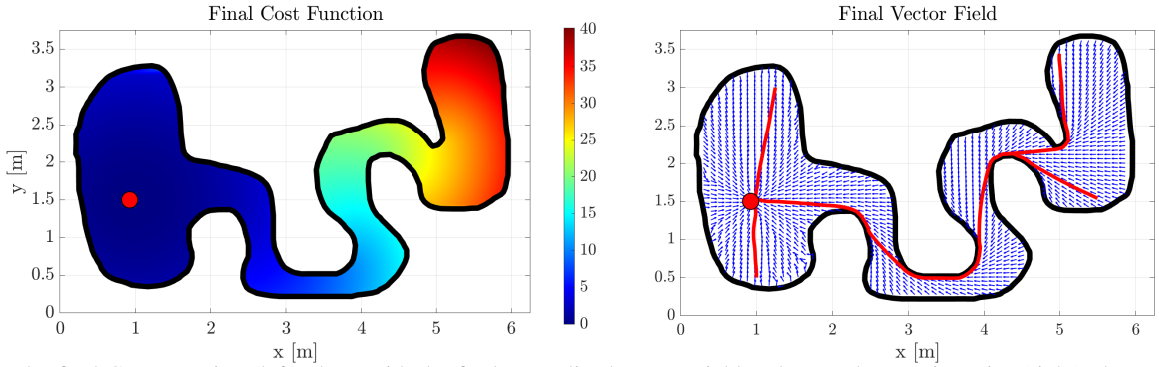

Fig. 2: The final Cost Function (left) along with the final normalized Vector Field and exemplary Trajectories (right), the goal position is depicted with a red disk.

TABLE I: RRT* Method vs Proposed Method.

$p(0)$	Length Ours	Length RRT*				Cost Ours	Cost [23]	Cost RRT*			
		Mean	Median	Min	Max			Mean	Median	Min	Max
Workspace Fig. 1											
$[1.47, 0.22]^\text{T}$	**4.23**	4.54	4.52	4.38	4.90	**13.71**	14.78	15.0	14.96	14.25	17.45
$[0.42, 1.33]^\text{T}$	**4.13**	4.38	4.38	4.26	4.57	**15.20**	15.88	16.14	16.02	15.42	17.36
$[1.99, 4.55]^\text{T}$	**4.57**	4.93	4.94	4.74	5.18	**18.39**	23.25	19.92	19.95	18.82	21.46
$[2.59, 3.54]^\text{T}$	**3.89**	4.28	4.27	4.04	4.67	**13.15**	16.3	14.56	14.43	13.46	16.21
$[4.64, 3.75]^\text{T}$	**3.47**	3.77	3.75	3.62	3.91	**10.39**	11.28	11.38	11.27	10.85	12.28
$[4.42, 2.97]^\text{T}$	**1.48**	1.57	1.57	1.54	1.64	**2.34**	2.36	2.41	2.40	**2.34**	2.60
$[3.28, 0.53]^\text{T}$	**1.16**	1.25	1.24	1.21	1.32	**1.46**	1.47	1.52	1.51	**1.46**	1.69
$[3.81, 4.76^\text{T}$	**3.50**	3.73	3.72	3.62	4.05	**11.95**	12.87	12.7	12.67	12.14	14.46
$[0.55, 3.69]^\text{T}$	**4.22**	4.55	4.55	4.41	4.81	**17.41**	21.06	18.51	18.49	17.92	19.58
Workspace Fig. 2											
$[1.0, 0.5]^\text{T}$	1.02	1.01	1.01	**1.00**	1.04	**1.00**	1.01	1.01	1.01	**1.00**	1.05
$[5.5, 1.54]^\text{T}$	6.40	6.42	6.41	**6.32**	6.58	31.08	36.34	31.40	31.35	**31.01**	32.26
$[5.0, 3.44]^\text{T}$	7.07	7.10	7.11	**7.02**	7.17	**36.90**	44.37	37.47	37.49	36.97	38.23
$[1.25, 3.0]^\text{T}$	**1.49**	1.53	1.53	1.52	1.56	**2.31**	2.38	2.33	2.32	**2.31**	2.40

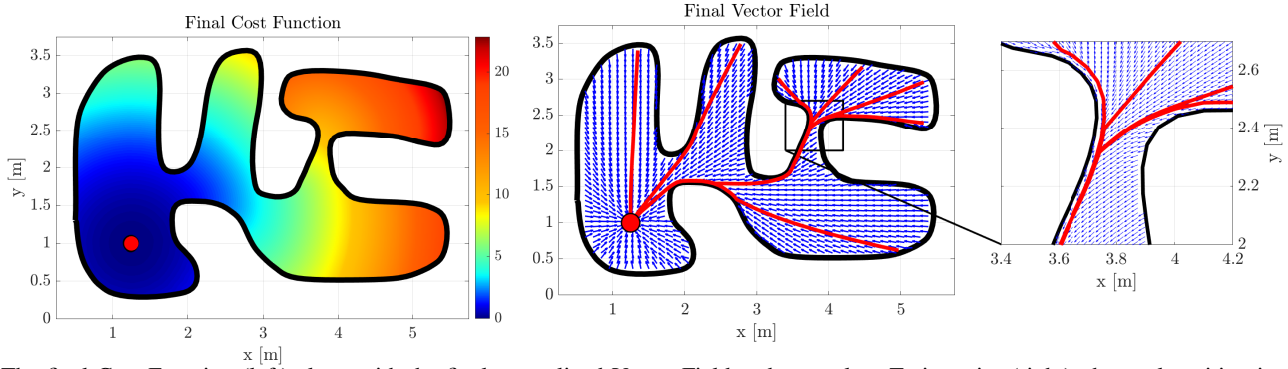

Fig. 3: The final Cost Function (left) along with the final normalized Vector Field and exemplary Trajectories (right), the goal position is depicted with a red disk.

REFERENCES

[1] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0196885890090017S

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

[3] X. Liu and D. Gong, "A comparative study of a-star algorithms for search and rescue in perfect maze," in *International Conference on Electric Information and Control Engineering*, 2011, pp. 24–27.

[4] N. Anastopoulos, K. Nikas, G. Goumas, and N. Koziris, "Early experiences on accelerating dijkstra's algorithm using transactional memory," in *Proceedings of the 2009 IEEE International Parallel and Distributed Processing Symposium*, 2009.

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

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

[7] I. Noreen, A. Khan, and Z. Habib, "Optimal path planning using rrt* based approaches: A survey and future directions," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, pp. 1–16, 2016.

[8] J. Wang, M. Q.-H. Meng, and O. Khatib, "Eb-rrt: Optimal motion planning for mobile robots," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 2063–2073, 2020.

[9] J. D. Gammell and M. P. Strub, "Asymptotically optimal sampling-based motion planning methods," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 295–318, 2021.

[10] Z. Wang, Y. Li, H. Zhang, C. Liu, and Q. Chen, "Sampling-based optimal motion planning with smart exploration and exploitation," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2376–2386, 2020.

[11] A. Francis, A. Faust, H.-T. L. Chiang, J. Hsu, J. C. Kew, M. Fiser, and T.-W. E. Lee, "Long-range indoor navigation with prm-rl," 2020, (arXiv Preprint).

[12] C. Paxton, V. Raman, G. D. Hager, and M. Kobilarov, "Combining neural networks and tree search for task and motion planning in challenging environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 6059–6066.

[13] J. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," in *IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 1991, pp. 790–796.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] S. Sundar and Z. Shiller, "Optimal obstacle avoidance based on the hamilton-jacobi-bellman equation," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 305–310, 1997.

[19] M. B. Horowitz and J. W. Burdick, "Optimal navigation functions for nonlinear stochastic systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 224–231.

[20] J. Amiryan and M. Jamzad, "Adaptive motion planning with artificial potential fields using a prior path," in *3rd RSI International Conference on Robotics and Mechatronics*, 2015, pp. 731–736.

[21] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00*, 2000, pp. 256–263.

[22] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, p. 387–424, 02 2022.

[23] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Optimal robot motion planning in constrained workspaces using reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 6917–6922.

[24] P. Rousseas, C. Bechlioulis, and K. Kyriakopoulos, "Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2005–2011, 2021.

[25] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Optimal motion planning in unknown workspaces using integral reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6926–6933, 2022.

[26] R. E. Kalman, "Contributions to the theory of optimal control." in *Bolétın de la Sociedad MatematicaMexicana*, vol. 5, 1960, pp. 102–119.

[27] M. Abu-Khalaf and F. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach," *Automatica*, vol. 41, pp. 779–791, 05 2005.

[28] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

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

[30] *Reinforcement Learning and Optimal Adaptive Control*. John Wiley and Sons, Ltd., 2012, ch. 11, pp. 461–517.

[31] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[32] R. Song, F. L. Lewis, Q. Wei, and H. Zhang, "Off-policy actor-critic structure for optimal control of unknown systems with disturbances," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1041–1050, 2016.

[33] H. Wendland, "Computational aspects of radial basis function approximation," in *Topics in Multivariate Approximation and Interpolation*, ser. Studies in Computational Mathematics, K. Jetter, M. D. Buhmann, W. Haussmann, R. Schaback, and J. Stöckler, Eds. Elsevier, 2006, vol. 12, pp. 231–256. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570579X06800108

[34] 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