




An Efficient Solution to Optimal Motion Planning With Provable Safety and Convergence

PANAGIOTIS ROUSSEAS ¹ (Student Member, IEEE),
CHARALAMPOS BECHLIOLIS ² (Senior Member, IEEE), AND KOSTAS KYRIAKOPOULOS ³ (Fellow, IEEE)

¹Control Systems Lab, School of Mechanical Engineering, National Technical University of Athens, 11144 Athens, Greece

²Division of Systems and Control, School of Electrical and Computer Engineering, University of Patras, 26504 Patra, Greece

³Center of AI & Robotics (CAIR), New York University, Abu Dhabi 129188, UAE

CORRESPONDING AUTHOR: PANAGIOTIS ROUSSEAS (e-mail: prousseas@mail.ntua.gr)

This work was supported by the Hellenic Foundation for Research and Innovation (HFRI) through the 4th Call for HFRI Ph.D. Fellowships.
[Fellowship Number: 9110].

ABSTRACT An innovative solution to the optimal motion planning problem is presented in this work. A novel parametrized actor structure is proposed, which guarantees safe and convergent navigation by construction. Concurrently, an efficient scheme for optimizing a mixed state and energy cost function is formulated. The proposed method inherits the positive traits of continuous methods, while at the same time providing sub-optimal –but close to optimal– results significantly faster and in more complex workspaces than previous ones. The scheme is demonstrated to outperform established relevant methods, while at the same time being competitive w.r.t. execution time. Extensive simulations to validate the effectiveness of the method are presented, along with relevant technical proofs for safety and convergence.

INDEX TERMS Optimal motion planning, optimization and optimal control.

I. INTRODUCTION

The field of Robotics has been at the forefront of engineering applications, from industrial ones, such as manufacturing and logistics, to civil ones, such as transportation of goods, search and rescue and research. Technological advances in both software and hardware continue to widen the scope of robotic applications, as real-world operation becomes the norm. In this context, Motion Planning (MP) and path planning, remain some of the most fundamental aspects of Robotics; the execution of almost any desired task in the real world requires safe navigation through physical space. The “embodied intelligence” aspect, present in any robotic platform, is a crucial feature that distinguishes such systems from other engineering disciplines [1].

It is thus evident why MP has been at the core of Robotics research, since the infancy of the field. Many solutions have been presented over the years, with two main classes emerging: Sampling-Based Methods (SBMs) and Continuous

Methods (CMs). SBMs have been widely employed in industry and research, owing to their relative non-complexity in implementation and effectiveness in providing fast asymptotically optimal solutions. On the contrary, CMs lack a universally accepted and adopted treatment of Optimal MP (OMP), in part due to tunability issues or due to computational complexity issues.

This work builds upon previous works and aims at providing an efficient and competitive CM for tackling the OMP problem in planar workspaces. Significant emphasis is placed on providing provable guarantees of safety and convergence, while at the same time overcoming the computationally intensive solution of an optimal control problem in complex geometries. The method is demonstrated to be competitive w.r.t. both cost function performance and execution time, while at the same time inheriting the advantages of state-feedback control (e.g., robustness, convergence from any initial condition, etc.).

A. RELATED WORK

As discussed previously, SBMs [2] and discrete/graph-based methods have been widely embraced by the community. Notable examples are Dijkstra's algorithm [3], A* [4], D*, D*-lite [5], Random Rapidly exploring Trees (RRT) [6], RRT* [7], Probabilistic Road Maps (PRMs) [8], Fast Marching Trees (FMT*) [9] and Visibility Graphs (VGs) [10]. Most importantly, many of the aforementioned methods have been significantly extended; for instance, RRT* extensions treat smart re-planning [11], dynamic workspaces [12], control barrier functions formulations [13], kino-dynamic planning [14] and heuristics to improve efficiency, such as RRT*-SMART [15], informed RRT* [16], etc. In treating optimality, Dijkstra's Algorithm, A*, D* and RRT* produce minimum length quasi-linear paths. More specifically, methods in the RRT* class, provide provable asymptotic optimality.

In contrast to SBMs, CMs treat MP from a control-theoretic perspective, through the design of continuous vector fields for state-feedback. Most commonly, such fields are extracted from real-valued potential functions defined over the robot's workspace. Seminal examples include Navigation Functions (NFs) [17] and Artificial Potential Fields (APFs), which focus on designing safe and convergent fields through appropriately designed potentials. Artificial Harmonic Potential Fields (AHPFs) [18] were later introduced in order to nullify tunability issues of seminal CMs, where the properties of harmonic functions imbue such potentials with safety and convergence by construction. Nevertheless, such methods are applied in the context of disk-worlds, thus, extending AHPFs to arbitrary workspaces, requires homeomorphic maps of arbitrary connected sets into such disk worlds [19], [20], [21]. More recently, the need for such homeomorphisms has been nullified through defining harmonic potentials over the physical workspace directly [22], [23], [24].

However, *continuous* OMP lacks solutions as widely adopted as SBMs, with only few works [25], [26], [27] that require solving a hard, non-linear Partial Differential Equation (PDE), or extensive parameter tuning. The authors have thus far focused on a mixed approach based on Policy Iteration (PI) [22], [28], [29], leveraging the properties of AHPFs, limiting however the space of possible policies. Our most recent works [30], [31], concentrate on a parameter-free solution, asymptotically yielding a globally optimal solution and overcoming the limitations of parametrizing the policy through AHPFs. Thus, solutions were extended to more complex dynamics [30] and more crucially, the Topological Perplexity (TP) [32] aspect of optimal control on multiply connected sets, which is vital for optimality, was addressed in [33]. Nevertheless, while these methods are competitive w.r.t. path length and cost function value they remain relatively computationally complex; even though CMs result in an infinite number of trajectories (solving the problem globally for any initial position), we believe that there is room for improvement.

Finally, Machine Learning (ML) methods have also been developed (an extensive review is provided in [34]), however

most such methods treat a single robotic configuration. For example, solutions such as [35], [36], [37] require high-level velocity commands, therefore, our class of methods is still required, serving as the necessary high-level planner. Thus, similar ML approaches differ in scope compared to this work, where we aim at solving the kinematic OMP problem in its general formulation.

B. MOTIVATION AND CONTRIBUTIONS

The proposed framework aims at leveraging the positive traits of CMs, while providing an efficient scheme for OMP. These positive traits become apparent when considering realistic applications; while SBMs are able to extract optimal paths (or even time-parametrized paths, i.e., trajectories), these are usually in the form of open-loop quasi-linear waypoints. While path-smoothing solutions have been developed [38] and even in cases where dynamics are accounted for through extracting kino-dynamically feasible paths [14], such solutions remain open-loop. Therefore, when applying such methods in realistic conditions, low-level planners are still required, and robotic platforms will most likely deviate from the nominal path.

In contrast, CMs, even in their simplest formulations, provide smooth velocity commands, which are more suitable for state (position)-feedback control of robotic platforms (e.g. [39]). Having a smooth velocity command for any position within the workspace renders real-world planning simpler, and opens up possibilities for reactive controllers for unseen and uncertain obstacles using the information encoded in the existing vector field.

Given the above context, special care is taken to provide provable guarantees of safety and convergence. Compared to the authors' previous works, the herein proposed method can be understood as a mid-point between the AHPF-parametrized fields [22], [23], [28], [29] and the non-parametric solutions [30], [31], [33]. Furthermore, the proposed scheme is computationally superior, while at the same time providing good performance compared to the asymptotically globally optimal solution [31], [33].

The main contributions of this work are outlined below:

- 1) A novel parametrized velocity field (actor structure), with provable guarantees of safety and convergent,
- 2) Incorporating the above actor structure into an efficient PI scheme for extracting the optimal velocity field,
- 3) A novel, fast and efficient mesh-free PDE solver, serving as an alternative way of calculating the cost function of a given policy in the context of the PI scheme.

II. NOTATION

The notation employed throughout the manuscript is as follows:

- Throughout the manuscript the matrix-notation is employed for defining vector valued functions and related mathematical objects, as well as the operations between them. Sub-section IV-F is an exception, where

the component-wise notation is employed. Hence operators (such as ∇) have their usual, multi-variate calculus meaning.

- Sets are usually defined through calligraphic characters (e.g., \mathcal{W}) with the exception of $\mathcal{T}(\cdot)$ which is a mapping. The sets $\Theta, \Theta_A, \Omega, \Omega_A$ do not follow this rule due to lack of Greek calligraphic symbols. Numerical sets are denoted through their classical symbols, i.e., \mathbb{R} for the set of real numbers. Notably, \mathbb{R}_+ denotes the set of real-positive numbers. Constant matrices are denoted through bold capital letters.
- Latin characters correspond to functions and constants -vectors or scalars-. This is further clarified in context, where each function is defined along with its argument and mapping domains. Constants are defined through their domain as well.
- Subscripts/superscripts are used in two ways: Either to denote different components of vector-valued functions (see Φ_1, Φ_2 for instance), or at times to denote dependence on some other object (e.g., $p_u(\cdot)$ is defined as a function that *implicitly* depends on the input signal u). This differentiates between the explicit domain of some function and other implicit dependencies. This is specifically clarified in text following the definition of any relevant object.
- The superscript “ T ” is an exception, denoting the transpose of a quantity.
- The symbol $\{\cdot, \cdot\}$ denotes a tuple.
- The parameters α (Greek letter) and a (Latin letter) are different (see footnote 3).

More specifically, $\mathcal{W}, \partial\mathcal{W}$ denote the workspace and its boundary, $p \in \mathcal{W}$ denotes the robot’s position and $p_d \in \mathcal{W}$ denotes the desired final position. The field v denotes a reference velocity field with $g(p; v)$ denoting a set of basis vectors based on the reference field. Finally, Φ denotes a multi-valued parametrized function.

III. PROBLEM FORMULATION

Consider a point robot,¹ operating within a bounded and fully connected workspace $\mathcal{G} \subset \mathbb{R}^2$ containing a number $M \in \mathbb{N}$ of non-overlapping internal obstacles denoted by $\mathcal{O}_m \subset \mathcal{G}, m \in \mathcal{M}$, where $\mathcal{M} = \{1, \dots, M\}$. Thus, the feasible workspace of the robot is denoted by $\mathcal{W} = \mathcal{G} \setminus \bigcup_{m \in \mathcal{M}} \mathcal{O}_m \subseteq \mathcal{G}$, and its boundary is denoted by $\partial\mathcal{W}$. Finally, consider a desired, final position $p_d \in \text{int}(\mathcal{W})$, where $\text{int}(\mathcal{W}) = \mathcal{W} \setminus \partial\mathcal{W}$. In this work, we adopt the single integrator model, i.e.:

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

where $p(t) : \mathbb{R}_+ \mapsto \mathcal{W}$ denotes the robot’s position, $\bar{p} \in \mathcal{W}$, denotes the robot’s initial position, and $u(t) : \mathbb{R}_+ \mapsto \mathbb{R}^2$ denotes the input (velocity) signal. The aim of this work is to find the state feedback input signal $u(t) \triangleq u(p(t))$ that minimizes

the cost function:

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

where $p_u(t; \bar{p})$ denotes the solution of (1) under the control input u starting from the initial position \bar{p} . The **state-related cost term** P and the **input-related cost term** R are defined as:

$$P(p; p_d) = \alpha \|p - p_d\|^2, \quad (3a)$$

$$R(u) = \beta \|u\|^2, \quad (3b)$$

where $\alpha, \beta \in \mathbb{R}_+$ are design/specification parameters and $\|\cdot\|$ is the Euclidean norm. Effectively, (3a) minimizes settling time, while (3b) minimizes energy expenditure.

In the context of the MP problem, the solution that minimizes (2) should also respect *safety constraints* and *asymptotic global convergence*, i.e.:

$$p_u(t; \bar{p}) \in \text{int}(\mathcal{W}), \quad \forall t > 0, \quad \forall \bar{p} \in \mathcal{W} \quad (4a)$$

$$\lim_{t \rightarrow \infty} (p_u(t; \bar{p})) = p_d, \quad \forall \bar{p} \in \mathcal{W}. \quad (4b)$$

Thus, we define an admissible input as follows:

Definition 1 (Admissible Control): A control signal $u \in \mathcal{U}_{\mathcal{W}}$, where $\mathcal{U}_{\mathcal{W}}$ denotes the set of *admissible inputs* for the workspace \mathcal{W} , is defined as the *reactive* input under which:

- 1) System (1) obeys (4),
- 2) The cost function (2) is continuous and finite for any initial position $p \in \mathcal{W}$.

IV. PROPOSED METHOD

In this section, the proposed method will be presented for the case of simply-connected workspaces. In [33], we have presented a methodology that transforms workspaces with obstacles to ones without obstacles, while also providing close to optimal results. Therefore, while workspaces with obstacles will be presented in the results Section, we will not go into further detail here, directing the reader to [33].

A. PROPOSED ACTOR STRUCTURE

In this work, we propose a novel parametrized actor structure (controller), based on safe and convergent velocity fields. Consider the admissible velocity field $v(p) \in \mathcal{U}_{\mathcal{W}}$ over a simply-connected workspace \mathcal{W} . This field defines a curvilinear coordinate transformation $\{\tau, s\} = \mathcal{T}(p) : \mathcal{W} \mapsto [0, 1] \times \mathcal{S}^1 = \mathcal{D}^1$. The inverse of the transformation is defined as:

$$p = \mathcal{T}^{-1}(\{\tau, s\}) = \int_0^\tau v(p(t)) dt + \partial\mathcal{W}(s), \quad (5)$$

where $\partial\mathcal{W}(s) : \mathcal{S}^1 \mapsto \partial\mathcal{W}$ is a parametrization of the Jordan curve that consists the boundary of the workspace, through the variable s . Eq. (5) can be understood as choosing the initial point on the workspace’s boundary (through s) whose trajectory at time τ intersects the point p . This transformation can be shown to be a Homeomorphism [30], thus its inverse is well-defined.

A parametric controller can be designed as:

$$u(p; v; \theta) = g(p; v)\Phi(p; \theta) : \mathcal{W} \times \mathcal{U}_{\mathcal{W}} \times \Theta \mapsto \mathbb{R}^2, \quad (6)$$

¹A disk robot with radius $R > 0$ can also be considered though inflating the workspace boundary by a distance equal to R .

where $\theta \in \Theta \subseteq \mathbb{R}^m$ denotes a set of m actor parameters, $\Phi(p; \theta) : \mathcal{W} \times \Theta \mapsto \mathbb{R}^2$ denotes a parametrization structure, e.g., a Neural Network and $g(p; v) : \mathcal{W} \times \mathcal{U}_{\mathcal{V}} \mapsto \mathbb{R}^{2 \times 2}$ denotes a vector basis:

$$g(p; v) = [\hat{v}(p), \hat{v}_{\perp}(p)]. \quad (7)$$

where $\hat{v}(p)$ and $\hat{v}_{\perp}(p)$ denote the co-linear vector to the velocity field v and a modified normal vector to the velocity field v respectively. The vector basis (7) defines two directions, which correspond to the radial and tangential ones of the transformed coordinates $\{s, \tau\}$ (5).

B. SAFETY AND CONVERGENCE

The controller of Subsection IV.IV-A is employed to provide a provably convergent and safe policy. We begin by providing the form of the modified normal vector $\hat{v}_{\perp}(p)$, which is given by:

$$\hat{v}_{\perp}(p) = h(p) (\mathbf{R}\hat{v}(p)), \quad (8)$$

where $\mathbf{R} = \begin{bmatrix} 0, & -1 \\ 1, & 0 \end{bmatrix}$ denotes the 90° rotation matrix, and

$$h(p) = \begin{cases} 1 - \exp\left(-\left(\frac{d(p)}{d(p)-a}\right)^2\right), & d(p) \leq a \\ 1, & d(p) > a \end{cases}, \quad (9)$$

with $a \in \mathbb{R}_+^2$ while the function $d : \mathcal{W} \mapsto \mathbb{R}_+$ computes the distance of the robot to the boundary:

$$d(p) = \min_{z \in \partial\mathcal{W}} \{\|p - z\|\}. \quad (10)$$

The bump function $h(p)$ 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. Therefore, (8) smoothly nullifies the normal vector to $\hat{v}(p)$, at a distance to the boundary lesser than a . In order to render the actor (6) safe, it suffices that:

$$u(p; v; \theta) \propto \hat{v}(p) \forall p \in \partial\mathcal{W}, \quad (11)$$

since the velocity field $v \in \mathcal{U}_{\mathcal{V}}$ is safe by definition. This is a conservative approach –there exist safe policies that are not aligned with v at the boundary–, however, this formulation significantly simplifies the safety constraints imposed on the actor structure. More specifically, denoting $\Phi(p; \theta) = [\Phi_1(p; \theta), \Phi_2(p; \theta)]^T$, $\Phi_i(p; \theta) : \mathcal{W} \times \Theta \mapsto \mathbb{R}$, $i = 1, 2$, a sufficient condition for safety is: $\Phi_1(z; \theta) > 0, \forall z \in \partial\mathcal{W}$. According to the above decomposition of Φ , the actor structure becomes:

$$\begin{aligned} u(p; v; \theta) &= g(p; v)\Phi(p; \theta) = [\hat{v}(p), \hat{v}_{\perp}(p)] \begin{bmatrix} \Phi_1(p; \theta) \\ \Phi_2(p; \theta) \end{bmatrix} \\ &= \Phi_1(p; \theta)\hat{v}(p) + \Phi_2(p; \theta)\hat{v}_{\perp}(p), \end{aligned} \quad (12)$$

²The use of the English letter a in (9) is not to be mistaken for the Greek letter α in (3a).

which is employed in the following propositions.

Proposition 1 (Actor Safety): The condition $\Phi_1(z; \theta) > 0, \forall z \in \partial\mathcal{W}$ is sufficient to satisfy safety of the velocity field (actor) (6).

Proof: A sufficient condition for safety is:

$$p^T \hat{n}(z) > 0, \forall z \in \partial\mathcal{W}, \quad (13)$$

where $\hat{n}(z) : \partial\mathcal{W} \mapsto \mathcal{S}^1$ denotes the inwards-pointing unitary vector that is normal to the boundary at the point z . Combining (1) and (13) yields:

$$\begin{aligned} u^T(z; v; \theta) \hat{n}(z) &> 0 \stackrel{(6)}{\iff} \\ \hat{n}^T(z)g(z; v)\Phi(z; \theta) &> 0 \stackrel{(8),(12)}{\iff} \\ \Phi_1(z; \theta) (\hat{n}^T(z)\hat{v}(z)) &> 0, \forall z \in \partial\mathcal{W}, \end{aligned} \quad (14)$$

since $\hat{v}_{\perp}(z) = h(z)(\mathbf{R}\hat{v}(z)) = 0, \forall z \in \partial\mathcal{W}$. However, note that $v \in \mathcal{U}_{\mathcal{V}}$, hence $(\hat{n}^T(z)\hat{v}(z)) > 0$ by definition, thus:

$$\Phi_1(z; \theta) > 0, \forall z \in \partial\mathcal{W}, \quad (15)$$

which concludes the proof. While in this proof, the condition appears to be necessary and sufficient, the necessity stems from Eq. (11). ■

We conclude this subsection by providing the conditions for Global Asymptotic Stability (GAS) of the actor:

Proposition 2 (Actor GAS): Assuming $v \in \mathcal{U}_{\mathcal{V}}$ in (6) is a conservative vector field, i.e., $v(p) = -\nabla\Psi(p)$, $\Psi : \mathcal{W} \mapsto \mathbb{R}_+$, System (1) under the input (6) is GAS, if:

$$\begin{aligned} \Phi_1(p; \theta) &> 0, \forall p \in \mathcal{W}, \\ \Phi_2(p; \theta) &\text{ is bounded, } \forall p \in \mathcal{W}. \end{aligned} \quad (16)$$

Proof: Given the field $v \in \mathcal{U}_{\mathcal{V}}$, consider its associated potential function $\Psi(p)$. This is a valid Lyapunov candidate function, since $\Psi(p) \geq 0, \forall p \in \mathcal{W}$, $\Psi(p) = 0 \iff p = p_d$. Consider now the derivative of $\Psi(p)$, for System (1), under the input (6):

$$\begin{aligned} \dot{\Psi}(p) &= \nabla^T \Psi(p) \dot{p} = \nabla^T \Psi(p) g(p; v) \Phi(p; \theta) \stackrel{(12)}{=} \\ \Phi_1(p; \theta) (\nabla^T \Psi(p) \hat{v}(p)) &+ \Phi_2(p; \theta) (\nabla^T \Psi(p) \hat{v}_{\perp}(p)) \stackrel{(8)}{=} \\ \Phi_1(p; \theta) \nabla^T \Psi(p) \hat{v}(p) &+ h(p) \Phi_2(p; \theta) \nabla^T \Psi(p) \mathbf{R} \hat{v}(p) = \\ -\Phi_1(p; \theta) \|\nabla \Psi(p)\|^2 &- h(p) \Phi_2(p; \theta) \nabla^T \Psi(p) \mathbf{R} \nabla \Psi(p). \end{aligned} \quad (17)$$

However, through the definition of \mathbf{R} :

$$\nabla^T \Psi(p) \mathbf{R} \nabla \Psi(p) = -\partial_x \Psi \partial_y \Psi + \partial_x \Psi \partial_y \Psi = 0, \quad (18)$$

and (17), assuming $\Phi_2(p; \theta)$ is bounded, yields:

$$\dot{\Psi}(p) = -\Phi_1(p; \theta) \|\nabla \Psi(p)\|^2. \quad (19)$$

Hence, $\Phi_1(p; \theta) \Rightarrow \dot{\Psi}(p) \leq 0, \forall p \in \mathcal{W}$, and $\dot{\Psi}(p) = 0 \iff p = p_d$, and under this condition, System (1) is proven to be GAS, concluding the proof. ■

Finally, we note that in order to render the cost function (2) continuous and finite, the solution to the ODE (1) under the

input (6) should be well-defined. To ensure this, the function $\Phi(p; \theta)$ should be Lipschitz continuous. In summary, if:

- 1) $\Phi(p; \theta)$ is Lipschitz continuous,
- 2) $\Phi_1(p; \theta) > 0, \forall p \in \mathcal{W}$,

then the actor structure (6) is *admissible* as per Def. 1. Since the function Φ is parametrized, the above conditions may be enforced through appropriate constraints on the parameters $\theta \in \Theta$. Thus, we denote as $\Theta_{\mathcal{A}} \subseteq \Theta$ the set of admissible parameters, and any $\theta \in \Theta_{\mathcal{A}}$ will henceforth be called an *admissible parameter vector*.

Remark 1 (Non-Conservative Actor): Note that, while the reference velocity field is conservative, the resulting actor structure is necessarily not so. First, the conservative term $\hat{v}(p)$ is multiplied by the scalar Φ_1 , yielding a non-conservative term. Most importantly however, the normal term \hat{v}_{\perp} , which results from rotating the conservative term by $\pi/2$, results in imbuing the actor's vector field with *vorticity*. This is a crucial aspect of the proposed actor, which previous works by the authors [22], [28], [29] lacked, and yields the close-to-optimal results presented in this work.

C. POLICY ITERATION

Having formulated a provably safe and convergent actor structure, we present a PI scheme that enables optimizing the parameters of the latter. The PI scheme is based upon *admissible policies*, hence Propositions 1 and 2 should hold in order to render the scheme implementable. **Assuming admissibility**, the cost function (2) assumes a differential form that is employed to extract the following Hamiltonian [40]:

$$H(p, u, \nabla V) = \nabla^T V_u(p)u + P(p; p_d) + R(u). \quad (20)$$

The minimum cost function is thus extracted through solving the following problem:

$$\min_u \{H(p, u, \nabla V^*)\} = 0, \quad (21)$$

where V^* denotes the optimal (minimum) cost function. By applying the stationary condition to the Hamiltonian (20), the optimal input is extracted:

$$u^* = -\frac{1}{2\beta} \nabla V^*(p). \quad (22)$$

As discussed extensively in [31], [40], [41], [42], in order to avoid solving the Hamilton Jacobi Bellman (HJB) Partial Differential Equation (PDE), a PI scheme is employed to successively improve the input, via the cost function approximation. Briefly, starting from a policy (input) $u^{(i=0)} \in \mathcal{U}_{\mathcal{W}}$, its associated cost $V^{(i=0)} \triangleq V_{u^{(i=0)}}$ is approximated. Afterwards, the policy is updated as:

$$u^{(i+1)}(p) = -\frac{1}{2\beta} \nabla V^{(i)}(p). \quad (23)$$

This scheme can be proven to converge asymptotically to the optimal input/cost function pair, [17], [40], even in case of safety constraints [31]. However, in our case we will employ

the actor structure instead of (23), inspired by actor-critic schemes.

Consider a critic approximation structure, denoted by $C(p; \omega) : \mathcal{W} \times \Omega \mapsto \mathbb{R}_+$, where $\omega \in \Omega \subseteq \mathbb{R}^n$ denotes a set of n critic parameters. Then, the associated cost function of an input $u^{(i)} \in \mathcal{U}_{\mathcal{W}}$ can be approximated as:

$$C^{(i)}(p) \triangleq C(p; \omega^{(i)}) = V^{(i)}(p) + \epsilon^{(i)}(p), \quad p \in \mathcal{W} \quad (24)$$

where $\epsilon^{(i)}(p) : \mathcal{W} \mapsto \mathbb{R}$ denotes the approximation error, through the HJB equation:

$$\begin{aligned} H(p, u, \nabla C^{(i)}(p)) &= 0 \Rightarrow \\ \nabla^T C^{(i)}(p)u^{(i)}(p) + P(p; p_d) + R(u^{(i)}(p)) &= 0, \end{aligned} \quad (25)$$

which can be solved numerically as:

$$\omega^{(i)} = \arg \min_{\omega \in \Omega} \{ \|\nabla^T C(p; \omega)u^{(i)} + P + R\|^2 \}, \quad (26)$$

$\forall p \in \mathcal{W}$, where the dependence on p and $u^{(i)}$ was dropped for brevity. Then, the actor structure (6) can be updated through the following optimization problem:

$$\theta^{(i+1)} = \arg \min_{\theta \in \Theta_{\mathcal{A}}} \left\{ \left\| u(p; v; \theta) + \frac{1}{2\beta} \nabla C(p; \omega^{(i)}) \right\|^2 \right\}, \quad (27)$$

$\forall p \in \mathcal{W}$. Note that the parameters in (27) are restricted to the set of admissible parameter vectors, which is crucial for the PI scheme to be well-defined. The above scheme entails obtaining an approximation for the current cost function through the critic network, and afterwards optimizing the actor's parameters to "match" the best implementable policy, as encoded in the latest cost function gradient. The scheme is then repeated until convergence, i.e., $\|\theta^{(i+1)} - \theta^{(i)}\| \leq E$, where $E > 0$.

Remark 2 (Differentiability of the Cost Function): In general, the cost function (2), as a solution to the Hamilton-Jacobi-Bellman equation might not be continuous, let alone differentiable. Even in case of the single integrator dynamics for instance, internal obstacles imply that the cost function is continuous everywhere, but not differentiable in a lesser-dimensional subset of the workspace. However, in our case, the single integrator model along with simple connectivity of the workspace, render (2) differentiable everywhere. We direct the reader to [33] for a more detailed discussion on the topological properties of AHPFs.

1) SINGLE-LAYERED PARAMETRIZATION STRUCTURES

The actor and critic parametrization structures in (27), (26) can assume many forms. In practice, we have observed that linear regressors through an appropriate basis function vector selection –e.g. Radial Basis Functions (RBFs)– work quite well, as problems (27), (26) become Constrained Quadratic Optimization problems. There exist a plethora of methods for solving the latter with bounded solution time [43]. At the same time, the safety and convergence constraints for the actor structures become linear constraints, as the set $\Theta_{\mathcal{A}}$ becomes a

convex polytope. More specifically, for such linear regressors:

$$\begin{aligned}\Phi(p; \theta) &= \phi^T(p)\theta, \\ C(p; \omega) &= \psi^T(p)\omega,\end{aligned}\quad (28)$$

where $\phi(p) : \mathcal{W} \mapsto \mathbb{R}^m$, $\psi(p) : \mathcal{W} \mapsto \mathbb{R}^n$, the weight update problems (26), (27) become:

$$\omega^{(i)} = \arg \min_{\omega \in \Omega} \{ \|A_C^T \omega + B_C\|^2 \}, \quad (26^*)$$

and

$$\theta^{(i+1)} = \arg \min_{\theta \in \Theta_A} \{ \|A_a^T \theta + B_a\|^2 \}, \quad (27^*)$$

where A_C , A_a , B_C , B_a are matrices that are computed by sampling the expressions (26), (27) over the workspace. Concerning safety, if RBFs are employed, positivity of Φ_1 can be sufficiently guaranteed through positivity of all of the parameters, i.e., the set Θ_A becomes:

$$\Theta_A = \{ \theta_j > 0, \forall j = \{1, 2, \dots, m\} \}, \quad (29)$$

where $\theta = [\theta_1, \theta_2, \dots, \theta_m]^T$, which renders (27*) a linearly constrained optimization problem.

2) NEURAL NETWORK PARAMETRIZATIONS

In case of NNs, safety and convergence can be ensured through appropriately choosing the final activation function for Φ_1 , e.g., ReLU or softPLus, such that the former is positive definite. Afterwards, Eqs. (27), (26) can be solved through back-propagation. Additionally, for the assumptions of Proposition 1 to hold, the NN should be Lipschitz continuous, which is, in general, not true. Nevertheless, a choice of smooth activation functions and the fact that the NN's domain is bounded, suffice to imply that the NN will be Lipschitz. Furthermore, recently, imposing Lipschitz continuity explicitly has been investigated [44]. Hence, the set Θ_A is arbitrary, according to the NN's parameters' definition.

3) INITIAL POLICY

The presented PI scheme requires an initial policy, which can be extracted through the actor structure (6). Consider (1) under the velocity field $v \in \mathcal{U}_V$:

$$\dot{p} = v(p). \quad (30)$$

The field $v(p)$ can be extracted by methods such as in [22], [23], [28], [29] as the gradient of an AHPF. This further satisfies the assumption of Proposition 2, i.e., that the initial velocity field is conservative. Note that Eq. (30) is equivalent to the following actor structure:

$$\begin{aligned}u^{(0)} &\triangleq u(p; v; \theta^{(0)}), \\ \text{such that: } \Phi(p; \theta^{(0)}) &= [1, 0]^T,\end{aligned}\quad (31)$$

which is identical to (30). Regardless of the choice for the actor parametrization structure Φ (e.g., linear regressor or NN), this can be trivially achieved by including a constant term to the structure and fixing the relevant parameter, while setting the rest of the parameters equal to zero.

D. CONTROL IMPROVEMENT

In this sub-section, the control improvement aspect of the proposed scheme is investigated w.r.t. the approximation error of the actor structure. In the best case, Eq. (22) holds exactly. In practice however, approximation errors arise due to two factors: 1) The approximation of the cost function is subject to errors itself, 2) Due to the employed actor structure, the error in (27) might be nonzero. Both effects are incorporated into a single error function as:

$$u^{(i+1)}(p) \triangleq u(p; v; \theta^{(i+1)}) = -\frac{1}{2\beta} \nabla V^{(i)}(p) + \varepsilon(p), \quad (32)$$

where $\varepsilon(p) : \mathcal{W} \mapsto \mathbb{R}^2$ is the error due to cost function approximation errors and actor-structure related errors. The following proposition provides an error bound for the successive cost improvement of the scheme.

Proposition 3: Consider the approximation error for the actor (6) as defined in (32), for the cost function (2). Then, each iteration of the PI scheme results in improvement of the cost function if:

$$\|\varepsilon\| < \frac{1}{2} \frac{|(2u^{(i+1)})^T(u^{(i)} - u^{(i+1)}) - (\|u^{(i+1)}\|^2 - \|u^{(i)}\|^2)|}{\|u^{(i)} - u^{(i+1)}\|}.$$

Proof: The difference between the successive cost function values $V^{(i)}(p)$, $V^{(i+1)}(p)$ between iterations will be computed by integrating both functions' time derivative along a trajectory of System (1) under the input $u^{(i+1)}$. Note that, since $\nabla V^{(i)}$, $\nabla V^{(i+1)}$ are conservative vector fields, the integrated value between two points is independent of the path along which the integral is taken (this process is similar to [45]). Hence,

$$\begin{aligned}V_{u^{(i+1)}}^{(i+1)}(\bar{p}) - V_{u^{(i+1)}}^{(i)}(\bar{p}) &= \\ \int_0^\infty \left[\left(\nabla V_{u^{(i+1)}}^{(i+1)}(\tau; \bar{p}) - \nabla V_{u^{(i+1)}}^{(i)}(\tau; \bar{p}) \right)^T u^{(i+1)}(\tau) \right] d\tau,\end{aligned}\quad (33)$$

where the index is employed to denote the dependence of the cost function gradient on each input—which implicitly defines a specific trajectory of (1)—. Evaluating the Hamiltonian (20) for the i -th and $i+1$ -th iterations, invoking (32) and after some algebra, (33) becomes:

$$V_{u^{(i+1)}}^{(i+1)}(\bar{p}) - V_{u^{(i+1)}}^{(i)}(\bar{p}) = -\int_0^\infty B(\tau) d\tau, \quad (34)$$

where

$$\begin{aligned}B &= -2\beta (u^{(i+1)} - \varepsilon)^T (u^{(i)} - u^{(i+1)}) \\ &\quad + \beta (\|u^{(i+1)}\|^2 - \|u^{(i)}\|^2) \\ &= 2\beta \varepsilon^T (u^{(i)} - u^{(i+1)}) - 2\beta (u^{(i+1)})^T (u^{(i)} - u^{(i+1)}) \\ &\quad + \beta (\|u^{(i+1)}\|^2 - \|u^{(i)}\|^2).\end{aligned}\quad (35)$$

The sum of the second and third terms of B can be shown to be positive by invoking the mean value theorem. Hence, a

Algorithm 1: PI METHOD.

- **Inputs:** Workspace \mathcal{W} , Actor and Critic structures $\Phi(p; \theta)$, $C(p; \omega)$, Convergence Threshold $E > 0$
- Compute an admissible initial input $v \in \mathcal{U}_{\mathcal{W}}$ [31].
- Sample \mathcal{W} yielding a set $\mathcal{S} = \{p_1, p_2, \dots, p_S\}$ of samples.
- Set $i = 0$, $\epsilon = \infty$.
- Initialize $\theta^{(0)}$ such that $u(p; \theta^{(0)}) = v(p)$.

while $\epsilon \geq E$ **do**

- Compute the expressions of (26), (27) over the set \mathcal{S} ,
- Update the Critic structure parameter values (26) $\rightarrow \omega^{(i)}$,
- Update the Actor structure parameter values (27) $\rightarrow \theta^{(i+1)}$,
- Set $\epsilon = \|\theta^{(i+1)} - \theta^{(i)}\|$,
- Set $i \leftarrow i + 1$,

end

- The final control law is given by

$$u(p; v; \theta^{(i)})$$

sufficient condition for control improvement is:

$$|\epsilon^T (u^{(i)} - u^{(i+1)})| < \frac{1}{2} \left| (2\tilde{u}^{(i+1)})^T (u^{(i)} - u^{(i+1)}) - (\|u^{(i+1)}\|^2 - \|u^{(i)}\|^2) \right|,$$

or:

$$\|\epsilon\| < \frac{1}{2} \frac{|(2\tilde{u}^{(i+1)})^T (u^{(i)} - u^{(i+1)}) - (\|u^{(i+1)}\|^2 - \|u^{(i)}\|^2)|}{\|u^{(i)} - u^{(i+1)}\|}, \quad (36)$$

which concludes the proof, since for $B > 0$, then from (34):

$$\begin{aligned} V_{u^{(i+1)}}^{(i+1)}(\bar{p}) - V_{u^{(i+1)}}^{(i)}(\bar{p}) &< 0 \Leftrightarrow \\ V^{(i+1)}(\bar{p}) &< V^{(i)}(\bar{p}), \quad \forall \bar{p} \in \mathcal{W}. \end{aligned}$$

E. PI ALGORITHM

The aforementioned elements are combined in Algorithm 1, which entails successively approximating the cost function and updating the actor structure, over a set of pre-defined collocation points within the workspace.

F. NOVEL COST FUNCTION PDE SOLVER

While the framework of Subsection IV-C is very effective in simple workspaces, while overcoming some computationally expensive aspects of previous works [30], [31], the solution of Eq. (26) provides poor results as the complexity of the workspace increases (the critic update step of Algorithm 1). Hence, this step can be interchanged with a custom PDE solver, which results in the cost function values at specific collocation points. Additionally, the corresponding cost function gradient can be extracted through numerical differentiation, and the actor update step of Algorithm 1 can proceed as previously.

In summary, a novel mesh-free method for the approximation of (2) is presented. Our method is inspired by Fast Marching Methods (FFMs) [46] and Level-Set Methods [47] for solving PDEs. Consider the differential cost form of (2), written explicitly as:

$$\begin{aligned} \frac{\partial V}{\partial x} u_x + \frac{\partial V}{\partial y} u_y &= -r(p, u), \\ \text{s.t. } V(p_d) &= 0, \end{aligned} \quad (37)$$

where $p = [x, y]^T \in \mathcal{W}$, $u = [u_x, u_y]^T \in \mathcal{U}_{\mathcal{W}}$ and $r(p, u) = P(p; p_d) + R(u)$. This PDE can be solved through the method of characteristics [48] through the following three ordinary differential equations:

$$\frac{dx}{u_x(p)} = \frac{dy}{u_y(p)} = -\frac{dV}{r(p, u)} = dt. \quad (38)$$

The solution of (38) begins through the solution of the first two terms, resulting in characteristic curves $c(x, y) = t$, $t \in \mathbb{R}_+$. Then, the final equation is solved across the trajectories $x(t)$, $y(t)$ that yield the characteristic curves.

Consider now a set of collocation points $\mathcal{P} = \{p^0, p^1, p^2, \dots, p^N\}$, $p^i \in \mathcal{W}$, $\forall i = \{1, 2, \dots, N\}$, containing the goal position $p^0 \triangleq p_d$, distributed over the workspace, along with a corresponding graph $\mathcal{G}_{\mathcal{P}} = \{\mathcal{P}, \mathcal{E}\}$, where \mathcal{P} , \mathcal{E} denote the vertices and edges of the graph respectively. The edges of $\mathcal{G}_{\mathcal{P}}$ are constructed according to a K-nearest neighbour principle for each point, i.e., each vertex (point) is connected to its nearest neighbours according to a maximum distance $d > 0$. The collocation points are split into three **disjoint** sets, “solved”, “boundary” and “far-away”, denoted by $\mathcal{P}_S \subset \mathcal{P}$, $\mathcal{P}_B \subset \mathcal{P}$ and $\mathcal{P}_F \subset \mathcal{P}$ respectively, such that $\mathcal{P}_S \cup \mathcal{P}_B \cup \mathcal{P}_F = \mathcal{P}$. The sets are initialized as $\mathcal{P}_S = \{p^0\}$, $\mathcal{P}_B = \mathcal{N}(p^0)$ and $\mathcal{P}_F = \mathcal{P} / (\mathcal{P}_B \cup \mathcal{P}_S)$, where $\mathcal{N}(\cdot)$ denotes the set of neighbours of a vertex.

The solution of (37) through (38) is accomplished as follows: Consider a “boundary” point $p^k \in \mathcal{P}_B$. According to (38), the value of the cost function at this point is influenced only by one or more points positioned “downstream” according to the flow of the field $u(p)$. Hence such points can be identified through propagating the solution:

$$\bar{p}^k = p^k + \int_0^{T_k} u(p(\tau)) d\tau, \quad (39)$$

where $T_k > 0$. In practice, this can be approximated well enough through Euler integration

$$\bar{p}^k = p^k + T_k u(p^k), \quad (40)$$

where T_k is computed as $T_k = \frac{d}{\|u(p^k)\|}$. Thus, the corresponding cost value is computed as:

$$V_k = \frac{1}{K} \sum_{\bar{k} \in \bar{\mathcal{K}}} \left[V_{\bar{k}} + r(p^{\bar{k}}, u(p^{\bar{k}})) T_{\bar{k}} \right], \quad (41)$$

where $V_{\bar{k}}$ denotes the computed cost of a point $p^{\bar{k}}$, for $p^{\bar{k}} \in \bar{\mathcal{K}} \subset \mathcal{P}_S$ denoting the set of K-Nearest Neighbours of the point

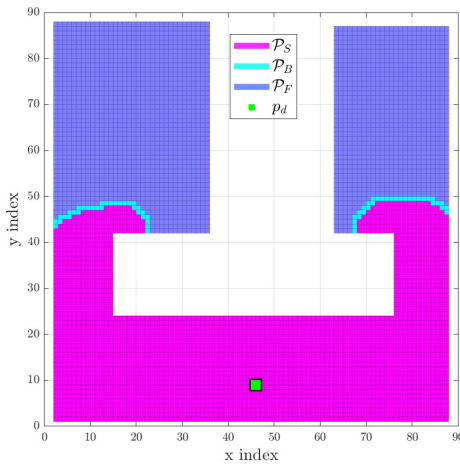


FIGURE 1. Depiction of the solution process of the proposed PDE solver, the three sets are depicted through different colors, while the desired final position is depicted through the green square.

\bar{p}^k . Eq. (41) essentially corresponds to solving the third ODE in (38) backwards in time through Euler integration. Then, the point p^k is removed from the set \mathcal{P}_B and appended to the set \mathcal{P}_S . Furthermore, the set \mathcal{P}_B is augmented with the “unsolved” neighbours of p^k , i.e., the set $\mathcal{N}(p^k) \cap \mathcal{P}_F$, which are removed from the set \mathcal{P}_F .

A crucial aspect of the method rests on propagating the “boundary” points appropriately in order to satisfy the HJB optimality condition. This is achieved via setting a reference value:

$$\bar{V} = \max_{i \in \mathcal{I}(\mathcal{P}_S)} \{V_i\}, \quad (42)$$

where $\mathcal{I}(\cdot)$ is a function that outputs the index set of a set. Then, this reference value is kept constant, and only the points $p^k \in \mathcal{P}_B$ such that $V_k < \bar{V}$ are added to the set \mathcal{P}_S . Once the former set is empty, then the reference value is recomputed and the scheme continues.

1) IMPLEMENTATION DETAILS

In practice, this method can be rendered extremely efficient if programmed with similar principles as with FMMs. For instance, if a uniformly distributed set of points is employed, then finding the nearest neighbours at a specific distance, as well as finding the set $\bar{\mathcal{K}}$ for any point, necessitates only (integer) matrix index operations. Finally, the value \bar{V} necessitates some book-keeping, but can ultimately be computed in $\mathcal{O}(1)$ operations, through a similar data structure to the min-heap one [46] used in FMMs.

An example of the method is depicted in Fig. 1, where all associated sets that were previously defined are depicted during the cost PDE solution for a simple workspace. Finally, regarding the computational complexity of the method, a graph of the execution time versus the number of solution nodes is depicted in Fig. 2. The method is shown to scale linearly w.r.t. the number of nodes, while the slope of the curve is

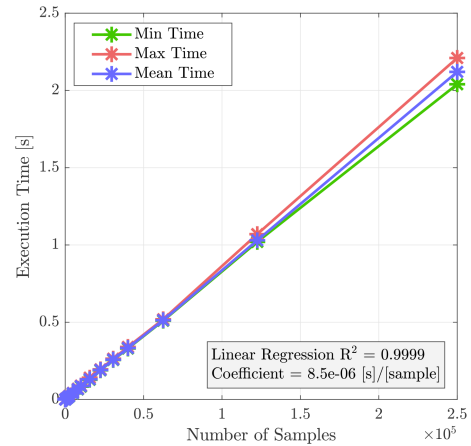


FIGURE 2. The execution time of the PDE solver vs. the number of solution points. Statistical data over a linear fit, along with the minimum maximum and mean values for all data are depicted.

approximately $8.5[\mu\text{s}]/\text{point}$. Concerning Fig. 2, 10 trials were carried out for the workspace of Fig. 1, and the mean, min and max curves are depicted.

Remark 3 (Optimality): It is shown in [31], that a policy iteration scheme similar to the one presented here results asymptotically in the globally optimal solution to the proposed problem. However, owing to limiting the policy to a very specific parametric structure –in contrast to [31] where the policy stems from a functional space–, no such guarantees can be made. Nevertheless, if the actor structure is able to closely match the gradient of the approximated cost function (23), then the cost improvement lemma in [31] will still apply. We will show in the Results section how the proposed method is effective in extracting close to the optimal policy, even in complex cases. In summary, this method sacrifices provable optimality for execution time and capacity to tackle complex workspaces, through the proposed actor structure (6).

V. RESULTS

In this section, results for the proposed method and a plethora of SBMs are presented. All simulations were carried out on a PC with 50 Gb RAM and an Intel-i7 processor running Ubuntu version 18.04LTS. The simulations for the proposed method and RRT* were implemented in the environment of MATLAB 2022a, while for the rest of the SBMs Python implementations were used. In order to provide a fair comparison with the quasi-linear paths extracted from SBMs, the asymptotically optimal paths are imbued with the closed-form optimal velocity norm w.r.t. (2) [31]. Notably, this places our method at a disadvantage, as implementing the optimal velocity over discontinuous paths is in practice infeasible (as the direction of the velocity is discontinuous). Nevertheless, this results in effective benchmarking of our method w.r.t. the globally optimal solution (as such SBMs provide asymptotically optimal paths), also taking into account the execution time.

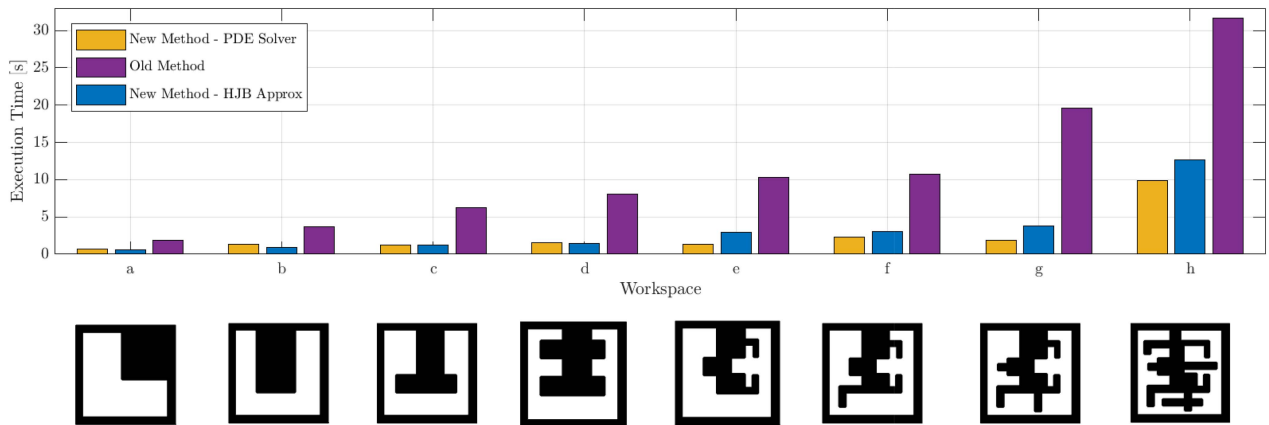


FIGURE 3. Execution times comparison between the proposed method, with the HJB framework and with the custom PDE solver framework, as well as our previous method in [31]. It is evident that the newer method scales more favorably as the complexity of the workspace increases. The workspaces are shown below each bar plot set, with each boundary depicted in black.

Concerning the proposed method, Gaussian RBF parametrization structures were employed for the actor and critic (where relevant) structures (see [31] for more details), uniformly distributed over each workspace. In practice, tuning the RBF structure entails starting from a less dense placement of RBFs and increasing the density if necessary. However, in the simulations carried out in this work, the scheme did not require a lot of tuning to work close to optimally. Furthermore, the shape parameters of the RBFs were chosen such that there is an 80% overlap between consecutive RBFs. The initial policy was similarly computed through an AHPF as in [31].

A. PROPOSED METHOD RUN-TIME

We begin by comparing the run-time of the proposed compared to our previous work [31] in Fig. 3. A series of workspaces of similar size, with increasing boundary complexity was investigated, and a feasible solution was found for each one. In Fig. 3, it is shown how the proposed scheme, both with the HJB formulation, as well as the PDE solver one, outperform the previous method in computational time as the complexity of the workspace increases. Notably, the more complex the workspace, the largest the discrepancy between [31] and the herein proposed method. Nevertheless, this is to the detriment of optimality, as [31] is shown to asymptotically converge to the **globally optimal** solution, whereas no such guarantee is provided here (see Remark 3).

B. WORKSPACE WITH OBSTACLES

In Fig. 4, comparative results between the proposed method and a variety of SBMs are presented, namely an FMT, RRT*, an RRT*-SMART and an Informed RRT* one, as well as with our previous method [31]. The workspace, whose boundary is depicted with black, contains various obstacles. This aspect was tackled through the method in [33], we will thus not go into further detail for the sake of brevity. The difference between the cost of the proposed one and each one of the rest of the methods are depicted through heat maps. Negative

values indicate that our method outperforms the latter. The absolute difference in cost values varies from ~ -5 to ~ 1.5 , with the cost function attaining values from 0 at the goal to ~ 10 . In most cases, with the exception of our previous method [31], the proposed scheme is superior, as demonstrated in the aforementioned figure. The RRT* method is the best out of the SBMs, therefore subsequent comparisons focus on the latter.

C. CURVED WORKSPACE

In Fig. 5, a workspace with a curved outer boundary is evaluated. We compare our method against RRT*, and present several trajectories for both, as well as the resulting normalized velocity vector field for our method. Trajectory lengths and cost function values comparisons are presented in bar plots. Ten trials for each starting point were carried out for statistical significance, and minimum, mean and maximum values for the RRT* case are presented (our method is deterministic and thus, no such statistical data are necessary). It is evident that our method outperforms the RRT* one in all tested cases.

D. WORKSPACE WITH NARROW PASSAGES

Similarly, in Fig. 6, we present results over a workspace with several narrow passages, which are known to be problematic for SBMs, owing to the low probability of sampling within the slim regions, and thus obtaining safe trajectories, resulting in long runtimes and/or low success rates. Twenty trials for each starting point were carried out for statistical significance, and minimum, mean and maximum values for the RRT* case are presented (our method is deterministic and thus, no such statistical data are necessary). Once again, our method outperforms the RRT* one in all tested cases.

E. COMPLEX WORKSPACE

Finally, we provide a comparative evaluation of both methods in a very complex workspace in Fig. 7, where twenty trials were carried out. Similarly to previous cases, our method

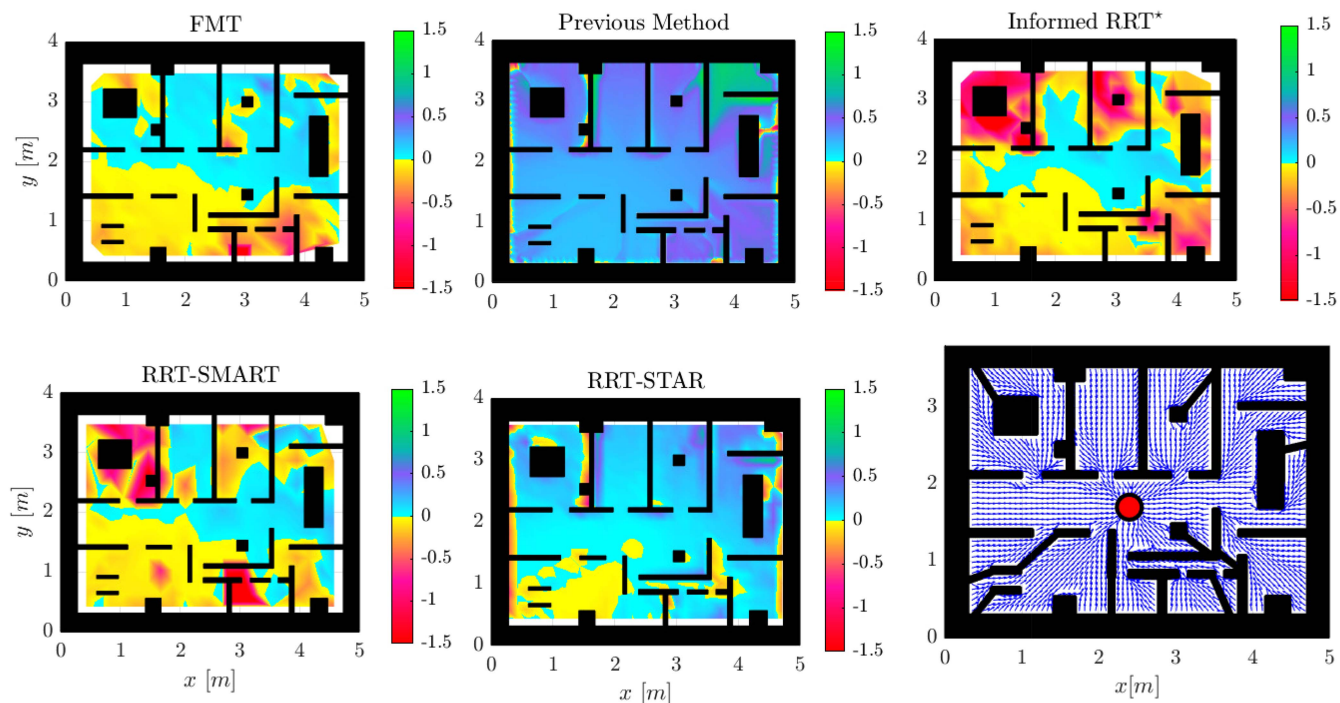


FIGURE 4. Comparative results between the proposed method, and an FMT, RRT*, an RRT*-SMART and an Informed RRT* one, as well as with our previous method [31]. The cost function differences for the entire workspace are depicted. The boundary of the workspace is depicted with black, while the goal position is depicted in the right-bottom figure (red disk) along with the final, normalized velocity vector field. Negative values of the cost difference plots indicate superior performance of our method.

proves to be superior, with a single exception for a point that is in the line of sight of the goal position. Hence, reactive planning is trivial in this case, as radial convergence is the optimal, state-feedback solution. Notably, in the last subfigure of Fig. 7, the scale is logarithmic, hence the apparent differences are in reality larger (this choice was made to render all bar plot visible across different scales, which is an issue that arises due to the size of the workspace).

F. EXECUTION TIME AND SUCCESS RATE

To conclude our evaluation, we present data over the execution times and success rate of our method versus the SBMs. Note that each method was fine-tuned prior to extracting the presented data, in order to provide a fair evaluation. We note that comparing execution times for SBMs and continuous methods is not straight-forward; continuous methods result in a one-shot solution for the entire workspace, while SBMs usually only provide a single starting-ending position pair. Hence, while SBMs are faster over a single trajectory, if more than one paths are required, their run-time increases (linearly at worst, with a slower rate if re-planning over the existing structure is employed). Thus, different applications might benefit from each class of methods differently. The results are presented in Table 1. We note that the execution times for our method also include the execution times for computing the initial, AHPF-based policy.

Nevertheless, in order to gain insight on each method's complexity, for the case of Fig. 4, the whole workspace was

covered with a sufficiently dense number of starting positions, and a single instance (the best one) for each sample was timed, resulting in the provided times. Our method proves to be far superior in obtaining a solution for the whole workspace, as expected.

Furthermore, in more challenging cases, such as the ones in Figs. 5, 6, 7, our method proves faster even when the entirety of the workspace is not sufficiently covered. The best case for RRT* is the curved-boundary case, where our method presents an improvement of $\sim 35\%$, while for the complex workspace of Fig. 7, our method presents an improvement of $\sim 64\%$. Most notably, for the case of narrow passages, where SBMs are known to struggle, the proposed method presents an improvement in execution time of $\sim 90\%$. Note that, even methods such as informed RRTs, which have been employed in the past to counter such problematic cases, would not improve the run-time for workspaces such as Fig. 6, as the corresponding ellipse would necessarily cover almost the entire workspace. Finally, while the SBM's performance deteriorates as the corridor size decreases, our method remains mostly unaffected, only necessitating making the solution grid for the PDE solver denser. We underline that the number of trajectories in these cases do not suffice to cover the entire workspace. Doing so, would result in greater deterioration of the aforementioned SBM performance.

We conclude this section with a discussion over the success rate of RRT*, depicted in the final column of Table 1. Our method, owing to the provable guarantees of the actor

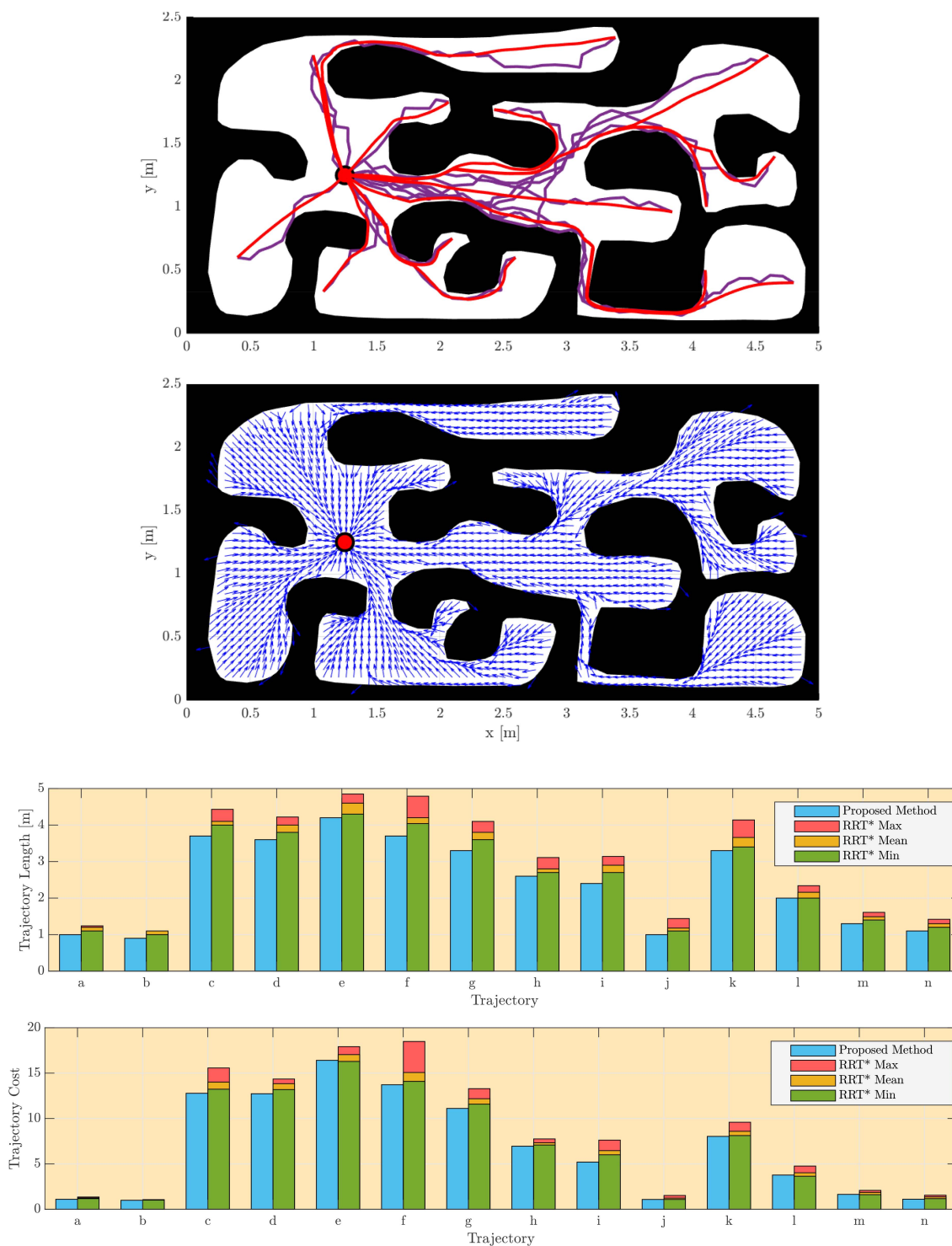


FIGURE 5. Comparison between our method and RRT* for a workspace with curved boundary. Trajectories from both methods are depicted in the top figure (red for our method, magenta for RRT*), and the final, normalized velocity field is depicted in the second figure. The goal position is depicted through a red disk. In the last two figures, comparative bar plots for path lengths and cost function values are depicted.

structure, as well as to the deterministic nature of the method, is successful in 100% of trials. RRT* on the other hand, presents similar performance in simple cases, however, as the complexity increases, several trials start to present significant

deterioration. Note that in Table 1, cumulative success rates are presented. That is, for the workspace of Fig. 7, the starting positions farther to the goal presented a minimum success rate of 55% (in two instances), while for the case of Fig. 6 the

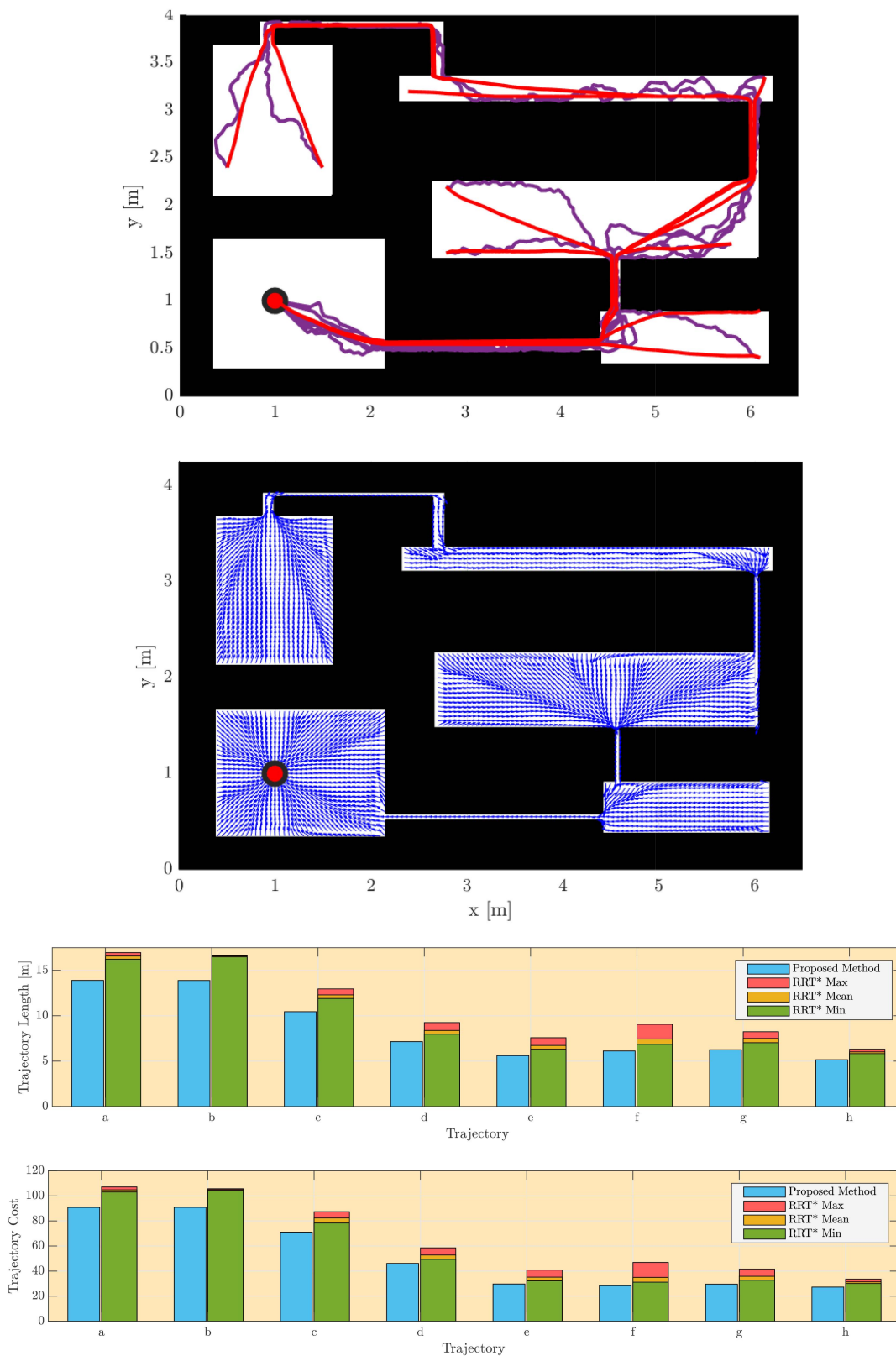


FIGURE 6. Comparison between our method and RRT* for a workspace with narrow passages. Trajectories from both methods are depicted in the top figure (red for our method, magenta for RRT*), and the final, normalized velocity field is depicted in the second figure. The goal position is depicted through a red disk. In the last two figures, comparative bar plots for path lengths and cost function values are depicted.

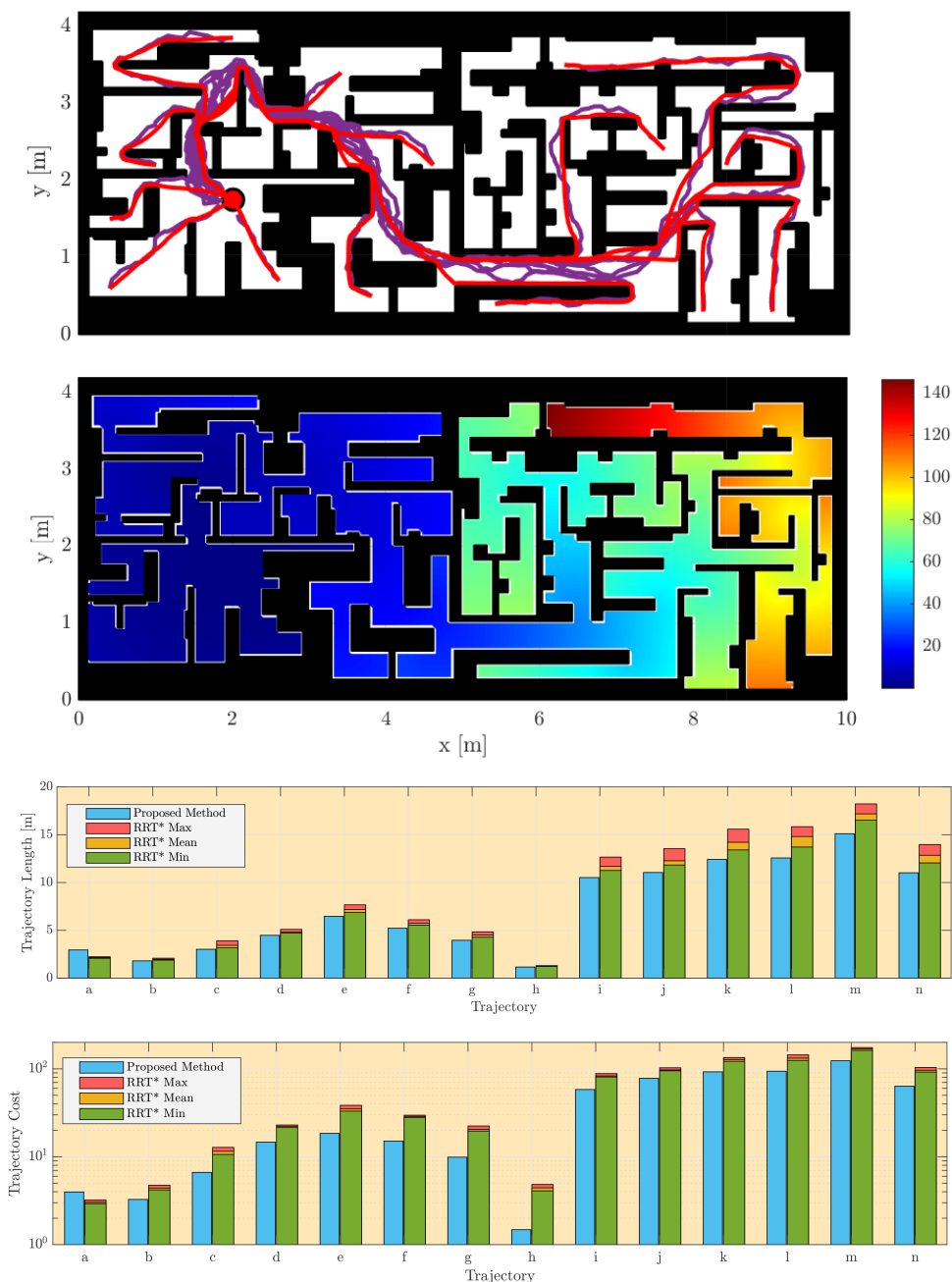


FIGURE 7. Comparison between our method and RRT* for a complex workspace. Trajectories from both methods are depicted in the top figure (red for our method, magenta for RRT*), and the final, cost function for our method is depicted in the second figure. The goal position is depicted through a red disk. In the last two figures, comparative bar plots for path lengths and cost function values are depicted.

success rate fell as low as 15% and 20%. This demonstrates a clear advantage of our method for safe and successful planning in known workspaces.

VI. LIMITATIONS

While the proposed method and the accompanying technical results yield a useful tool for tackling OMP, some limitations persist. First of all, while employing the single integrator dynamics model is useful in yielding minimum-length paths, our method can not be directly implemented for more complex

dynamical systems. Nevertheless, methods such as Explicit Reference Governors (ERGs) [49] necessitate such velocity commands to drive more complex systems safely, hence our method can be used therein. Additionally, High-Order Control Barrier Function with Quadratic Programming (HOCBF-QP) methods are also dependent on a reference control signal for safe planning in dynamical systems [50], [51], where the proposed method can also prove useful.

This limitation mainly stems from two factors: 1) The form of the actor is tailored for the single integrator model, 2)

TABLE 1. Execution times and success rates comparison.

Workspace of Fig. 4						
	# of Trajs.	# Iterations	Mean Time per Traj. [s]	Total Time [mins]		Success Rate%
Ours	-	-	-	5.58		100%
				Time for TP Regions	Time for OMP	
				3.9	1.68	
Method in [33]	-	-	-	13.90		100%
				Time for TP Regions	Time for OMP	
				3.9	10.00	
RRT*	656	2500	3.98	43.5		100%
Smart RRT*	289	1000	25.94	124.9		100%
Informed RRT*	288	2000	26.92	129.2		100%
FMT*	288	2000	9.06	43.5		100%
Workspace of Fig. 5						
	# of Trajs.	# Iterations	Mean Time per Traj. [s]	Total Time [mins]		Success Rate%
Ours	-	-	-	3.45		100%
RRT*	150	5000	2.13	5.33		100%
Workspace of Fig. 6						
	# of Trajs.	# Iterations	Mean Time per Traj. [s]	Total Time [mins]		Success Rate%
Ours	-	-	-	3.93		100%
RRT*	140	20000	17.32	40.27		77%
Workspace of Fig. 7						
	# of Trajs.	# Iterations	Mean Time per Traj. [s]	Total Time [mins]		Success Rate%
Ours	-	-	-	21.67		100%
RRT*	255	30000	14.01	60		90%

as the number of dimensions of the state space increases, along with changes in its topology, the scheme might suffer both technically and in computational load. This is due to the “curse of dimensionality” of such PI methods, while topological considerations necessitate more careful construction of an actor structure. Both of the above aspects can motivate future research endeavors for extending this work.

Furthermore, lack of finite time convergence might be critical for some applications. The adopted infinite-horizon cost functional can only yield asymptotically converging trajectories, hence finite-time convergence requires severe modifications to the scheme. Nevertheless, this aspect can be somewhat amended by altering the rate at which the robot converges to the goal, through modifying the norm of the velocity field as a post-processing step. More details can be found in [31].

Additionally, while computing the admissible parameter set Θ_A might be easy in case of the proposed parametrized structures, in general this might not be straightforward, especially considering future extensions to higher-order dynamical systems. Finally, providing a constructive way to choose a parametrized structure possibly based on the topology and the dynamics of the workspace is an interesting research direction.

VII. CONCLUSION

In this article, a novel, PI scheme for continuous, state-feedback optimal motion planning was presented. A novel parametrized actor structure was proposed, such that safety and convergence are guaranteed by construction. Additionally, a novel PDE solver for the computation of the cost function was presented, in order to enable tackling more complex workspaces. Most crucially, our method was demonstrated to provide a significantly more efficient solution to

previous continuous methods, while not sacrificing optimality significantly, and was effective even in very complex environments. Compared to existing, proven methods, our method is superior w.r.t. path length, cost function value as well as success rate and execution times.

Finally, concerning future research efforts, the limitations outlined in the previous section provide promising research directions. For instance, the proposed actor structure will be extended for more complex dynamics and environments, such as mechanical systems and three-dimensional workspaces. Additionally, the proposed scheme will be combined with [33] in order to ameliorate the limitations of the latter, with the intent to also tackle moving obstacles. These directions will extend the applicability, flexibility and scope of the proposed scheme paving the way for applications in realistic operating conditions.

REFERENCES

- [1] N. Roy et al., “From machine learning to robotics: Challenges and opportunities for embodied intelligence,” 2021, *arXiv:2110.15245*.
- [2] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [3] N. Anastopoulos, K. Nikas, G. Goumas, and N. Koziris, “Early experiences on accelerating Dijkstra’s algorithm using transactional memory,” in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, 2009, pp. 1–8.
- [4] X. Liu and D. Gong, “A comparative study of A-star algorithms for search and rescue in perfect maze,” in *Proc. IEEE Int. Conf. Electric Inf. Control Eng.*, 2011, pp. 24–27.
- [5] S. Koenig and M. Likhachev, “D*lite,” in *Proc. 18th Nat. Conf. Artif. Intell.*, 2002, pp. 476–483.
- [6] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [7] I. Noreen, A. Khan, and Z. Habib, “Optimal path planning using RRT* based approaches: A survey and future directions,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, pp. 1–16, 2016.
- [8] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

- [9] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 883–921, 2015, doi: [10.1177/0278364915577958](https://doi.org/10.1177/0278364915577958).
- [10] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, no. 10, pp. 560–570, Oct. 1979, doi: [10.1145/359156.359164](https://doi.org/10.1145/359156.359164).
- [11] Z. Wang, Y. Li, H. Zhang, C. Liu, and Q. Chen, "Sampling-based optimal motion planning with smart exploration and exploitation," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 5, pp. 2376–2386, Oct. 2020.
- [12] J. Wang, M. Q.-H. Meng, and O. Khatib, "EB-RRT: Optimal motion planning for mobile robots," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 4, pp. 2063–2073, Oct. 2020.
- [13] G. Yang, M. Cai, A. Ahmad, C. Belta, and R. Tron, "Efficient LQR-CBF-RRT*: Safe and optimal motion planning," 2023, *arXiv:2304.00790*.
- [14] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *Int. J. Robot. Res.*, vol. 35, no. 5, pp. 528–564, 2016, doi: [10.1177/0278364915614386](https://doi.org/10.1177/0278364915614386).
- [15] J. Nasir et al., "RRT*-Smart: A rapid convergence implementation of RRT*," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 7, 2013, Art. no. 299, doi: [10.5772/56718](https://doi.org/10.5772/56718).
- [16] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2997–3004.
- [17] E. Rimón and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Automat.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
- [18] S. G. Loizou, "Closed form navigation functions based on harmonic potentials," in *Proc. IEEE 50th Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 6361–6366.
- [19] P. Vlantis, C. Vrohidis, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Robot navigation in complex workspaces using harmonic maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1726–1731.
- [20] L. Fan, J. Liu, W. Zhang, and P. Xu, "Robot navigation in complex workspaces using conformal navigation transformations," *IEEE Robot. Automat. Lett.*, vol. 8, no. 1, pp. 192–199, Jan. 2023, doi: [10.1109/LRA.2022.3222953](https://doi.org/10.1109/LRA.2022.3222953).
- [21] L. Fan, J. Liu, and W. Zhang, "Using conformal navigation transformations for robot navigation in complex polygonal workspaces," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 70, no. 12, pp. 4464–4468, Dec. 2023, doi: [10.1109/TCSII.2023.3285658](https://doi.org/10.1109/TCSII.2023.3285658).
- [22] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2005–2011, Apr. 2021.
- [23] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Trajectory planning in unknown 2D workspaces: A smooth, reactive, harmonics-based approach," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1992–1999, Apr. 2022.
- [24] J. O. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, pp. 790–796.
- [25] M. B. Horowitz and J. W. Burdick, "Optimal navigation functions for nonlinear stochastic systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 224–231.
- [26] J. Amiryani and M. Jamzad, "Adaptive motion planning with artificial potential fields using a prior path," in *Proc. IEEE 3rd RSI Int. Conf. Robot. Mechatronics*, 2015, pp. 731–736.
- [27] P. Vadakkepat, K. C. Tan, and W. Ming-Liang, "Evolutionary artificial potential fields and their application in real time robot path planning," in *Proc. IEEE Congr. Evol. Computation*, 2000, pp. 256–263.
- [28] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Optimal robot motion planning in constrained workspaces using reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6917–6922.
- [29] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Optimal motion planning in unknown workspaces using integral reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6926–6933, Jul. 2022.
- [30] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Reactive optimal motion planning for a class of holonomic planar agents using reinforcement learning with provable guarantees," *Front. Robot. AI*, vol. 10, 2023, Art. no. 1255696.
- [31] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "A continuous off-policy reinforcement learning scheme for optimal motion planning in simply-connected workspaces," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10247–10253.
- [32] Y. Bryshnikov, "Topological perplexity of feedback stabilization," *J. Appl. Comput. Topol.*, vol. 7, pp. 75–87, 2022.
- [33] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "State-feedback optimal motion planning in the presence of obstacles," *IEEE Robot. Automat. Lett.*, vol. 8, no. 12, pp. 8406–8413, Dec. 2023.
- [34] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *J. Intell. Manuf.*, vol. 33, pp. 387–424, 2022.
- [35] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, 2020, Art. no. eabc5986, doi: [10.1126/scirobotics.abc5986](https://doi.org/10.1126/scirobotics.abc5986).
- [36] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, 2022, Art. no. eabk2822, doi: [10.1126/scirobotics.abk2822](https://doi.org/10.1126/scirobotics.abk2822).
- [37] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 2497–2503.
- [38] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, 2018, Art. no. 3170. [Online]. Available: <https://www.mdpi.com/1424-8220/18/9/3170>
- [39] P. Rousseas, G. C. Karras, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Indoor visual exploration with multi-rotor aerial robotic vehicles," *Sensors*, vol. 22, no. 14, 2022, Art. no. 5194. [Online]. Available: <https://www.mdpi.com/1424-8220/22/14/5194>
- [40] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012.
- [41] F. L. Lewis, D. Vrabie, and V. L. Syrmos, "Reinforcement learning and optimal adaptive control," in *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012, ch. 11, pp. 461–517.
- [42] D. Vrabie and F. L. Lewis, "Adaptive optimal control algorithm for continuous-time nonlinear systems based on policy iteration," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 73–79.
- [43] Tse and Edison, "An extension of Karmarkar's projective algorithm for convex quadratic programming," *Math. Program.*, vol. 44, pp. 157–179, 1989.
- [44] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing lipschitz continuity," *Mach. Learn.*, vol. 110, no. 2, pp. 393–416, Feb. 2021, doi: [10.1007/s10994-020-05929-w](https://doi.org/10.1007/s10994-020-05929-w).
- [45] 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.
- [46] J. A. Sethian, "Fast marching methods," *SIAM Rev.*, vol. 41, no. 2, pp. 199–235, 1999, doi: [10.1137/S0036144598347059](https://doi.org/10.1137/S0036144598347059).
- [47] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0021999188900022>
- [48] A. K. Nandakumaran and P. S. Datti, *First-Order Partial Differential Equations: Method of Characteristics* (Cambridge IISc Series). Cambridge, U.K.: Cambridge Univ. Press, 2020.
- [49] M. M. Nicotra and E. Garone, "The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems," *IEEE Control Syst. Mag.*, vol. 38, no. 4, pp. 89–107, Aug. 2018.
- [50] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3655–3662, Jul. 2022.
- [51] S. Fukuda, Y. Satoh, and O. Sakata, "Trajectory-tracking control considering obstacle avoidance by using control barrier function," in *Proc. Int. Autom. Control Conf.*, 2020, pp. 1–6.