Approximate Optimal Motion Planning to Avoid Unknown Moving Avoidance Regions

Patryk Deptula, Hsi-Yuan Chen, Ryan A. Licitra, Joel A. Rosenfeld, and Warren E. Dixon, *Fellow, IEEE*

***Abstract*—In this article, an infinite-horizon optimal regulation problem is considered for a control-affine nonlinear autonomous agent subject to input constraints in the presence of dynamic avoidance regions. A local model-based approximate dynamic programming method is implemented to approximate the value function in a local neighborhood of the agent. By performing local approximations, prior knowledge of the locations of avoidance regions is not required. To alleviate the *a priori* knowledge of the number of avoidance regions in the operating domain, an extension is provided that modifies the value function approximation. The developed feedback-based motion planning strategy guarantees uniformly ultimately bounded convergence of the approximated control policy to the optimal policy while also ensuring the agent remains outside avoidance regions. Simulations are included to demonstrate the preliminary development for a kinematic unicycle and generic nonlinear system. Results from three experiments are also presented to illustrate the performance of the developed method, where a quadcopter achieves approximate optimal regulation while avoiding three mobile obstacles. To demonstrate the developed method, known avoidance regions are used in the first experiment, unknown avoidance regions are used in the second experiment, and an unknown time-varying obstacle directed by a remote pilot is included in the third experiment.**

***Index Terms*—Data-based control, learning and adaptive systems, motion and path planning, neural and fuzzy control, optimization and optimal control.**

I. Introduction

MANY challenges exist for real-time navigation in uncertain environments. To operate safely in an uncertain environment, an autonomous agent must identify and react to possible collisions. In practice, challenges come from limitations in computational resources, sensing, communication, and mobility. Hence, robot navigation, motion planning, and path planning continues to be an active research area (cf., [1] and references therein).

Because motion and path-planning strategies need to account for environmental factors with various uncertainties, they can be divided into two groups—global and local approaches [2]. Global planners seek the best trajectory by using models of the entire environment, are computed before a mission begins, and tend to provide high-level plans (cf., [3]–[7]). Local planners (sometimes referred to as reactive methods) plan only a few time steps forward based on limited knowledge using sensory data; hence, they have the advantage of providing optimal feedback if the agent is forced off of its original path, but they may need to be recomputed online (cf., [7]–[10]). Since complex operating conditions present significant navigation, guidance, and control challenges (i.e., agents' dynamics, obstacles, disturbances, or even faults), online feedback-based control/guidance algorithms with online learning and adaptation capabilities are essential for replanning and execution in dynamically changing and uncertain environments. Constrained optimization methods can be leveraged to generate guidance/control laws for agents operating in complex environments. However, agents often exhibit nonlinear dynamics and navigate in environments with uncertain dynamics or constraints, which makes the determination of analytical solutions to constrained optimization problems difficult. Traditional guidance/control solutions exploit numerical methods to generate approximate optimal solutions. For instance, approaches may use pseudoscpectral methods, they may solve the Hamilton–Jocobi–Bellman (HJB) equation offline via discretization and interpolation, or viscosity solutions can be solved offline before a mission begins (cf., [3], [11]–[14]). Such results may provide performance guarantees; however, numerical nonlinear optimization problems are typically computationally expensive (often preventing real-time implementation), especially as the dimension of the system increases. Generally, numerical methods are unable to consider uncertainty in the dynamics or environment, and are ill suited for dynamically changing environments because new guidance/control solutions would need to be recalculated offline in the event of a change in the environment. Such challenges motivate the use of approximate optimal control methods that use parametric function approximation techniques capable of approximating the solution to the HJB online (cf., [15]–[26]).

Manuscript received February 7, 2019; accepted November 12, 2019. Date of publication December 10, 2019; date of current version April 2, 2020. This article was recommended for publication by Associate Editor S.-J. Chung and Editor P. Robuffo Giordano upon evaluation of the reviewers' comments. This work was supported in part by National Science Foundation (NSF) under Award 1509516, in part by the Office of Naval Research under Grant N00014-13-1-0151, and in part by the Air Force Office of Scientific Research (AFOSR) under Award FA9550-19-1-0169. The work of P. Deptula was done prior to joining The Charles Stark Draper Laboratory, Inc. The work of H.-Y. Chen was done prior to joining Amazon Robotics. *(Corresponding author: Patryk Deptula.)*

P. Deptula is with the Perception and Autonomy Group, The Charles Stark Draper Laboratory, Inc., Cambridge, MA 02139 USA (e-mail: pdeptula@draper.com).

H.-Y. Chen is with the Amazon Robotics, North Reading, MA 01864 USA (e-mail: hsiyuc@amazon.com).

R. A. Licitra and W. E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: rlicitra@ufl.edu; wdixon@ufl.edu).

J. A. Rosenfeld is with the Department of Mathematics and Statistics, University of South Florida, Tampa, FL 33620 USA (e-mail: rosenfeldj@usf.edu).

This article has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2019.2955321

Further complicating the task of optimal motion planning are agent actuator constraints and state constraints (e.g., static or mobile avoidance regions) often present en route to an objective. Certain avoidance regions may remain undiscovered until they fall into a given detection range. The concept of avoidance control was introduced in [7] for two-player pursuit-evasion games. However, results such as [9], [10], and [27]–[29] have used navigation functions for low-level control with collision avoidance in applications, such as multiagent systems. Other results, such as [30]–[32] have considered collision avoidance in multiagent systems with limited sensing by using bounded avoidance functions in the controller which are only active when agents are within a defined sensing radius. The results in [9] and [28]–[32] do not consider optimal controllers, and in certain cases do not consider control constraints. Compared to such results which do not consider optimality, work such as [33] utilizes unbounded avoidance functions to explicitly compute optimal controllers for cooperative avoidance for multiagent systems. Moreover, results such as [34]–[36], develop sets of feasible states along with safe controllers using reachability methods such as [14] by developing differential games between two players. Moreover, results such as [37]–[41] approach collision avoidance problems through the use of collision cones in conjunction with other methods based on engagement geometry between two point objects. In such works, dynamically moving objects are modeled by quadric surfaces and collision conditions are derived for dynamic inversion-based avoidance strategies between agents. Despite the progress, the results in [33] rely on explicitly computed controllers, which are unknown when the optimal value function is unknown, and while results such as [37]–[40] establish a framework for providing collision cones, they are still combined with methods which may not necessarily be optimal, cf., [41]. However, although results such as [14] and [34]–[36] provide optimality guarantees, they rely on numerical techniques, which tend to be computationally intensive, and need to be resolved when conditions change.

Over the last several years, model predictive control (MPC) has gained attention for its capability to solve finite horizon optimal control problems in real-time (cf., [8], [42]–[45]). Moreover, MPC has been applied in a plethora of optimization problems; MPC is known for handling complex problems, such as of multiobjective problems, point-to-point trajectory generation problems, and collision avoidance (cf., [8], [42]–[45]). Specifically, works such as [8] consider multiobjective MPC frameworks for autonomous underwater vehicles with different prioritized objectives where the main objective is path convergence, while the secondary objective is different (i.e., speed assignment, which can be sacrificed at times in lieu of better performance on path convergence), or the objective is purely trajectory generation, such as [42] and [43], where the goal is point-to-point trajectory generation (i.e., offline multiagent trajectory generation or trajectory generation for constrained linearized agent models). Unlike, the aforementioned MPC results, results such as [44] and [45] take advantage of MPC's ability for fast optimization to combine it with other methods when considering collision avoidance problems. Although MPC has shown to be effective in motion/path planning and obstacle avoidance problems, the system dynamics are generally considered to be discretized and at each time-step, a finite horizon optimal control problem needs to be solved where a sequence of control inputs is generated. Even in the absence of obstacles, MPC methods generally do not yield an optimal policy over the complete trajectory since new solutions need to be recomputed at the end of each time horizon. Specifically, limited horizon methods, such as MPC, often require linear dynamics (cf., [42], [43]) or at least known dynamics (cf., [8], [42]–[45]). Since in practice, the environment and agents are prone to uncertainties, motivation exists to use parametric methods, such as neural-networks (NNs), to approximate optimal controllers online in continuous state nonlinear systems.

In recent years, approximate dynamic programming (ADP) has been successfully used in deterministic autonomous control-affine systems to solve optimal control problems [15]–[18], [46], [47]. By utilizing parametric approximation methods, ADP methods approximate the value function, which is the solution to the HJB and is used to compute the online forward-in-time optimal policy. Input constraints are considered in [19]–[21] by using a nonquadratic cost function [48] to yield a bounded approximate optimal controller.

For general nonlinear systems, generic basis functions, such as Gaussian radial basis functions, polynomials, or universal kernel functions are used to approximate the value function. One limitation of these generic approximation methods is that they only ensure an approximation over a compact neighborhood of the origin. Once outside the compact set, the approximation error tends to either grow or decay depending on the selected functions. Consequently, in the absence of domain knowledge, a large number of basis functions, and hence, a large number of unknown parameters, are required for value function approximation. A recent advancement in ADP utilizes computationally efficient state-following (StaF) kernel basis functions for local approximation of the value function around the current state, thereby reducing the number of basis functions required for sufficient value function approximation [22], [49]–[51]. The authors in [49] utilized the StaF approximation method to develop an approximate optimal online path planner with static obstacle avoidance. However, the development in [49] used a transitioning controller which switched between the approximate controller and a robust controller when the obstacles where sensed.

Inspired by advances in [22]–[26], [49], and [50], an approximate local optimal feedback-based motion planner is developed in this article that considers input and state constraints with mobile avoidance regions. The developed method differs from numerical approaches, such as [15]–[26], or MPC approaches, such as [42] and [43], because this article provides an online closed-loop feedback controller with computational efficiency provided by the local StaF approximation method. Moreover, the agent's trajectory is not computed offline, but instead the agent adjusts its trajectory online when it encounters an obstacle. Compared to works such as [9] and [28]–[32], which do not consider optimality, the controller designed in this article is based on an optimal control formulation that provides an

approximate optimal control solution. In addition, unlike [49] and other path planners, this method tackles the challenge of avoiding dynamic avoidance regions within the control strategy without switching between controllers. Since the StaF method uses local approximations, it does not require knowledge of uncertainties in the state space outside an approximation window. Local approximations of the StaF kernel method can be applied when an agent is approaching avoidance regions represented as $(n-1)$-spheres, not known *a priori*, in addition to state and system constraints. Because the avoidance regions become coupled with the agent in the HJB, their respective states must be incorporated when approximating the value function. Hence, a basis is given for each region which is zero outside of the sensing radius but is active when the avoidance region is sensed. In applications, such as station keeping of marine craft (e.g., [52]), knowledge of the weights for an avoidance region may provide useful information, as the approximation of the value function can be improved every time the region is encountered. To prevent collision, a penalizing term is added to the cost function which guarantees that the agent stays outside of the avoidance regions. A Lyapunov-based stability analysis is presented and guarantees uniformly ultimately bounded convergence while also ensuring that the agent remains outside of the avoidance regions. This work extends from the preliminary results in [53]. Unlike the preliminary work in [53], this article provides a unique value function representation and approximation, the actor update law is modified, and a more detailed stability analysis is included. The significance of this work over [53], is the mathematical development that considers an uncertain number of avoidance regions by transforming the autonomous value function approximation into a nonautonomous approximation. Because time does not lie on a compact set, it cannot be used in the StaF NNs, a transformation is performed so that a bounded signal of time is leveraged in the NNs. Moreover, experimental validations are presented to illustrate the performance of the developed path planning strategy.

Notation

In the following development, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote the sets of real n-vectors and $n\times m$ matrices, and $\mathbb{R}_{\geq a}$ and $\mathbb{R}_{>a}$ denote the sets of real numbers greater than or equal to a and strictly greater than a, respectively, where $a\in\mathbb{R}$. The $n\times n$ identity matrix, column vector of ones of dimension j, and the zeros matrix or dimension $m\times n$ are denoted by I_n, 1_j, and $0_{m\times n}$, respectively; hence, if $n=1$, $0_{m\times n}$ reduces to a vector of zeros. The partial derivative of k with respect to the state x is denoted by $\nabla k(x,y,\ldots)$, while the transpose of a matrix or vector is denoted by $(\cdot)^T$. For a vector $\xi\in\mathbb{R}^m$, the notation $\mathrm{Tanh}(\xi)\in\mathbb{R}^m$ and $\mathrm{sgn}(\xi)\in\mathbb{R}^m$ are defined as $\mathrm{Tanh}(\xi)\triangleq[\tanh(\xi_i),\ldots,\tanh(\xi_m)]^T$ and $\mathrm{sgn}(\xi)\triangleq[\mathrm{sgn}(\xi_i),\ldots,\mathrm{sgn}(\xi_m)]^T$, respectively, where $\tanh(\cdot)$ denotes the hyperbolic tangent function and $\mathrm{sgn}(\cdot)$ denotes the signum function. The notation $U[a,b]1_{n\times 1}$ denotes a n-dimensional vector selected from a uniform distribution on $[a,b]$, and $1_{n\times m}$ denotes a $n\times m$ matrix of ones.

II. Problem Formulation

Consider an autonomous agent with control-affine nonlinear dynamics given by

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

for all $t\in\mathbb{R}_{\geq t_0}$, where $x:\mathbb{R}_{\geq t_0}\to\mathbb{R}^n$ denotes the state, $f:\mathbb{R}^n\to\mathbb{R}^n$ denotes the drift dynamics, $g:\mathbb{R}^n\to\mathbb{R}^{n\times m}$ denotes the control effectiveness, $u:\mathbb{R}_{t\geq t_0}\to\mathbb{R}^m$ denotes the control input, and $t_0\in\mathbb{R}_{\geq 0}$ denotes the initial time. In addition, consider dynamic avoidance regions with nonlinear dynamics given by

$$\dot{z}_i(t)=h_i\left(z_i(t)\right) \tag{2}$$

for all $t\in\mathbb{R}_{\geq t_0}$, where $z_i:\mathbb{R}_{t\geq t_0}\to\mathbb{R}^n$ denotes the state of the center of the ith avoidance region and $h_i:\mathbb{R}^n\to\mathbb{R}^n$ denotes the drift dynamics for the ith zone in $\mathcal{M}\triangleq\{1,2,\ldots,M\}$, where $\mathcal{M}$ is the set of avoidance regions in the state space $\mathbb{R}^n$.[1] The dynamics in (2) are modeled as autonomous and isolated systems to facilitate the control problem formulation. The representation of the dynamics in (2) would require that complete knowledge of the dynamics over the entire operating domain are used. However, motivated by real systems where agents may only have local sensing, it is desired to only consider the zone inside a detection radius. Therefore, to alleviate the need for the HJB to require knowledge of the avoidance region dynamics outside of the agents' ability to sense the obstacles, the avoidance regions are represented as

$$\dot{z}_i(t)=\mathscr{F}_i\left(x(t),z_i(t)\right)h_i\left(z_i(t)\right) \tag{3}$$

for all $t\in\mathbb{R}_{\geq t_0}$. In (3), $\mathscr{F}_i:\mathbb{R}^n\times\mathbb{R}^n\to[0,1]$ is a smooth transition function that satisfies $\mathscr{F}_i(x,z_i)=0$ for $\|x-z_i\|>r_d$ and $\mathscr{F}_i(x,z_i)=1$ for $\|x-z_i\|\leq\bar{r}$, where $r_d\in\mathbb{R}_{>0}$ denotes the detection radius of the system in (1), and $\bar{r}\in(r_a,r_d)$ where $r_a\in\mathbb{R}_{>0}$ denotes the radius of the avoidance region. From the agent's perspective, the dynamics of the obstacles do not affect the agent outside of the sensing radius.

Remark 1: In application, a standard practice is to enforce a minimum avoidance radius to ensure safety [30], [31]. In addition, the detection radius r_d and avoidance radius r_s depend on the system parameters such as the maximum agent velocity limits.

Assumption 1: The number of dynamic avoidance regions M is known; however, the locations of the states of each region is unknown until it is within the sensing radius of the agent. Section VII presents an approach to alleviate Assumption 1.

Assumption 2: The drift dynamics f, h_i, and control effectiveness g are locally Lipschitz continuous, and g is bounded such that $0<\|g(x(t))\|\leq\overline{g}$ for all $x\in\mathbb{R}^n$ and all $t\in\mathbb{R}_{\geq t_0}$ where $\overline{g}\in\mathbb{R}_{>0}$. Furthermore, $f(0)=0$, and $\nabla f:\mathbb{R}^n\to\mathbb{R}^n\times\mathbb{R}^n$ is continuous.

Assumption 3: The equilibrium points z_i^e for the obstacles given by the dynamics in (3) lie outside of a ball of radius r_d centered at the origin. That is, the origin is sufficiently clear of obstacles. Furthermore, obstacles do not trap the agent, meaning

[1] The terms avoidance regions and obstacles are used interchangeably.

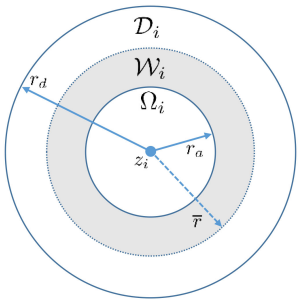

Fig. 1. Augmented regions around each avoidance region.

the obstacles do not completely barricade the agent in the sense that the agent has a free, unblocked, path to the goal location. Moreover, the agent is assumed to be sufficiently agile to be able to outmaneuver the moving obstacles. Specifically, the obstacle velocities must be appropriately equal or less than the agent for the agent to have capability to avoid the obstacle in general.

Remark 2: Assumption 3 limits pathological scenarios where obstacle avoidance is not possible. Specifically, scenarios may arise where obstacles move faster than the agent. In such scenarios, it may be infeasible for agents using this method, or other existing approaches, to avoid the obstacle without colliding. However, given an upper bound on the obstacles velocities, the sensing radius can be sized large enough for the agent to respond accordingly.

Remark 3: To facilitate the development, let $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ denote a distance metric defined as $d(v, w) \triangleq \|v - w\|$ for $v, w \in \mathbb{R}^n$. Moreover, the centers of the avoidance regions, shown in Fig. 1, are augmented with the following.[2]

1) The total detection set is defined as $\mathcal{D} = \cup_{i\in\mathcal{M}}\mathcal{D}_i$, where
$$\mathcal{D}_i = \{x \in \mathbb{R}^n \mid d(x, z_i) \le r_d\}.$$
2) The total conflict set is defined as $\mathcal{W} = \cup_{i\in\mathcal{M}}\mathcal{W}_i$, where
$$\mathcal{W}_i = \{x \in \mathbb{R}^n \mid r_a < d(x, z_i) \le \bar{r}\}.$$
3) The total avoidance set is $\Omega = \cup_{i\in\mathcal{M}}\Omega_i$, where each local avoidance region is
$$\Omega_i = \{x \in \mathbb{R}^n \mid d(x, z_i) \le r_a\}.$$

Furthermore, the avoidance region and agent dynamics can be combined to form the following system:

$$\dot{\zeta}(t) = F(\zeta(t)) + G(\zeta(t))\, u(t) \tag{4}$$

for all $t \in \mathbb{R}_{\ge t_0}$, where $\zeta = [x^T, z_1^T, \ldots, z_M^T]^T \in \mathbb{R}^{\mathcal{N}}$, $\mathcal{N} = (M+1)n$ and

$$F(\zeta) = \begin{bmatrix} f(x) \\ \mathscr{F}_1(x, z_1)\, h_1(z_1) \\ \vdots \\ \mathscr{F}_M(x, z_M)\, h_M(z_M) \end{bmatrix} \qquad G(\zeta) = \begin{bmatrix} g(x) \\ 0_{Mn\times m} \end{bmatrix}.$$

[2]The size of the regions also depends on the dynamics of the obstacles.

The goal is to simultaneously design and implement a controller u which minimizes the cost function

$$J(\zeta, u) \triangleq \int_{t_0}^{\infty} r(\zeta(\tau), u(\tau))\, d\tau \tag{5}$$

subject to (4) while obeying $\sup_t(u_i) \le \mu_{\text{sat}}\ \forall i = 1, \ldots, m$, where $\mu_{\text{sat}} \in \mathbb{R}_{>0}$ is the control effort saturation limit. In (5), $r : \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^m \to [0, \infty]$ is the instantaneous cost defined as

$$r(\zeta, u) = Q_x(x) + \sum_{i=1}^{M} s_i(x, z_i)\, Q_z(z_i) + \Psi(u) + P(\zeta) \tag{6}$$

where Q_x, $Q_z : \mathbb{R}^n \to \mathbb{R}_{\ge 0}$ are user-defined positive definite functions that penalize the agent and obstacle states. The $Q_z(z_i)$ term in (6) only influences the cost when the obstacles are sensed. The smooth scheduling function $s_i : \mathbb{R}^n \times \mathbb{R}^n \to [0, 1]$ that allows the avoidance region states in the detection radius to be penalized, satisfies $s_i = 0$ for $\|x - z_i\| > r_d$ and $s_i = 1$ for $\|x - z_i\| \le \bar{r}$. In (6), $\Psi : \mathbb{R}^m \to \mathbb{R}$ is a positive definite function penalizing the control input u, defined as

$$\Psi(u) \triangleq 2\sum_{i=1}^{m}\left[\int_0^{u_i}\left(\mu_{\text{sat}} r_i \tanh^{-1}\left(\frac{\xi_{u_i}}{\mu_{\text{sat}}}\right)\right) d\xi_{u_i}\right] \tag{7}$$

where u_i is the ith element of the control u, ξ_{u_i} is an integration variable, and r_i is the diagonal elements which make up the symmetric positive definite weighting matrix $R \in \mathbb{R}^{m\times m}$ where $R \triangleq \text{diag}\{\overline{R}\}$, and $\overline{R} \triangleq [r_1, \ldots, r_m] \in \mathbb{R}^{1\times m}$ [19], [21], [48]. The selection of the input penalizing function in (7) is motivated such that a bounded form of control policy can be derived from the HJB [48]. Moreover, $\tanh(\cdot)$ is used in (7) because it is a continuous one-to-one real-analytic function, $\tanh(0_m) = 0_m$, and $\tanh^{-1}(\frac{\xi_{u_i}}{\mu_{\text{sat}}})$ is monotonically increasing. The function $P : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}$ in (6), called the avoidance penalty function, is a positive semidefinite compactly supported function defined as

$$P(\zeta) \triangleq \sum_{i=1}^{M}\left(\min\left\{0, \frac{d(x, z_i)^2 - r_d^2}{\left(d(x, z_i)^2 - r_a^2\right)^2}\right\}\right)^2. \tag{8}$$

Remark 4: The avoidance penalty function in (8) is zero outside of the compact set $\mathcal{D}$, and yields an infinite penalty when $\|x - z_i\| = r_a$ for any $i \in \mathcal{M}$. Other penalty/avoidance functions can be used; see [33] for a generalization of avoidance functions. The avoidance penalty function in (8) modifies the one found in [33], which studies a generalization of avoidance penalty functions. Since the term in the denominator has quartic growth compared to only quadratic growth, the function in (8) is scaled differently compared to the one found in [33]. Other growth factors can also be used which affect the rate at which the agent penalizes the avoidance regions once it detects them.

Assumption 4: There exist constants $\underline{q}_x$, $\overline{q}_x$, $\underline{q}_z$, $\overline{q}_z \in \mathbb{R}_{>0}$ such that $\underline{q}_x\|x\|^2 \le Q_x(x) \le \overline{q}_x\|x\|^2$ for all $x \in \mathbb{R}^n$, and $\underline{q}_z\|z_i\|^2 \le Q_z(z_i) \le \overline{q}_z\|z_i\|^2$ for all $z_i \in \mathbb{R}^n$ and $i \in \mathcal{M}$.

The infinite-horizon scalar value function for the optimal value function, denoted by $V^*:\mathbb{R}^{\mathcal{N}}\to\mathbb{R}_{\geq 0}$, is expressed as

$$V^*(\zeta)=\min_{u(\tau)\in U|\tau\in\mathbb{R}_{\geq t}}\int_t^{\infty} r\left(\zeta(\tau),u(\tau)\right)d\tau \tag{9}$$

where $U\subset\mathbb{R}^m$ denotes the set of admissible inputs. For the stationary solution, the HJB equation, which characterizes the optimal value function is given by

$$\begin{aligned}0&=\frac{\partial V^*(\zeta)}{\partial\zeta}\left(F(\zeta)+G(\zeta)u^*(\zeta)\right)+r\left(\zeta,u^*(\zeta)\right)\\&=\frac{\partial V^*(\zeta)}{\partial x}\left(f(x)+g(x)u^*(\zeta)\right)\\&\quad+\sum_{i=1}^{M}\frac{\partial V^*(\zeta)}{\partial z_i}\left(\mathscr{F}_i(x,z_i)h_i(z_i)\right)+r\left(\zeta,u^*(\zeta)\right)\end{aligned} \tag{10}$$

with the condition $V^*(0)=0$, where $u^*:\mathbb{R}^{\mathcal{N}}\to\mathbb{R}^m$ is the optimal control policy. Taking the partial derivative of (10) with respect to $u^*(\zeta)$, setting it to zero (i.e., $u^*(\zeta)$ is the minimizing argument) and solving for $u^*(\zeta)$ results in

$$u^*(\zeta)=-\mu_{\text{sat}}\text{Tanh}\left(\frac{R^{-1}G(\zeta)^T}{2\mu_{\text{sat}}}\left(\nabla V^*(\zeta)\right)^T\right). \tag{11}$$

The HJB in (10) uses both the agent and avoidance region dynamics.[3] However, because each avoidance region is modeled as in (3), the terms that include them are zero when the regions are not detected; hence, they do not affect the HJB. Furthermore, the analytical expression in (11) requires knowledge of the optimal value function. However, the analytical solution for the HJB, i.e., the value function, is not feasible to compute in general cases. Therefore, an approximation is sought using a neural network approach.

III. Value Function Approximation

Recent developments in ADP have resulted in computationally efficient StaF kernels to approximate the value function [22]. To facilitate the development let $\chi\subset\mathbb{R}^{\mathcal{N}}$ be a compact set, with x and all z_i in the interior of χ. Based on the StaF method in [22] and [50], after adding and subtracting a bounded avoidance function $P_a(\zeta)$, the optimal value function and controller can be approximated as

$$V^*(y)=P_a(y)+W(y)^T\sigma\left(y,c(\zeta)\right)+\epsilon(\zeta,y) \tag{12}$$

$$\begin{aligned}u^*(y)=&-\mu_{\text{sat}}\,\text{Tanh}\left(\frac{R^{-1}G(y)^T}{2\mu_{\text{sat}}}\right.\\&\times\left(\nabla P_a(y)^T+\nabla\sigma\left(y,c(\zeta)\right)^T W(\zeta)\right.\\&\left.\left.+\nabla W(\zeta)^T\sigma\left(y,c(\zeta)\right)+\nabla\epsilon\left(y,\zeta\right)^T\right)\right)\end{aligned} \tag{13}$$

where $c(\zeta)\in(\overline{B_r(\zeta)})^L$ are centers around the current concatenated state ζ, $L\in Z_{>0}$ is the number of centers, and $y\in\overline{B_r(\zeta)}$ where $\overline{B_r(\zeta)}$ is a small compact set around the current state $\zeta\in\chi$. In (12), $W:\chi\to\mathbb{R}^L$ is the continuously differentiable ideal StaF weight function that changes with the state dependent centers, $\epsilon:\chi\to\mathbb{R}$ is the continuously differentiable bounded function reconstruction error, and $\sigma:\chi\to\mathbb{R}^L$ is a concatenated vector of StaF basis functions such that

$$\sigma\left(\zeta,c(\zeta)\right)=\begin{bmatrix}\sigma_0\left(x,c_0(x)\right)\\s_1\left(x,z_1\right)\sigma_1\left(z_1,c_1\left(z_1\right)\right)\\\vdots\\s_M\left(x,z_M\right)\sigma_M\left(z_M,c_M\left(z_M\right)\right)\end{bmatrix} \tag{14}$$

where $\sigma_0(x,c_0(x)):\mathbb{R}^n\to\mathbb{R}^{P_x}$ and $\sigma_i(z_i,c_i(z_i)):\mathbb{R}^n\to\mathbb{R}^{P_{z_i}}$ for $i\in\mathcal{M}$ are strictly positive definite, continuously differentiable StaF kernel function vectors, $c_i:\mathbb{R}^n\to\mathbb{R}^n$ for $i\in\{0,1,\ldots,M\}$ are state-dependent centers, and the dimension of the concatenated vector of StaF basis functions σ is $L=P_x+\sum_{i=1}^{M}P_{z_i}$. The formation of the vector of basis functions in (14) allows for certain weights of the approximation to be constant when the agent and no-entry zones are not in the detection regions. This formulation introduces a sparse-like approach because the basis functions that correlate to the no-entry zones are off due to the scheduling function s_i, when they are outside of the detection regions. Hence, approximation of the value function is only influenced by the no-entry zones when they are in the detection regions $\mathcal{D}_i$. However, the optimal value function and controller are not known in general; therefore, approximations $\hat{V}:\mathbb{R}^{\mathcal{N}}\times\mathbb{R}^{\mathcal{N}}\times\mathbb{R}^L\to\mathbb{R}$ and $\hat{u}:\mathbb{R}^{\mathcal{N}}\times\mathbb{R}^{\mathcal{N}}\times\mathbb{R}^L\to\mathbb{R}^m$ are used where

$$\hat{V}\big(y,\zeta,\hat{W}_c\big)\triangleq P_a(y)+\hat{W}_c^T\sigma\left(y,c(\zeta)\right) \tag{15}$$

$$\begin{aligned}\hat{u}\big(y,\zeta,\hat{W}_a\big)\triangleq&-\mu_{\text{sat}}\,\text{Tanh}\left(\frac{R^{-1}G(y)^T}{2\mu_{\text{sat}}}\right.\\&\left.\times\left(\nabla\sigma\left(y,c(\zeta)\right)^T\hat{W}_a+\nabla P_a^T(y)\right)\right).\end{aligned} \tag{16}$$

In (15) and (16), $\hat{V}$ and $\hat{u}$ are evaluated at a point $y\in B_r(\zeta)$ using StaF kernels centered at ζ, while $\hat{W}_c$, $\hat{W}_a\in\mathbb{R}^L$ are the weight estimates for the ideal weight vector W. In actor-critic architectures, the estimates $\hat{V}$ and $\hat{u}$ replace the optimal value function V^* and optimal policy u^* in (10) to form a residual error $\delta:\mathbb{R}^{\mathcal{N}}\times\mathbb{R}^{\mathcal{N}}\times\mathbb{R}^L\times\mathbb{R}^L\to\mathbb{R}$ known as the Bellman error (BE), which is defined as

$$\begin{aligned}\delta\big(y,\zeta,\hat{W}_c,\hat{W}_a\big)\triangleq&\nabla\hat{V}\big(y,\zeta,\hat{W}_c\big)\Big(F(y)\\&+G(y)\hat{u}\big(y,\zeta,\hat{W}_a\big)\Big)+r\left(y,\hat{u}\big(y,\zeta,\hat{W}_a\big)\right).\end{aligned} \tag{17}$$

The aim of the actor and critic is to find a set of weights which minimize the BE for all $\zeta\in\mathbb{R}^{\mathcal{N}}$.

Remark 5: Unlike the function P, which is not finite when $\|x-z_i\|=r_a$, for any $i\in\mathcal{M}$, the function P_a satisfies $P_a=0$ when x, $z_i\notin\mathcal{D}_i$ for each $i\in\mathcal{M}$, and for all $0\leq P(\zeta)\leq\overline{P}_a$, and $\|\nabla P_a(\zeta)\|\leq\overline{\|\nabla P_a\|}$ for all $\zeta\in\mathbb{R}^{\mathcal{N}}$. An example of $P_a(\zeta)$ includes $P_a(\zeta)\triangleq\sum_{i=1}^{M}P_{a,i}(x,z_i)$ where $P_{a,i}\triangleq(\min\{0,\frac{\|x-z_i\|^2-r_d^2}{(\|x-z_i\|^2-r_a^2)^2+r_\varepsilon}\})^2$ for $r_\varepsilon\in\mathbb{R}_{>0}$, or see [30]–[32] for other examples of bounded avoidance functions.

[3]The following Lyapunov-based stability analysis indicates that the states $\zeta(t)$ remain outside of Ω, i.e. $\zeta(t)\notin\Omega$. Hence, the gradient is never taken over the discontinuity.

IV. Online Learning

To implement the approximations online, at a given time instance t, the BE $\delta_t : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is evaluated as

$$\delta_t(t) \triangleq \delta\big(\zeta(t), \zeta(t), \hat{W}_c(t), \hat{W}_a(t)\big) \tag{18}$$

where ζ denotes the state of the system in (4) starting at initial time t_0 with initial condition ζ_0, while $\hat{W}_c(t)$ and $\hat{W}_a(t)$ denote the critic weight and actor weight estimates at time t, respectively. The controller which influences the state $x(t) \subset \zeta(t)$ is

$$u(t) = \hat{u}\big(\zeta(t), \zeta(t), \hat{W}_a(t)\big). \tag{19}$$

Simulation of experience is used to learn online by extrapolating the BE to unexplored areas of the state space [22], [23]. Off-policy trajectories $\{x_k : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n\}_{k=1}^N$ are selected by the critic such that each x_k maps the current state $x(t)$ to a point $x_k(x(t), t) \in B_r(x(t))$. The extrapolated BE $\delta_k : \mathbb{R}_{\geq 0} \to \mathbb{R}$ for each ζ_k takes the form

$$\delta_k(t) = \hat{W}_c^T(t)\omega_k(t) + \omega_{Pk}(t) + r\left(\zeta_k(t), \hat{u}_k(t)\right) \tag{20}$$

where $\zeta_k = [\, x_k^T,\ Z(t)\,]^T$

$$\omega_{Pk}(t) \triangleq \nabla P_a\left(\zeta_k(t)\right)\Big(F\left(\zeta_k(t)\right) + G\left(\zeta_k(t)\right)\hat{u}\Big(\zeta_k(t), \zeta(t), \hat{W}_a(t)\Big)\Big)$$

$$\omega_k(t) \triangleq \nabla\sigma\left(\zeta_k(t), c\left(\zeta(t)\right)\right)\Big(F\left(\zeta_k(t)\right) + G\left(\zeta_k(t)\right)\hat{u}\Big(\zeta_k(t), \zeta(t), \hat{W}_a(t)\Big)\Big)$$

and the extrapolated policies are

$$\hat{u}_k(t) \triangleq -\mu_{\text{sat}}\,\text{Tanh}\left(\frac{R^{-1}G\left(\zeta_k(t)\right)}{2\mu_{\text{sat}}} \times \Big(\nabla\sigma\big(\zeta_k(t), c\left(\zeta(t)\right)\big)^T\hat{W}_a(t) + \nabla P_a^T\left(\zeta_k(t)\right)\Big)\right). \tag{21}$$

The concurrent learning-based least squares update laws are designed as

$$\dot{\hat{W}}_c(t) = -\Gamma(t)\left(\frac{k_{c1}\omega(t)}{\rho(t)}\delta(t) + \frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{\omega_k(t)}{\rho_k(t)}\delta_k(t)\right) \tag{22}$$

$$\dot{\Gamma}(t) = \beta\Gamma(t) - k_{c1}\Gamma(t)\frac{\omega(t)\omega^T(t)}{\rho^2(t)}\Gamma(t) - \frac{k_{c2}}{N}\Gamma(t)\sum_{k=1}^{N}\frac{\omega_k(t)\omega_k^T(t)}{\rho_k^2(t)}\Gamma(t), \quad \Gamma\left(t_0\right) = \Gamma_0. \tag{23}$$

Furthermore, in (22) and (23) $\rho(t) \triangleq 1 + \gamma_1\omega(t)^T\omega(t)$, $\rho_k(t) \triangleq 1 + \gamma_1\omega_k(t)^T\omega_k(t)$ are normalizing factors, $k_{c1}, k_{c2}, \gamma_1 \in \mathbb{R}_{>0}$ are adaptation gains, $\beta \in \mathbb{R}_{>0}$ is a forgetting factor, and

$$\omega(t) \triangleq \nabla\sigma\big(\zeta(t), c\left(\zeta(t)\right)\big)\Big(F\left(\zeta(t)\right) + G\left(\zeta(t)\right)\hat{u}\big(\zeta(t), \zeta(t), \hat{W}_a(t)\big)\Big).$$

The policy weights are updated to follow the critic weights using the actor update law designed as

$$\dot{\hat{W}}_a(t) = -\Gamma_a\Bigg(k_{a1}\Big(\hat{W}_a(t) - \hat{W}_c(t)\Big) + k_{a2}\hat{W}_a(t) + k_{c1}G_{a1}(t)\frac{\omega^T(t)}{\rho(t)}\hat{W}_c(t) + \frac{k_{c2}}{N}\sum_{k=1}^{N}G_{a1,k}(t)\frac{\omega_k^T(t)}{\rho_k(t)}\hat{W}_c(t)\Bigg) \tag{24}$$

where $k_{a1}, k_{a2} \in \mathbb{R}_{>0}$ are adaptation gains, $\Gamma_a \in \mathbb{R}^{L\times L}$ is a positive definite constant matrix, and

$$G_{a1}(t) \triangleq \mu_{\text{sat}}\nabla\sigma\left(\zeta(t), c\left(\zeta(t)\right)\right)G\left(\zeta(t)\right) \times \left(\text{Tanh}\left(\frac{1}{k_u}\hat{\bar{D}}(t)\right) - \text{Tanh}\left(\frac{R^{-1}}{2\mu_{\text{sat}}}\hat{\bar{D}}(t)\right)\right)$$

$$G_{a1,k}(t) \triangleq \mu_{\text{sat}}\nabla\sigma\left(\zeta_k(t), c\left(\zeta(t)\right)\right)G(\zeta_k(t)) \times \left(\text{Tanh}\left(\frac{1}{k_u}\hat{\bar{D}}_k(t)\right) - \text{Tanh}\left(\frac{R^{-1}}{2\mu_{\text{sat}}}\hat{\bar{D}}_k(t)\right)\right)$$

where $k_u \in \mathbb{R}_{>0}$ is a constant, $\hat{\bar{D}}(t) \triangleq G^T(\zeta(t))(\nabla\sigma^T(\zeta(t), c(\zeta(t)))\hat{W}_a(t) + \nabla P_a^T(\zeta(t)))$, and $\hat{\bar{D}}_k(t) \triangleq G^T(\zeta_k(t))(\nabla\sigma^T(\zeta_k(t), c(\zeta(t)))\hat{W}_a(t) + \nabla P_a^T(\zeta_k(t)))$. Similar to the preliminary work in [53], a projection-based update law for the actor weight estimates can be used to simplify the stability analysis. In such a case, (24) would become $\dot{\hat{W}}_a(t) = \text{proj}\{-\Gamma_a k_{a1}(\hat{W}_a(t) - \hat{W}_c(t))\}$, where proj$\{\cdot\}$ denotes a smooth projection operator which bounds the weight estimates, see [54, Ch. 4] for details of the projection operator.

Remark 6: Rather than extrapolating the entire state vector of the system, as designed in [22], [23], and [51], only the controlled states, i.e., the agent's states, are extrapolated to perform simulation of experience. Compared to experience replay results such as [21], which record a history stack of prior input–output pairs, the simulation of experience approach in this result only uses extrapolated states within a time-varying neighborhood of the current agent state. This is motivated by the StaF approximation method, which only provides a sufficient approximation of the value function a neighborhood of the current agent state.

V. Stability Analysis

For notational brevity, time dependence of functions are henceforth suppressed. Define $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$ as the weight estimation errors, and let $\overline{\|(\cdot)\|} \triangleq \sup_{\pi\in B_\xi}\|(\cdot)\|$, where $B_\xi \subset \chi \times \mathbb{R}^L \times \mathbb{R}^L$ is a compact set. Then, the

BEs in (18) and (20) can be expressed as

$$\delta_t = -\omega^T \tilde{W}_c + G_{a1}^T \tilde{W}_a + G_{a2}^T \tilde{W}_a + \Delta(\zeta)$$
$$\delta_k = -\omega_k^T \tilde{W}_c + G_{a1,k}^T \tilde{W}_a + G_{a2,k}^T \tilde{W}_a + \Delta_k(\zeta).$$

The terms G_{a2} and G_{a2k} are defined as $G_{a2} \triangleq \mu_{\text{sat}} \nabla\sigma G (\text{sgn}(\hat{\tilde{D}}) - \text{Tanh}(\frac{1}{k_u}\hat{\tilde{D}}))$ and $G_{a2,k} \triangleq \mu_{\text{sat}} \nabla\sigma_k G_k (\text{sgn}(\hat{\tilde{D}}_k) - \text{Tanh}(\frac{1}{k_u}\hat{\tilde{D}}_k))$. The functions $\Delta,\ \Delta_k : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}$ are uniformly bounded over χ such that the residual bounds $\overline{\|\Delta\|}$, $\overline{\|\Delta_k\|}$ decrease with decreasing $\overline{\|\nabla W\|}$ and $\overline{\|\nabla\epsilon\|}$.[4]

To facilitate the analysis, the system states x and selected states x_k are assumed to satisfy the following inequalities.

Assumption 5: There exists constants $T \in \mathbb{R}_{>0}$ and $\underline{c}_1, \underline{c}_2, \underline{c}_3 \in \mathbb{R}_{\geq 0}$, such that

$$\underline{c}_1 I_L \leq \frac{1}{N}\sum_{k=1}^{N} \frac{\omega_k(t)\omega_k^T(t)}{\rho_k^2(t)}$$
$$\underline{c}_2 I_L \leq \int_t^{t+T} \left(\frac{1}{N}\sum_{k=1}^{N} \frac{\omega_k(\tau)\omega_k^T(\tau)}{\rho_k^2(\tau)}\right) d\tau \qquad \forall t \in \mathbb{R}_{\geq t_0}$$
$$\underline{c}_3 I_L \leq \int_t^{t+T} \left(\frac{\omega(\tau)\omega^T(\tau)}{\rho^2(\tau)}\right) d\tau \qquad \forall t \in \mathbb{R}_{\geq t_0}$$

where at least one of the constants $\underline{c}_1$, $\underline{c}_2$, or $\underline{c}_3$ is strictly positive [22].

In general, $\underline{c}_1$ can be made strictly positive by sampling redundant data, i.e, choosing $N \gg L$, and $\underline{c}_2$ can be made strictly positive by sampling extrapolated trajectories at a high frequency. Generally, $\underline{c}_3$ is strictly positive provided the system is persistently excited (PE), which is a strong assumption that cannot be verified online. Since only one constant has to be strictly positive, ω_k can be selected such that $\underline{c}_1 > 0$ or $\underline{c}_2 > 0$, since ω_k is a design variable. Unlike the strong PE given by the third inequality in Assumption 5, the first two inequalities can be verified online.

Remark 7: Instead of injecting potentially destabilizing dither signals into the physical system to satisfy the PE condition, virtual excitation can be obtained by using the sample states. Specifically, the sample states $x_k(t)$ can be selected from a sampling distribution, such as a normal or uniform distribution, or they can be selected to follow a highly oscillatory trajectory.

Lemma 1: Provided Assumption 5 is satisfied and $\lambda_{\min}\{\Gamma_0^{-1}\} > 0$, the update law in (23) ensures that the least squares gain matrix Γ satisfies

$$\underline{\Gamma} I_L \leq \Gamma(t) \leq \overline{\Gamma} I_L \tag{25}$$

where the bounds $\underline{\Gamma}$ and $\overline{\Gamma}$ are defined as

$$\underline{\Gamma} = \frac{1}{\left(\lambda_{\max}\left\{\Gamma_0^{-1}\right\} + \frac{k_{c1}+k_{c2}}{4\gamma_1\beta}\right)}$$
$$\overline{\Gamma} = \frac{1}{\min\left\{\left(k_{c1}\underline{c}_3 + k_{c2}\max\left\{\underline{c}_1 T, \underline{c}_2\right\}\right), \lambda_{\min}\left\{\Gamma_0^{-1}\right\}\right\} e^{-\beta T}}$$

[4]For an arbitrary function ϕ, ϕ_k is defined as $\phi_k \triangleq \phi(\zeta_k(t))$.

where $\lambda_{\min}\{\cdot\}$, $\lambda_{\max}\{\cdot\}$ denote the minimum and maximum eigenvalues, respectively [22].

To facilitate the analysis, consider a candidate Lyapunov function $V_L : \mathbb{R}^{\mathcal{N}+2L} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ given by

$$V_L(Y,t) = V^*(\zeta) + \frac{1}{2}\tilde{W}_c^T \Gamma^{-1}(t)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T \Gamma_a^{-1}\tilde{W}_a + \frac{1}{2}\sum_{i=1}^{M} z_i^T z_i \tag{26}$$

where V^* is the optimal value function, and $Y = [\zeta^T, \tilde{W}_c^T, \tilde{W}_a^T]^T$. Since the optimal value function is positive definite, using (25) and [55, Lemma 4.3], (26) can be bounded as

$$\underline{\nu}_l(\|Y\|) \leq V(Y,t) \leq \overline{\nu}_l(\|Y\|) \tag{27}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and for all $Y \in \mathbb{R}^{n+1+2^{\hat{}}L}$, where $\underline{\nu}_l, \overline{\nu}_l : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are class $\mathcal{K}$ functions. To facilitate the following analysis, let $\nu_l : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a class $\mathcal{K}$ function such that

$$\nu_l(\|Y\|) \leq \frac{q}{2}\|x\|^2 + \frac{\underline{q}_z}{4}\sum_{i=1}^{M} s_i(x, z_i)\|z_i\|^2 + \left(\frac{k_{a1}+k_{a2}}{8}\right)\left\|\tilde{W}_a\right\|^2 + \frac{k_{c2}\underline{c}}{8}\left\|\tilde{W}_c\right\|^2 \tag{28}$$

and let $\underline{c} \in \mathbb{R}_{>0}$ be a constant defined as

$$\underline{c} \triangleq \frac{\beta}{2k_{c2}\overline{\Gamma}} + \frac{\underline{c}_1}{2}. \tag{29}$$

The sufficient conditions for the subsequent analysis are given by

$$\frac{k_{a1}+k_{a2}}{2} \geq \max\left\{\varphi_{ac}, \frac{\overline{\|\nabla W\|}G_R}{\lambda_{\min}\{\Gamma_a\}}\overline{\|\nabla\sigma^T\|}\right\} \tag{30}$$
$$k_{c2}\underline{c} \geq \varphi_{ac} \tag{31}$$
$$\frac{1}{2}\underline{q}_z \geq L_z \tag{32}$$
$$\nu_\iota^{-1}(\iota) < \overline{\nu}_\iota^{-1}(\underline{\nu}_\iota(\xi)) \tag{33}$$

where L_z is the Lipschitz constant such that $\|h_i(z_i)\| \leq L_z\|z_i\|$ satisfying assumption (2) and $\varphi_{ac} \in \mathbb{R}_{>0}$ is defined in the appendix.

Theorem 1: Consider the augmented dynamic system (4) and the dynamic systems in (1) and (3). Provided Assumptions 1–5 are satisfied along with the sufficient conditions in (30)–(33), then system state $\zeta(t)$, input $u(t)$, and weight approximation errors $\tilde{W}_a$ and $\tilde{W}_c$ are Uniformly Ultimately Bounded (UUB); furthermore, states $\zeta(t)$ starting outside of Ω remain outside of Ω.

Proof: Consider the Lyapunov function candidate in (26). The time derivative is given by

$$\dot{V}_L = \dot{V}^* + \tilde{W}_c^T\Gamma^{-1}\left(\dot{W} - \dot{\hat{W}}_c\right) + \tilde{W}_a^T\Gamma_a^{-1}\left(\dot{W} - \dot{\hat{W}}_a\right) - \frac{1}{2}\tilde{W}_c^T\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}\tilde{W}_c + \sum_{i=1}^{M} z_i^T\left(\mathscr{F}_i h_i\right).$$

Using the chain rule, the time derivative of the ideal weights $\dot{W}$ can be expressed as

$$\dot{W} = \nabla W \left(F + Gu \right). \tag{34}$$

Substituting in (22)–(24) with (34) yields

$$\begin{aligned}
\dot{V}_L = {} & \nabla V^* F + \nabla V^* G u + \sum_{i=1}^{M} z_i^T \left(\mathscr{F}_i h_i \right) \\
& + \tilde{W}_c^T \Gamma^{-1} \left(k_{c1} \Gamma \frac{\omega}{\rho} \delta_t + \frac{k_{c2}}{N} \Gamma \sum_{k=1}^{N} \frac{\omega_k}{\rho_k} \delta_k \right) \\
& + \tilde{W}_a^T k_{a1} \left(\hat{W}_a - \hat{W}_c \right) + \tilde{W}_a^T k_{a2} \hat{W}_a(t) \\
& + \tilde{W}_a^T \left(k_{c1} G_{a1} \frac{\omega^T}{\rho} - \frac{k_{c2}}{N} \sum_{k=1}^{N} G_{a1,k} \frac{\omega_k^T}{\rho_{ik}} \right) \hat{W}_c(t) \\
& + \left(\tilde{W}_c^T \Gamma^{-1} + \tilde{W}_a^T \right) \nabla W \left(F + Gu \right) - \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} \\
& \times \left(\beta \Gamma - k_{c1} \Gamma \frac{\omega \omega^T}{\rho^2} \Gamma - \frac{k_{c2}}{N} \Gamma \sum_{k=1}^{N} \frac{\omega_k \omega_k^T}{\rho_k^2} \Gamma \right) \Gamma^{-1} \tilde{W}_c.
\end{aligned}$$

Using (6) with (10), (18)–(21), Young's inequality, and Lemma 1, the Lyapunov derivative can be bounded as

$$\begin{aligned}
\dot{V}_L \leq {} & - \underline{q}_x \left\| x \right\|^2 - \frac{\underline{q}_z}{2} \sum_{i=1}^{M} s_i \left(x, z_i \right) \left\| z_i \right\|^2 \\
& - 2 \left(\frac{k_{a1} + k_{a2}}{8} \right) \left\| \tilde{W}_a \right\|^2 - 2 \left(\frac{k_{c2} \underline{c}}{8} \right) \left\| \tilde{W}_c \right\|^2 \\
& - \begin{bmatrix} \left\| \tilde{W}_c \right\| & \left\| \tilde{W}_a \right\| \end{bmatrix} \begin{bmatrix} \frac{k_{c2} \underline{c}}{2} & -\frac{\varphi_{ac}}{2} \\ -\frac{\varphi_{ac}}{2} & \frac{k_{a1} + k_{a2}}{4} \end{bmatrix} \begin{bmatrix} \left\| \tilde{W}_c \right\| \\ \left\| \tilde{W}_a \right\| \end{bmatrix} \\
& - \frac{\underline{q}_z}{2} \sum_{i=1}^{M} s_i \left(x, z_i \right) \left\| z_i \right\|^2 + \sum_{i=1}^{M} z_i^T \left(\mathscr{F}_i h_i \right) + \iota
\end{aligned}$$

where $\iota \in \mathbb{R}_{>0}$ is the positive constant defined in the appendix. Using (28), (30), and (31), the Lyapunov derivative reduces to

$$\begin{aligned}
\dot{V}_L \leq {} & - \nu_l \left(\left\| Y \right\| \right) - \left(\nu_l \left(\left\| Y \right\| \right) - \iota \right) \\
& - \frac{\underline{q}_z}{2} \sum_{i=1}^{M} s_i \left(x, z_i \right) \left\| z_i \right\|^2 + \sum_{i=1}^{M} z_i^T \left(\mathscr{F}_i h_i \right).
\end{aligned}$$

For the case when $x, z_i \notin \mathcal{D} \ \ \forall i \in \mathcal{M}$, the avoidance region dynamics in (3) can be used conclude that, $\frac{\underline{q}_z}{2} \sum_{i=1}^{M} s_i(x, z_i) \|z_i\|^2 + \sum_{i=1}^{M} z_i^T (\mathscr{F}_i h_i) = 0$; therefore

$$\dot{V}_L \leq -\nu_l \left(\left\| Y \right\| \right) - \left(\nu_l \left(\left\| Y \right\| \right) - \iota \right).$$

Provided the sufficient conditions in (30), (31), and (33) are met, then

$$\dot{V}_L \leq -\nu_l \left(\left\| Y \right\| \right), \quad \forall Y \in \chi \, \forall \left\| Y \right\| \geq \nu_l^{-1} \left(\iota \right).$$

For the case when $\zeta \in \mathcal{W}$, Assumption 2 is used to conclude that

$$\begin{aligned}
\dot{V}_L \leq {} & - \nu_l \left(\left\| Y \right\| \right) - \left(\nu_l \left(\left\| Y \right\| \right) - \iota \right) \\
& - \frac{\underline{q}_z}{2} \sum_{i=1}^{M} s_i \left(x, z_i \right) \left\| z_i \right\|^2 + \sum_{i \in \mathcal{M}} L_z \left\| z_i \right\|^2.
\end{aligned}$$

Using the fact that $\inf_{x, z_i \in \mathcal{W}_i} s_i(x, z_i) = 1$ for any $i \in \mathcal{M}$, and provided the sufficient conditions in (30)–(33) hold,

$$\dot{V}_L \leq -\nu_l \left(\left\| Y \right\| \right) \quad \forall \left\| Y \right\| \geq \nu_l^{-1} \left(\iota \right). \tag{35}$$

Hence, (26) is nonincreasing.

If $\|x - z_i\| \to r_a$ for some $i \in \mathcal{M}$, then $P(\zeta) \to \infty$, and $V^*(\zeta) \to \infty$. If $V^*(\zeta) \to \infty$ then $V_L(Y) \to \infty$. Since this is a contradiction to (26) being nonincreasing, then $\forall \zeta(t_0) \notin \Omega$, $\zeta(t) \notin \Omega \, \forall t \geq t_0$. Hence, $V^*(\zeta)$ is finite and $\nabla V^*(\zeta)$ exists for all $\|x - z_i\| \neq r_a$.

After using (27), (33), and (35), [55, Th. 4.18] can be invoked to conclude that Y is uniformly ultimately bounded such that $\limsup_{t \to \infty} \|Y(t)\| \leq \underline{\nu_\iota}^{-1}(\overline{\nu_\iota}(\nu_\iota^{-1}(\iota)))$. Since $Y \in L_\infty$, it follows that $\zeta, \tilde{W}_c, \tilde{W}_a \in L_\infty$. Since W is a continuous function of ζ, $W \circ \zeta \in L_\infty$. Hence, $\hat{W}_a, \hat{W}_c \in L_\infty$ which implies $u \in L_\infty$. ■

Remark 8: The sufficient condition in (30) can be satisfied by increasing the gain k_{a2} and selecting a gain Γ_a such that $\lambda_{\min}\{\Gamma_a\}$ is large. This will not affect the sufficient conditions in (31) and (32). Selecting extrapolated trajectories x_k such that $\underline{c}$ is sufficiently large will aid in satisfying (31) without affecting (30) or (32). In addition, selecting StaF basis such that $\overline{\|\nabla \sigma\|}$ is small will help satisfy the conditions in (30) and (31). To satisfy the sufficient condition in (32) without affecting (30) or (31), it suffices to select a function Q_z according to Assumption 4 such that $\underline{q}_x$ is larger than the Lipschitz constant L_z. Provided the StaF basis functions are selected such that $\overline{\|\epsilon\|}$, $\overline{\|\nabla \epsilon\|}$, and $\overline{\|\nabla W\|}$ are small, and k_{a2} and $\underline{c}$ are selected to be sufficiently large, then the sufficient condition in (33) can be satisfied.

Remark 9: The value function V^* is dependent on the no-entry zone states, and since it is used as a candidate Lyapunov function, (26) is also dependent on the states z_i. Therefore, through proper construction of (6), it is shown in Theorem 1 that since V_L is nonincreasing there is no collision between the agent x and no-entry zones z_i. Other than Assumptions 2 and 3, there is no restriction on the movement of the obstacles. Rather, the states of the obstacles are included in the candidate Lyapunov function because the controlled agent must move such that collision is avoided, making the candidate Lyapunov function nonincreasing.

VI. Simulations

A. Mobile Robot

To demonstrate the developed approach in Sections II–V, a simulation is provided for unicycle kinematic equations, where

TABLE I
INITIAL CONDITIONS AND PARAMETERS SELECTED FOR THE MOBILE ROBOT SIMULATION

Agent Initial conditions at $t_0 = 0$ $x(0) = [\ 1.4, \quad -1.5, \quad 3.14\ ]^T$,
Penalizing parameters and input saturation $Q_x(x) = x^T q_x x$, $Q_z(z_i) = z_i^T q_z z_i$, $R = 5I_3$, $q_x = \text{diag}\{2.5, 2.5, 2.5\}$, $q_z = \text{diag}\{2, 2, 2\}$, $\mu_{sat} = 1$,
Gains for ADP update laws $k_{c1} = 0.001$, $k_{c2} = 3$, $k_{a1} = 0.75$, $k_{a2} = 0.75$, $\gamma_1 = 1$, $\beta = 0.002$, $k_u = 0.005$,
Radii $r_d = 0.6$, $\bar{r} = 0.55$, $r_a = 0.1$.

$f(x(t)) = 0_{3\times 1}$ and

$$g(x(t)) = \begin{bmatrix} \cos(x_3(t)) & -\sin(x_3(t)) & 0 \\ \sin(x_3(t)) & \cos(x_3(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Three heterogeneous no-entry zones are considered with oscillatory linear dynamics; the third state was selected to be stationary for each no-entry zone for the entirety of the simulation. The function $\mathscr{F}_i(x, z_i)$ is selected as

$$\mathscr{F}_i(x, z_i) = \begin{cases} 0, & \|x - z_i\| > r_d \\ T_i(x, z_i), & r_d \geq \|x - z_i\| > \bar{r} \\ 1, & \|x - z_i\| \leq \bar{r} \end{cases} \tag{36}$$

where $T_i(x, z_i) \triangleq \frac{1}{2} + \frac{1}{2}\cos(\pi(\frac{\|x-z_i\|-\bar{r}}{r_d-\bar{r}}))$ with the smooth scheduling function $s_i(x, z_i) = \mathscr{F}_i(x, z_i)$, and $P_a(\zeta)$ is selected as $P_a(\zeta) = 0$. For value function approximation, the StaF basis $\sigma_0(x, c(x)) = [k_{0,1},\ k_{0,2},\ k_{0,3},\ k_{0,4}]^T$ is used where $k_{0,i} \triangleq k(x, c_i(x)) = e^{x^T(x+0.05d_i)} - 1$, $i = 1, 2, 3, 4$, and the offsets are selected as $d_1 = [1,\ 0,\ 0]^T$, $d_2 = [-0.333,\ 0.943,\ 0]^T$, $d_3 = [-0.333,\ -0.471,\ 0.471]^T$, and $d_4 = [-0.333,\ -0.471,\ -0.471]^T$. The StaF basis σ_i for each obstacle is selected to be the same as the agent, except that the state changes from x to z_i. To perform BE extrapolation, five points are selected at random each time step from a $0.05\nu(x(t)) \times 0.05\nu(x(t))$ uniform distribution centered at the current state, where $\nu(x(t)) \triangleq \frac{x(t)^T x(t)}{1+x(t)^T x(t)}$. The initial critic and actor weights and gains are selected as $W_c(0) = W_a(0) = 0.4 \times 1_{16\times 1}$, $\Gamma_c(0) = 300 \times I_{16}$, and $\Gamma_a = I_{16}$. Table I summarizes the selected parameters.

B. Results

Figs. 2(a) and (b) shows that the agent and policy converge while detecting and navigating around the no-entry zones. Specifically, Fig. 2(b) shows that the agent's policy changes when the agent detects each no-entry zone shown in Fig. 2(e), and hence, modifies the agent's trajectory shown in Fig. 2(f). Fig. 2(c) and (d) shows that the critic and actor weights for value function approximation remain bounded. However, they can not be compared to the ideal values since they are unknown due to the StaF nature of the function approximation method.

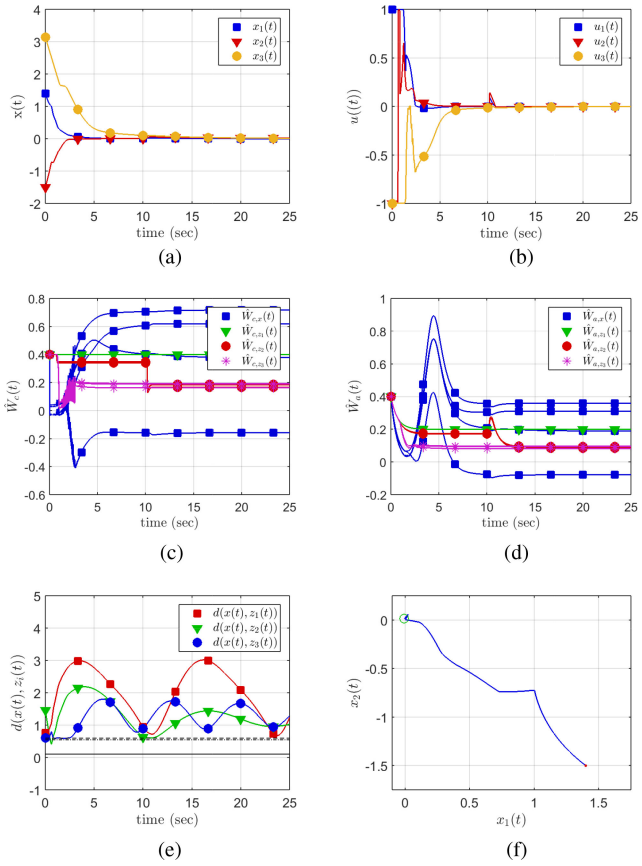

Fig. 2. States, control policy, and weight estimates are shown in addition to the distances between the agent and each avoidance region center and the agent's phase space portrait. Fig. 2(a) shows that the agents states converge to the origin. The input, shown in Fig. 2(b), causes the agent to steer off course as shown by the trajectory change of x_2 in Fig. 2(a). The distance between the agent's center and each avoidance region is shown in Fig. 2(e); the solid horizontal line represents $r_a = 0.1$, and the two dashed horizontal lines represent $r_d = 0.6$ and $\bar{r} = 0.55$, respectively. (a) The agent states. (b) The agent approximate optimal input. (c) The critic weight estimates. (d) The actor weight estimates. (e) The distance between the agent and avoidance regions. (f) The phase space portrait for $x_1(t)$ and $x_2(t)$ of the agent.

C. Nonlinear System

In addition to the mobile robot simulation, a simulation for a nonlinear system is performed with system dynamics (see [56, Ch. 5.2]) given by

$$f(x(t)) = \begin{bmatrix} -x_1(t) + x_2(t), \\ -\frac{1}{2}x_1(t) \\ -\frac{1}{2}x_2(t)\left(1 - (\cos(2x_1(t)) + 2)^2\right) \end{bmatrix}$$

and

$$g(x(t)) = \begin{bmatrix} \sin(2x_1(t)) + 2, & 0, \\ 0, & \cos(2x_1(t)) + 2 \end{bmatrix}.$$

Three heterogeneous obstacles are considered. The first and second obstacles were designed to converge to $z_1^e = [-0.8,\ 0.5]^T$ and $z_2^e = [-0.1,\ -1.1]^T$, respectively, while the third obstacle oscillated around $z_3^e = [0,\ 0]^T$ at a radius of 1.13. The functions $\mathscr{F}_i(x, z_i)$, $s_i(x, z_i)$, and $P_a(\zeta)$ are

TABLE II
INITIAL CONDITIONS AND PARAMETERS SELECTED FOR THE NONLINEAR SYSTEM SIMULATION

Agent Initial conditions at $t_0 = 0$ $x(0) = \begin{bmatrix} -1, & 1 \end{bmatrix}^T$,
Penalizing parameters and input saturation $Q_x(x) = x^T q_x x$, $Q_z(z_i) = z_i^T q_z z_i$, $R = 3I_3$, $q_x = \text{diag}\{1, 1\}$, $q_z = \text{diag}\{5, 5\}$, $\mu_{sat} = 4$,
Gains for ADP update laws $k_{c1} = 0.001$, $k_{c2} = 1$, $k_{a1} = 0.9$,$\gamma_1 = 1$, $\beta = 0.002$,
Radii $r_d = 0.7$, $\overline{r} = 0.5$, $r_a = 0.1$.

selected to be the same as the mobile robot simulation. The basis used for value function approximation for the agent is selected as $\sigma_0(x, c(x)) = [k_{\sigma,1},\ k_{\sigma,2},\ k_{\sigma,3}]^T$, where $k_{\sigma,i} \triangleq k_\sigma(x, c_i(x)) = e^{x^T(x+0.005\nu(x(t))d_i)} - 1, i = 1, 2, 3$ where $\nu(x(t))$ is defined in the mobile robot simulation and the offsets are selected as $d_1 = [0,\ 1]^T$, $d_2 = [-0.866,\ -0.5]^T$, $d_3 = [0.866\ -0.5]^T$. Moreover, the basis used for the each obstacles is selected to be the same as for the agent. A single point was selected from a $0.005\nu(x(t)) \times 0.005\nu(x(t))$ uniform distribution centered at the current state and is used to perform BE extrapolation. A projection algorithm was used on the actor weight estimates. Table II shows the selected parameters, while the initial actor, critic weights, and least-squares gains are selected as $W_c(0) = W_a(0) = 0.4 \times 1_{12\times 1}$, $\Gamma_c(0) = 1000 \times I_{12}$, and $\Gamma_a = I_{12}$, the selected parameters are shown in the table.

D. Results

Fig. 3(a) and (b) shows that the agent and policy converge to the origin. However, when a no-entry zone comes into the sensing radius of the agent, shown in Fig. 3(e), the input in Fig. 3(b) steers the agent off course. This is seen by the change in the agent's trajectory and is shown in Fig. 3(f). Moreover, when the agent senses the no-entry zones, their basis is turned ON and the corresponding actor and critic weights are updated as seen in Fig. 3(c) and (d). It is seen that the weights remain bounded. Similar to the previous simulation, the weights can not be compared to the ideal values since they are unknown.

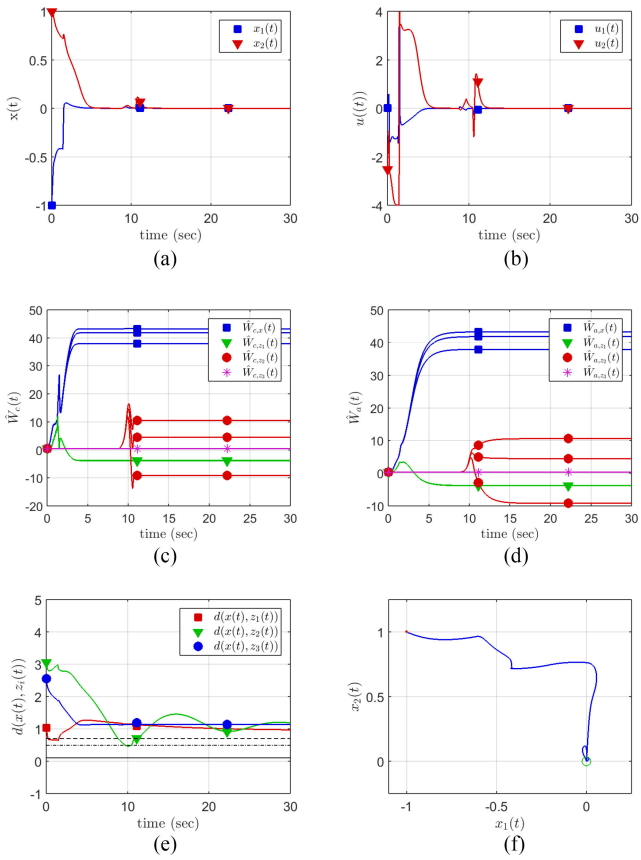

Fig. 3. States, control policy, and weight estimates for the nonlinear system simulation are shown in addition to the distances between the agent and each avoidance region center and the agent's phase space portrait. Fig. 3(a) shows that the agents states converge to the origin, but go off course when obstacles are sensed. Fig. 3(b) shows the input for the agent, which acts abruptly as obstacles are sensed. The distance between the agent's center and each avoidance region is shown in Fig. 3(e); the solid horizontal line represents $r_a = 0.1$, and the two dashed horizontal lines represent $r_d = 0.7$ and $\overline{r} = 0.5$, respectively. (a) The agent states. (b) The agent approximate optimal input. (c) The critic weight estimates. (d) The actor weight estimates. (e) The distance between the agent and avoidance regions. (f) The phase space portrait for $x_1(t)$ and $x_2(t)$ of the agent.

VII. EXTENSION TO UNCERTAIN NUMBER OF AVOIDANCE REGIONS AND UNCERTAIN SYSTEMS

In Section II–V, the HJB in (10) required the number of no-entry zones in the operating domain to be known, which may not always be available. In this section, an extension is provided which alleviates the need to know how many no-entry zones are in the operating domain. Furthermore, by adding and subtracting $P_a(x, Z)$, the following value function is introduced:

$$V^*(x(t), Z(t)) = P_a(x(t), Z(t)) + V^\#(x(t), Z(t)) \quad (37)$$

where $V^\#(x(t), Z(t))$ is an approximation error of the optimal value function. Furthermore, the function $V^\#(x, Z)$ can be interpreted as time-varying map $V_t^\# : \mathbb{R}^n \times \mathbb{R}_{\geq t_0}$ such that $V_t^\#(x, t) = V^\#(x, Z)$ [57]. Therefore, (37) is rewritten as

$$V^*(x(t), Z(t)) = P_a(x(t), Z(t)) + V_t^\#(x(t), t). \quad (38)$$

The optimal controller u^* is admissible; hence, the value function $V^*(x, Z)$ is finite and $x, Z \notin \Omega$. Therefore, $P_a(x, Z)$ is continuous for $x, Z \notin \Omega$, hence (38) can be approximated via the StaF approximation method. However, because time does not lie on a compact domain, $V_t^\#$ can not be approximated directly using time as an input to the NN. To address this technical challenge, the mapping $\phi : \mathbb{R}_{\geq t_0} \to [0, \alpha]$, $\alpha \in \mathbb{R}_{>0}$ is introduced such that $V_t^\#(x(t), t) = V_t^\#(x(t), \phi^{-1}(\kappa)) = V_\kappa^\#(x(t), \kappa)$ where $\kappa = \phi(t)$. Now, κ lies on a compact set and the function $V_\kappa^\#(x, \kappa)$ can be approximated using the StaF method as

$$V^*(x(t), Z(t)) = P_a(x(t), Z(t)) + W^T(\zeta^\#(t))\,\sigma(y(t), c(\zeta^\#(t))) + \varepsilon(y(t), \zeta^\#(t))$$

with $\sigma(\zeta^{\#}, c(\zeta^{\#})) = [\begin{smallmatrix} \sigma_0(x, c_0(x)) \\ s_0(x)\sigma_1(\kappa, c_1(\kappa)) \end{smallmatrix}]$, where $\zeta^{\#} \triangleq [x^T, \kappa]^T$, $y \triangleq [y_x^T, y_\kappa]^T \in \overline{B_r(\zeta^{\#})}$, and $s_0 : \mathbb{R}^n \to [0,1]$ is a smooth function such that $s_0(0_{2\times 1}) = 0$.

Moreover, since $P_a(x, Z) = \sum_{i\in\mathcal{M}} P_{a,i}(x, z_i)$ is designed to be a bounded positive semidefinite symmetric function, it follows that $\frac{\partial P_{a,i}(x,z_1,\ldots,z_m)}{\partial x} = -\frac{\partial P_{a,i}(x,z_1,\ldots,z_m)}{\partial z_i}$ for all $i \in \mathcal{M}$; hence, the HJB is represented as

$$0 = r(x, Z, u) + \frac{\partial V_\kappa^{\#}\left(\zeta^{\#}\right)}{\partial \zeta^{\#}}\left(F^{\#}\left(\zeta^{\#}\right) + G^{\#}\left(\zeta^{\#}\right) u\right) + \sum_{i=1}^{M} \frac{\partial P_{a,i}}{\partial x}\left(f(x) + g(x)u - \mathscr{F}_i\left(x, z_i\right) h_i\left(z_i\right)\right) \quad (39)$$

where $F^{\#}(\zeta^{\#}) \triangleq [\, f(x)^T,\ \frac{\partial \kappa}{\partial t}\,]^T$, and $G^{\#}(\zeta^{\#}) \triangleq [g(x)^T, 0_{m\times 1}]^T$. The HJB in (39) requires the knowledge of the uncertain dynamics $f(x)$ and $h_i(z_i)$. Using a NN approximator, the time derivative of P_a is written as

$$\dot{P}_a = \sum_{i=1}^{M} \frac{\partial P_{a,i}}{\partial x}\left(f(x) + g(x)u - \mathscr{F}_i\left(x, z_i\right) h_i\left(z_i\right)\right) = Y_p(x, Z)\theta + \varepsilon_p(x, Z)$$

where $Y_p : \mathbb{R}^n \times \mathbb{R}^{Mn} \to \mathbb{R}^{1\times l_p}$ is a selected basis such that $Y_p(x, Z) = 0_{1\times l_p}$ when $\|x - z_i\| > r_d$, for all $i \in \mathcal{M}$, $\theta \in \mathbb{R}^{l_p}$ is an unknown weight, and $\varepsilon_p : \mathbb{R}^n \times \mathbb{R}^{Mn} \to \mathbb{R}$ is the unknown function approximation error. Likewise the agent drift dynamics can be represented as $f(x(t)) = Y_f(x(t))\Xi + \varepsilon_f(x(t))$ with $Y_f : \mathbb{R}^n \to \mathbb{R}^{n\times l_f}$ being a known basis, $\Xi \in \mathbb{R}^{l_f}$ an unknown weight, and $\varepsilon_f : \mathbb{R}^n \to \mathbb{R}^n$ the function approximation error.[5]

Assumption 6: There exists constants $\overline{\varepsilon}_p, \overline{\varepsilon}_f, \overline{Y}_f, \overline{Y}_p, \overline{\theta}, \overline{\Xi} \in \mathbb{R}_{>0}$ such that $\sup_{\zeta\in\chi} \|Y_p(x, Z)\| \le \overline{Y}_p$, $\sup_{\zeta\in\chi} \|\varepsilon_p(x, Z)\| \le \overline{\varepsilon}_p$, $\sup_{x\in\chi} \|Y_f(x)\| \le \overline{Y}_f$, $\sup_{x\in\chi} \|\varepsilon_f(x)\| \le \overline{\varepsilon}_f$, $\|\theta\| \le \overline{\theta}$, and $\|\Xi\| \le \overline{\Xi}$ [23], [58].

Using the estimates $\hat{W}_c$, $\hat{W}_a$, $\hat{\theta}$, and $\hat{\Xi}$ in (39), the approximate BE $\hat{\delta} : \mathbb{R}^{n+1} \times \mathbb{R}^{n+1} \times \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^{l_f+l_p} \to \mathbb{R}$ is defined as

$$\hat{\delta}\left(y, \zeta^{\#}, Z, \hat{W}_c, \hat{W}_a, \hat{\theta}, \hat{\Xi}\right) \triangleq Y_p\left(y_x, Z\right)\hat{\theta} + \omega^{\#}\left(y, \zeta^{\#}, Z, \hat{W}_a, \hat{\Xi}\right)^T \hat{W}_c + r\left(y_x, Z, \hat{u}\left(y, \zeta^{\#}, Z, \hat{W}_a\right)\right) \quad (40)$$

where $\omega^{\#}(y, \zeta^{\#}, Z, \hat{W}_a, \hat{\Xi}) \triangleq \nabla\sigma(y, c(\zeta^{\#}))(Y^{\#}(y)\hat{\Xi}^{\#} + G^{\#}(y)\hat{u}(y, \zeta^{\#}, Z, \hat{W}_a))$, $Y^{\#}(y) \triangleq [\, Y_f(y_x)^T,\ \frac{\partial y_\kappa}{\partial t}\,]^T$, $\hat{\Xi}^{\#} \triangleq [\begin{smallmatrix} \hat{\Xi} & 0_{l_f\times 1} \\ 0_{1\times 1} & 1 \end{smallmatrix}]$, and

$$\hat{u}\left(y, \zeta^{\#}, Z, \hat{W}_a\right) \triangleq -\mu_{\text{sat}} \operatorname{Tanh}\left(\frac{1}{2\mu_{\text{sat}}} R^{-1} G^{\#T}(y) \times \left(\nabla P_a^T\left(y_x, Z\right) + \nabla\sigma^T\left(y, c\left(\zeta^{\#}\right)\right)\hat{W}_a\right)\right) \quad (41)$$

where $\nabla P_a(y_x, Z) \triangleq [\frac{\partial P_a(y_x,Z)}{\partial x},\ \frac{\partial P_a(y_x,Z)}{\partial \kappa}] = [\frac{\partial P_a(y_x,Z)}{\partial x},\ 0]$. Using $\hat{\delta}$, the instantaneous BEs and approximate policies in (18)–(21) are redefined as $\delta_t(t) \triangleq \hat{\delta}(\zeta^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_c(t), \hat{W}_a(t), \hat{\theta}(t), \hat{\Xi}(t))$, $\delta_k(t) \triangleq \hat{\delta}(\zeta_k^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_c(t), \hat{W}_a(t), \hat{\theta}(t), \hat{\Xi}(t))$, $u(t) \triangleq \hat{u}(\zeta^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_a(t))$, and $\hat{u}_k(t) \triangleq \hat{u}(\zeta_k^{\#}(t), \zeta^{\#}(t), Z(t), \hat{W}_a(t))$, respectively.

Assumption 7. [22], [23]: There exists a compact set $\Theta \subset \mathbb{R}^{l_p+l_f}$, known *a priori*, which contains the unknown parameter vectors θ and Ξ. Let $\tilde{X} \triangleq [\tilde{\Xi}^T, \tilde{\theta}^T]^T = [(\Xi - \hat{\Xi})^T, (\theta - \hat{\theta})^T]^T$ and $\hat{X} = [\hat{\Xi}^T, \hat{\theta}^T]^T$ denote the total concatenated vector of parameter estimate errors and parameter estimates, respectively. The estimates $\hat{X} : \mathbb{R}_{\ge t_0} \to \mathbb{R}^{l_p+l_f}$ are updated based on switched update laws of the form

$$\dot{\hat{X}}(t) = f_{Xs}\left(\hat{X}(t), t\right), \quad \hat{X}\left(t_0\right) \in \Theta \quad (42)$$

where $s \in \mathbb{N}$ is the switching index with $\{f_{Xs} : \mathbb{R}^{l_p+l_f} \times \mathbb{R}_{\ge t_0} \to \mathbb{R}^{l_p+l_f}\}_{s\in\mathbb{N}}$ being a family of continuously differentiable functions. There exist a continuously differentiable function $V_\theta : \mathbb{R}^{l_p+l_f} \times \mathbb{R}_{\ge t_0} \to \mathbb{R}_{\ge 0}$ that satisfies

$$\underline{\nu}_\theta\left(\left\|\tilde{X}\right\|\right) \le V_\theta\left(\tilde{X}, t\right) \le \overline{\nu}_\theta\left(\left\|\tilde{X}\right\|\right) \quad (43)$$

$$\frac{\partial V_\theta\left(\tilde{X}, t\right)}{\partial \tilde{X}}\left(-f_{Xs}\left(\tilde{X}(t), t\right)\right) + \frac{\partial V_\theta\left(\tilde{X}, t\right)}{\partial t} \le -K_\theta\left\|\tilde{X}\right\|^2 + D\left\|\tilde{X}\right\| \quad (44)$$

for all $t \in \mathbb{R}_{\ge t_0}$, $s \in \mathbb{N}$, and $\tilde{X} \in \mathbb{R}^{l_p+l_f}$. In (43), $\underline{\nu}_\theta, \overline{\nu}_\theta : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ are class $\mathcal{K}$ functions. In (44), $K_\theta \in \mathbb{R}_{>0}$ is an adjustable parameter, $D \in \mathbb{R}_{>0}$ is a positive constant, and the ratio $\frac{D}{K_\theta}$ is sufficiently small.[6]

Remark 10: If $f(x(t)) = 0_{n\times 1}$, then $Y^{\#}(y)$ and $\hat{\Xi}^{\#}$ simplify to $Y^{\#}(y) \triangleq [\, 0_{l_f\times n},\ \frac{\partial y_\kappa}{\partial t}\,]^T$ and $\hat{\Xi}^{\#} \triangleq [\begin{smallmatrix} 0_{l_f\times 1} & 0_{l_f\times 1} \\ 0_{1\times 1} & 1 \end{smallmatrix}]$, respectively. Furthermore, Ξ does not need to be estimated for single integrator dynamics and the concatenated systems then reduce to $\tilde{X} \triangleq \tilde{\theta}$ and $f_{Xs}(\hat{X}(t), t) \triangleq f_{\theta s}(\hat{\theta}(t), t)$.

The conditions (43) and (44) in Assumption 7 imply that V_θ can be used as a candidate Lyapunov function to show the parameter estimates $\hat{\theta}$ and $\hat{\Xi}$ converge to a neighborhood of the true values. Update laws using CL-based methods can be designed to satisfy Assumption 7; examples of such update laws can be found in [59]–[62]. The main result for the extension to systems with uncertainties and an unknown number of avoidance regions uses $V_\theta + V_L$ as a candidate Lyapunov function and is summarized in the following theorem.

Theorem 2: Provided Assumptions 2–7 along with the sufficient conditions in (30)–(33) are satisfied, and StaF kernels are selected such that ∇W, ε, $\nabla\varepsilon$, are sufficiently small, then the update laws in (22)–(24) with (41), $\delta_t(t)$, and $\delta_k(t)$ ensure that the state x and input $u(t)$, and weight approximation errors $\tilde{W}_a$, $\tilde{W}_c$, $\tilde{\theta}$, $\tilde{\Xi}$ are UUB; furthermore, states $x(t)$, $z_i(t)$ starting outside of Ω remain outside of Ω.

Proof: The proof is a combination of Assumption 7 with Theorem 1 by using $V_L + V_\theta$ as a candidate Lyapunov function; hence, the proof is omitted to alleviate redundancy. ■

[5]If the agent dynamics $f(x(t))$ are assumed to be single integrator dynamics such that $f(x(t)) = 0_{n\times 1}$, system identification for the agent is not necessary.

[6]The positive constant D can possibly depend on the parameter K_θ.

VIII. Simulation-in-the-Loop Experiments

In Section VI, simulations where performed to demonstrate the developed approach. To demonstrate the robustness of the developed method, experiments are performed on a quadcopter avoiding virtual obstacles. Specifically, three experiments are conducted to demonstrate the ability of an aerial vehicle to be autonomously regulated to the origin while avoiding dynamic avoidance regions. For each experiment, a Parrot Bebop 2.0 quadcopter is used as the aerial vehicle. The developed quadcopter controller requires feedback of its and the obstacle's position and orientation (pose). The pose of the quadcopter is obtained by a NaturalPoint, Inc. OptiTrack motion capture system at 120 Hz. Using the robotic operating system (ROS) Kinetic framework and the *bebop_*autonomy *package* developed by [63] running on Ubuntu 16.04, the control policies are calculated for the quadcopter. The control policy is communicated from a ground station which broadcasts velocity commands at 120 Hz over the 5-GHz Wi-Fi channel. The developed control policy is implemented as a velocity command to the quadcopter. While this allows an effective demonstration of the underlying strategy, improved performance could be obtained by implementing the policies through acceleration commands that do not rely on the onboard velocity tracking controller. Such an implementation could also have additional implications due to input constraints for acceleration commands.

The experiments are performed using two-dimensional (2-D) Euclidean coordinates (without the inclusion of altitude) for the state $x(t)$ for ease of experimental execution and implementation. However, since the development does not restrict the state dimension, experiments can also be extended to use 3-D Euclidean coordinates. All three experiments use simplified quadcopter dynamics represented by (1) with $f(x(t)) = 0_{2\times 1}$ and $g(x(t)) = I_2$ so that $\dot{x} = u$, where, without a loss of generality, $x(t) \in \mathbb{R}^2$ is the composite vector of the 2-D Euclidean coordinates, with respect to the inertial frame and $u \in \mathbb{R}^2$ are velocity commands broadcast to the quadcopter. A supplementary video of the experiment accompanies this article, available for download[7] (included in the submitted files). For the first two experiments, virtual spheres are used as the dynamic avoidance regions. The virtual spheres, which evolve according to linear oscillatory dynamics, are generated using ROS via Ubuntu on the ground station. The positions of the virtual spheres in the inertial frame are used in the designed method to interact with the vehicle, only when each position is within the detection radius of the quadcopter. The supplementary video shows how the quadcopter interacts with the virtual spheres. For the third experiment, one of the virtual spheres is replaced by a remotely controlled (i.e., human-piloted) quadcopter.

A. Experiment One

The first experiment is performed using the method developed in Sections II–V. Three virtual avoidance regions are generated using heterogeneous oscillatory linear dynamics. The functions $\mathscr{F}_i(x, z_i)$, $s_i(x, z_i) = \mathscr{F}_i(x, z_i)$, are selected to be the same as in Section VI while $P_a(\zeta)$ is selected to be $P_a(\zeta) = \sum_{i=1}^{M} (\min\{0, \frac{\|x-z_i\|^2 - r_d^2}{(\|x-z_i\|^2 - r_d^2)^2 + r_\varepsilon}\})^2$. For value function approximation, the agent is selected to have the StaF basis $\sigma_0(x, c(x)) = [x^T c_1(x), x^T c_2(x), x^T c_3(x)]^T$, where $c_i(x) = x + \nu(x) d_i$, $i = 1, 2, 3$, where $\nu(x)$ is redefined as $\nu(x) \triangleq \frac{0.5 x^T x}{1 + x^T x}$ and the offsets are selected as $d_1 = [0, -1]^T$, $d_2 = [0.866, -0.5]^T$, and $d_3 = [-0.866, -0.5]^T$. The StaF basis σ_i for each obstacle is selected to be the same as the agent, except that the state changes from x to z_i. Assumption 5 discussed how the extrapolated regressors ω_k are design variables. Thus, instead of using input–output data from a persistently exciting system, the dynamic model can be used and evaluated at a single time-varying extrapolated state to achieve sufficient excitation. It was shown in [22, Sec. 6.3] that the use of a single time-varying extrapolated point results in improved computational efficiency when compared using a large number of stationary extrapolated states. Motivated by this insight, at each time a single point is selected at random from a $0.2\nu(x(t)) \times 0.2\nu(x(t))$ uniform distribution centered at the current state. The initial critic and actor weights and gains are selected as $W_c(0) = U[0, 4] 1_{12\times 1}$, $W_a(0) = 1_{12\times 1}$, $\Gamma_c(0) = I_{12}$, and $\Gamma_a = I_{12}$, and the selected parameters are shown in Table III.

TABLE III
Initial Conditions and Parameters Selected for the Experiment

Agent Initial conditions at $t_0 = 0$ $x(0) = [-6.3, 1.5]^T$,
Penalizing parameters and input saturation $Q_x(x) = x^T q_x x$, $Q_z(z_i) = z_i^T q_z z_i$, $R = 10 I_2$, $q_x = \text{diag}\{2.0, 1.0\}$, $q_z = \text{diag}\{2.0, 2.0\}$, $\mu_{sat} = 0.5$,
Gains for ADP update laws $k_{c1} = 0.05$, $k_{c2} = 0.75$, $k_{a1} = 0.75$, $k_{a2} = 0.01$, $\gamma_1 = 1$, $\beta = 0.001$, $k_u = 1$,
Radii (m) $r_d = 0.7$, $\overline{r} = 0.45$, $r_a = 0.2$, $r_\varepsilon = 0.15$.

B. Experiment Two

The second experiment is performed using the extension in Section VII and similar to experiment one, three virtual avoidance regions are generated with heterogeneous oscillatory linear dynamics. The agent has the same basis $\sigma_0(x)$ as the first experiment, while the basis $\sigma_1(\kappa, c(\kappa))$ is selected as $\sigma_1(\kappa, c(\kappa)) = [\kappa^T c_1(\kappa), \kappa^T c_2(\kappa),]^T$, where $\kappa = \phi(t) \triangleq \frac{0.25}{0.01t+1}$ and $c_i(\kappa) = \kappa + \nu(\kappa) d_i$, $i = 1, 2$ where $\nu(\kappa)$ is the same function as in the first experiment except evaluated at κ and the offsets are selected as $d_1 = 0.25$, and $d_2 = 0.05$. For the total basis $\sigma(\zeta^\#, c(\zeta^\#))$, the function $s_0(x)$ is selected as $s_0(x) = \frac{\nu(x)}{0.5}$. The initial critic and actor weights and adaptive gains are selected as as $W_c(0) = U[0, 4] 1_{5\times 1}$, $W_a(0) = 1_{5\times 1}$, and $\Gamma_a = I_5$. The rest of the parameters are selected to remain the same as in the first experiment and are shown in Table III. Since the agent dynamics are modeled as single integrator dynamics with $f(x(t)) = 0_{2\times 1}$, system identification was not performed on the agent. However, to approximate θ in Section VII, the ICL method in [62, Sec. IV.B] was utilized with the basis $Y_p(x, Z) = \text{Tanh}(V_p^T \nabla P_a(y_x, Z)^T)$, where $V_p = U[-5, 5] 1_{3\times 10}$ is a constant weight matrix. To keep

[7][Online]. Available: http://ieeexplore.ieee.org

the weight estimates bounded, a projection algorithm was used similar to [62, Sec. IV.B] and the update laws were turned OFF when no avoidance regions were sensed. Not performing system identification on the agent reduces redundancy in parameter identification because the unknown weight θ in the function in the time derivative of P_a is already being approximated. Furthermore, as stated in Footnote 5, if the agent is implemented using single integrator dynamics, then system identification can be ignored on the agent drift dynamics $f(x(t))$.

C. Experiment Three

The third experiment is performed using the extension in Section VII where the first avoidance region, denoted by the state z_1 and represented by another Parrot Bebop quadcopter, is flown/controlled manually by hand. The virtual avoidance regions with states z_2 and z_3 are simulated as in the previous experiments. The radii were changed to $r_d = 1.0$, $\bar{r} = 0.7$, and $r_a = 0.45$ meters (m) to reduce the chance of the quadcopters colliding, the gains q_x, q_z where changed to $q_x = \text{diag}\{0.5, 0.5\}$ and $q_z = \text{diag}\{3.0, 3.0\}$, and the rest of the parameters remained the same as in the second experiment.

D. Results

The first experimental validation for the development in Sections II–V are shown in Figs. 4 and 5. Fig. 4(a) and (b) shows that the agent, as well as the agent's control policy, remains bounded around the origin. Fig. 4(b) shows that the control of the agent is bounded by $0.5\frac{m}{s}$ even in the presence of the mobile avoidance regions. The input does not converge to zero because of aerodynamic disturbances, when the quadcopter reaches the origin. The critic and actor weight estimates remain bounded and converge to steady-state values, as shown in Fig. 4(c) and (d). However, because of the StaF nature of the StaF approximation method, the ideal weights are unknown; hence the estimate cannot be compared to their ideal values. Even though the agent enters the detection region as shown by Figs. 4(e) and 5(a), the developed method drives the agent away from the avoidance regions and toward the origin. When encountering avoidance region z_2 between the 8th and 12th seconds, the agent was able to maneuver around the avoidance region without collision despite multiple encounters with it because the avoidance region was moving close to the origin and obstructing the path. Moreover, Fig. 6(a) shows the change in velocity when the agent encounters the avoidance region.

The second and third experiments were performed to validate the development in Section VII with the results shown in Figs. 5–8. Specifically, the second experiment was performed using similar conditions and parameters as in the first experiment. Fig. 5(b) shows that the agent is capable of adjusting its path when it encounters the avoidance regions and the agent is regulated to the origin without colliding with the avoidance regions, while Fig. 6(b) shows the relative velocities between the agent and each avoidance region. The approximate value function and total cost for the first two experiments are shown in Fig. 7. Both experiments resulted in similar costs and approximate value functions. Specifically, Fig. 7(a) shows that the approximate

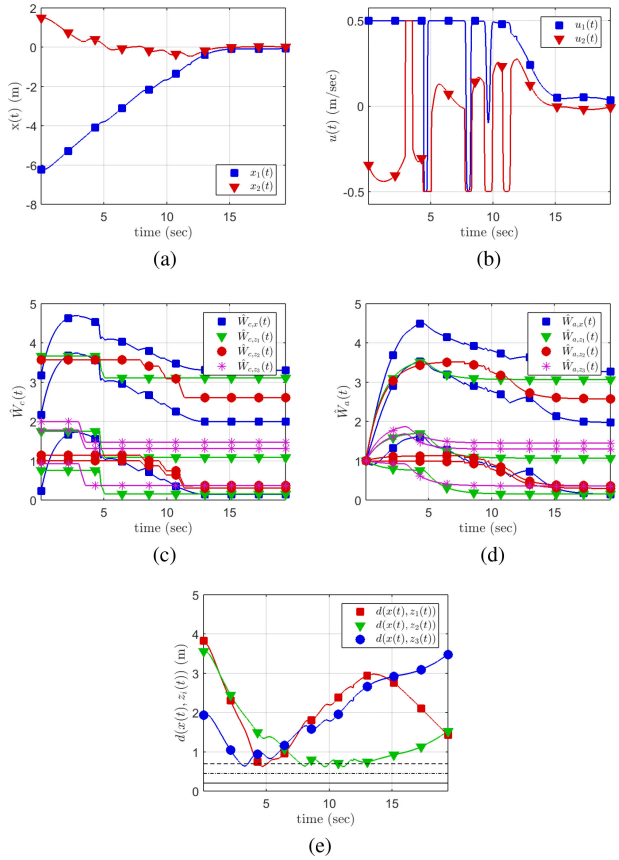

Fig. 4. States, control policy, and weight estimates are shown in addition to the distances between the agent and each avoidance region center for the first experiment. Fig. 4(a) shows that the agents states converge to a close neighborhood of the origin. When the agent detects the avoidance regions, the commanded input, shown in Fig. 4(b), causes the agent to steer off course as shown by the change in the trajectory of x_2 in Fig. 4(a). The distance between the center of the agent and each avoidance region is shown in Fig. 4(e); the two dashed horizontal lines represent the detection radius and conflict radius denoted by $r_d = 0.7$ m and $\bar{r} = 0.45$ m, respectively, while the solid horizontal line represents the radius of the avoidance region denoted by $r_a = 0.2$ m. (a) The agent states. (b) The agent approximate optimal input. (c) The critic weight estimates. (d) The actor weight estimates. (e) The distance between the agent and avoidance regions.

value function remains positive and converges to zero when the agent reaches the origin.

Furthermore, the third experiment extends the second experiment further by substituting one of the autonomous avoidance regions for a nonautonomous one. Specifically, a manually controlled avoidance region is used, which is controlled to approach the agent throughout the experiment. Figs. 5(c)–8 show the results of the experiment. In Fig. 5(c), the agent is forced away from the direction of the origin, but still manages to redirect itself without colliding with the avoidance regions. The relative velocity for the third experiment is shown in Fig. 6(c), which changes abruptly as each avoidance regions is sensed. The approximate value function and total cost for the third experiment are also shown in Fig. 7. Since one of the avoidance regions was remotely controlled, its trajectory was nonautonomous; hence, the agent's trajectory differed when interacting with it and the applied control policy did not saturate as much compared to

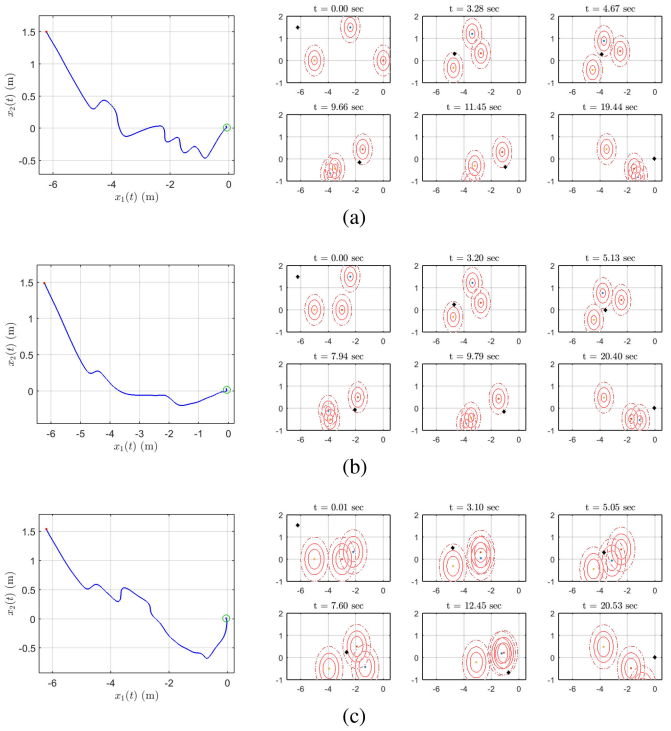

Fig. 5. Phase-space portrait for the agent and the positions of the agent and avoidance regions for each experiment. In each figure, the left plot shows the agent's phase-space portrait where the green circle is the agent's final position. The plots on the right of each figure show the agent's and avoidance regions positions at certain time instances where the diamond represents the agent state and the circles represent the avoidance regions. (a) The agent phase-space portrait (left) and the positions of the agent and avoidance regions (right) for the first experiment. (b) The agent phase-space portrait (left) and positions of the agent and avoidance region (right) for the second experiment. (c) The agent phase-space portrait (left) and positions of the agent and avoidance region (right) for the third experiment.

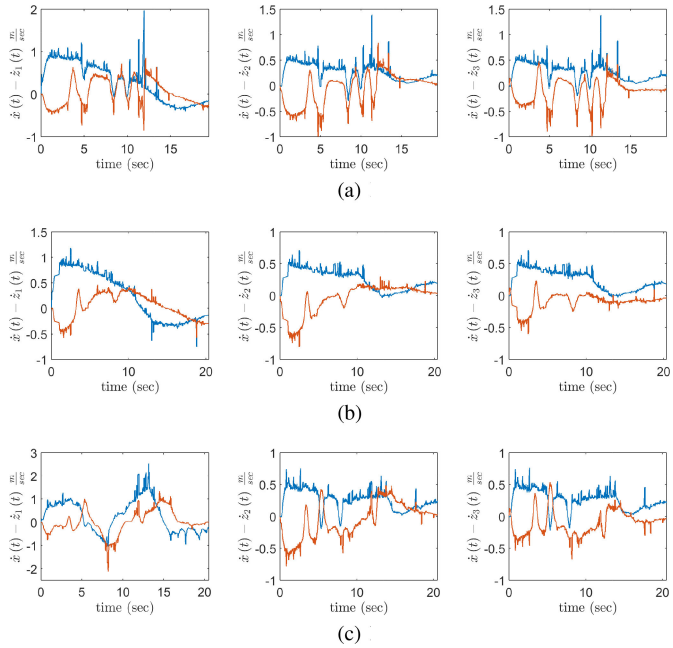

Fig. 6. Relative velocities for each experiment. The relative velocities were numerically computed and filtered using a moving average filter with a window size of ten time-steps. In each figure, the blue line represents the relative velocity of the first state and the red line represents the relative velocity of the second state for each obstacle (i.e., $\dot{x}_1(t) - \dot{z}_{i,1}(t)$ and $\dot{x}_2(t) - \dot{z}_{i,2}(t)$ for $i = 1, 2, 3$, respectively). (a) Relative velocities for z_1 (left), z_2 (middle), and z_3 (right) for the first experiment. (b) Relative velocities for z_1 (left), z_2 (middle), and z_3 (right) for the second experiment. (c) Relative velocities for z_1 (left), z_2 (middle), and z_3 (right) for the third experiment.

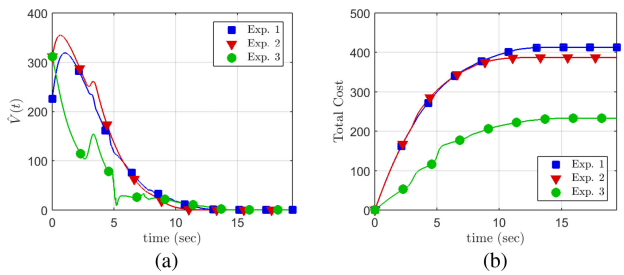

Fig. 7. Approximate value functions and total costs for the three experiments. (a) Approximate value function. (b) Total cost.

the first experiment, resulting in a smaller total cost. Fig. 8(a) shows that the agent is regulated to the origin and that its state is adjusted online in real time by the input, as shown in Fig. 8(b), when it encounters the avoidance regions. The input remains bounded by the controller saturation of $0.5\frac{\text{m}}{\text{s}}$ and converges to a small bounded residual of the origin. The estimates of the unknown weights θ are shown in Fig. 8(c), which remain bounded, but since the ideal basis is unknown and the ideal weights are unknown, the estimates cannot be compared to the actual weights. Fig. 8(d) shows the distance between the agent and each avoidance region center, and shows that the agent does not get within r_a of the avoidance regions. Moreover, as soon as the agent gets within $\bar{r}$ of the avoidance region, it moves away from z_i. Additionally, when the agent detects the avoidance region, i.e., $\|x - z_i\| \leq r_d$, the control policy is adjusted, which can be seen from Fig. 8(b) and (d). Moreover, the critic and actor weights estimates using the transformation in Section VII are shown in Fig. 8(e) and (f), respectively. The figures show that the estimates remain bounded and converge to steady-state values. Similar to the first experiment, the ideal weights are unknown, thus the weight estimates cannot be compared to the ideal weights.

The results in Figs. 4–8 show that the developed method is capable of handling uncertain dynamic avoidance regions while regulating an autonomous agent. The agent locally detects the avoidance regions and then adjusts its path online. While experiments one and two used radii selected as $r_d = 0.7$, $\bar{r} = 0.45$, $r_a = 0.2$ meters, the radii in experiment three were increased to $r_d = 1.0$, $\bar{r} = 0.7$, and $r_a = 0.45$ m, respectively. The increase in radii was because one of the obstacles moved at a higher relative velocity, and a larger distance was required to enable the agent to avoid collision. The optimal selection of the size of the detection region (e.g., as a function of the maximum agent speed, the obstacle relative velocity, and the sensing rate) including detection radii changing with relative agent and avoidance region velocities remains a subject for future research.

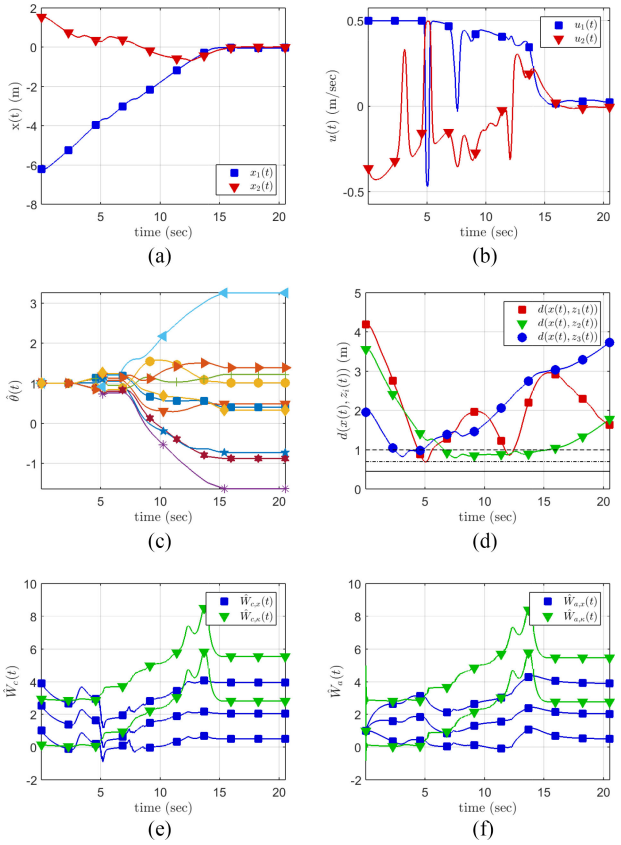

Fig. 8. States, control policy, and weight estimates of the agent are shown in addition to the distances between the agent and each avoidance region center for the third experiment. (a) The agent states. (b) The agent approximate optimal input. (c) The estimates of θ. (d) The distance between the agent and avoidance regions. (e) The critic weight estimates. (f) The actor weight estimates.

IX. Conclusion

In this article, an online approximate motion planning strategy in the presence of mobile avoidance regions was developed. Because the avoidance regions need to only be known inside a detection radius, they were modeled using local dynamics. Since the avoidance regions were coupled with the agent in the HJB, the basis of the approximation also used the avoidance region state when approximating the value function. Because the states were not always known, a scheduling function was used to turn-OFF the basis, which then stopped updating the weight approximations for the avoidance regions when they were not detected. Theorem 1 showed the UUB of the states and that the states of the coupled system remain outside of the avoidance set. An extension to systems with uncertain dynamics and an unknown number of avoidance regions was presented, and Theorem 2 summarized the overall stability for the system with uncertainties. Simulations and three experiments were performed that demonstrated successful implementation of the developed motion planning and avoidance region evasion strategy.

Some possible topics of future research include—determining the size of the avoidance region based on the sampling rate of nearby obstacles, investigating the use of collision cones (cf., [39], [40]) instead of spherical-shaped avoidance regions, investigating methods to alleviate Assumption 3, and extending the developed approach to multiple agents which cooperate but avoid other mobile obstacles. Investigating the relationships between the speed of the agent and avoidance regions and the respective sensing radius including dynamically changing radii also remains a topic for future research.

Moreover, in the presence of uncertainty, it is unclear how to develop a finite-time convergent update law. However, recent developments, such as [64] and [65], could potentially provide insight into developing finite-time approximate optimal controllers. In addition, during the learning phase of adaptive systems, it is difficult to ensure safety guarantees are met, especially in safety-critical systems. Results such as [66]–[68] could provide insight into designing RL constrained approaches with safety specifications for motion planning. Such investigations into finite-time and safety-critical approximate optimal controllers are subjects of future research.

Appendix A
Stability Analysis Constants

In Section V, the positive constants $\iota, \varphi_{ac} \in \mathbb{R}_{>0}$ are introduced, which are defined as $\iota \triangleq \frac{\iota_c^2}{k_{c2}\underline{c}} + \frac{(\iota_{a1}+\iota_{a2})^2}{k_{a1}+k_{a2}} + \overline{\|W^T\nabla\sigma + \sigma^T\nabla W + \nabla\epsilon + \nabla P_a\|}\frac{G_R}{2}\overline{\|\nabla W^T\sigma + \nabla\varepsilon^T\|}m$, and $\varphi_{ac} \triangleq k_{a1} + \frac{\overline{\|\nabla W\|}}{\underline{\Gamma}}\frac{G_R}{2}\overline{\|\nabla\sigma^T\|} + k_{c1}\sqrt{\frac{m}{\gamma_1}}\mu_{\text{sat}}\overline{\|\nabla\sigma G\|} + \frac{k_{c2}}{N}\sum_{k=1}^{N}\sqrt{\frac{m}{\gamma_1}}\mu_{\text{sat}}\overline{\|\nabla\sigma_k G_k\|}$, where $\iota_{a1} \triangleq \overline{\|W^T\nabla\sigma + \sigma^T\nabla W + \nabla\epsilon + \nabla P_a\|}\frac{G_R}{2}\overline{\|\nabla\sigma^T\|} + \frac{\overline{\|\nabla WF\|}}{\lambda_{\min}\{\Gamma_a\}} + \frac{\overline{\|\nabla W\|}}{\lambda_{\min}\{\Gamma_a\}}\frac{G_R}{2}\overline{\|\nabla P_a + \nabla\sigma^T W\|}$, $\iota_{a2} \triangleq k_{a2}\|W\| + k_{c1}\mu_{\text{sat}}\overline{\|\nabla\sigma G\|}\sqrt{\frac{m}{\gamma_1}}\|W\| + \frac{k_{c2}}{N}\sum_{k=1}^{N}\mu_{\text{sat}}\overline{\|\nabla\sigma_k G_k\|}\sqrt{\frac{m}{\gamma_1}}\overline{\|W\|}$, and $\iota_c \triangleq \frac{\overline{\|\nabla WF\|}}{\underline{\Gamma}} + \frac{\overline{\|\nabla W\|}}{\underline{\Gamma}}\frac{G_R}{2}\overline{\|\nabla P_a + \nabla\sigma^T W\|} + k_{c1}\frac{1}{2\sqrt{\gamma_1}}\overline{\|\Delta\|} + \frac{k_{c2}}{N}\sum_{k=1}^{N}\frac{1}{2\sqrt{\gamma_1}}\overline{\|\Delta_k\|}$.

Acknowledgment

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.

References

[1] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.
[2] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Auton. Robot.*, vol. 35, no. 1, pp. 51–76, 2013.
[3] 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.
[4] 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.
[5] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. R*, vol. 30, pp. 846–894, 2011.
[6] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 418–429, Apr. 2004.

[7] G. Leitmann and J. Skowronski, "Avoidance control," *J. Optim. Theory Appl.*, vol. 23, no. 4, pp. 581–591, 1977.

[8] C. Shen, Y. Shi, and B. Buckham, "Path-following control of an AUV: A multiobjective model predictive control approach," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1334–1342, May 2019.

[9] Z. Kan, A. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1827–1832, Jul. 2012.

[10] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.

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

[12] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, and D. Lane, "Path planning for autonomous underwater vehicles," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 331–341, Apr. 2007.

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

[14] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[15] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive Dynamic Programming for Control Algorithms and Stability*, (Communications and Control Engineering). London, U.K.: Springer-Verlag, 2013.

[16] H. Zhang, L. Cui, and Y. Luo, "Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 206–216, Feb. 2013.

[17] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[18] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Third Quarter 2009.

[19] K. G. Vamvoudakis, M. F. Miranda, and J. P. Hespanha, "Asymptotically stable adaptive–optimal control algorithm with saturating actuators and relaxed persistence of excitation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2386–2398, Nov. 2016.

[20] X. Yang, D. Liu, D. Wang, and H. Ma, "Constrained online optimal control for continuous-time nonlinear systems using neuro-dynamic programming," in *Proc. IEEE Chin. Control Conf.*, 2014, pp. 8717–8722.

[21] 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.

[22] R. Kamalapurkar, J. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, Dec. 2016.

[23] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.

[24] D. Liu, X. Yang, D. Wang, and Q. Wei, "Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1372–1385, July. 2015.

[25] D. Wang, D. Liu, Q. Zhang, and D. Zhao, "Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1544–1555, Nov. 2016.

[26] R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for infinite-horizon approximate optimal tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 753–758, Mar. 2017.

[27] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, pp. 412–442, Dec. 1990.

[28] Z. Kan, J. R. Klotz, E. Doucette, J. Shea, and W. E. Dixon, "Decentralized rendezvous of nonholonomic robots with sensing and connectivity constraints," *ASME J. Dyn. Syst., Meas., Control*, vol. 139, no. 2, 2017, Art. no. 024501.

[29] T.-H. Cheng, Z. Kan, J. A. Rosenfeld, and W. E. Dixon, "Decentralized formation control with connectivity maintenance and collision avoidance under limited and intermittent sensing," in *Proc. Amer. Control Conf.*, 2014, pp. 3201–3206.

[30] E. J. Rodríguez-Seda, C. Tang, M. W. Spong, and D. M. Stipanović, "Trajectory tracking with collision avoidance for nonholonomic vehicles with acceleration constraints and limited sensing," *Int. J. Robot. Res.*, vol. 33, no. 12, pp. 1569–1592, 2014.

[31] E. J. Rodríguez-Seda, D. M. Stipanović, and M. W. Spong, "Guaranteed collision avoidance for autonomous systems with acceleration constraints and sensing uncertainties," *J. Optim. Theory Appl.*, vol. 168, no. 3, pp. 1014–1038, 2016.

[32] E. J. Rodríguez-Seda and M. W. Spong, "Guaranteed safe motion of multiple lagrangian systems with limited actuation," in *Proc. IEEE Conf. Decis. Control*, 2012, pp. 2773–2780.

[33] D. M. Stipanović, P. F. Hokayem, M. W. Spong, and D. D. Šiljak, "Cooperative avoidance control for multiagent systems," *J. Dyn. Syst., Meas., Control*, vol. 129, no. 5, pp. 699–707, 2007.

[34] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. Int. Conf. Hybrid Syst., Comput. Control*, 2015, pp. 11–20.

[35] J. Ding, E. Li, H. Huang, and C. J. Tomlin, "Reachability-based synthesis of feedback policies for motion planning under bounded disturbances," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2160–2165.

[36] R. Takei, H. Huang, J. Ding, and C. J. Tomlin, "Time-optimal multi-stage motion planning with guaranteed collision avoidance via an open-loop game formulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 323–329.

[37] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Trans. Syst., Man, Cybern.—Part A: Syst. Human*, vol. 28, no. 5, pp. 562–574, Sep. 1998.

[38] A. Chakravarthy and D. Ghose, "Generalization of the collision cone approach for motion safety in 3-D environments," *Auton. Robots*, vol. 32, no. 3, pp. 243–266, 2012.

[39] A. Chakravarthy and D. Ghose, "Collision cones for quadric surfaces in n-dimensions," *IEEE Robot. Autom. Lett.*, vol. 3, no. 1, pp. 604–611, Jan. 2018.

[40] V. Sunkara, A. Chakravarthy, and D. Ghose, "Collision avoidance of arbitrarily shaped deforming objects using collision cones," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2156–2163, Apr. 2019.

[41] A. Ferrara and C. Vecchio, "Second order sliding mode control of vehicles with distributed collision avoidance capabilities," *Mechatronics*, vol. 19, no. 4, pp. 471–477, 2009.

[42] C. E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 6375–382, Apr. 2019.

[43] M. M. G. Ardakani, B. Olofsson, A. Robertsson, and R. Johansson, "Model predictive control for real-time point-to-point trajectory generation," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 972–983, Apr. 2019.

[44] Z. Huang, D. Chu, C. Wu, and Y. He, "Path planning and cooperative control for automated vehicle platoon using hybrid automata," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 959–974, Mar. 2019.

[45] H. Wang, Y. Huang, A. Khajepour, Y. Rasekhipour, Y. Zhang, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3313–3323, Sep. 2019.

[46] T. Dierks, B. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Netw.*, vol. 22, no. 5/6, pp. 851–860, 2009.

[47] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

[48] S. Lyshevski, "Optimal control of nonlinear continuous-time systems: Design of bounded controllers via generalized nonquadratic functionals," in *Proc. Amer. Control Conf.*, 1998, pp. 205–209.

[49] P. Walters, R. Kamalapurkar, and W. E. Dixon, "Approximate optimal online continuous-time path-planner with static obstacle avoidance," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 650–655.

[50] J. A. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "The state following approximation method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1716–1730, Jun. 2019.

[51] P. Deptula, J. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "Approximate dynamic programming: Combining regional and local state following approximations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2154–2166, Jun. 2018.

[52] P. Walters, R. Kamalapurkar, F. Voight, E. Schwartz, and W. E. Dixon, "Online approximate optimal station keeping of a marine craft in the presence of an irrotational current," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 486–496, Apr. 2018.

[53] P. Deptula, R. Licitra, J. A. Rosenfeld, and W. E. Dixon, "Online approximate optimal path-planner in the presence of mobile avoidance regions," in *Proc. Amer. Control Conf.*, 2018, pp. 2515–2520.
[54] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Cambridge, MA, USA: Birkhauser, 2003.
[55] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[56] R. Kamalapurkar, P. S. Walters, J. A. Rosenfeld, and W. E. Dixon, *Reinforcement Learning for Optimal Feedback Control: A Lyapunov-Based Approach*. Cham, Switzerland: Springer, 2018.
[57] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, Jan. 2015.
[58] F. L. Lewis, R. Selmic, and J. Campos, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia, PA, USA: SIAM, 2002.
[59] P. Deptula, Z. I. Bell, E. Doucette, J. W. Curtis, and W. E. Dixon, "Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with partial loss of control effectiveness," in *Proc. Amer. Control Conf.*, 2018, pp. 2521–2526.
[60] Z. Bell, P. Deptula, H.-Y. Chen, E. Doucette, and W. E. Dixon, "Velocity and path reconstruction of a moving object using a moving camera," in *Proc. Amer. Control Conf.*, 2018, pp. 5256–5261.
[61] R. Licitra, Z. I. Bell, E. Doucette, and W. E. Dixon, "Single agent indirect herding of multiple targets: A switched adaptive control approach," *IEEE Control Syst. Lett.*, vol. 2, no. 1, pp. 127–132, Jan. 2018.
[62] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Target tracking in the presence of intermittent measurements via motion model learning," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 805–819, Jun. 2018.
[63] M. Monajjemi, "bebop_autonomy library," 2015. [Online]. Available: http://bebop-autonomy.readthedocs.io
[64] Z. Kan, T. Yucelen, E. Doucette, and E. Pasiliao, "A finite-time consensus framework over time-varying graph topologies with temporal constraints," *J. Dyn. Syst., Meas., Control*, vol. 139, no. 7, 2017, Art. no. 071012.
[65] E. Arabi, T. Yucelen, B. C. Gruenwald, M. Fravolini, S. Balakrishnan, and N. T. Nguyen, "A neuroadaptive architecture for model reference control of uncertain dynamical systems with performance guarantees," *Syst. Control Lett.*, vol. 125, pp. 37–44, 2019.
[66] I. Papusha, J. Fu, U. Topcu, and R. M. Murray, "Automata theory meets approximate dynamic programming: Optimal control with temporal logic constraints," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 434–440.
[67] Q. Gao, D. Hajinezhad, Y. Zhang, Y. Kantaros, and M. M. Zavlanos, "Reduced variance deep reinforcement learning with temporal logic specifications," in *Proc. ACM/IEEE Int. Conf. Cyber Phys. Syst.*, 2019, pp. 237–248.
[68] Y. Yang, Y. Yin, W. He, K. G. Vamvoudakis, H. Modares, and D. C. Wunsch, "Safety-aware reinforcement learning framework with an actor-critic-barrier structure," in *Proc. IEEE Amer. Control Conf.*, 2019, pp. 2352–2358.

Patryk Deptula received the B.Sc. degree in mechanical engineering (major) and mathematics (minor) from Central Connecticut State University, New Britain, CT, USA, in 2014. He received the M.S. and Ph.D. degrees in mechanical engineering from the Department of Mechanical and Aerospace Engineering, the University of Florida, Gainesville, FL, USA, in 2017 and 2019, respectively.

In 2019, he joined The Charles Stark Draper Laboratory, Inc., Cambridge, MA, USA. His current research interests include learning-based and adaptive control, multiagent systems, human–machine interaction, vision-based navigation and control, biomedical systems, biomechanics, and robotics applied to a variety of fields.

Hsi-Yuan (Steven) Chen received the Ph.D. degree in mechanical engineering from the University of Florida, Gainesville, FL, USA, in 2018.

In 2019, he joined Amazon Robotics, North Reading, MA, USA, where he is an Applied Scientist with a focus on autonomous mobility. His main research interests include the development and application of Lyapunov-based state estimation and control methods for autonomous vehicles.

Ryan A. Licitra received the B.S., M.S., and doctoral degrees in mechanical engineering from the University of Florida, Gainesville, FL, USA, in 2013, 2015, and 2017, respectively.

He joined the Nonlinear Controls and Robotics group in 2014 to pursue his doctoral research. He is currently with the University of Florida Research and Engineering Education Facility to facilitate extensive collaboration with Air Force Research Laboratory (AFRL) research staff. His research interests include the study of network control systems, including systems with changing topologies, and multitasked agents.

Joel A. Rosenfeld received the Ph.D. in mathematics from the University of Florida, Gainesville, FL, USA, in 2013.

He joined the Nonlinear Controls and Robotics group in 2013 as a Postdoctoral Researcher with the Department of Mechanical and Aerospace Engineering, University of Florida, focusing on approximation problems in control theory. In 2017, he joined the VeriVital Laboratory, Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA, as a Postdoctoral Researcher and later became a Senior Research Scientist Engineer with the Institute for the Study of Software Integrated Systems (ISIS), Vanderbilt University. He is currently an Assistant Professor with the Department of Mathematics and Statistics, University of South Florida. His research interest includes data science problems in dynamical systems theory with an emphasis on kernel techniques.

Warren E. Dixon (F'16) received the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, Clemson University, Clemson, CS, USA, in 2000.

He was an Eugene P. Wigner Fellow with Oak Ridge National Laboratory (ORNL). In 2004, he joined the Mechanical and Aerospace Engineering Department, University of Florida. He has authored or coauthored three books, an edited collection, 13 chapters, and over 130 journals, and 230 conference papers. His main research interest include the development and application of Lyapunov-based control techniques for uncertain nonlinear systems.

Dr. Dixon was the recipient of the 2001 Oak Ridge National Laboratory (ORNL) Early Career Award for Engineering Achievement in 2001, the Department of Energy Outstanding Mentor Award in 2004, the IEEE Robotics and Automation Society (RAS) Early Academic Career Award in 2006, the American Automatic Control Council (AACC) O. Hugo Schuck (Best Paper) Award in 2009 and 2015, the National Science Foundation (NSF) CAREER Award during 2006 to 2011, the American Society of Mechanical Engineers (ASME) Dynamics Systems and Control Division Outstanding Young Investigator Award in 2011, the University of Florida College of Engineering Doctoral Dissertation Mentoring Award during 2012 to 2013, and the Fred Ellersick Award for Best Overall MILCOM Paper in 2013. He is an ASME Fellow, an IEEE Control Systems Society (CSS) Distinguished Lecturer, and served as the Director of Operations for the Executive Committee of the IEEE CSS Board of Governors (2012–2015). He currently serves as a member of the U.S. Air Force Science Advisory Board. He is currently or formerly an Associate Editor for *ASME Journal of Dynamic Systems, Measurement and Control*, *Automatica*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: PART B CYBERNETICS, and the *International Journal of Robust and Nonlinear Control*.