Approximate Optimal Online Continuous-Time Path-Planner with Static Obstacle Avoidance

Patrick Walters, Rushikesh Kamalapurkar, and Warren E. Dixon

***Abstract*—Online approximation of the optimal path for a control affine nonlinear autonomous agent subject to input and state constraints (e.g., actuator saturation, obstacles, no-enter zones) is considered. A model-based adaptive dynamic programming technique is implemented to locally estimate the unknown value function associated with the optimal path-planning problem. By performing a local approximation, the locations of the static obstacles do not need to be known until the obstacles are within a defined approximation window. The developed feedback policy guarantees ultimately bounded convergence of the approximated path to the optimal path without the requirement of persistence of excitation, typically required for online adaptive dynamic programming. Simulation results are presented to illustrate the performance of the proposed method.**

I. Introduction

Path-planning approaches can be divided into two types, pregenerative and reactive [1]. Pregenerative methods compute a path before a mission begins (c.f [1]–[4]), while reactive methods determine a path as the agent progresses through its environment. Planning optimal paths is of particular interest. Often pregenerative methods are used in planning optimal paths, which in the event of a disturbance requires the agent to take a non-optimal trajectory to return to the original optimal path or continually execute the planner. In contrast, an optimal reactive method (feedback motion planner) has the advantage of generating a policy that provides optimal feedback if the agent is forced off its original path.

In developing an optimal path for autonomous agents, it is often necessary to consider the agent's dynamics. In general, it is difficult to develop optimal path-planing strategies for nonlinear dynamics. One method of dealing with the challenges of path-planning under differential constraints is to pose the problem as an of optimal control problem. The corresponding Hamilton-Jacobi-Bellmen (HJB) equation can be numerically approximated to produce feedback policies. Dynamic programming with interpolation has been used as a feedback motion planner to compute an approximate optimal path through value iteration in results such as [5]. However, similar to pregenerative graph search methods (e.g., A*, Dijkstra), difficulties arise related to the state discretization as the order of the dynamics increase [6]. In results such as [7] and [8], feedback-based path-planning is generated offline by solving the HJB numerically. In the event of a change in the environment, such results would be required to recalculate a new optimal plan offline. In this result, we consider adaptive dynamic programming (ADP), which has been used to approximate the solution to HJB equation of general nonlinear systems online using parametric function approximation techniques (e.g., [9]–[13]).

Further complicating the task of optimal path-planning, state constraints (e.g., obstacles, no-enter zones) are often present en route to an objective. Autonomous agents are not able to sense all obstacles a priori, e.g., obstacles may remain undiscovered until they fall within a given sensing range. A recent advance in ADP bases the parametric approximation of the solution to the HJB on state-following (StaF) kernels [14]. These StaF kernels yield a local approximation of the HJB around the current state. By only utilizing information near the current state to approximate the solution, StaF does not require knowledge of obstacles outside an approximation window.

In addition to state constraints, input constraints inherent to the agent (e.g., maximum speed) are also important to consider. Results such as [12], [13], [15] have considered input constraints within the ADP framework. Utilizing a generalized non-quadratic local cost [16], the results in [12], [13], [15] yield a bounded approximate optimal controller.

Inspired by the advances in [12]–[15], an optimal feedback-based path-planner is developed in this paper that respects input and state constraints. The developed planner differs from optimal controllers in that it tackles the challenges specific to obstacle avoidance within the path-planning strategy. The local approximation in [14] enables the handling of obstacles not known a priori, but also introduces time-varying parameters in the parametric representation of the solution to the HJB. The time-varying parameters cause the estimation of the parameters to be in a near constant transient state making it difficult to prove that the generated feedback policy avoids the obstacles. This technical challenge motivates the introduction of an auxiliary feedback term to assist in navigating an agent around obstacles, and a scheduling function to switch between the approximate optimal feedback plan and the auxiliary feedback plan. Switching to the auxiliary feedback plan when the agent risks hitting an obstacle ensures obstacle avoidance. The proposed model-based ADP method approximates the optimal path using a combination of on-policy and off-policy

Patrick Walters, Rushikesh Kamalapurkar, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA. Email: {walters8, rkamalapurkar, wdixon}@ufl.edu

This research is supported in part by NSF award number 1509516 and ONR grant number N00014-13-1-0151. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring agency.

data, eliminating the need for physical exploration of the state space often associated with ADP. A Lyapunov-based stability analysis is presented which guarantees ultimately bounded convergence of the approximated path to the optimal path. Simulation results validate the proposed method.

II. Online Approximate Optimal Path-Planning

The objective of the proposed path-planner is to optimally navigate to a setpoint while avoiding static obstacles and adhering to the agent's actuation constraints.

A. Problem Formulation

Consider a nonlinear control affine dynamical system of the form

$$\dot{\zeta}\left(t\right)=\overline{f}\left(\zeta\left(t\right)\right)+\overline{g}\left(\zeta\left(t\right)\right)\overline{u}\left(t\right), \tag{1}$$

$t\in\left[t_{0},\infty\right)$, where t_{0} denotes the initial time, $\zeta:\left[t_{0},\infty\right)\rightarrow\mathbb{R}^{n}$ denotes the system state, $\overline{f}:\mathbb{R}^{n}\rightarrow\mathbb{R}^{n}$ denotes the drift dynamics, $\overline{g}:\mathbb{R}^{n}\rightarrow\mathbb{R}^{n\times m}$ denotes the control effectiveness, and $\overline{u}:\left[t_{0},\infty\right)\rightarrow\mathbb{R}^{m}$ denotes the control input.

The path-planning problem may be posed as a constrained infinite-horizon nonlinear regulation problem, i.e., to design a control signal $\overline{u}$ to minimize a subsequently defined cost function subject to the dynamic constraint in (1), while avoiding the obstacles and obeying $\sup_{t}\left(\overline{u}_{i}\right)\leq u_{sat}\ \forall i=1,\ldots,m$, where $\overline{u}=\left[\overline{u}_{1},\cdots,\overline{u}_{m}\right]^{T}$ and $u_{sat}\in\left[0,\infty\right)$ is the control saturation constant.

Static obstacles represent hard constraints that must be taken into account in the development of the approximate optimal path-planner. To this end, an auxiliary controller is subsequently developed to assist in navigating an agent around obstacles. The auxiliary controller is denoted by $u_{s}:\mathbb{R}^{n}\rightarrow\mathbb{R}^{m}$, where $u_{s}=\left[u_{s_{1}},\cdots,u_{s_{m}}\right]^{T}$ and $\sup_{t}\left(u_{s_{i}}\right)\leq u_{sat}\ \forall i=1,2,\ldots,m$. To facilitate the development of u_{s}, obstacles are augmented with a perimeter that extends from their borders denoting an unsafe region as illustrated in Figure 1. A smooth scheduling function $s:\mathbb{R}^{n}\rightarrow\left[0,a\right]$, where $a<1$, is used to transition between the approximate optimal controller $u:\left[t_{0},\infty\right)\rightarrow\mathbb{R}^{m}$ and the auxiliary controller u_{s} without introducing discontinuities to the system dynamics. The scheduling function and auxiliary controller are designed such that they are functions of the state and that they drive all state trajectories away from obstacles. In Section IV, the auxiliary controller u_{s} and the scheduling function s are designed for a specific system.

The control input $\overline{u}$ is defined as

$$\overline{u}\left(\zeta,t\right)\triangleq s\left(\zeta\right)u_{s}\left(\zeta\right)+\left(1-s\left(\zeta\right)\right)u\left(t\right), \tag{2}$$

where u is the subsequently designed approximate optimal controller. Based on (2), the agent dynamics can be rewritten as

$$\dot{\zeta}\left(t\right)=f\left(\zeta\left(t\right)\right)+g\left(\zeta\left(t\right)\right)u\left(t\right), \tag{3}$$

where $f:\mathbb{R}^{n}\rightarrow\mathbb{R}^{n}$ denotes the augmented drift dynamics defined as

$$f\left(\zeta\right)\triangleq\overline{f}\left(\zeta\right)+\overline{g}\left(\zeta\right)s\left(\zeta\right)u_{s}\left(\zeta\right)$$

and $g:\mathbb{R}^{n}\rightarrow\mathbb{R}^{n\times m}$ denotes the augmented control effectiveness defined as

$$g\left(\zeta\right)\triangleq\overline{g}\left(\zeta\right)\left(1-s\left(\zeta\right)\right).$$

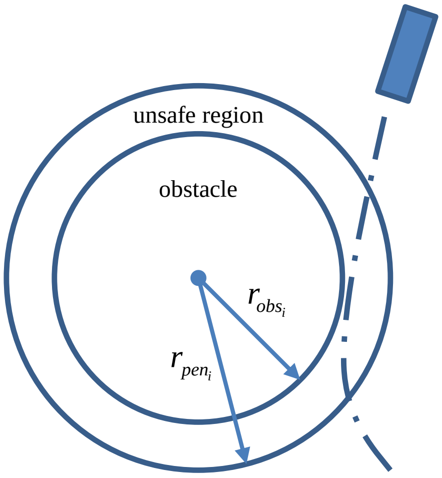

Figure 1. Obstacles are augmented with an unsafe region that extends from it's border.

To account for actuator saturation and the unsafe regions, the cost function is defined as

$$J\left(\zeta,u\right)\triangleq\int_{t_{o}}^{\infty}r\left(\zeta\left(\tau\right),u\left(\tau\right)\right)d\tau, \tag{4}$$

where $r:\mathbb{R}^{n}\times\mathbb{R}^{m}\rightarrow\left[0,\infty\right)$ is the local cost defined as

$$r\left(\zeta,u\right)\triangleq\zeta^{T}Q\zeta+P\left(\zeta\right)+U\left(u\right), \tag{5}$$

subject to the dynamic constraint in (3), where $Q\in\mathbb{R}^{n\times n}$ is a constant, user defined, symmetric positive definite weighting matrix, $P:\mathbb{R}^{n}\rightarrow\mathbb{R}$ is a non-negative continuous function penalizing trajectories that enter the unsafe region, and $U:\mathbb{R}^{m}\rightarrow\mathbb{R}$ is a positive definite function penalizing control effort. The matrix Q has the property $\underline{q}\left\Vert \xi_{q}\right\Vert ^{2}\leq\xi_{q}^{T}Q\xi_{q}\leq\overline{q}\left\Vert \xi_{q}\right\Vert ^{2},\ \forall\xi_{q}\in\mathbb{R}^{n}$ where $\underline{q}$ and $\overline{q}$ are positive constants.

The positive definite function U in (5) is defined as [12], [17]

$$U\left(u\right)\triangleq2\sum_{i=1}^{m}\left[\int_{0}^{u_{i}}\left(u_{sat}r_{i}\tanh^{-1}\left(\frac{\xi_{u_{i}}}{u_{sat}}\right)\right)d\xi_{u_{i}}\right] \tag{6}$$

where u_{i} is the i^{th} element of u, $\xi_{u_{i}}$ is an integration variable, and $R\in\mathbb{R}^{m\times m}$ is a diagonal positive definite, user defined, weighting matrix given as $R=\mathrm{diag}\left(\left[r_{1},r_{2},\cdots,r_{m}\right]\right)$.

The infinite-time scalar value function $V:\mathbb{R}^{n}\rightarrow\left[0,\infty\right)$ for the optimal solution is written as

$$V\left(\zeta\right)=\min_{u}\int_{t_{0}}^{\infty}r\left(\zeta\left(\tau\right),u\left(\tau\right)\right)d\tau. \tag{7}$$

The objective of the optimal path-planner is to find the optimal policy $u^*: \mathbb{R}^n \to \mathbb{R}^m$ that minimizes the performance index (4) with the local cost (5) subject to the dynamic constraint in (3).

The optimal value function is characterized by the HJB, which is given as

$$\frac{\partial V(\zeta)}{\partial \zeta}\left(f(\zeta)+g(\zeta)u^*(\zeta)\right)+r\left(\zeta,u^*(\zeta)\right)=0 \quad (8)$$

with the boundary condition $V(0)=0$. The optimal control policy can be determined from (8) as

$$u^*(\zeta)=-u_{sat}\tanh\left(\frac{1}{2u_{sat}}R^{-1}g^T\left(\frac{\partial V(\zeta)}{\partial \zeta}\right)^T\right). \quad (9)$$

The analytical expression for the optimal path in (9) requires knowledge of the value function which is the solution to the HJB equation in (8). The HJB equation is a partial differential equation which is generally infeasible to solve analytically; hence, an approximate solution is sought.

B. Local Approximation of the Value Function

The subsequent development is based on an approximation of the value function and optimal policy. Differing from previous ADP literature (e.g., [?], [10], [12], [13], [18]) that seeks a global policy, the following development seeks only a local policy. Instead of generating an approximation of the value function over the entire operating region, we aim to approximate a small region about the current state. With the region of approximation limited to a small range about the current state, one only needs to assume that there may exist an obstacle or obstacles outside the local approximation. Once inside the local approximation window, the optimal policy will adapt to avoid the obstacle. Despite the uncertainty of distant obstacles, the following development yields guaranteed stability of the state and convergence to the optimal path.

Leveraging the results of [14], StaF kernels are employed to approximate the local policy on some small compact set $B_r(\zeta)$, i.e., the approximation window, around the state ζ. The StaF representation of the value function and optimal policy are given as

$$V(\zeta)=W(\zeta)^T\sigma(\zeta,c(\zeta))+\epsilon(\zeta),$$

$$u^*(\zeta)=-u_{sat}\tanh\left(\frac{R^{-1}g(\zeta)^T}{2u_{sat}}\left(\sigma'(\zeta,c(\zeta))^T W(\zeta)\right.\right. \\ \left.\left.+\epsilon'(\zeta)\right)\right), \quad (10)$$

respectively, where $W: \mathbb{R}^n \to \mathbb{R}^l$ is the ideal weight vector, $\sigma: \mathbb{R}^n \to \mathbb{R}^l$ is a continuously differentiable kernel function, and $\epsilon: \mathbb{R}^n \to \mathbb{R}$ is the continuously differentiable function reconstruction error, and $\sigma': \mathbb{R}^n \to \mathbb{R}^{l\times n}$ and $\epsilon': \mathbb{R}^n \to \mathbb{R}^n$ are the partial derivatives with respect to the state. Note that the centers of the kernel function change as the system state changes; therefore, the ideal weight vector W is a time-varying function. The approximations of the value function and the optimal policy are defined as

$$\hat{V}\left(\zeta,\hat{W}_c\right)\triangleq\hat{W}_c^T\sigma(\zeta,c(\zeta)), \quad (11)$$

$$\hat{u}\left(\zeta,\hat{W}_a\right)\triangleq-u_{sat}\tanh\left(\frac{R^{-1}g(\zeta)^T}{2u_{sat}}\sigma'(\zeta,c(\zeta))^T\hat{W}_a\right), \quad (12)$$

where $c(\zeta)\in B_r(\zeta)$ is the StaF kernel center, and $\hat{W}_c, \hat{W}_a \in \mathbb{R}^l$ are estimates of the ideal weight vector W.

Substituting the approximations from (11) and (12) into (8), results in a residual error $\delta: \mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^l\to\mathbb{R}$ called the Bellman error given by

$$\delta\left(\zeta,\hat{W}_c,\hat{W}_a\right)=r\left(\zeta,\hat{u}\left(\zeta,\hat{W}_a\right)\right) \\ +\frac{\partial\hat{V}\left(\zeta,\hat{W}_c\right)}{\partial\zeta}\left(f(\zeta)+g(\zeta)\hat{u}\left(\zeta,\hat{W}_a\right)\right). \quad (13)$$

C. Online Learning

At a given time instant t, the Bellman error $\delta_t: [0,\infty)\to\mathbb{R}$ is

$$\delta_t(t)=\delta\left(\zeta(t),\hat{W}_c(t),\hat{W}_a(t)\right),$$

where $\hat{W}_c(t)$ and $\hat{W}_a(t)$ denote the estimates of the ideal weights at time t, and $\zeta(t)$ denotes the state of the system in (1) at time t starting from initial time t_o and initial state ζ_o under the influence of the state feedback controller

$$u(t)=\hat{u}\left(\zeta(t),\hat{W}_a(t)\right).$$

The Bellman error is extrapolated to off-policy sampled states $\{\zeta_k(t)\in B_r(\zeta(t))\,|k=1,2,\ldots,N\}$ that follow the system state. The extrapolated Bellman error $\delta_k: [0,\infty)\to\mathbb{R}$ is given as

$$\delta_k(t)=\hat{W}_c^T(t)\,w_k(t)+r(\zeta_k(t),\hat{u}_k(t)),$$

where

$$\hat{u}_k(t)=-u_{sat}\tanh\left(\frac{R^{-1}g}{2u_{sat}}(\zeta_k(t))^T\right. \\ \left.\sigma'(\zeta_k(t),c(\zeta(t)))^T\hat{W}_a(t)\right)$$

and

$$\omega_k(t)=\sigma'(\zeta_k(t),c(\zeta(t)))\left[f(\zeta_k(t))\right. \\ \left.+g(\zeta_k(t))\,\hat{u}\left(\zeta(t),\hat{W}_a(t)\right)\right].$$

The value function least squares update law based on the minimization of the instantaneous and extrapolated Bellman error is given by

$$\dot{\hat{W}}_c(t)=-\Gamma(t)\left(k_{c1}\frac{\partial\delta_t(t)}{\partial\hat{W}_c}\frac{\delta_t(t)}{\rho(t)}\right. \\ \left.+\frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{\partial\delta_k(t)}{\partial\hat{W}_c}\frac{\delta_k(t)}{\rho_k(t)}\right), \quad (14)$$

$$\dot{\Gamma}\left(t\right)=\begin{cases}\beta\Gamma\left(t\right)-k_{c1}\Gamma\left(t\right)\frac{\omega(t)\omega(t)^{T}}{\rho(t)}\Gamma\left(t\right), & \left\Vert \Gamma\left(t\right)\right\Vert \leq\overline{\Gamma}\\0, & \text{otherwise}\end{cases},\tag{15}$$

where $k_{c1}, k_{c2}\in\mathbb{R}$ are positive adaptation gains, $\left\Vert\Gamma\left(t_0\right)\right\Vert = \left\Vert\Gamma_0\right\Vert\leq\bar{\Gamma}$ is the initial adaptation gain, $\bar{\Gamma}\in\mathbb{R}$ is a positive saturation gain, $\beta\in\mathbb{R}$ is a positive forgetting factor,

$$\omega\left(t\right)=\sigma'\left(\zeta\left(t\right),c\left(\zeta\left(t\right)\right)\right)\left[f\left(\zeta\left(t\right)\right)\right.\\ \left.+g\left(\zeta\left(t\right)\right)\hat{u}\left(\zeta\left(t\right),\hat{W}_{a}\left(t\right)\right)\right]$$

is the instantaneous regressor matrix,

$$\rho\left(t\right)=1+k_{\rho}\omega\left(t\right)^{T}\omega\left(t\right)$$

is the instantaneous normalization constant,

$$\rho_{k}\left(t\right)=1+k_{\rho}\omega_{k}\left(t\right)^{T}\omega_{k}\left(t\right)$$

is the extrapolated normalization constant, and $k_{\rho}\in\mathbb{R}$ is a positive gain.

The policy update law is given by

$$\dot{\hat{W}}_{a}\left(t\right)=\text{proj}\left\{-k_{a}\left(\hat{W}_{a}\left(t\right)-\hat{W}_{c}\left(t\right)\right)\right\},\tag{16}$$

where $k_a\in\mathbb{R}$ is an positive gain, and proj $\{\cdot\}$ is a smooth projection operator[1] used to bound the weight estimates. The weight estimation errors are then defined as $\tilde{W}_c\left(t\right)\triangleq W\left(\zeta\left(t\right)\right)-\hat{W}_c\left(t\right)$ and $\tilde{W}_a\left(t\right)\triangleq W\left(\zeta\left(t\right)\right)-\hat{W}_a\left(t\right)$.

In Section III, Bellman error extrapolation is employed to establish ultimately bounded convergence of the approximate policy to the optimal policy without requiring persistence of excitation provided the following assumption is satisfied.

Assumption 1. There exists a strictly positive constant $\underline{c}$ such that

$$\underline{c}=\inf_{t\in[t_{0},\infty)}\left[\lambda_{min}\left(\sum_{k=1}^{N}\frac{\omega_{k}\left(t\right)\omega_{k}\left(t\right)^{T}}{\rho_{k}\left(t\right)^{2}}\right)\right].$$

In general, Assumption 1 cannot be guaranteed to hold a priori; however, heuristically, the condition can be met by sampling redundant data, i.e., $N\gg l$.

III. Stability Analysis

For notational brevity, all function dependencies from previous sections are henceforth suppressed. Let the notation F_k denote the function $F\left(\zeta,\cdot\right)$ evaluated at the sampled state, i.e., $F_k\left(\cdot\right)=F\left(\zeta_k,\cdot\right)$. An unmeasurable form of the Bellman error can be written as

$$\delta=-\tilde{W}_{c}^{T}\omega+W^{T}\sigma'g\left(\hat{u}-u^{*}\right)+U\left(\hat{u}\right)-U\left(u^{*}\right)\\-\epsilon'\left(f+gu^{*}\right).\tag{17}$$

Similarly, the Bellman error at the extrapolated points can be written as

$$\delta_{k}=-\tilde{W}_{c}^{T}\omega_{k}+W^{T}\sigma_{k}'g_{k}\left(\hat{u}_{k}-u_{k}^{*}\right)+U\left(\hat{u}_{k}\right)-U\left(u_{k}^{*}\right)\\-\epsilon_{k}'\left(f_{k}+g_{k}u_{k}^{*}\right).\tag{18}$$

To facilitate the subsequent stability analysis, consider the candidate Lyapunov function $V_L:\mathbb{R}^n\times\mathbb{R}^l\times\mathbb{R}^l\to[0,\infty)$ given as

$$V_{L}\left(Z\right)=V+\frac{1}{2}\tilde{W_{c}}^{T}\Gamma^{-1}\tilde{W}_{c}+\frac{1}{2}\tilde{W}_{a}^{T}\tilde{W}_{a},$$

where $Z\triangleq\left[\begin{array}{ccc}\zeta^{T} & \tilde{W}_{c}^{T} & \tilde{W}_{a}^{T}\end{array}\right]^{T}\in\chi\times\mathbb{R}^{l}\times\mathbb{R}^{l}$. Since the value function V in (7) is positive definite [21] using Lemma 4.3 of [22], V_L can be bounded by

$$\underline{\upsilon_{L}}\left(\left\Vert Z\right\Vert\right)\leq V_{L}\left(Z\right)\leq\overline{\upsilon_{L}}\left(\left\Vert Z\right\Vert\right),\tag{19}$$

where $\underline{\upsilon_L}$, $\overline{\upsilon_L}:[0,\infty)\to[0,\infty)$ are class $\mathcal{K}$ functions. Define the constant $K\in\mathbb{R}$ as

$$K\triangleq\sqrt{\frac{\iota_{c}^{2}}{\alpha\left(2\underline{c}-k_{a}\right)}+\frac{\iota_{a}^{2}}{\alpha k_{a}}+\frac{\iota}{\alpha}}$$

where $\alpha\triangleq\min\left\{\frac{\underline{q}}{2},\left(\frac{\underline{c}}{2}-\frac{k_{a}}{4}\right),\frac{k_{a}}{4}\right\}$ and ι_c, ι_a, ι are positive constants.

Theorem 1. *Provided Assumption 1 is satisfied along with the sufficient conditions*

$$\underline{c}>\frac{k_{a}}{2},\tag{20}$$

$$K<\underline{\upsilon_{L}}^{-1}\left(\overline{\upsilon_{L}}\left(r\right)\right),\tag{21}$$

where $r\in\mathbb{R}$ is the radius of the compact set β_L, then the policy in (12) *with the update laws in* (14)-(16) *guarantee ultimately bounded regulation of the state ζ and of the approximated policies $\hat{u}$ to the optimal policy u^*.*

Proof: The time derivative of the candidate Lyapunov function is

$$\dot{V}_{L}=\frac{\partial V}{\partial\zeta}f+\frac{\partial V}{\partial\zeta}g\hat{u}-\frac{1}{2}\tilde{W}_{c}^{T}\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_{c}\\-\tilde{W}_{c}^{T}\Gamma^{-1}\left(\dot{W}-\dot{\hat{W}}_{c}\right)-\tilde{W}_{a}^{T}\left(\dot{W}-\dot{\hat{W}}_{a}\right).\tag{22}$$

Using Theorem 2 in [14], the time derivative of the ideal weights can be expressed as

$$\dot{W}=W'\left(f+g\hat{u}\right).\tag{23}$$

Substituting (14)-(18) and (23) yields

$$\dot{V}_{L}=\frac{\partial V}{\partial\zeta}f+\frac{\partial V}{\partial\zeta}g\hat{u}\\-\frac{1}{2}\tilde{W}_{c}^{T}\Gamma^{-1}\left[\left(\beta\Gamma-k_{c1}\Gamma\frac{\omega\omega^{T}}{\rho}\Gamma\right)\mathbf{1}_{\left\Vert\Gamma\right\Vert\leq\overline{\Gamma}}\right]\Gamma^{-1}\tilde{W}_{c}\\+\tilde{W}_{c}^{T}\left[k_{c1}\frac{\omega}{\rho}\delta+\frac{k_{c2}}{N}\sum_{j=1}^{N}\frac{\omega_{k}}{\rho_{k}}\delta_{k}\right]-\tilde{W}_{c}^{T}\Gamma^{-1}W'\left(f+g\hat{u}\right)\\+\tilde{W}_{a}^{T}k_{a}\left(\hat{W}_{a}-\hat{W}_{c}\right)-\tilde{W}_{a}^{T}W'\left(f+g\hat{u}\right).$$

[1] See Section 4.4 in [19] or Remark 3.6 in [20] for details of the projection operator.

Using Young's inequality, (12), (17), and (18) the Lyapunov derivative can be upper bounded as

$$\dot{V}_L \le -\underline{q}\left\|\zeta\right\|^2 - \left(\underline{c} - \frac{k_a}{2}\right)\left\|\tilde{W}_c\right\|^2 - \frac{k_a}{2}\left\|\tilde{W}_a\right\|^2 + \iota_c\left\|\tilde{W}_c\right\| + \iota_a\left\|\tilde{W}_a\right\| + \iota.$$

Completing the squares, the upper bound on the Lyapunov derivative may be written as

$$\dot{V}_L \le -\frac{\underline{q}}{2}\left\|\zeta\right\|^2 - \left(\frac{\underline{c}}{2} - \frac{k_a}{4}\right)\left\|\tilde{W}_c\right\|^2 - \frac{k_a}{4}\left\|\tilde{W}_a\right\|^2 + \frac{\iota_c^2}{2\underline{c} - k_a} + \frac{\iota_a^2}{2k_a} + \iota,$$

which can be further upper bounded as

$$\dot{V}_L \le -\alpha\left\|Z\right\|, \; \forall \left\|Z\right\| \ge K > 0. \tag{24}$$

Using (19), (20), (21), and (24), Theorem 4.18 in [22] is invoked to conclude that Z is ultimately bounded, in the sense that $\limsup_{t\to\infty}\left\|Z\left(t\right)\right\| \le \underline{v_L}^{-1}\left(\overline{v_L}\left(K\right)\right)$.

Based on the definition of Z and the inequalities in (19) and (24), $\zeta, \tilde{W}_c, \tilde{W}_a \in \mathcal{L}_\infty$. Since $\zeta \in \mathcal{L}_\infty$ and W is a continuous function of ζ, $W\left(\zeta\right) \in \mathcal{L}_\infty$. Hence, $\hat{W}_c$, $\hat{W}_a \in \mathcal{L}_\infty$, which implies $\hat{u} \in \mathcal{L}_\infty$. From the definitions of u_s and s, $\overline{u} \in \mathcal{L}_\infty$. ■

IV. Simulation Results

Simulation results are provided to demonstrate the performance of the developed ADP-based path-planner. The simulation is performed for the control affine system given in (3), where $\overline{f}\left(\zeta\right) = \mathbf{0}$ and $\overline{g}\left(\zeta\right) = I_{2\times 2}$.

For this particular example, the smooth scheduling function is defined as

$$s\left(\zeta\right) \triangleq \sum_{i=1}^{M}\begin{cases} 0.95, & \left\|\zeta - c_{obs_i}\right\| \le r_{obs_i} \\ 0.95T\left(\zeta\right), & r_{obs_i} < \left\|\zeta - c_{obs_i}\right\| \le r_{pen_i} \\ 0, & \text{otherwise} \end{cases}, \tag{25}$$

where

$$T\left(\zeta\right) \triangleq \left(\frac{1}{2} + \frac{1}{2}\cos\left(\frac{\pi}{r_{pen_i} - r_{obs_i}}\left(\left\|\zeta - c_{obs_i}\right\| - r_{obs_i}\right)\right)\right),$$

M is the number of obstacles, r_{obs_i} is a positive constant indicating the radius of the i^{th} obstacle, r_{pen_i} is a positive constant indicating the radius corresponding to the unsafe region surrounding the i^{th} obstacle, and $c_{obs_i} \in \mathbb{R}^n$ denotes the center corresponding to the i^{th} obstacle. With this formulation of the smooth scheduling function, it is assumed that the obstacles are selected such that unsafe regions do not overlap[2].

The continuous auxiliary controller u_s is defined as

$$u_s\left(\zeta\right) \triangleq \frac{u_{sat}\left(\zeta - c_{obs_i}\right)}{\left\|\zeta - c_{obs_i}\right\|}. \tag{26}$$

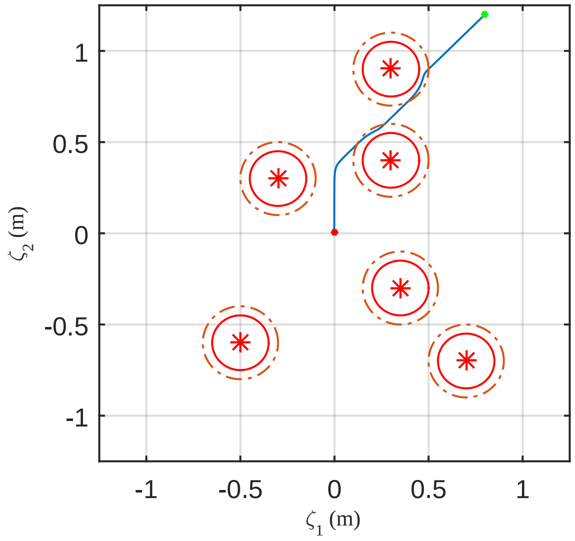

Figure 2. The path generated by the developed method is shown, where the dashed lines around the obstacles denote the boundary of the unsafe region and the solid lines denote the boundary of the obstacle.

In Appendix A , the auxiliary controller in (26) is shown to prevent the agent from entering the interior of an obstacle.

The unsafe region is where the penalty function P begins to effect the agent's cost. The non-negative function P is given as

$$P\left(\zeta\right) = \sum_{i=1}^{M}\begin{cases} 40 & \left\|\zeta - c_{obs_i}\right\| \le r_{obs_i} \\ 40T\left(\zeta\right), & r_{obs_i} < \left\|\zeta - c_{obs_i}\right\| \le r_{pen_i} \\ 0, & \text{otherwise} \end{cases}.$$

The initial state is selected as $\zeta_0 = \begin{bmatrix} 0.8 & 1.2 \end{bmatrix}^T$. The StaF basis for the value function approximation is selected as

$$\sigma = \begin{bmatrix} \zeta^T\left(\zeta + d_1\right) & \zeta^T\left(\zeta + d_2\right) & \zeta^T\left(\zeta + d_3\right) \end{bmatrix}^T,$$

where the centers of the kernels are selected as $d_1 = 0.005 \cdot \begin{bmatrix} 0 & 1 \end{bmatrix}^T$, $d_2 = 0.005 \cdot \begin{bmatrix} 0.8660 & -0.5 \end{bmatrix}^T$, $d_3 = 0.005 \cdot \begin{bmatrix} -0.8660 & -0.5 \end{bmatrix}^T$. The Bellman error is extrapolated to 25 sampled data points that are selected on a uniform 5×5 grid that spans a square of size 0.01, and is centered about the current state. The weighting matrices are selected as $\mathcal{Q} = I_{2\times 2}$ and $\mathcal{R} = I_{2\times 2}$. The learning gains are selected as $k_{c1} = 0.25$, $k_{c2} = 0.15$, $k_a = 0.5$, $\beta = 0.3$, and $k_\rho = 0.05$. The least squares update law's initial condition is selected as $\Gamma_0 = 300 \cdot I_{3\times 3}$. The policy and value function weight estimates are arbitrarily initialized to

$$\hat{W}_c\left(0\right) = \hat{W}_a\left(0\right) = \begin{bmatrix} 0.4 & 0.4 & 0.4 \end{bmatrix}^T.$$

The generated path for both simulation trials are shown in Figure 2. Note that the state trajectories in Figure 2 briefly enter the unsafe region, where the auxiliary controller successfully escorts the agent away from the obstacle.

[2] If a group of obstacles are close enough for the unsafe regions to overlap, then the group may be considered as one large obstacle.

V. Conclusion

An online approximation of a robust optimal path-planning strategy is developed. The solution to the HJB equation is approximated using adaptive dynamic programming. Since the unknown value function is approximated locally, the locations of the static obstacles do not need to be known until the obstacles are within an approximation window. The developed feedback policy guarantees ultimately bounded convergence of the approximated path to the optimal path without the requirement of persistent excitation, typically required for online adaptive dynamic programming. The results are validated with simulations. Future work will focus on comparisons with existing pregenerative literature to evaluate the accuracy of the developed method.

Appendix A
Auxiliary Controller Analysis

Consider a change of coordinates, where $\overline{\zeta} = \zeta - c_{obs_i}$. A positive definite function $V_{obs} : [0, \infty) \to \mathbb{R}^n$ is given as

$$V_{obs} = \overline{\zeta}^T \overline{\zeta}. \tag{27}$$

The time derivative of (27) is

$$\dot{V}_{obs} = 2\overline{\zeta}^T \dot{\zeta}.$$

Substituting the dynamics (1) with the definitions of $\overline{f}$ and $\overline{g}$ provided in Section IV, and the controller in (2) yields

$$\dot{V}_{obs} = 2\overline{\zeta}^T \left(s\left(\zeta\right) u_s + \left(1 - s\left(\zeta\right)\right) u\left(t\right)\right).$$

Substituting the auxiliary controller defined in (26) and the fact that the norm of the optimal controller u is bounded by $\sqrt{2}u_{sat}$, the derivative is lower bounded by

$$\dot{V}_{obs} \geq 2u_{sat} \left\|\zeta\right\| \left(s\left(\zeta\right) - \frac{\sqrt{2}}{1+\sqrt{2}} \right). \tag{28}$$

Let B_{obs_i} denote the local domain of the obstacle centered at c_{obs_i} defined as $B_{obs_i} \triangleq \left\{\zeta | \overline{\zeta} \leq r_{obs_i}\right\}$. By the definition of the scheduling function in (25),

$$\inf_{\zeta \in B_{obs_i}} s\left(\zeta\right) = 0.95. \tag{29}$$

Consider the inequality in (28) on the local domain B_{obs_i}, then (28) is further bounded by

$$\dot{V}_{obs} \geq 2u_{sat} \left\|\zeta\right\| \left(\sqrt{2} - 1.05 \right).$$

Substituting the function V_{obs}, the derivative may be written as

$$\dot{V}_{obs} \geq 2u_{sat} V_{obs}^{\frac{1}{2}} \left(\sqrt{2} - 1.05 \right).$$

Solving the differential equation using separation of variables, yields

$$V_{obs} \geq \left(\sqrt{2} - 1.05 \right)^2 u_{sat}^2 t^2$$

Hence, the obstacle center c_{obs_i} is unstable in the local domain B_{obs_i}. Furthermore, a state trajectory starting outside the local domain B_{obs_i} will not enter the interior of B_{obs_i}.

References

[1] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," vol. 29, pp. 418–429, 2004.

[2] A. V. Rao, D. A. Benson, C. L. Darby, M. A. Patterson, C. Francolin, and G. T. Huntington, "Algorithm 902: GPOPS, A MATLAB software for solving multiple-phase optimal control problems using the Gauss pseudospectral method," *ACM Trans. Math. Softw.*, vol. 37, no. 2, pp. 1–39, 2010.

[3] K. Yang, S. K. Gan, and S. Sukkarieh, "An efficient path planning and control algorithm for RUAVs in unknown and cluttered environments," *J. Intell. Robot Syst.*, vol. 57, pp. 101–122, 2010.

[4] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. R*, vol. 30, pp. 846–894, 2011.

[5] S. M. LaValle and P. Konkimalla, "Algorithms for computing numerical optimal feedback motion strategies," *Int. J. Rob. Res.*, vol. 20, pp. 729–752, 2001.

[6] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[7] A. Shum, K. Morris, and A. Khajepour, "Direction-dependent optimal path planning for autonomous vehicles," *Robot. and Auton. Syst.*, 2015.

[8] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," vol. 23, pp. 331–341, 2007.

[9] K. Vamvoudakis and F. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[10] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Netw.*, vol. 22, no. 3, pp. 237 – 246, 2009.

[11] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, 2013.

[12] H. Modares, F. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, 2013.

[13] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[14] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, "State following (StaF) kernel functions for function approximation Part II: Adaptive dynamic programming," in *Proc. Am. Control Conf.*, 2015, pp. 521–526.

[15] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Hamilton-Jacobi-Isaacs formulation for constrained input nonlinear systems," in *Proc. IEEE Conf. Decis. Control*, vol. 5, 2004, pp. 5034–5040.

[16] S. Lyshevski, "Optimal control of nonlinear continuous-time systems: design of bounded controllers via generalized nonquadratic functionals," in *Proc. Am. Control Conf.*, 1998.

[17] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, 2009.

[18] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Concurrent learning-based approximate optimal regulation," in *Proc. IEEE Conf. Decis. Control*, Florence, IT, Dec. 2013, pp. 6256–6261.

[19] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[20] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhauser: Boston, 2003.

[21] M. Abu-Khalaf and F. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[22] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.