2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

An Actor-Critic Reinforcement Learning Scheme for Reactive 3D Optimal Motion Planning Based on Fluid Dynamics

Marios Malliaropoulos[1], Panagiotis Rousseas[1], Charalampos P. Bechlioulis[2], and Kostas J. Kyriakopoulos[3] .

***Abstract*— This work proposes a novel and provably correct method for three-dimensional optimal motion planning in complex environments. Our approach models the 3D motion planning problem by solving streamlines of the potential fluid flow, filling a gap in traditional motion planning techniques by guaranteeing a closed-loop, smooth and natural-looking navigation solution. Special emphasis is given to an inherent challenge of artificial potential field (APF) methods, namely establishing proofs of safety and stability over the entire optimization process. A model-based actor-critic reinforcement learning algorithm is introduced to approximate the optimal solution to the Hamilton-Jacobi-Bellman equation and update the controller parameters in a deterministic manner. Through a series of ROS-Gazebo software-in-the-loop simulations the proposed methodology demonstrates robustness and outperforms widely used methods such as the RRT*, highlighting its contribution to the field of 3D optimal motion planning.**

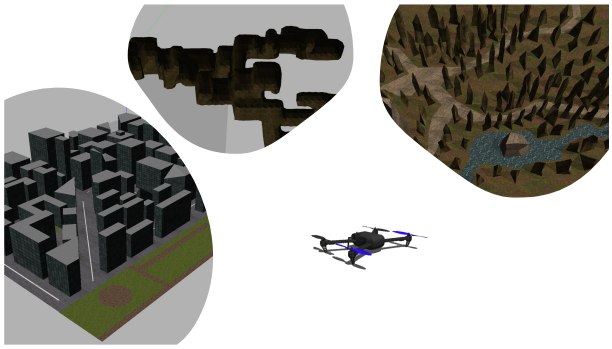

Fig. 1. The simulation environments for UAV applications of the proposed method.

I. Introduction

The motion planning problem has gathered significant attention from researchers throughout the years and is pivotal for the future of robotics [1]. Motion planners are involved in a wide range of applications, spanning from drone delivery to search and rescue missions, with the objective to generate a feasible control sequence that navigates the robot towards a desired target position. While methodologies on the motion planning problem have long been established, pursuing optimality remains challenging. So far, optimal motion planning solutions rely mainly on open-loop sampling-based methods, which in complex environments lead to discretization errors and sub-optimal results [2]. Hence, we propose a continuous-time reactive navigation method, for platforms such as unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUVs) as depicted in Fig. 1, allowing to treat optimality in a deterministic manner. In particular, we make an analogy to fluid streamlines by modeling the robot as a fluid particle that follows an ideal flow towards a sink element. This analogy is very powerful not only because it aligns the motion planning problem with the fundamental laws of physics but it also enables the integration of efficient and widely-used potential flow solvers from the aerospace field [3]. Previous nature-inspired studies appear promising [4]–[7], yet they have neither demonstrated results in realistic 3D environments nor addressed optimality formally. In this work, a reinforcement learning (RL) scheme is employed to learn the controller's optimal parameters, while control theoretical analysis enables formulating a provably safe and convergent solution. Notably, the controller acts as a high-level motion planner, where dynamic obstacles can be handled through low-level replanning techniques [8]. Additionally, the topological perplexity problem [9], [10] that imposes optimality restrictions in multiply-connected workspaces is of lesser importance for the majority of 3D navigation problems, since they can be modeled through a simply-connected workspace representation (i.e., absent of floating objects and internal cavities). Even though we examine fully known workspaces, this family of reactive methods can be expanded to unknown exploration problems, as presented in [11], [12] for the two-dimensional case.

A. Related Work

In the literature, motion planning algorithms are mainly divided into two categories: discrete open-loop methods and continuous reactive methods. Path planning in discrete environments is generally classified into graph-based-methods (GBMs) and sampling-based methods (SBMs). The former includes many popular algorithms, e.g., A* [13] and Dijkstra's algorithm [14], that construct a graph representation of the discretized space and then search for a feasible path. SBMs utilize an additional obstacle detection module to explore the environment by generating nodes in a tree-like structure, with the most popular approaches being Probabilistic Roadmaps [15] and Rapidly Exploring Random Trees (RRT) [16]. Besides lacking optimality proofs, discrete-time methods provide insufficient results or may fail to find a path at all, as the discretization resolution and the map complexity grow. A noteworthy contribution in SBMs, which

*This work was not supported by any organization

[1]The authors are with the School of Mechanical Engineering, Control Systems Laboratory, National Technical University of Athens. [2]The author is with the Department of Electrical and Computer Engineering, University of Patras. [3]The author is with the Center of AI & Robotics (CAIR), New York University Abu Dhabi. Emails: {mmalliaropoulos, prousseas}@mail.ntua.gr, chmpechl@upatras.gr, kk4812@nyu.edu

offers optimality guarantees, is the introduction of RRT* by Karaman and Frazzoli [17]. Numerous extensions of RRT* have been developed over the years aiming to generate smoother trajectories [18], address kino-dynamic constraints [19] and deal with slow convergence and high memory consumption [20], [21].

On the other hand, Artificial Potential Fields (APFs) employ continuous potential functions over the entire workspace, enabling the robot to navigate on the potential's gradient through closed-loop reactive velocity commands. The application of APFs is mainly limited due to the presence of local minima within the workspace that can trap the robot in an undesired location. To overcome this issue in two-dimensional spaces, Rimon and Koditschek introduced Navigation Functions (NF) [22] using a family of APFs that is applied in a spherical transformation of the workspace. Recent studies explored the use of Artificial Harmonic Potential fieds (AHPFs) in 2D environments [23]–[25], as they are by construction free of local minima. However, they utilize a homeomorphic transformation that can not trivially be extended to higher-dimensional cases

B. Contribution

In this work, we extend [26] to three-dimensional workspaces and present a distinct formulation of harmonic functions that can be effectively applied to any 3D workspace geometry to provide a safe and convergent motion planning solution. The novelty of this works lies also in an actor-critic reinforcement learning architecture that introduces an optimization framework for 3D state-feedback navigation. In contrast to model-free RL algorithms, the model-based nature of our approach overcomes the challenges of high sample complexity of the action and state space, memory buffer storage techniques and the exploration-exploitation trade off. By combining fluid flow equations, optimal control theory, and reinforcement learning techniques, this study offers a unique approach that inherits the positive attributes of the aforementioned paradigms.

II. Problem formulation

Consider a point robot operating within a three-dimensional bounded and simply-connected workspace $\mathcal{W} \subset \mathbb{R}^3$, along with a desired goal position $\mathbf{p_d} \in \mathcal{W}$. Also, assume that the robot has fully knowledge of the workspace and is subjected to single integrator dynamics of the form:

$$\dot{\mathbf{p}}(t) = \mathbf{u}(t), \qquad \mathbf{p}(0) = \mathbf{p_0}, \tag{1}$$

where $\mathbf{p(t)} = [x, y, z]^\top : \mathbb{R}_+ \to \mathcal{W}$ denotes the state of the robot, $\mathbf{p_0} \in \mathcal{W}$ denotes the robot's initial position and $\mathbf{u}(t)$ denotes the control policy, i.e., a velocity command input. Next, define the unconstrained infinite horizon cost function:

$$V(\mathbf{p_0}; \mathbf{p_d}) = \int_0^\infty Q\left(\mathbf{p}(\tau); \mathbf{p_d}\right) + R\left(\mathbf{u}(\tau)\right)\, d\tau, \tag{2}$$

as the weighted sum of a state-related term $Q(\mathbf{p}; \mathbf{p_d}) = \alpha\|\mathbf{p} - \mathbf{p_d}\|^2 + \gamma L(\mathbf{p})$, and an input-related term $R(\mathbf{u}) = \beta\|\mathbf{u}\|^2$, where α, β, γ are real-valued weighing parameters and $\|\cdot\|$ denotes the Euclidean norm. The term $L(\mathbf{p})$ is employed in order to penalize the robot for operating above an undesired altitude, and is motivated by UAVs operating within environments where such constraints are imposed either for maintaining visibility with operators or by existing legislation. To this end, $L(\mathbf{p})$ is:

$$L(\mathbf{p}) = \begin{cases} \exp\left(-\left(\frac{|z - z_{max}|}{|z - z_{max}| - c}\right)^2\right), & |z - z_{max}| \leq c \\ 0, & |z - z_{max}| > c \end{cases}, \tag{3}$$

where z_{max} is the altitude limit and $c \in \mathbb{R}_+$ is a positive threshold. The aim of this work is to derive a control policy $\mathbf{u}(t)$ that safely drives the robot to the desired position while at the same time minimizing the cost function (2).

III. Proposed Controller

The proposed AHPF controller consists of a continuous potential function Φ that satisfies the Laplace equation $\nabla^2\Phi = 0$. Hence, owing to the minimum-maximum principle of harmonic functions, the robot is guaranteed to converge to the goal position by following the negated gradient of the potential field. The reactive motion planning controller is defined as follows:

$$\mathbf{u}(\mathbf{p}) = -\nabla\Phi. \tag{4}$$

A. Environment Model

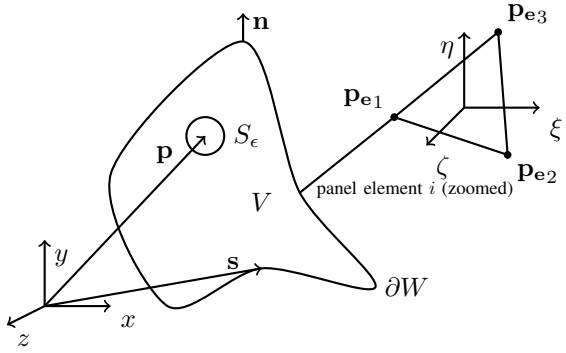

Fig. 2. Workspace modeling through panel elements.

The robot's workspace with volume V is enclosed by the boundary surface $\partial\mathcal{W}$, with its corresponding normal vector always pointing outwards, as illustrated in Fig. 2. In order to solve the Laplace equation we begin by employing Green's second identity:

$$\begin{aligned} &\iiint_V \left(\Phi_1\nabla^2\Phi_2 - \Phi_2\nabla^2\Phi_1\right)\, dV = \\ &\oiint_{\partial\mathcal{W}} \left(\Phi_1\nabla\Phi_2 - \Phi_2\nabla\Phi_1\cdot\right)\mathbf{n}\, d(\partial\mathcal{W}) \end{aligned} \tag{5}$$

and selecting $\Phi_1 = \frac{1}{\|\mathbf{p}-\mathbf{s}\|}, \Phi_2 = \Phi$. The singularity that occurs as the robot approaches a boundary point $\mathbf{s} \in \partial\mathcal{W}$

is treated by excluding from the workspace the volume enclosed by a small spherical surface S_e, and Eq. (5) becomes:

$$0 = \oiint_{\partial\mathcal{W}} (\Phi_1 \nabla\Phi_2 \cdot \mathbf{n} - \Phi_2 \nabla\Phi_1 \cdot \mathbf{n})\, d(\partial\mathcal{W}) - \oiint_{S_\epsilon} (\Phi_1 \nabla\Phi_2 \cdot \mathbf{n}_\epsilon - \Phi_2 \nabla\Phi_1 \cdot \mathbf{n}_\epsilon)\, d(\partial\mathcal{W}) \Rightarrow$$
$$\Phi = \frac{1}{4\pi} \oiint_{\partial\mathcal{W}} \left(\frac{1}{\|\mathbf{p}-\mathbf{s}\|} \nabla\Phi \cdot \mathbf{n} - \Phi \nabla \frac{1}{\|\mathbf{p}-\mathbf{s}\|} \cdot \mathbf{n} \right) d(\partial\mathcal{W})$$
$$= \frac{1}{4\pi} \oiint_{\partial\mathcal{W}} \left(\frac{\partial\Phi}{\partial\mathbf{n}}(\mathbf{s}) \frac{1}{\|\mathbf{p}-\mathbf{s}\|} - \Phi(\mathbf{s}) \frac{(\mathbf{p}-\mathbf{s})\cdot\mathbf{n}(\mathbf{s})}{\|\mathbf{p}-\mathbf{s}\|^3} \right) d(\partial\mathcal{W}). \tag{6}$$

Eq. (6) does not pose a unique representation of the potential field, resulting in a policy that is paremeterized based on the chosen boundary conditions $\frac{\partial\Phi}{\partial n}(\mathbf{s})$ and $\Phi(\mathbf{s})$, i.e., the weight factors of the fundamental solutions. Analogous to fluid mechanic applications, the first term in (6) corresponds to a distribution of sources-sinks and the second term to a distribution of doublets on the workspace boundary. In the context of motion planning, we choose to construct a repulsive potential through a distribution of source-sink elements, thus we set $\Phi(\mathbf{s}) = 0$. Furthermore, a point sink is placed at the goal position to render the potential field attractive, which results in the following expression for the AHPF:

$$\Phi(\mathbf{p};\mathbf{p_d};\mathbf{s}) = \frac{1}{4\pi} \oiint_{\partial\mathcal{W}} \left(\frac{\partial\Phi}{\partial\mathbf{n}}(\mathbf{s}) \frac{1}{|\mathbf{p}-\mathbf{s}|} \right) d(\partial\mathcal{W}) - \frac{\partial\Phi}{\partial\mathbf{n}}(\mathbf{p_d}) \frac{1}{4\pi|\mathbf{p}-\mathbf{p_d}|}. \tag{7}$$

B. Workspace Discretization

Having constructed the fundamental harmonic terms, an analytical expression for the potential field is derived by discretizing the workspace surface ∂W into two-dimensional triangular panels [27]. Each panel $i = 1, \ldots, K$ is defined by the position of the triangle's vertices $\mathbf{p_e}_j^{(i)} \in \mathbb{R}^3$, $j = 1, \ldots, 3$, as depicted in Fig 2, and is associated with a constant weight factor w_i. The induced total velocity is then given by:

$$\nabla_p \Phi(\mathbf{p};\mathbf{p_d};\mathbf{s}) = \sum_{i=1}^{K} \left(\frac{1}{4\pi} w_i \oint_{\Delta\partial\mathcal{W}_i} \frac{\mathbf{p}-\mathbf{s}}{|\mathbf{p}-\mathbf{s}|^3}\, d(\partial\mathcal{W}) \right) + w_0 \frac{\mathbf{p}-\mathbf{p_d}}{4\pi|\mathbf{p}-\mathbf{p_d}|^3}. \tag{8}$$

C. Control Policy Structure

The parameterized controller takes the following form:

$$\mathbf{u}(\mathbf{p}) = -\nabla_p \Phi(\mathbf{p};\mathbf{p_d};\mathbf{p_e}) = -\mathbf{w}^\top \mathbf{v}(\mathbf{p};\mathbf{p_d};\mathbf{p_e}), \tag{9}$$

where $\mathbf{w} \triangleq \left[w_0, w_1, \ldots w_K\right]^\top \in \mathbb{R}^{K+1}$ are the tunable parameters of the control policy, which correspond to the basis function:

$$\mathbf{v}(\mathbf{p};\mathbf{p_d};\mathbf{p_e}) \triangleq [\mathbf{v}_0(\mathbf{p};\mathbf{p_d}), \mathbf{v}_1(\mathbf{p};\mathbf{p_e}), \ldots, \ldots, \mathbf{v}_K(\mathbf{p};\mathbf{p_e})]^\top : \mathbb{R}^3 \to \mathbb{R}^{K+1}. \tag{10}$$

The first term represents the attractive sink $\mathbf{v}_0(\mathbf{p};\mathbf{p_d})^\top = \frac{1}{4\pi}\frac{\mathbf{p}-\mathbf{p_d}}{|\mathbf{p}-\mathbf{p_d}|^3}$, while the rest of the terms $\mathbf{v}_i(\mathbf{p};\mathbf{p_e}) = \frac{1}{4\pi}\oint_{\Delta\partial\mathcal{W}_i}\frac{\mathbf{p}-\mathbf{s}}{|\mathbf{p}-\mathbf{s}|^3}\, d(\partial\mathcal{W}), i = 1, \ldots, K$ form a distribution of repulsive harmonic terms on the workspace boundary. In order to calculate the integral over each panel's surface, we utilize a homogenous transformation matrix ${}^0\mathbf{T}_i$ from the global coordinate system $O_0(x, y, z)$ to the local coordinate system $O_i(\xi, \eta, \zeta)$, centered on the panel's centroid with $\xi, \eta \perp \mathbf{n}$ and $\zeta \parallel \mathbf{n}$. This way, the contribution of each panel on the robot's velocity is analytically calculated as follows:

$$\mathbf{v}_i(\mathbf{p};\mathbf{p_e}) = \left[v_{i_x}(\mathbf{p};\mathbf{p_e}^{(i)}), v_{i_y}(\mathbf{p};\mathbf{p_e}^{(i)}), v_{i_z}(\mathbf{p};\mathbf{p_e}^{(i)})\right] = {}^i\mathbf{T}_0 \cdot \left[v_{i_\xi}(\mathbf{p}';\mathbf{p'_e}^{(i)}), v_{i_\eta}(\mathbf{p}';\mathbf{p'_e}^{(i)}), v_{i_\zeta}(\mathbf{p}';\mathbf{p'_e}^{(i)})\right]^\top. \tag{11}$$

By expressing the transformed robot and the panel edges position as $\mathbf{p}' = \left[\xi, \eta, \zeta\right] \in \mathcal{W}$ and $\mathbf{p'_e}^{(i)} = \left[\xi_j^{(i)} \eta_j^{(i)} \zeta_j^{(i)}\right] \in \partial\mathcal{W}$ respectively, the velocity components induced by the i^{th} panel w.r.t. its local coordinate system are given as:

$$v_{i_\xi} = \frac{1}{4\pi}\left(S_{12}^{(i)} Q_{12}^{(i)} + S_{23}^{(i)} Q_{23}^{(i)} + S_{31}^{(i)} Q_{31}^{(i)}\right), \tag{12}$$
$$v_{i_\eta} = \frac{1}{4\pi}\left(-C_{12}^{(i)} Q_{12}^{(i)} - C_{23}^{(i)} Q_{23}^{(i)} - C_{31}^{(i)} Q_{31}^{(i)}\right), \tag{13}$$
$$v_{i_\zeta} = \frac{1}{4\pi}\left(J_{12}^{(i)} + J_{23}^{(i)} + J_{31}^{(i)}\right), \tag{14}$$

where:

$$S_{ab}^{(i)} = \frac{\eta_b^{(i)} - \eta_a^{(i)}}{d_{ab}^{(i)}},$$
$$C_{ab}^{(i)} = \frac{\xi_b^{(i)} - \xi_a^{(i)}}{d_{ab}^{(i)}},$$
$$Q_{ab}^{(i)} = \ln \frac{r_a^{(i)} + r_b^{(i)} - d_{ab}^{(i)}}{r_a^{(i)} + r_b^{(i)} + d_{ab}^{(i)}},$$
$$J_{ab}^{(i)} = \tan^{-1}\left(\frac{m_{ab}^{(i)} e_a^{(i)} - h_a^{(i)}}{\zeta r_a^{(i)}}\right) - \tan^{-1}\left(\frac{m_{ab}^{(i)} e_b^{(i)} - h_b^{(i)}}{\zeta r_b^{(i)}}\right),$$
$$d_{ab}^{(i)} = \sqrt{(\xi_b^{(i)} - \xi_a^{(i)})^2 + (\eta_b^{(i)} - \eta_a^{(i)})^2},$$
$$m_{ab}^{(i)} = \frac{\eta_b^{(i)} - \eta_a^{(i)}}{\xi_b^{(i)} - \xi_a^{(i)}},$$
$$r_a^{(i)} = \sqrt{(\xi - \xi_a^{(i)})^2 + \zeta^2},$$
$$e_a^{(i)} = (\xi - \xi_a^{(i)})^2 + \zeta^2,$$
$$h_a^{(i)} = (\xi - \xi_a^{(i)})(\eta - \eta_a^{(i)}).$$

D. Safety and Convergence Criteria

In order to provide a suitable navigation solution, we first define a set of admissible control policies.

Definition 1 (Admissible Policy): A control policy u(p) is defined as admissible with respect to the cost function (2) on $\mathcal{W}$ if $u(p)$ is continuous on $\mathcal{W}$, $u(p_d) = 0$, $u(p)$ stabilizes

(1) on $\mathcal{W}$, $V(p)$ is finite $\forall p \in \mathcal{W}$, and the trajectories under the control $u(p)$ are safe.

To address collision avoidance, we impose the following safety constraints to the proposed motion planning policy:

$$\mathbf{n}(\mathbf{p})^\top \mathbf{u}(\mathbf{p}) \leq 0, \forall \mathbf{p} \in \partial\mathcal{W}, \tag{15}$$

which dictates that the velocity command at the boundary always points inwards. As proven in [26], this property can be relaxed if the safety condition holds for a finite set of boundary points. Hence, the safety condition (15) can be expressed as a set of N linear inequalities with respect to the controller parameters:

$$\mathbf{A} \cdot \mathbf{w} \leq -\epsilon \mathbf{I}_{(N+1)\times 1} \Rightarrow \begin{bmatrix} A_1^0 & A_1^1 & \dots & A_1^K \\ A_2^0 & A_2^1 & \dots & A_2^K \\ \vdots & \vdots & \ddots & \vdots \\ A_N^0 & A_N^1 & \dots & A_N^K \\ -1 & 0 & \dots & 0 \end{bmatrix} \cdot \begin{bmatrix} w_0 \\ \vdots \\ w_K \end{bmatrix} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}, \tag{16}$$

where $A_b^i = \mathbf{n}^\top(\mathbf{p}_b)\mathbf{v}_i(\mathbf{p_b}; \mathbf{p_e})$, $i = 0, \dots, K$ corresponds to the i^{th}'s harmonic term influence on the induced normal velocity at the boundary point $\mathbf{p}_b \in S$, $b = 1, \dots, N$ and ϵ is a small positive scalar constant. Besides the safety criteria, note that matrix $\mathbf{A}$ is augmented to include the constraint $w_0 > 0$, which addresses the convergence of the robot towards the goal position. An initial motion planning policy can be extracted by solving the following problem:

$$\begin{aligned} \mathbf{w}^0 &= \arg\min_{\mathbf{w}} \{\mathbf{w}^\top \mathbf{w}\} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{w} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}. \end{aligned} \tag{17}$$

The parameters $\mathbf{w}^0$ establish a control policy that satisfies both the safety and the convergence requirements, which will be henceforth referred to as an admissible policy per Def. 1. The metric in (17) is trivial and serves the purpose of initializing the optimization algorithm, as described in Section IV. Notice that there exists various libraries with efficient solvers for the aforementioned quadratic optimization problem with linear constraints.

IV. Optimal Control Policy

To tackle the optimal motion planning problem we define the Hamiltonian:

$$H(\mathbf{p}, \mathbf{u}, \nabla_p V) = \nabla_p V^\top \mathbf{u} + Q(\mathbf{p}; \mathbf{p_d}) + R(\mathbf{u}), \tag{18}$$

which yields the following Hamilton-Jacobi-Bellman (HJB) equation for the optimal value function V^*:

$$H(\mathbf{p}, \mathbf{u}^*, \nabla_p V^*) = 0. \tag{19}$$

The closed-form solution for the optimal control policy is then given by the stationarity condition $\frac{\partial H(\mathbf{p}, \mathbf{u}, \nabla_p V^*)}{\partial u}|_{\mathbf{u}=\mathbf{u}^*} = 0$ as follows:

$$\mathbf{u}^* = -\frac{1}{2\beta} \nabla_p V^*. \tag{20}$$

Obtaining the optimal policy $\mathbf{u}^*$ requires an analytical expression of the value function V^*. However, this entails solving a hard non-linear partial differential equations problem. To mitigate these challenges we employ a reinforcement learning algorithm, which successively approximates the solution of the HJB equation, effectively merging the advantages of optimal control theory and machine learning techniques.

A. Reinforcement Learning Framework

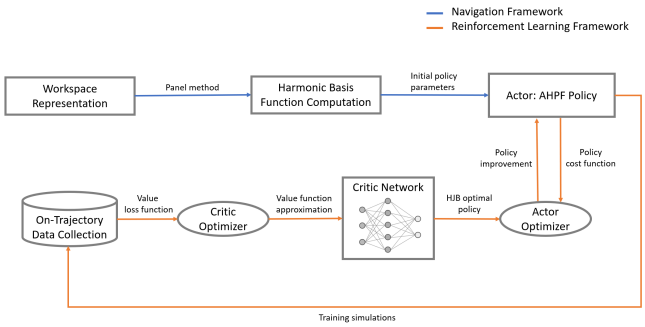

Fig. 3. The proposed optimal motion planning framework using an actor-critic reinforcement learning architecture.

The proposed method introduces an off-policy, continuous, model-based reinforcement learning algorithm originating from policy iteration [28], [29]. The algorithm is structured in an actor-critic fashion, where the critic Deep Neural Network (DNN) is responsible for approximating the value function (2), while the actor successively learns the optimal control parameters without violating the admissibility criteria. Our methodology is schematically illustrated in Figure 3. In particular, the iterative optimization procedure is initialized with an admissible control policy, as presented in Section III. By utilizing the model-based nature of this approach, multiple simulations are executed and on-trajectory samples of the value function (2) under the current control policy (9) are collected. Subsequently an approximation for the value function is derived using the critic DNN. The trained critic network is fed into the actor optimizer and, alongside with the optimal policy of the HJB equation (20), forms a quadratic constrained minimization problem to update the controller's parameters.

B. Proposed Critic Structure

As a universal approximator for the cost function (2), a multi-layer fully connected DNN $\hat{V}(\phi)$ is employed, where ϕ are the network's tunable parameters. To construct the training dataset of the critic network, an iterative simulation process takes place, adjusting the robot's initial position until a set of position-cost function pairs is uniformly distributed within the workspace. The accumulated cost $\overline{V}(\mathbf{p}; \mathbf{p_d})$ from any initial position $\mathbf{p_0}$ to a position $\mathbf{p}$ along the robot's trajectory is computed by solving the following system of

ordinary differential equations:

$$\begin{aligned}\frac{d\mathbf{p}(t)}{dt} &= \mathbf{u}(\mathbf{p}(t)),\\ \frac{d\overline{V}(\mathbf{p}(t);\mathbf{p_d})}{dt} &= Q\left(\mathbf{p}(t);\mathbf{p_d}\right) + R\left(\mathbf{u}(\mathbf{p}(t))\right),\end{aligned} \tag{21}$$

with initial conditions $\mathbf{p}(0) = \mathbf{p_0}$ and $\overline{V}(\mathbf{p_0};\mathbf{p_d}) = 0$. According to Bellman's principle of optimality the value function for any on-trajectory point $\mathbf{p}$ is derived as follows:

$$V(\mathbf{p};\mathbf{p_d}) = \overline{V}(\mathbf{p_d};\mathbf{p_d}) - \overline{V}(\mathbf{p};\mathbf{p_d}). \tag{22}$$

Note that the optimal solution of the HJB equation (18) necessitates an expression for the positional derivative of the value function, which is derived analytically in a back-propagating fashion.

C. Proposed Actor Structure

Having approximated the cost function gradient, we continue by providing the policy improvement step, where the AHPF (9) is employed as an actor structure. In each iteration k, an improved policy is obtained by minimizing the following expression:

$$\begin{aligned}\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}}\{\frac{1}{2}\|\mathbf{u}^{(k)} - \mathbf{u}^{*(k)}\|^2\}\\ \text{s.t.} \quad \mathbf{A}\mathbf{w} \leq -\epsilon\mathbf{I}_{(N+1)\times 1}.\end{aligned} \tag{23}$$

The actor optimizer (23) is formulated such that the improved policy strives to reach as close to the unconstrained optimal solution of the HJB equation (20) while at the same time satisfying the admissibility criteria (16). The metric in (23) can be expanded to a quadratic form as follows:

$$\begin{aligned}&\frac{1}{2}\|\mathbf{u}^{(k)} - \mathbf{u}^{*(k)}\|^2 =\\ &\frac{1}{2}\mathbf{w}^\top(-\mathbf{v}(\mathbf{p}))(-\mathbf{v}(\mathbf{p})^\top)\mathbf{w} - \left(\frac{1}{2\beta}\nabla_p\hat{V}(\mathbf{p})^{(k)}\right)^\top\mathbf{v}(\mathbf{p})\mathbf{w} +\\ &\left(-\frac{1}{2\beta}\nabla_p\hat{V}(\mathbf{p})^{(k)}\right)^\top\left(-\frac{1}{2\beta}\nabla_p\hat{V}(\mathbf{p})^{(k)}\right),\end{aligned} \tag{24}$$

which is minimized over a number of samples $\mathbf{p_m} \in \mathbb{R}^3, m = 1,\dots,L$ within the workspace. Substituting (24) into (23) yields the following constrained quadratic optimization problem to update the actor parameters in each step:

$$\begin{aligned}\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}}\{\frac{1}{2}\mathbf{w}^\top\overline{\mathbf{H}}(\mathbf{p})\mathbf{w} + \overline{\mathbf{B}}(\mathbf{p})^{(k)}\mathbf{w}\}\\ \text{s.t.} \quad \mathbf{A}\mathbf{w} \leq -\epsilon\mathbf{I}_{(N+1)\times 1},\end{aligned} \tag{25}$$

where:

$$\begin{aligned}\overline{\mathbf{H}}(\mathbf{p}) &= \sum_{m=1}^{L}\left(\mathbf{H}(\mathbf{p_m})^\top\mathbf{H}(\mathbf{p_m})\right) \in \mathbb{R}^{(K+1)\times(K+1)},\\ \overline{\mathbf{B}}(\mathbf{p})^{(k)} &= \sum_{m=1}^{L}\left(-\mathbf{H}(\mathbf{p_m})^\top\mathbf{B}(\mathbf{p_m})^{(k)}\right)^\top \in \mathbb{R}^{1\times(K+1)},\\ \mathbf{H}(\mathbf{p}) &= -\mathbf{v}(\mathbf{p}),\\ \mathbf{B}(\mathbf{p})^{(k)} &= -\frac{1}{2\beta}\nabla_p\hat{V}(\mathbf{p})^{(k)}.\end{aligned} \tag{26}$$

The algorithm of the optimal motion planning problem is summarized in Algorithm 1. Notably, as the selected number of panels K increases, the convergence of the algorithm as well as the resolution of the motion planning policy improve, at the expense of an increase in computational time.

Algorithm 1 Optimal Motion Planning Algorithm

1: Discretize the workspace boundary $\partial\mathcal{W}$ into K total triangular panels.
2: Select a number of N control points $\mathbf{p}_b \in S$, $b = 1,\dots,N$ on the boundary surface and construct the safety set of linear inequalities $\mathbf{A}$ (16).
3: Find an initial admissible policy $\mathbf{u}^{(0)}(\mathbf{p}) = -\mathbf{w}^{(0)\top}\mathbf{v}(\mathbf{p};\mathbf{p_d};\mathbf{p_e})$ by solving (17). The basis function $\mathbf{v}$ is computed through (11), (12), (13) and (14).
4: Set $k \leftarrow 0$
5: **while** $\sum_{i=1}^{K}\frac{1}{K}|w_i^{(k)} - w_i^{(k+1)}|$ *has not converged* **do**
- **Model-Based Simulations**: Define a set of initial positions to run multiple simulations under the control policy $\mathbf{u}^{(k)}(\mathbf{p})$ and collect a set of H total on-trajectory samples $\{((\mathbf{p}_1, V(\mathbf{p}_1;\mathbf{p_d})),\cdots,(\mathbf{p}_H, V(\mathbf{p}_H;\mathbf{p_d}))\}$ using (21), (22).
- **Critic**: Approximate the critic network by obtaining the DNN parameters ϕ that minimize the loss function:

$$\phi = \arg\min_{\phi}\frac{1}{H}\sum_{h=1}^{H}\left(V(\mathbf{p_h}) - \hat{V}(\mathbf{p_h};\phi)\right)^2 \tag{27}$$

- **Actor**: Select a sample of L points within the workspace and compute $\overline{\mathbf{H}}(\mathbf{p}), \overline{\mathbf{B}}(\mathbf{p})^{(k)}$ through (26). Update the actor parameters:

$$\begin{aligned}\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}}\{\frac{1}{2}\mathbf{w}^\top\overline{\mathbf{H}}(\mathbf{p})\mathbf{w} + \overline{\mathbf{B}}(\mathbf{p})^{(k)}\mathbf{w}\}\\ \text{s.t.} \quad \mathbf{A}\mathbf{w} \leq -\epsilon\mathbf{I}_{(N+1)\times 1}.\end{aligned} \tag{28}$$

Acquire an improved policy: $\mathbf{u}^{(k+1)}(\mathbf{p}) = -\mathbf{w}^{(k+1)\top}\mathbf{v}(\mathbf{p};\mathbf{p_d};\mathbf{p_e})$.
- $k \leftarrow k+1$

6: **end while**

V. Technical Results

In this section a technical analysis for the proposed methodology is carried out. In particular, we establish the existence of the solution for the motion planning problem and conduct a stability analysis for the corresponding controller. Additionally, we prove that every control policy of the successive approximation scheme satisfies Def. 1.

Lemma 1 (Existence of Solution): There exists a set of control parameters $\mathbf{w}$ for the proposed motion planning policy $\mathbf{u} = -\nabla\Phi$ (9), such that: **1)** the potential field satisfies the Laplace equation $\nabla^2\Phi = 0$ and **2)** the problem constraints (16) are satisfied.

Proof: Assume a set of parameters $\mathbf{w}$ that provide a Laplacian potential field by definition, as presented in Section III-A. We will prove that under certain conditions $\mathbf{w}$ also satisfies the motion planning constrains $\mathbf{Aw} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}$. Owing to Gauss's theorem:

$$\oiint_{\partial \mathcal{W}} (\mathbf{u} \cdot \mathbf{n}) \, d(\partial \mathcal{W}) = \iiint_{\mathcal{W}} (\nabla \cdot \mathbf{u}) \, d\mathcal{W}, \tag{29}$$

and substituting the control policy (9) in (29) results in:

$$\oiint_{\partial \mathcal{W}} (-\nabla \Phi \cdot \mathbf{n}) \, d(\partial \mathcal{W}) = \iiint_{\mathcal{W}} -\nabla^2 \Phi \, d\mathcal{W} \Rightarrow$$
$$\oiint_{\partial \mathcal{W}} (-\nabla \Phi \cdot \mathbf{n}) \, d(\partial \mathcal{W}) = 0. \tag{30}$$

Eq. (30) represents the compatibility principle and indicates that the flux of the vector field that traverses a loop enclosed by the workspace boundary is zero. This implies that if the vector field, due to the safety constraints, points inwards toward the workspace, a sink term within the workspace is essential to compensate for the fulfillment of (30). By excluding from the workspace a closed contour $\mathcal{G}$ around the goal position, (30) becomes: $\oiint_{\partial \mathcal{W}} (-\nabla \Phi \cdot \mathbf{n}) \, d(\partial \mathcal{W}) - \oiint_{\mathcal{G}} (-\nabla (\Phi \cdot \mathbf{n}) \, d\mathcal{G} = 0 \Rightarrow \oiint_{\partial \mathcal{W}} (-\nabla \Phi \cdot \mathbf{n}) \, d(\partial \mathcal{W}) - \oiint_{\mathcal{G}} (-\nabla (\Phi_{Repulsive} \cdot \mathbf{n}) \, d\mathcal{G} - w_0 = 0$, where $\Phi_{Repulsive}$ denotes the potential field of the boundary panels and w_0 is the flux of the point source through the spherical surface around the goal position. Moreover, the safety and convergence criteria (16) impose the following conditions:

$$\oiint_{\partial \mathcal{W}} (-\nabla \Phi \cdot \mathbf{n}) \, d(\partial \mathcal{W}) > 0, \tag{31}$$
$$w_0 > 0. \tag{32}$$

By substituting the compatibility principle in (31) and (32) we conclude that a solution for the path planning problem exists if:

$$w_0 > \oiint_{\mathcal{G}} (\mathbf{u}_{Repulsive} \cdot \mathbf{n}) \, d\mathcal{G} > 0 \tag{33}$$

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

Lemma 2 (Stability Analysis): The controller $\mathbf{u}(\mathbf{p})$ (9) stabilizes the system (1) in the desired position $\mathbf{p}_d$.

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

$$\lim_{p \to p_d} \Phi(\mathbf{p}) = \lim_{p \to p_d} \left(\frac{w_0}{4\pi} \frac{1}{\|\mathbf{p} - \mathbf{p_d}\|} \right) = -\infty, \tag{34}$$

which results in:

$$\lim_{p \to p_d} \exp(\Phi(\mathbf{p})) = \lim_{\Phi \to -\infty} \exp(\Phi) = 0. \tag{35}$$

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

$$\frac{d\Psi}{dt} = \frac{d\Psi}{d\Phi} \frac{d\Phi}{dp} \frac{dp}{dt} = \exp(\Phi(\mathbf{p})) \nabla \Phi^\top \mathbf{u}(\mathbf{p}) \overset{(9)}{=}$$
$$- \exp(\Phi(\mathbf{p})) \nabla \Phi^\top \nabla \Phi = - \exp(\Phi(\mathbf{p})) |\nabla \Phi|^2 \tag{36}$$

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

Lemma 3 (Policy Update Admissibility): Consider the admissible control policy $u^{(k)}$ (9) at the k-th iteration step along with the corresponding value function $V^{(k)}$ (2). The improved policy $u^{(k+1)}$ that is obtained following the proposed methodology will be admissible.

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

- Since the motion planning policy strictly preserves the AHPF-structure (9) in every iteration step, $u^{(k+1)}$ is continuous.
- According to **Lemma** 2 $u^{(k+1)}$ stabilizes (1) on the goal position p_d.
- Finiteness of the cost function along with the requirement $u^{(i+1)}(p_d) = 0$ follow from asymptotic convergence of the robot towards the target.
- The set of linear inequalities $\mathbf{Aw} \leq -\epsilon \mathbf{I}_{(N+1)\times 1}$ is integrated as hard constraints in the policy update step (25), thus rendering the navigation safe with respect to boundary collisions.

∎

VI. Results

In this section, we present a simulation study for a broad-range of applications, namely the navigation in: 1) a complex urban environment, 2) a tunnel-maze environment with narrow passages, and 3) a densely populated forest map.

A. Comperative Study with RRT*

All simulations were implemented in MATLAB on a Windows 10 PC with an eight-core Ryzen 7 CPU and 32GB of RAM. An extensive comparison study against RRT* is carried out with 25 trials for each initial position. In order to evaluate the cost function for RRT* we set the optimal velocity norm as $\frac{\alpha}{\beta} \|p - p_d\|$, which imposes a more strict baseline for this comparison since the proposed method optimizes both the path shape and the velocity magnitude simultaneously. The constant parameters of the cost function (2) are chosen as $\alpha, \beta, \gamma = 0.04$ for all workspaces. The UAV trajectories from different initial positions are illustrated in Fig. 4, Fig. 5, and Fig. 6, while the corresponding costs and path lengths are summarized in Table I. It is evident that the proposed motion planning algorithm outperforms the RRT* method both in terms of path lengths and in terms of cost w.r.t. a quadratic form infinite horizon value function. Regarding the computational time, our method appears to be 10% faster compared to the total run-time of the considered RRT* trajectories.

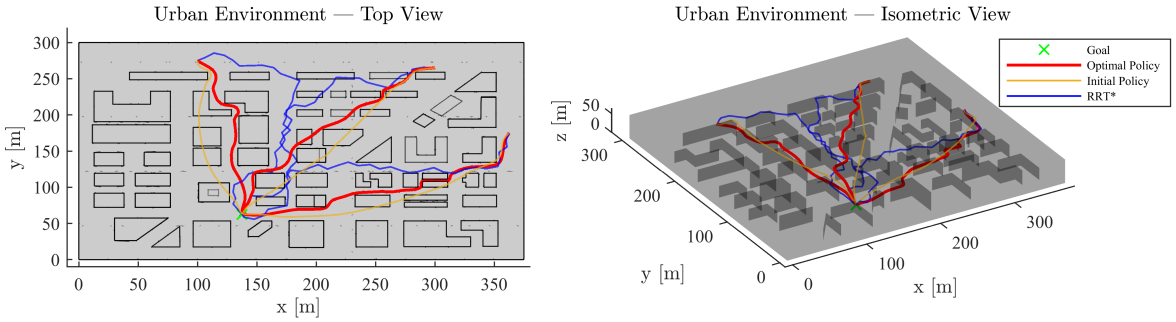

Fig. 4. Comparative trajectories in an urban environment from various initial positions. The figure depicts the initial trajectory (orange line), the optimal trajectory (red line) of the proposed method and the optimal trajectory of the RRT* method (blue line).

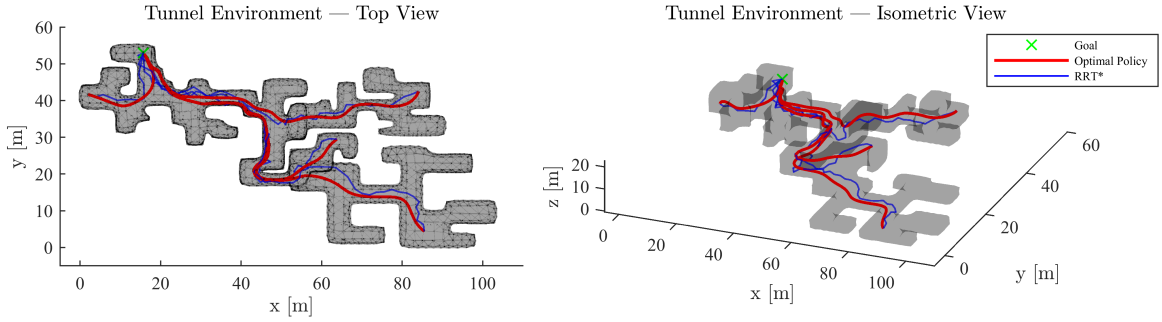

Fig. 5. Comparative trajectories in a tunnel environment from various initial positions. The figure depicts the the optimal trajectory (red line) of the proposed method and the optimal trajectory of the RRT* method (blue line).

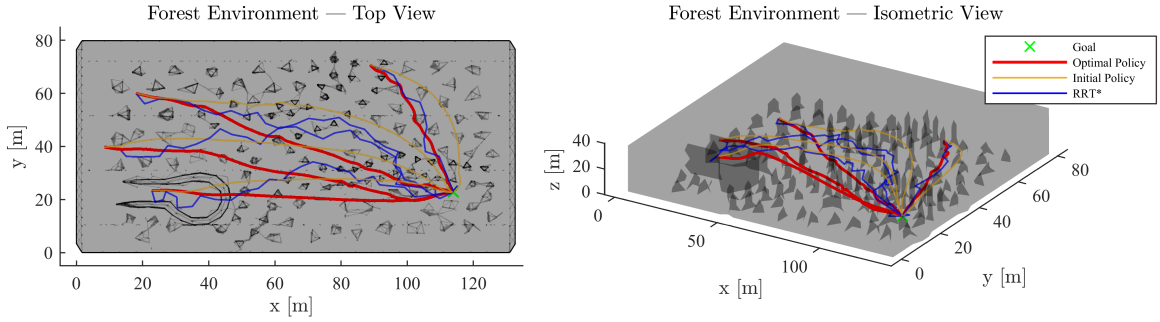

Fig. 6. Comparative trajectories in a forest environment from various initial positions. The figure depicts the initial trajectory (orange line), the optimal trajectory (red line) of the proposed method and the optimal trajectory of the RRT* method (blue line).

TABLE I
COMPERATIVE RESULTS

p(0)	Distance Traveled [m]					Cost				
	Ours	RRT*				Ours	RRT*			
		Mean	Med.	Min	Max		Mean	Med.	Min	Max
Urban Environment Fig. 4										
[362,175,12.5]	**290**	391	372	338	605	20.2	**16.4**	14.7	13.4	28.3
[100,275,7.5]	**233**	437	402	326	660	**15.6**	19.1	15.7	12.0	34.2
[300,265,25]	**291**	404	389	350	541	22.0	**17.8**	16.9	14.9	26.1
Tunnel Environment Fig. 5										
[85,4,2.5]	**105**	144	143	130	157	**10.0**	12.0	12.0	10.9	13.1
[84,42,20]	**82**	105	105	97	113	**5.7**	7.4	7.5	6.6	8.0
[64,30,12]	**79**	105	103	96	123	**6.0**	6.7	6.7	6.1	7.74
Forest Environment Fig. 6										
[8,40,3]	**111**	157	152	127	240	**9.6**	11.2	10.8	9.2	16.3
[18,60,3]	**105**	150	146	131	202	**8.5**	10.4	10.2	8.8	13.0
[89,71,3	**56**	100	94	64	192	**2.9**	4.2	3.7	2.5	9.8
[23,23,12]	**100**	174	166	125	258	**8.1**	14.2	12.3	8.6	24.9

B. ROS-Gazebo Simulations

The proposed motion planning controller is employed in a high-fidelity ROS-Gazebo-Ardupilot environment using an Extended Kalman Filter to fuse GPS, barometer, and IMU data for state estimation. We incorporate sensors perception zero-mean Gaussian white noise with a standard deviation ranging from 0.05 to 0.7 meters, while external disturbances are modeled using the Von Dryden wind gust model [30] for wind speeds ranging from 1 to 10 m/s. To evaluate the controller's robustness, a parametric study conducts 15 simulations in each environment for various standard deviation pairs of perception noise and wind disturbance. As depicted in Fig. 7 both the traveled distance and the cost function remain consistent even for high values of sensors noise and external disturbances. A video of the SIL validation is avail-

able in this link: https://youtu.be/OWUBSgkhoDI.

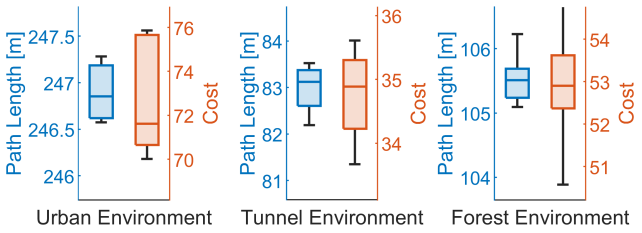

Fig. 7. Box plot of the path length and cost over 15 Gazebo simulations in each workspace for various perception noise and actuator disturbance levels.

VII. Conclusion - Future Work

In this work we present a complete framework for optimal motion planning in 3D environments. Owing to the reactive nature of this approach we are able to integrate a novel actor-critic RL structure and seek optimality in a deterministic manner. Additionally, the analogy of the motion planner with fluid particles guarantees safe and smooth trajectories that converge towards the target from **anywhere** within the workspace. A comparison with a widely-used path planning method that has proven asymptotically optimal, namely RRT*, demonstrates an advancement towards achieving global optimality by simultaneously optimizing the robot motion for all possible states. An important element of this work is our emphasis on ensuring stability and safety. This is achieved by strictly maintaining a harmonic basis function throughout the entire optimization process, thereby restricting the margin for achieving globally optimal solutions. Exploring future directions includes implementing complex formulations of the harmonic potential field either through a higher order weight distribution on the panels or by adding doublet and vortex elements. Moreover, model-free RL methods appear promising to address the trade-off between a harmonic planning policy and global optimality.

References

[1] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, and et al., "The grand challenges of *Science Robotics*," *Science Robotics*, vol. 3, no. 14, p. eaar7650, 2018.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *ICLR*, 2016.

[3] J. Katz and A. Plotkin, *Low-Speed Aerodynamics*, 2nd ed., ser. Cambridge Aerospace Series. Cambridge University Press, 2001.

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

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

[6] D. Keymeulen and J. Decuyper, "The fluid dynamics applied to mobile robot motion: the stream field method," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 378–385 vol.1.

[7] Y. Wang and G. Chirikjian, "A new potential field method for robot path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 977–982 vol.2.

[8] R. E. Allen and M. Pavone, "A real-time framework for kinodynamic planning in dynamic environments with application to quadrotor obstacle avoidance," *Robotics and Autonomous Systems*, vol. 115, pp. 174–193, 2019.

[9] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "State-feedback optimal motion planning in the presence of obstacles," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8406–8413, 2023.

[10] M. D. Kvalheim and D. E. Koditschek, "Necessary conditions for feedback stabilization and safety," *Journal of Geometric Mechanics*, pp. 659–693, 2022.

[11] 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.

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

[13] D. Foead, A. Ghifari, M. B. Kusuma, N. Hanafiah, and E. Gunawan, "A systematic literature review of a* pathfinding," *Procedia Computer Science*, vol. 179, pp. 507–514, 2021, 5th International Conference on Computer Science and Computational Intelligence 2020.

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

[15] 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*, 1996.

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

[17] 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.

[18] D. Lee, H. Song, and D. H. Shim, "Optimal path planning based on spline-rrt* for fixed-wing uavs operating in three-dimensional environments," in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, 2014, pp. 835–839.

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

[20] O. Arslan and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal motion planning," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2421–2428.

[21] F. Islam, J. Nasir, U. Malik, Y. Ayaz, and O. Hasan, "Rrt∗-smart: Rapid convergence implementation of rrt∗ towards optimal solution," in *2012 IEEE International Conference on Mechatronics and Automation*, 2012, pp. 1651–1656.

[22] 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.

[23] 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.

[24] 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.

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

[26] P. Rousseas, C. Bechlioulis, and K. J. 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.

[27] F. T. Johnson, "A general panel method for the analysis and design of arbitrary configurations in incompressible flows," NASA, Tech. Rep., 1980.

[28] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

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

[30] T. Beal, "Digital simulation of atmospheric turbulence for dryden and von karman models," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 1, pp. 132–138, 1993.