2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Optimal Robot Motion Planning in Constrained Workspaces Using Reinforcement Learning

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

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

I. INTRODUCTION

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

In this work, we intend to explore the application of Reinforcement Learning methods in optimizing the motion of a robot moving in a two-dimensional constrained, but fully known workspace with internal fixed obstacles. In particular, an offline solution to the underlying optimization problem is formulated in such a way that ensures safety and convergence with mathematical rigor using robust principles and tools from the successive approximation theory [20]. Subsequently, we establish an on-line reinforcement learning approach for optimizing the motion of a moving robot with respect to a specific utility function, with great emphasis on the rigorous proof of safety and convergence. The motivation behind the online approach stems from the fact that in real-world problems, not all trajectories of a workspace are of use, but rather specific starting-ending point combinations are needed. Therefore, an online approach is not only sufficient, but also advantageous with respect to computational complexity. Finally, the prospect of implementing the online scheme in unknown workspaces further motivates the latter's formulation.

*This work was not supported by any organization.

The authors are with the Control System Lab, School of Mechanical Engineering, National Technical University of Athens, Greece. rousseas.p@gmail.com {chmpechl, kkyria}@mail.ntua.gr.

II. RELATED WORK

Since the early days of robotics, many research efforts have been devoted to the motion planning problem and thus many approaches have been formulated. Such approaches can be generally classified as discrete methods, e.g., Configuration Space Decomposition methodologies [1]-[3], Probabilistic Sampling methods, e.g., Rapidly Exploring Random Trees [4]-[5] or Probabilistic Roadmaps [6]-[7] and others such as Manifold Samples [8]. On the other hand, the Optimal motion planning problem has been approached via Receding Horizon control [9]-[10] and Path Homotopy Invariants [11]-[12].

A specific class of solutions to the motion planning problem, and one that aims at addressing both safety and convergence aspects are the APFs, as introduced in [13]. This class of solutions encompasses both information for safety and convergence in the form of the gradient of a potential field. However, APFs entail problems of unwanted local equilibria due to their inherent construction and the topology of the workspace [14]. Rimon and Koditschek managed to produce a family of APFs, namely Navigation Functions (NF) that are applied to a transformed version of the physical workspace in the form of a sphere world[1]. Along with providing a constructive transformation for mapping workspaces with star-shaped obstacles (sets with a point from which any ray crosses the boundary once) to the aforementioned sphere worlds, Rimon and Koditschek aleviated some of the issues of the APFs as well. However, extensive tuning is required to get rid of local minima and in practice these functions prove difficult to be implemented (see [15]).

Aiming at tackling the shortcomings of APFs, a specific sub-category of the latter was introduced, namely the Artificial Harmonic Potential Fields (AHPF) [17]-[23]. The AHPFs are free of local minima by construction, and negate many of the issues of previous NFs. In the present work, the natural progress of previous research efforts [18], leads to

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

inheriting all the strong points of AHPFs and introducing a robust solution to the optimal motion planning problem. In order to accomplish this, a novel approach will be introduced, encompassing past work on Reinforcement Learning Optimization [16], re-framed and adapted for a specific family of AHPF-inspired motion controllers. Our work provides a deterministic and mathematically rigorous approach that exceeds the capabilities of previous probabilistic approaches. The implementation of Reinforcement Learning is pivotal in the current approach, as it overcomes the need for solving a very hard non-linear partial differential equation for calculating the cost function. Additionally, the latter is rigorously proven to converge under mild assumptions.

III. PROBLEM FORMULATION

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

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

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

$$V(\bar{p}; p_0) = \int_0^\infty \left[Q(p(\tau; \bar{p}); p_0) + R(u(\tau))\right] d\tau, \quad \forall \bar{p} \in \mathcal{W} \tag{2}$$

where $\bar{p}$ is the initial state of the system $\bar{p} = p(0)$ and p_0 denotes the goal position.

A. A Set of Parametrized Control Policies

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

$$h(p, k) \triangleq P(p) \cdot k = -\mathcal{J}_f^{-1}(p) \cdot g(p) \cdot A \cdot k, \tag{3}$$

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

$$A = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(M+1)\times(M+1)} \tag{4}$$

[2]The adopted transformation is the composition of a diffeomorphism that maps all points inside $\mathcal{W}$ onto the open punctured unit disk [18], with a diffeomorphism that maps the unit disk onto $\mathbb{R}^2$ [17].

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

$$g(p) = \prod_{i=0}^{M} tanh\left(\|f(p) - \mathcal{V}_i\|^2\right) \cdot \cdot \left[\frac{f(p) - \mathcal{V}_0}{\|f(p) - \mathcal{V}_0\|^2}, \frac{-f(p) + \mathcal{V}_1}{\|f(p) - \mathcal{V}_1\|^2}, ..., \frac{-f(p) + \mathcal{V}_M}{\|f(p) - \mathcal{V}_M\|^2}\right] \tag{5}$$

with $\mathcal{J}_f(p)$ denoting the Jacobian of the transformation $f(p)$. The above formulation is a direct analog to the gradient of a classic harmonic potential field, enhanced in a way that fits the needs of the optimization process that follows. We will later show that this formulation, besides safety, ensures convergence as well. Furthermore, we have further simplified the problem of stability and safety of the robot incorporating the matrix A in (3). As shown in [17], for safe navigation the weight of the attractive term has to be greater than the sum of the weights of all the repulsive ones. It is evident that such formulation can be quite tedious especially when considering an optimization approach. Nevertheless, in our formulation, the equivalent constraint boils down to:

$$k_i > 0, i = 0, .., M \tag{6}$$

owing to the adopted form of matrix A in (4) (notice that the weight of the attractive term is the sum of all k_i). We will prove analytically how our method will ensure safety and convergence after discussing the optimization problem, as our solution for the vector k will ensure both optimality and (6). Hence, we consider the following value function:

$$V(\bar{p}; p_0) = \int_0^\infty \left[Q(p(\tau; \bar{p}); p_0) + W(k(\tau))\right] d\tau \tag{7}$$

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

$$Q(p; p_0) = \beta \cdot \|p - p_0\|^2, \beta > 0 \tag{8}$$

and

$$W(k) = \gamma \cdot \sum_{i=0}^{M} \int_{\alpha(p)}^{k_i} \left(\frac{v_i}{\alpha^2(p)} - \frac{1}{v_i}\right) dv_i, \gamma > 0 \tag{9}$$

with $\alpha(p) = \frac{1}{\sqrt{1+M}} \frac{\bar{u}}{\sqrt{\|P(p)\|^2+1}}$ for an upper bound of the velocity $\bar{u}$. The state-related term is a classical control problem's form. In such problems, the input-related cost is quadratic. However, in our approach, (9) is used to ensure safety and convergence. Notice that the selection of W is not heuristic. This term has no physical significance, however, the form of the function $W(k)$ ensures that through its minimization all components of k remain positive. We have already discussed how the positivity of these components ensure safety and convergence. Moreover, the lower bound of the integral in (9) ensures that the parameters k_i are such that an upper bound of the velocity control signal is minimized.

IV. OFFLINE OPTIMIZATION

Let us define the Hamiltonian associated with the adopted value function (7) as:

$$H(p, k, \nabla V(p)) = \nabla V(p)^T P(p) k + Q(p; p_0) + W(k) \tag{10}$$

Hence, the Bellman optimality equation is formed as follows:

$$\nabla V^*(p)^T P(p)k^* + Q(p;p_0) + W(k^*) = 0 \qquad (10^*)$$

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

$$\frac{k^*}{\alpha^2(p)} - \frac{1}{k^*} = -\frac{1}{\gamma}(\nabla V^*(p))^T P(p) \qquad (11)$$

that forms a simple quadratic equation. Solving (11) for k^* and keeping only the positive roots to establish safe navigation, we obtain the optimal control vector $k^* = [k_0^*, k_1^*, ..., k_M^*]$ as:

$$k_i^* = \frac{\alpha^2(p)\Gamma_i(p)+\sqrt{(\alpha^2(p)\Gamma_i(p))^2+4\alpha^2(p)}}{2}, i = 0, 1, .., M \quad (12)$$

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

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

Furthermore, it is evident that with this formulation all elements k_i^* of this vector are strictly positive by construction. Additionally, notice that if we had an analytical expression for the value function $V^*(p)$ we would be able to directly compute the optimal control vector k^*. However that is not the case, since in our formulation, one should replace the term k^* from (12) in (10*) and then solve a non-linear partial differential equation, which is rather hard to solve. Nevertheless, we shall remedy this issue employing, first the successive approximation theory [20] in an offline setting, and then RL to provide an online solution.

A. Successive Approximation of the Value Function

In this section, we prove that a method for successively approximating the Value Function of (2) is valid. First, we define an admissible control policy.

Definition 1 (Admissible Control): A control vector $k(p)$ is defined to be admissible with respect to (2) on $\mathcal{W}$, denoted by $k(p) \in \Psi(\mathcal{W})$, if $k(p)$ is continuous on $\mathcal{W}$, $k(p)$ stabilizes the system on $\mathcal{W}$ and $V(p)$ is finite for all $p \in \mathcal{W}$.

It is evident that the proposed parametrized control policies are admissible. Moreover, the Hamilton-Jacobi-Bellman equation is linear w.r.t. the value function, which motivates why we adopt successive approximation. The latter was introduced by [20] and later expanded by [19] for bounded controls. Nevertheless, we shall further prove the validity of this approach in our case, effectively expanding it for a control vector obeying only lower bounds, through the use of the appropriately selected function $W(k)$ in (9). Notice that the successive approximation technique is applied to (10) and (12). Hence, the following lemma proves how (12) can be used to improve the tuning policy for the control vector $k(p)$.

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

$$k_i^{(j+1)} = \frac{\alpha^2(p)\Gamma_i^{(j)}(p)+\sqrt{\left(\alpha^2(p)\Gamma_i^{(j)}(p)\right)^2+4\alpha^2(p)}}{2}, \quad i = 0, \cdots, M \qquad (13)$$

is an admissible control vector for (1) on $\mathcal{W}$.

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

$$\dot{V}^{(j)}(p, k^{(j+1)}) = \left(\nabla V_p^{(j)}\right)^T P(p)k^{(j+1)} \qquad (14)$$

Writing the HJB equation for this control yields:

$$H(p, k^{(j)}, \nabla V^{(j)}) = -\nabla V^{(j)}(p)^T P(p)k^{(j)} - Q(p;p_0) - W(k^{(j)}) = 0 \qquad (15)$$

Adding the above expression to (14) and invoking the fact that the quantity $-Q(p;p_0)$ is always negative away from the desired point p_0 and

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

owing to the Mean Value Theorem, it is clear that $\dot{V}^{(j)}(p, k^{(j+1)}) < 0$ and $V^{(j)}(p)$ is a Lyapunov function for $k^{(j+1)}$ on $\mathcal{W}$. Therefore, from Definition 1, $k^{(j+1)}$ is admissible on $\mathcal{W}$. ■

Notice that we can also prove that each successive approximation decreases the value function, through following a process similar to [19], i.e.:

$$V^*(p) \leq V^{(j+1)}(p) \leq V^{(j)}(p) \ , \forall p \in \mathcal{W} \qquad (16)$$

Finally, the stability and convergence properties of the robot trajectories may be proven through Lyapunov arguments with the cost function acting as the candidate Lyapunov function. Notice that his holds for all $p \in \mathcal{W}$ except for a measure zero subset $\Omega \subset \mathcal{W}$, since for any admissible initial control policy, the corresponding set Ω obtains zero measure and (16) holds true.

B. Neural Network Successive Approximation

Neural Networks have long been used to approximate sufficiently well functions within certain compact sets [21]. In our case $V^{(j)}(p)$ is approximated as:

$$V^{(j)}(p) = \sum_{i=1}^{L} w_i^{(j)}\phi_i(p) = (w^{(j)})^T \cdot \phi(p) \qquad (17)$$

which is a single-hidden-layer neural network with L neurons with activation functions $\phi_i(p) \in C^1(\mathcal{W})$ and $w_i^{(j)}$ weights.

The corresponding weights are then tuned in order to minimize the error of the approximation - in a least squares sense - over a number of samples taken on the workspace $\mathcal{W}$ as defined in Algorithm 1. To prove, i) convergence in the mean, ii) existence of the approximation in the least-squares sense and iii) uniqueness of the approximation as well as admissibility of $k^{(j+1)}$, we refer the reader to [19]. Finally, notice that the process described in Algorithm 1 is obviously an offline process, where the critic estimation weights are calculated in advance of the implementation of the motion planning policy.

Algorithm 1: Algorithm for the Neural Network Approximation of the Value Function

- **Sampling**;

Select N points $p_i, i = 1, \cdots, N$ within the workspace $\mathcal{W}$.

- **Initialize**;

Select an initial control vector $k(0) = [1, 1, ..., 1]^T \in \mathbb{R}^{(M+1)}$, which is an admissible policy.

while *Weights have not Converged* **do**

- **Weights Improvement Step:** Solve the following linear regression problem:

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

where

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

and

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

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

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

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

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

$$j \leftarrow j + 1$$

end

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

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

V. ONLINE OPTIMIZATION

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

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

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

$$k(w) = -\frac{\alpha^2(p)}{2\gamma}P^T(p)\nabla\phi^T(p)w + \\ + \sqrt{\left(\frac{\alpha^2(p)}{16\gamma}P(p)^T\nabla\phi^T(p)w\right)^2 + \alpha^2(p)} \tag{18}$$

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

$$e(\hat{w}) = \hat{w}^T\nabla\phi(p)P(p)k(\hat{w}) + Q(p;p_0) + W(k(\hat{w})) \tag{19}$$

where $k(\hat{w})$ denotes the estimation of the control vector provided by (18) based on the estimation of the NN weights. Hence, we formulate the tuning law for the NN weight estimates to minimize the cost function:

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

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

$$\dot{\hat{w}} = -a\frac{\sigma_2}{m_s}\left[\hat{w}^T\nabla\phi(p)P(p)k(\hat{w}) + Q(p;p_0) + W(k(\hat{w}))\right] \tag{20}$$

with $a > 0$, where

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

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

Theorem 1: The closed loop system $\dot{p} = P(p)\cdot k(\hat{w})$ with the adaptive law (20) guarantees that the trajectory for almost any initial position in the workspace converges safely to the desired position p_0.

Proof: We adopt the Lyapunov candidate function:

$$L(p, \hat{w}) = V(p) + \frac{1}{2}\tilde{w}^T a^{-1}\tilde{w} \tag{21}$$

where $V(p)$ is the unknown value function and $\tilde{w} = w - \hat{w}$ denotes the parametric error. It is easy to see that the above is always positive except for $p = p_0$ and $\tilde{w} = 0$. Now consider the dynamics of the weight estimation (20) in the following compact form:

$$\dot{\hat{w}} = -a\frac{\sigma_2}{m_s}e(\hat{w}) \tag{22}$$

Notice that the error in (19) can be written via a Taylor series expansion around $\hat{w}$ as follows:

$$e = -\sigma_2^T\tilde{w} + e_1 \tag{23}$$

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

$$\dot{\tilde{w}} = \alpha\frac{\sigma_2}{m_s}e = -\frac{\alpha}{m_s}\sigma_2\sigma_2^T\tilde{w} + \frac{ae_1}{m_s}\sigma_2 \tag{24}$$

which leads to:

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

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

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

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

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

Notice that, assuming persistently excited neurons, the above expression provides essentially a lower bound to $\|\tilde{w}\|$ for which the Lyapunov candidate is negative, which provides convergence as shown in [22]. ■

VI. RESULTS

In this section we will present the results of the offline solution, followed by a comparison between the online approach and an RRT* method. For the proposed algorithm, a grid of 15×15 neurons, consisting of Radial Basis Functions (RBFs), were used. All simulations were implemented with Matlab on a PC running Windows 10, on an intel-i7 quad-core processor. For the RTT* approach, a traditional quadratic form for the input part of the cost function was used. An artificial workspace was designed, with a square outer boundary of side lengths equal to 10[m], and three inner disk obstacles, as presented in Fig. 2. The goal position was $p_0 = (1, 1)$ for all runs. In Fig. 1, we illustrate the approximation of the value function and the respective vector field that resulted from the successive value function approximation of Algorithm 1. The approximation exhibits the desired behaviour, with large values away from the minimum at the goal position. In Fig. 2 we present four trajectories that resulted from various starting points, along with the same trajectories for the RRT* method. In Fig. 3, we present the respective tree graphs for each trajectory of the RRT* method. Finally, Table I includes the results for every run, including the start-end point configurations, the computed cost for each method and the corresponding run time. It is evident that our method consistently outperforms the RRT* optimization method, both in cost function value, and in run time. Additionally, our method produces smooth trajectories. The offline method outperforms the online one as expected, however, the trajectories of the online approach tend to match the offline ones as time progresses and the learning process evolves towards the optimal parameter estimates. Finally, all of the aforementioned trajectories exhibit both safety and convergence, as it has been rigorously proven.

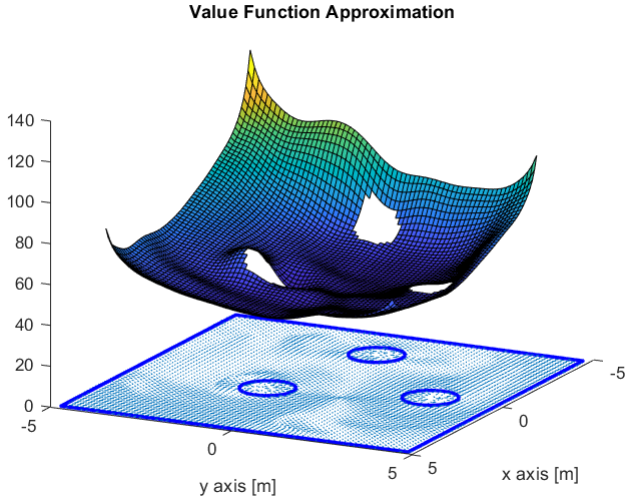

Fig. 1. The offline vector field and value function approximation.

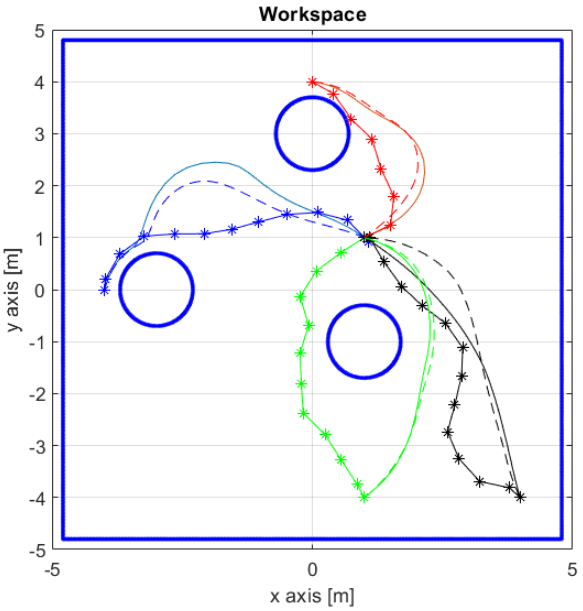

Fig. 2. The online trajectories (solid lines), the offline trajectories (dashed lines) and the RRT* trajectories (star points).

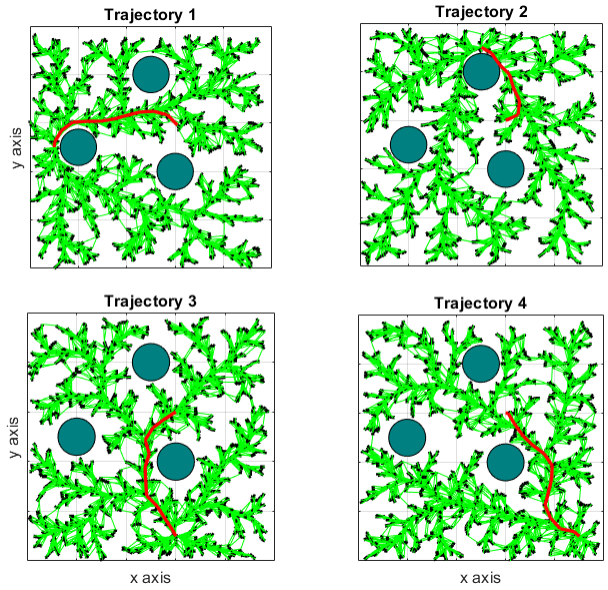

Fig. 3. The RRT* graphs and trajectories.

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

TABLE I
COMPARATIVE SIMULATION RESULTS

VII. FUTURE WORK

The results of this work are both promising and intriguing. As future directions, we intend to expand the application to unknown workspaces. Moreover, we intend to further generalize the above results with novel sets of parametrized control policies and more general controller forms.

REFERENCES

[1] J. T. Schwartz and M. Sharir, "On the "piano movers" problem. i: The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Communications on Pure and Applied Mathematics*, vol. 36, pp. 345 – 398, 05 1983.

[2] ——, "On the piano movers' problem: Iii. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers," *International Journal of Robotic Research - IJRR*, vol. 2, pp. 46–75, 09 1983.

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

[4] 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.

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

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

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

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

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

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

[11] S. Bhattacharya and R. Ghrist, "Path homotopy invariants and their application to optimal trajectory planning," *Annals of Mathematics and Artificial Intelligence*, August 2018, online-first.

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

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

[14] D. Koditschek, "Exact robot navigation by means of potential functions:Some topological considerations," in *Proceedings. 1987 IEEE Interna- tional Conference on Robotics and Automation*, vol. 4, March 1987, pp. 1–6.

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

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

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

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

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

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

[21] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos. Optimal Control, Third Edition. John Wiley & Sons, Inc, 2012

[22] J. Sun and P. A. Ioannou. Robust Adaptive Control. 2012

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