Approximate Optimal Influence Over an Agent Through an Uncertain Interaction Dynamic [⋆]

Patryk Deptula [a], Zachary I. Bell [b], Federico M. Zegers [b],
Ryan A. Licitra [b], and Warren E. Dixon [b]

[a] *The Charles Stark Draper Laboratory, Inc., Cambridge, MA, USA*

[b] *Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, USA*

Abstract

An approximate optimal indirect regulation problem is considered for two nonlinear uncertain agents. An influencing agent is tasked with optimally intercepting and directing a roaming agent to a goal location. The roaming agent is not directly controlled by the influencing agent, but instead moves based on some uncertain interaction dynamic. To overcome this challenge, a virtual controller is designed to yield a desired influence on the roaming agent. In addition, an approximate dynamic programming (ADP) strategy is used to develop an approximate optimal solution to the optimal control problem using a computationally efficient function approximation method. Because system uncertainties are considered in both agents, a data-based parameter identification method called integral concurrent learning (ICL) is used to identify uncertain dynamics. A Lyapunov-based stability analysis is performed which proves the closed-loop pursuing and roaming agent systems are uniformly ultimately bounded (UUB). Simulation and experimental results are provided to demonstrate the performance of the developed method.

1 Introduction

Multi-agent systems may be cooperative, where the agents aim to reach the same goal, or non-cooperative, where the agents aim to reach different goals such as pursuit-evasion or reach-avoid games. Game theory is concerned with the analysis of strategies that players can take based on particular conditions [1]. Pursuit-evasion games are problems motivated by predator-prey scenarios, where strategies for either evading or pursing agents are calculated using differential game theory (cf., [2–14] and references therein). Several different approaches have been considered in pursuit-evasion games. For instance, works such as [9] consider multi-player pursuit-evasion capture conditions, and cooperative control strategies are calculated for pursuing agents to capture evading agents. Results such as [10] develop escape strategies for evading agents using mathematical frameworks based on Apollonius circles. Results such as [11] consider pursing agents with uncertain speeds and calculate the strategies for the pursuing agent, while escape strategies are selected for evading agents based on how pursuing agents are approaching them. Results such as [8] and [15] consider reach-avoid games to determine strategies (i.e., winning and losing regions) computed numerically using level set methods to solve the Hamilton Jacobi Isaacs (HJI) equation. While the aforementioned and related literature provide foundational strategies for pursuit-evasion games, most results assume simple or known dynamics and generally only consider the problem in two-dimensions. In addition, assumptions about the knowledge of the opposing players strategies are required to gain an advantage. Moreover, such games are only focused at finding strategies for either capturing or evading aspects of the game.

While traditional pursuit-evasion problems focus on either the trapping or fleeing aspects of the game, a different class of problems, called herding, focus on directing uncontrolled agents to a goal location and have been investigated in results such as [7,16–22]. Unlike pursuit-evasion problems, in indirect herding problems, the influencing agent must pursue a roaming agent while also escorting it to a desired location through an inter-agent interaction. The indirectly influenced agent is called roam-

[⋆] This research is supported in part by NSF award 1509516, Office of Naval Research Grant N00014-13-1-0151, and AFOSR award number FA9550-19-1-0169. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of sponsoring agencies.

Email addresses: pdeptula@draper.com (Patryk Deptula), bellz121@ufl.edu (Zachary I. Bell), fredzeg@ufl.edu (Federico M. Zegers), rlicitra@ufl.edu (Ryan A. Licitra), wdixon@ufl.edu (Warren E. Dixon).

ing instead of evading since it does not necessarily pursue an optimal strategy or seek to escape from the pursing agent. Such problems are inspired by behaviors and practical considerations often seen in nature and have inspire works such as [7,16–22] to leverage such behaviors in controlling autonomous systems. Motivated by such results, the authors in [19] and [20] approach the indirect regulation (also known as indirect herding) problem via a switched-systems approach, where the influencing agent switches between target agents. In [19], a robust sliding mode approach is used to compensate for worst-case uncertainties of the target agent dynamics. Compared to [19], the result in [20] uses an adaptive control approach, where a data-based parameter estimation method called integral concurrent learning (ICL) is used to learn the linearly parametrized (LP) target dynamics by storing input-output data (cf., [23]). In [21] and [22], a forcing function based on geometric constraints is used to develop a controller for a group of agents that regulate other agents indirectly by forming an arc and forcing the targets to the desired location. Although major advancements have been made in multi-player problems, results such as [9–11] generally consider point mass systems and know the form of the agent dynamics, while results such as [19–22] rely on explicitly designed controllers for the influencing agent based on the target dynamics and do not consider optimality.

Herding-based problems using dynamic programming (DP) to find optimal strategies for pursuing agents to capture and regulate evading/roaming agents to goal locations have been investigated. Results such as [7,17,18] compute optimal policies for pursuing agents regulating multiple evading agents with known point mass dynamics. Specifically, the result in [17] uses the Sparse Nonlinear Optimizer (SNOPT) algorithm in [24,25] to compute numerical solutions offline. The works in [7,18] use DP and shortest-path algorithms over a finite graph to determine offline optimal policies that a pursing agent can take to drive an evading agent to a goal location. Although results such as [7,17,18] provide in-roads for optimal herding, the computational complexity associated with a large number of states renders the problems infeasible for online implementation, the agents are assumed to take simple one-step discrete actions over a finite grid, and the dynamics of the agents are generally known. Such results require numerical solutions, which can be computationally expensive for high dimensional systems, and do not consider system uncertainties; hence, the use of parametric methods, such as neural-networks (NNs), to yield computationally efficient approximate optimal controllers online is motivated.

Approximate dynamic programming (ADP) is a popular method which has been successfully used in deterministic autonomous control-affine systems to develop approximately optimal solutions [26–29]. ADP approximates the solution to the Hamilton Jacobi Bellman (HJB) equation, called the value function, and is used to compute the online forward-in-time optimal policy via NNs. ADP approaches such as [27,30,31] approximate the value function by using stationary basis functions representing the entire operating domain, which can be computationally expensive. Specifically, in the absence of domain knowledge, a large number of basis functions, and hence, a large number of unknown parameters, is required for value function approximation. However, by using computationally efficient state-following (StaF) kernel basis functions [32–34] for local approximation of the value function around the current state, the number of basis functions required for sufficient value function approximation can be reduced.

Numerous results have been developed using differential game formulations using ADP (cf., [27,35–40]). However, all of these results solve the multi-player problem by generating controllers and directly controlling each agent. Exceptions include the innovative work in [31] and [41]. Specifically, an ADP-based backstepping approach is developed in [31] and [41] for a class of known strict-feedback nonlinear systems containing a one-dimensional input. In [41], for each individual step of the backstepping approach, a virtual control is obtained using the Sontag formula [42] which is equivalent to the optimal control. While in [31], a quadratic term is injected into the optimal value function of each backstepping instance, and the mismatch between the quadratic term and unknown optimal value function is approximated using NNs. Despite such progress, results such as [31] and [41] both assume exact model knowledge of the agent dynamics and require the strict persistence of excitation (PE) condition to be satisfied.

Unlike typical pursuit-evasion problems or the aforementioned ADP results, this work explores an approximately optimal learning-based indirect regulation problem for two agents. The developed approach is model-based and an actor-critic-identifier [26] strategy is employed, where the adaptive estimates must converge to the actual parameters to yield the optimal policy. To alleviate the need for physical excitation of the system to satisfy the PE condition, the work in this paper uses ICL to identify both the pursuing and roaming agent uncertainties. Specifically, the contribution of this result is to approximately optimally regulate an agent to a goal location through an uncertain interaction with a controlled pursuing agent. The approximate optimal pursuer does not require exact model knowledge of either the agent dynamic or the interaction dynamic, and does not assume a policy for the roaming agent. The difficulty in the developed approach stems from the problem definition and resulting stability analysis. Moreover, unlike results such as [19] and [20], the drift dynamics of both agents are assumed to be unknown and an approximately optimal control strategy is developed. Compared to results such as [9–11], the agent dynamics in this paper are nonlinear and uncertain, where the developed technique does not rely on numerical methods and can produce a closed-form policy. Compared to results such as [7,17,18], the policy for the influencing agent is calculated online and does not use one-step discrete actions, and while the results in [7,17,18] tend to be computationally inefficient due to the curse of dimensionality associated with DP, the strategy in this work uses the computationally efficient StaF function approximation approach in a continuous space problem formulation. Furthermore, unlike the preliminary result in [43], this work includes redefined error signals. In addition, compared to the preliminary results in [43] and many works in ADP, this result includes an experimental study, where the developed controller is implemented online in real time on a quadcopter testbed.

Roadmap

The developed indirect approach in this work is based on the ADP framework. Hence, in Section 2 (Problem Formulation), the problem is introduced, where the primary objective for the influencing agent is to intercept and regulate a roaming agent to a goal location. However, since there is no direct input for the roaming agent, the influencing agent must direct the roaming agent by the use of an uncertain interaction dynamic. Moreover, the influencing agent state may be non-affine in the roaming agent dynamics; hence, a virtual state is introduced, whose time-derivative is the virtual input. The roaming agent is regulated to the goal location by regulating the virtual state. Section 2.1 (Optimal Control Development) defines the optimal control problem, where the influencing agent's input is designed to optimally minimize the mismatch between the influencing agent's actual versus virtual state. The solution to this problem is then approximated in Section 3 (Approximate Optimal Control), where the approximate optimal control is discussed. Specifically, in Section 3.1 (System Identification) system identification is introduced since both the influencing and roaming agent dynamics are uncertain. The ICL update laws are shown along with supporting assumptions, and Theorem 1 is provided, which shows that the estimated system weights exponentially converge to a neighborhood of the true weights. Using the estimated (identified) system uncertainties, the optimal value function is approximated in Section 3.2 (Value Function Approximation). The feedback error, called the Bellman error (BE), is introduced and used to adjust the actor and critic weights online. Using the BE from Section 3.2, the actor and critic weight update laws are introduced in Section 3.3 (Online Learning) along with specific assumptions regarding the regressors used for learning. Closed-loop stability is then shown using a Lyapunov-based approach in Section 4 (Stability Analysis), which shows uniformly ultimately bounded stability of the closed-loop system. This stability result is then validated via a simulation in Section 5 (Simulation). An experiment is included in Section 6 (Experiment), which demonstrates the performance of the developed strategy in real-time. Finally, Section 7 (Conclusion) summarizes the developed approach and provides possibilities for future work. Figure 1 illustrates the actor-critic-identifier architecture.

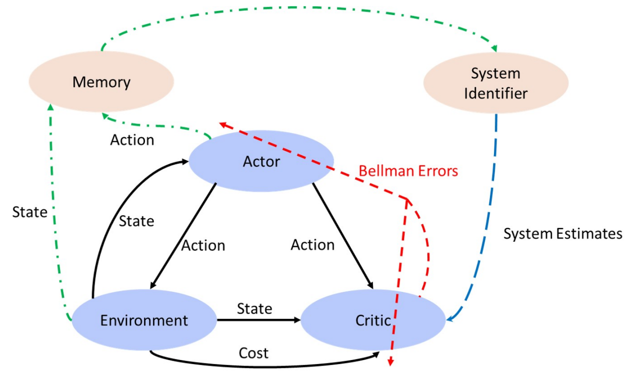


Figure 1. The state is measured from the environment, which is both saved in memory and passed to the critic estimator. Based on the actor's response, the critic uses the state, action, and cost to generate a new BE, which updates the critic and actor weights. As new state information and action information is produced, it is saved in memory and provided to the system identifier to perform system identification. The estimated system is provided to the critic to generate new actions. The system identifier does not have copies of the critic. It only has a copy of the actor output, samples of the system state, and structure of the policy.

Notation

In the following development, $\mathbb{R}$ denotes the set of real numbers while $\mathbb{Z}$ denotes the set of integers. The sets of numbers greater than or equal $a \in \mathbb{R}$ and strictly greater than a, are denoted by the subscript $\geq a$ and $> a$, respectively. For scalars $n, m \in \mathbb{Z}_{>0}$, the sets of real n-vectors and $n \times m$ matrices are denoted by $\mathbb{R}^n$ and $\mathbb{R}^{n \times m}$, respectively. The $n \times n$ identity matrix and the $m \times n$ zero matrices are denoted by I_n and $0_{m \times n}$, respectively. In addition, the j-dimensional column vector and $n \times m$ matrix of ones are denoted by 1_j and $1_{n \times m}$, respectively. The partial derivative of k with respect to the state x is denoted by $\nabla k(x, y, \ldots)$, the transpose of a matrix or vector is denoted by $(\cdot)^T$, and the trace of a square matrix is denoted as $\mathrm{tr}\,(\cdot)$. The vectorization operator of a matrix $A = [a_1,\ a_2, \ldots, a_m] \in \mathbb{R}^{n \times m}$ is denoted by $\mathrm{vec}\,(A) \triangleq \left[a_1^T,\ a_2^T, \ldots, a_m^T\right]^T$, where $a_i \in \mathbb{R}^n$ denotes the i^{th}-column of the matrix A. The notation $\|(\cdot)\|$ is defined as $\|d\| \triangleq \sup_{\xi \in \zeta} \|d\,(\xi)\|$, for some continuous function $h : \mathbb{R}^n \to \mathbb{R}^k$ and bounded set $\zeta \subseteq \mathbb{R}^n$. The notation $U\,[a, b]\, 1_{n \times m}$ denotes an $n \times m$-dimensional matrix, where each entry is selected from a uniform distribution on $[a, b]$. Finally, $\lambda_{min}\,\{\cdot\}$ and $\lambda_{max}\,\{\cdot\}$ denote the minimum and maximum eigenvalue, respectively.

2 Problem Formulation

In this section, the control problem is introduced as discussed in the roadmap in Section 1. In the subsequent development, the goal is to regulate a roaming agent to a desired user-defined goal location. [1] However, the roaming agent may not know where the goal location is or may not be cooperating to go there. The influencing agent knows the goal location, and simultaneously is tasked to optimally intercept and escort the roaming agent through an interaction dynamic [19, 20, 43]. Motivated by behaviors seen in nature, where the dynamics between predator and prey are dictated based on both agents' states, or in pursuit evasion problems, where the pursuer and evader dynamics are also coupled, consider a roaming agent governed by the drift dynamics

$$\dot{z}(t) = f(z(t), \eta(t)), \tag{1}$$

where $z : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is the roaming agent's state, $\eta : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ denotes the influencing agent's state, $t_0 \in \mathbb{R}_{\geq 0}$ is the initial time, and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is an uncertain locally Lipschitz function. The dynamics in (1) are not directly controllable; however, (1) can be influenced through interaction with the controlled pursuing agent governed by the uncertain dynamics

$$\dot{\eta}(t) = h(z(t), \eta(t)) + g(\eta(t)) u(t), \tag{2}$$

where $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is an unknown locally Lipschitz function representing the influencing agent drift dynamics, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m_\eta}$ is the known control effectiveness matrix, and $u : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{m_\eta}$ is the influencing agent's control input.

The form of the dynamics in (1) and (2) stem from behavior where the roaming agent tries to flee from pursuing the agent; hence it's dynamics are also dependent on the pursuer agent's state. The form of the dynamics in (2) are motivated by scenarios where the pursing agent may also be dependent on the roaming agent's state. For instance, such dynamics can model scenarios such as crowd control, where a controlling person (i.e, safety officer) may want to motivate another person to go to some desired location. Their respective dynamics are based on the coupled states, but in such a scenario it can be assumed that the officer dictates the situation (i.e, produces the control policy) to achieve the goal. Likewise, in nature, the predator will pursue some action to catch a prey, while the prey executes a responsive action to flee from the predator.

Assumption 1 *There exist class $\mathcal{K}$ functions $\overline{\alpha}_1, \overline{\alpha}_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that the uncertain dynamics in (1) can be bounded as $\|f(z(t), \eta(t))\| \leq \overline{\alpha}_1(\|z(t) - \eta(t)\|) + \overline{\alpha}_2(\|z(t) - z_g\|)$, where $z_g \in \mathbb{R}^n$ is a fixed goal location.*

Remark 1 *Assumption 1 indicates that the dynamics of the roaming agent in (1) depend on the distance between the influencing and roaming agents and the distance between the roaming agent and the goal location. The roaming agent dynamics in results such as [19, 20], can be shown to satisfy Assumption 1.*

Assumption 2 *The control effectiveness matrix $g(\eta)$ is bounded and full column rank for all $\eta \in \mathbb{R}^n$, and $g^+ : \mathbb{R}^n \to \mathbb{R}^{m_\eta \times n}$ is a bounded and locally Lipschitz pseudo inverse defined as $g^+ \triangleq \left(g^T g\right)^{-1} g^T$. [44].*

Remark 2 *Assumption 2 requires the control effectiveness matrix to be full column rank. There is a large class of systems which satisfy this assumption, such as fully actuated Euler-Lagrange systems with invertible inertia matrices [38, 44, 45].*

To quantify the objective, a regulation error $e_z : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is defined as

$$e_z(t) \triangleq z(t) - z_g. \tag{3}$$

Additional error system development is motivated by backstepping approaches, where the agent control input is designed based on a unique error system development that requires both the influencing and roaming agent errors to converge to the goal. Specifically, an auxiliary error, denoted by $e_\eta : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$, is defined as

$$e_\eta(t) \triangleq \eta(t) - \eta_d(t), \tag{4}$$

where $\eta_d : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is a desired virtual state. Because the influencing agent's state $\eta(t)$ may be non-affine in the roaming agent dynamics in (1), the aim of the virtual state $\eta_d(t)$ is to minimize the regulation error in (3). To quantify this aspect, another auxiliary error, denoted by $e_d : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$, is defined as

$$e_d(t) \triangleq \eta_d(t) - z_g - k_d e_z(t), \tag{5}$$

where $k_d \in \mathbb{R}$ is positive constant control gain, which is generally selected as $k_d \geq 1$. The error signals in (4) and (5) have been modified from the preliminary result in [43], and resemble those of backstepping approaches such as in [20]; however, compared to [20], optimality is considered for the overall system in this result. The virtual state $\eta_d(t)$ is injected into (5) with the goal of regulating $e_d(t)$. Based on (5) and the subsequent analysis, the time-derivative of $\eta_d(t)$ is designed as

$$\dot{\eta}_d(t) \triangleq \mu_d(t), \tag{6}$$

[1] The influencing agent in this work is synonymous with predator, pursuing, or herding agents, while the roaming agent is synonymous with prey, evading, or target agents in works such as [8–11, 15, 19, 20].

where $\mu_d : \mathbb{R}_{\geq t_0} \to \mathbb{R}^n$ is a subsequently designed virtual input.[2] Moreover, it will be shown in Theorem 2 that the errors $e_z(t)$, $e_d(t)$, and $e_\eta(t)$, and virtual controller $\mu_d(t)$ converge to a neighborhood containing the origin. Hence, the virtual state $\eta_d(t)$ converges to a region of the desired state z_g, implying that the roaming agent is regulated to a neighborhood of the desired location.

After taking the time-derivative of (5) and using (1) and (3)-(6), the error dynamics for $e_d(t)$ are $\dot{e}_d(t) = -k_d f(z(t), \eta(t)) + \mu_d(t)$. To determine the error dynamics for $e_\eta(t)$, (2) and (6) are substituted into the time-derivative of (4) to obtain

$$\begin{aligned}\dot{e}_\eta(t) &= h(z(t), \eta(t)) + g(\eta(t))\mu_\eta(t) \\ &\quad + g(\eta(t))u_d(t) - \mu_d(t),\end{aligned} \tag{7}$$

where $\mu_\eta(t) : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{m_\eta}$ is defined as $\mu_\eta(t) \triangleq u(t) - u_d(t)$, and $u_d(t) : \mathbb{R}_{\geq t_0} \to \mathbb{R}^{m_\eta}$ denotes a desired input. Based on (7) and the subsequent stability analysis, the desired input is designed as

$$u_d(t) \triangleq g(\eta_d(t))^+ (\mu_d(t) - h(z(t), \eta_d(t))). \tag{8}$$

Substituting (8) into (7) yields the following closed-loop system $\dot{e}_\eta(t) = h(z(t), \eta(t)) + g(\eta(t))\mu_\eta(t) - g(\eta(t))g^+(\eta_d(t))h(z(t), \eta_d(t)) + (g(\eta(t))g^+(\eta_d(t)) - I_n)\mu_d(t)$. To formulate the optimal control problem such that the errors in (3)-(5) are minimized, the influencing and roaming agent states are transformed. To facilitate this transformation, let $x(t) \triangleq \left[e_z^T(t), e_d^T(t), e_\eta^T(t)\right]^T$ and $x_d(t) \triangleq \left[e_z^T(t), e_d^T(t), 0_{1\times n}\right]^T$ denote the concatenated error state and desired concatenated state, respectively. In addition, we define the mappings $s_1, s_2 : \mathbb{R}^{3n} \to \mathbb{R}^n$ as $s_1(x(t)) \triangleq e_z(t) + z_g$, and $s_2(x(t)) \triangleq e_\eta(t) + e_d(t) + k_d e_z(t) + z_g$. Using (3)-(5), the roaming and influencing agent states are represented as $z(t) = s_1(x(t))$ and $\eta(t) = s_2(x(t))$, respectively. Using the mappings s_1 and s_2, the bounds in Assumption 1 can be represented such as $\|f(s_1(x(t)), s_2(x(t)))\| \leq \overline{\alpha}_1(\|(k_d - 1)e_z(t) + e_\eta(t) + e_d(t)\|) + \overline{\alpha}_2(\|e_z(t)\|)$. Hence, if $e_z(t), e_\eta(t), e_d(t) \to 0$ then $\|f(s_1(x(t)), s_2(x(t)))\| \leq \overline{\alpha}_1(0) + \overline{\alpha}_2(0)$.

Using these relationships, a composite autonomous error system can be written as

$$\dot{x}(t) = F(x(t)) + G(x(t))\mu(t), \tag{9}$$

where $\mu(t) \triangleq \left[\mu_\eta^T(t)\ \mu_d^T(t)\right]^T \in \mathbb{R}^m$ is the total vector of policies with $m = m_\eta + n$, while $F : \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ and $G : \mathbb{R}^{3n} \to \mathbb{R}^{3n \times m}$ are defined as

$$F(x(t)) \triangleq \begin{bmatrix} f(s_1(x(t)), s_2(x(t))), \\ -k_d f(s_1(x(t)), s_2(x(t))), \\ h(s_1(x(t)), s_2(x(t))) - F_{sd}(x(t)), \end{bmatrix},$$

and $$G(x(t)) \triangleq \begin{bmatrix} 0_{n\times m_\eta}, & 0_{n\times n}, \\ 0_{n\times m_\eta}, & I_n, \\ g(s_2(x(t))), & G_{sd}(x(t)), \end{bmatrix},$$

where $F_{sd}(x(t)) \triangleq g(s_2(x(t)))g^+(s_2(x_d(t))) \cdot h(s_1(x(t)), s_2(x_d(t)))$, and $G_{sd}(x(t)) \triangleq g(s_2(x(t)))g(s_2(x_d(t)))^+ - I_n$.

The goal is to ensure the roaming agent is driven to z_g. Hence, if the three errors $e_z(t), e_\eta(t), e_d(t)$ converge to 0, the total system achieves the goal such that $z(t) \to z_g$, $\eta(t) \to \eta_d(t)$, and $\eta_d(t) \to z_g$ by the use of (3)-(5). This implies that $\eta(t) \to z_g$.

2.1 Optimal Control Development

In this section, the optimal control is developed as specified in Section 1. Specifically, an analytical controller is derived from the cost function in (10).[3] Given (9), the goal is to design controllers $\mu_d(t)$ and $\mu_\eta(t)$ to minimize the cost function $J : \mathbb{R}^{3n} \times \mathbb{R}^m \times \mathbb{R}_{t_0} \to \mathbb{R}_{\geq 0}$ defined as

$$J(x, \mu) \triangleq \int_{t_0}^{\infty} r(x(\tau), \mu(\tau))\, d\tau, \tag{10}$$

subject to (9), where $r : \mathbb{R}^{3n} \times \mathbb{R}^m \to \mathbb{R}_{\geq 0}$ is the instantaneous cost defined as

$$r(x, \mu) \triangleq Q(x) + P(x) + \Psi(\mu). \tag{11}$$

In (11), $Q : \mathbb{R}^{3n} \to \mathbb{R}_{\geq 0}$ is a user-defined continuous positive-definite (PD) function (e.g., $x^T Q_x x$ where $Q_x \in \mathbb{R}^{3n\times 3n}$ is a PD matrix), which can be bounded as $\underline{q}\|x\|^2 \leq Q(x) \leq \overline{q}\|x\|^2$ for all $x \in \mathbb{R}^{3n}$ and $\underline{q}$, $\overline{q} \in \mathbb{R}_{>0}$. Furthermore, $\Psi(\mu) \triangleq \mu^T R \mu$, where $R = \operatorname{diag}\{R_\eta, R_d\}$ such that $R_\eta \in \mathbb{R}^{m_\eta \times m_\eta}$ and $R_d \in \mathbb{R}^{n\times n}$ are user-defined PD symmetric weighting matrices. In addition, $P : \mathbb{R}^{3n} \to \mathbb{R}$ is a user-defined continuous positive semi-definite (PSD) penalty function such that $P(x) = 0$ when $\|s_1(x) - s_2(x)\| \leq r_a(x)$ and $P(x) > 0$ when $\|s_1(x) - s_2(x)\| > r_a(x)$, where $r_a : \mathbb{R}^{3n} \to \mathbb{R}_{\geq 0}$ is a design parameter. Examples of functions that satisfy the conditions for $P(x)$ include $P(x) = e^{\frac{1}{2\alpha}(p_x(x))^2} - 1$, $P(x) =$

[2] Single integrator dynamics are used for the virtual state for simplicity. However, the virtual state can also evolve according to dynamics such as $\dot{\eta}_d(t) \triangleq -A_d \eta_d(t) + B_d \mu_d(t)$, where $A_d, B_d \in \mathbb{R}^{n\times n}$ are positive definite matrices.

[3] For notational brevity, unless otherwise specified, time dependence is suppressed in subsequent equations, trajectories, and definitions.

$\alpha \left(p_x \left(x\right)\right)^2$, $P\left(x\right) = \alpha \ln\left(\cosh\left(p_x\left(x\right)\right)^2\right)$, where $p_x\left(x\right) \triangleq \max\left\{0, \left\|\left(1-k_d\right)e_z - e_\eta - e_d\right\|^2 - r_a\left(x\right)^2\right\}$, and $\alpha \in \mathbb{R}_{>0}$. Piecewise continuous smooth functions that saturate at a constant may also be used.

Definition 1 *[31]: Let $\Omega \subseteq \mathbb{R}^n$ be a set containing the origin $x = 0$. A control policy $\mu\left(x\left(t\right)\right)$ is said to be admissible with respect to (9) in Ω, i.e., $\mu\left(x\left(t\right)\right) \in U\left(\Omega\right) \subset \mathbb{R}^m$, if $\mu\left(x\left(t\right)\right)$ is continuous in Ω with $\mu\left(0\right) = 0_{m\times 1}$, $\mu\left(x\left(t\right)\right)$ stabilizes (9) in Ω, and $V\left(x\left(t\right)\right) \triangleq J\left(x\left(t\right), \mu\left(x\left(t\right)\right)\right)$ is finite.*

The optimal value function, denoted by $V^* : \mathbb{R}^{3n} \to \mathbb{R}_{\geq 0}$, is expressed as

$$V^*\left(x\left(t\right)\right) = \inf_{\mu(\tau)\in U \left| \tau\in\mathbb{R}_{\geq t}\right.} \int_t^\infty r\left(x\left(\tau\right), \mu\left(\tau\right)\right) d\tau. \tag{12}$$

The HJB equation, which characterizes the optimal value function, is given by

$$\begin{aligned} 0 = {} & \nabla V^*\left(x\left(t\right)\right)\left(F\left(x\left(t\right)\right) + G\left(x\left(t\right)\right)\mu^*\left(x\left(t\right)\right)\right) \\ & + r\left(x\left(t\right), \mu^*\left(x\left(t\right)\right)\right), \end{aligned} \tag{13}$$

with $V^*\left(0\right) = 0$, where $\mu^* : \mathbb{R}^{3n} \to \mathbb{R}^m$ denotes the admissible optimal input policy, which is determined from (13) as

$$\mu^*\left(x\left(t\right)\right) = -\frac{1}{2}R^{-1}G\left(x\left(t\right)\right)^T \left(\nabla V^*\left(x\left(t\right)\right)\right)^T. \tag{14}$$

3 Approximate Optimal Control

This section discusses the approximate optimal formulation as briefed in Section 1. The optimal solution in (12)-(14) depends on the unknown value function and system dynamics. Hence, system identification (Section 3.1) is used to identify the dynamics, which is followed by value function approximation (Sections 3.2 and 3.3) to approximate the optimal solution. The implementation diagram is shown in Figure 1, where system identification and value function approximation are performed simultaneously. As mentioned in Section 1, data-based techniques are leveraged to implement the approximate controller online such that dithering signals do not need to be injected into the system to facilitate learning.

3.1 System Identification

The HJB equation in (13) and optimal controller in (14) require knowledge of both the drift dynamics $k_d f\left(z\left(t\right), \eta\left(t\right)\right)$ and $h\left(z\left(t\right), \eta\left(t\right)\right)$. Since these function are unknown, we approximately minimize the cost function in (10) while simultaneously learning these functions as shown in Figure 1. Various methods could be employed to learn the functions (cf., [23, 46–49]). The following is based on the ICL strategy in [23]. Using the universal function approximation property of single layer NNs [50–52], (1) and (2) can be represented as

$$\dot{\breve{x}}\left(t\right) = S\left(x\left(t\right)\right)\theta + \varepsilon\left(x\left(t\right)\right) + \breve{G}\left(x\left(t\right), u\left(t\right)\right), \tag{15}$$

where $\breve{x}\left(t\right) \triangleq \left[k_d z\left(t\right),\ \eta\left(t\right)\right]^T \in \mathbb{R}^{2\times n}$, $\theta \triangleq \begin{bmatrix}\theta_z^T\ \theta_\eta^T\end{bmatrix}^T \in \mathbb{R}^{p\times n}$, $\breve{G}\left(x\left(t\right), u\left(t\right)\right) \triangleq \begin{bmatrix}0_{n\times 1},\ g\left(x\left(t\right)\right)u\left(t\right)\end{bmatrix}^T \in \mathbb{R}^{2\times n}$, $S\left(x\left(t\right)\right) \triangleq \begin{bmatrix} S_z^T\left(x\left(t\right)\right) & 0_{1\times p_\eta} \\ 0_{1\times p_z} & S_\eta^T\left(x\left(t\right)\right)\end{bmatrix} \in \mathbb{R}^{2\times p}$, and $\varepsilon\left(x\left(t\right)\right) \triangleq \begin{bmatrix}\varepsilon_z\left(x\left(t\right)\right)\ \varepsilon_\eta\left(x\left(t\right)\right)\end{bmatrix}^T \in \mathbb{R}^{2\times n}$. In (15), $\theta_j \in \mathbb{R}^{p_j \times n}$ are the unknown weights, $S_j : \mathbb{R}^{3n} \to \mathbb{R}^{p_j}$ are the user-defined basis functions, and $\varepsilon_j : \mathbb{R}^{3n} \to \mathbb{R}^n$ is the function approximation error for $j = \{z, \eta\}$, and $p = p_z + p_\eta \in \mathbb{Z}_{>0}$ denotes the total number of rows of θ.[4] Moreover, if an exact basis is known for both agent dynamics, then $\varepsilon\left(x\left(t\right)\right) = 0_{2\times n}$. In addition, if $h\left(z\left(t\right), \eta\left(t\right)\right) = 0_{n\times 1}$ in (2), then the terms in (15) can be reduced to $\breve{x}\left(t\right) \triangleq \left[k_d z\left(t\right)\right]^T \in \mathbb{R}^{1\times n}$, $\breve{G}\left(x\left(t\right), u\left(t\right)\right) \triangleq \begin{bmatrix}0_{n\times 1}\end{bmatrix}^T \in \mathbb{R}^{1\times n}$, $\theta \triangleq \begin{bmatrix}\theta_z^T\end{bmatrix}^T \in \mathbb{R}^{p_z\times n}$, $S\left(x\right) \triangleq \begin{bmatrix}S_z^T\left(x\left(t\right)\right)\end{bmatrix} \in \mathbb{R}^{1\times p_z}$, and $\varepsilon\left(x\left(t\right)\right) \triangleq \begin{bmatrix}\varepsilon_z\left(x\left(t\right)\right)\end{bmatrix}^T \in \mathbb{R}^{1\times n}$, respectively.

Assumption 3 *There exist $\overline{\theta}$, $\overline{S}$, $\overline{\varepsilon} \in \mathbb{R}_{>0}$ such that $\left\|\theta\right\| \leq \overline{\theta}$, $\sup_{x\in\chi}\left\|S\left(x\right)\right\| \leq \overline{S}$, and $\sup_{x\in\chi}\left\|\varepsilon\left(x\right)\right\| \leq \overline{\varepsilon}$ in a compact set $\chi \subseteq \mathbb{R}^{3n}$. [30, 51].*

Based on the ICL strategy in [23], let $\Delta t_\theta \in \mathbb{R}_{>0}$ denote an integration time-window, where the integral of (15) at time $t_i \in \left[\Delta t_\theta, t\right]$ can be represented as $\breve{x}\left(t_i\right) - \breve{x}\left(t_i - \Delta t_\theta\right) = \mathcal{S}_i\theta + \mathcal{E}_i + \mathcal{G}_i$ such that $\mathcal{S}_i = \mathcal{S}\left(t_i\right) \triangleq \int_{t_i-\Delta t_\theta}^{t_i} S\left(x\left(\tau\right)\right)d\tau$, $\mathcal{E}_i = \mathcal{E}\left(t_i\right) \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \varepsilon\left(x\left(\tau\right)\right)d\tau$, and $\mathcal{G}_i = \mathcal{G}\left(t_i\right) \triangleq \int_{t_i-\Delta t_\theta}^{t_i} \breve{G}\left(x\left(\tau\right), u\left(\tau\right)\right)d\tau$. A least-squares based parameter estimate update law is designed as

$$\begin{aligned} \dot{\hat{\theta}}\left(t\right) = k_\theta\Gamma_\theta\left(t\right)\sum_{i=1}^{M}\mathcal{S}^T\left(t_i\right)\Big(&\breve{x}\left(t_i\right) - \breve{x}\left(t_i - \Delta t_\theta\right) \\ & - \mathcal{G}\left(t_i\right) - \mathcal{S}\left(t_i\right)\hat{\theta}\Big), \end{aligned} \tag{16}$$

$$\dot{\Gamma}_\theta\left(t\right) = \beta_\theta\Gamma_\theta\left(t\right) - k_\theta\Gamma_\theta\left(t\right)\sum_{i=1}^{M}\mathcal{S}^T\left(t_i\right)\mathcal{S}\left(t_i\right)\Gamma_\theta\left(t\right), \tag{17}$$

[4] The unknown weights θ_z and θ_η can be estimated independently using separate update laws. To alleviate redundancy, a combined approximation method is presented.

where $k_\theta, \beta_\theta \in \mathbb{R}_{>0}$ is an update gain and forgetting factor, respectively, and $M \in \mathbb{Z}_{>0}$ is the number of data points collected in the history stack.

Remark 3 *In general, M is specified a priori; however, it can also be determined online by checking Assumption 4 during runtime. In this manuscript, $M \geq \frac{p}{2}$ is selected to ensure enough data is gathered to facilitate learning, i.e., $\sum_{i=1}^{M} \mathcal{S}^T(t_i)\mathcal{S}(t_i)$ is full rank, where p denotes the total rows of θ.*

Assumption 4 *There exists $T_1 \in \mathbb{R}_{>0}$ such that $T_1 > \Delta t_\theta$ and a strictly positive constant $\lambda_1 \in \mathbb{R}_{>0}$ where $\lambda_1 I_p \leq \sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$, $\forall t \geq T_1$ [23].*

Provided $\lambda_{\min}\left\{\Gamma_\theta^{-1}(t_0)\right\} > 0$, and Assumption 4 is satisfied, Γ_θ satisfies $\underline{\Gamma}_\theta I_p \leq \Gamma_\theta(t) \leq \overline{\Gamma}_\theta I_p$, using similar arguments to [53, Corollary 4.3.2], where $\underline{\Gamma}_\theta, \overline{\Gamma}_\theta \in \mathbb{R}_{>0}$. Let $Z_\theta(t) = \operatorname{vec}\left(\tilde{\theta}(t)\right)$ denote a vector of parameter estimate errors with $\tilde{\theta}(t) \triangleq \theta - \hat{\theta}(t)$. Also let $V_\theta : \mathbb{R}^{np} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$ be a candidate Lyapunov functional defined as

$$V_\theta(Z_\theta, t) \triangleq \frac{1}{2}\operatorname{tr}\left(\tilde{\theta}^T \Gamma_\theta^{-1}(t)\,\tilde{\theta}\right), \tag{18}$$

which can be bounded as $\frac{1}{2\overline{\Gamma}_\theta}\|Z_\theta\|^2 \leq V_\theta(Z_\theta, t) \leq \frac{1}{2\underline{\Gamma}_\theta}\|Z_\theta\|^2$, for all $t \in \mathbb{R}_{\geq t_0}$ and $Z_\theta \in \mathbb{R}^{np}$.

Theorem 1 indicates that the estimation error $\tilde{\theta}$ remains bounded. Specifically, prior to Assumption 4 being satisfied, the estimation error is upper bounded by a constant based on the initial error and the NN approximation error. Once Assumption 4 is satisfied, the estimation error can be bounded by an exponential function. Stability is shown using a Lyapunov-based approach using the candidate Lyapunov functional in (18).

Theorem 1 *Provided Assumptions 3 and 4 are satisfied, the adaptive update laws in (16) and (17) ensure that the estimation error $\tilde{\theta}$ remains bounded for all $t \geq T_1$ such that*

$$\|Z_\theta(t)\| \leq c_\Gamma \sqrt{c_M e^{-\lambda_\theta(t-T_1)} + \left(1 - e^{-\lambda_\theta(t-T_1)}\right) c_B}, \tag{19}$$

where $c_\Gamma \triangleq \sqrt{\frac{\overline{\Gamma}_\theta}{\underline{\Gamma}_\theta}}$, $\lambda_\theta \triangleq \frac{k_\theta c_{\theta 2}\underline{\Gamma}_\theta}{2}$, $c_{\theta 1} \triangleq \frac{\beta_\theta}{k_\theta \overline{\Gamma}_\theta}$, $c_{\theta 2} \triangleq c_{\theta 1} + \lambda_1$, $c_M \triangleq \|Z_\theta(t_0)\|^2 + \frac{4\upsilon_1^2}{c_{\theta 1}^2}$, $c_B \triangleq \frac{4\upsilon_1^2}{c_{\theta 2}^2}$, and $\upsilon_1 \triangleq$ $\sup_{t \in \mathbb{R}_{\geq 0}}\left\|\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{E}_i\right\|$.

PROOF. Taking the time-derivative of (18), substituting in (16) and (17), using the fact that for $t < T_1$, $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i \geq 0$,

$$\dot{V}_\theta(Z_\theta, t) \leq -\frac{1}{2}k_\theta c_{\theta 1}\|Z_\theta\|^2 + k_\theta\|Z_\theta\|\upsilon_1. \tag{20}$$

Completing the squares, using the bounds on (18), and invoking the Comparison Lemma [54, Lemma 3.4] yields

$$\begin{aligned} V_\theta(Z_\theta(t), t) \leq{}& V_\theta(Z_\theta(t_0), t_0)\, e^{-\lambda_\theta \frac{k_\theta c_{\theta 1}\underline{\Gamma}_\theta}{2}(t-t_0)} \\ &+ \left(1 - e^{-\frac{k_\theta c_{\theta 1}\underline{\Gamma}_\theta}{2}(t-t_0)}\right)\frac{2\upsilon_1^2}{\underline{\Gamma}_\theta c_{\theta 1}^2}, \end{aligned} \tag{21}$$

for all $t \in [t_0, T_1)$. Then, $\|Z_\theta(t)\|^2 \leq \frac{\overline{\Gamma}_\theta}{\underline{\Gamma}_\theta}\left(\|Z_\theta(t_0)\|^2 + \frac{4\upsilon_1^2}{c_{\theta 1}^2}\right)$ follows for all $t \in \mathbb{R}_{\geq t_0}$.

After $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$ becomes full rank, (16), (17), and Assumption 4 are used in the time-derivative of (18) to yield

$$\dot{V}_\theta(Z_\theta, t) \leq -\frac{1}{4}k_\theta c_{\theta 2}\|Z_\theta\|^2 + \frac{k_\theta \upsilon_1^2}{c_{\theta 2}}, \tag{22}$$

for all $t \geq T_1$. Using the Comparison Lemma [54, Lemma 3.4], $\forall t \geq T_1$

$$\begin{aligned} V_\theta(Z_\theta(t), t) \leq{}& V_\theta(Z_\theta(T_1), T_1)\, e^{-\lambda_\theta(t-T_1)} \\ &+ \left(1 - e^{-\lambda_\theta(t-T_1)}\right)\frac{c_B}{2\underline{\Gamma}_\theta}. \end{aligned} \tag{23}$$

From (21), $V_\theta(Z_\theta(T_1), T_1) \leq V_\theta(Z_\theta(t_0), t_0)\, e^{-\lambda_\theta \frac{k_\theta c_{\theta 1}\underline{\Gamma}_\theta}{2}(t-t_0)} + \frac{2\upsilon_1^2}{\underline{\Gamma}_\theta c_{\theta 1}^2}$follows, and using (23) along with the bounds $\frac{1}{2\overline{\Gamma}_\theta}\|Z_\theta\|^2 \leq V_\theta(Z_\theta, t) \leq \frac{1}{2\underline{\Gamma}_\theta}\|Z_\theta\|^2$ results in (19). After $\sum_{i=1}^{M} \mathcal{S}_i^T \mathcal{S}_i$ becomes full rank, as $t \to \infty$, the residual bound in (23) (i.e., $\frac{c_B}{2\underline{\Gamma}_\theta}$) is smaller compared to the residual bound in (21) (i.e., $\frac{2\upsilon_1^2}{\underline{\Gamma}_\theta c_{\theta 1}^2}$), because c_B depends on $c_{\theta 2}$ and $c_{\theta 2} > c_{\theta 1}$. Moreover, while a single set of data can be collected, additional data selection and purging techniques such as in [47] and [55], can be used to select data that reduces the residuals even further. ■

Remark 4 *Theorem 1 shows that the estimation error $\tilde{\theta}$ remains bounded. The residual bound in (19) results from the function approximation errors $\varepsilon(x(t))$ in (15); however, if an exact basis is known in the neural network, then $\varepsilon(x(t)) = 0_{2\times n}$. Therefore, the bound in (19) can be reduced to $\|Z_\theta(t)\| \leq c_\Gamma\sqrt{c_M e^{-\lambda_\theta(t-T_1)}}$, which produces convergence to the origin. In addition, results such as [56–58] can be leveraged to possibly achieve asymptotic convergence to zero.*

3.2 Value Function Approximation

The value function $V^*(x)$, which is unknown, can be approximated via computationally efficient StaF kernels [32–34]. To facilitate the following development, let $\overline{B_r(x)}$ be defined as the closure of an open ball centered at $x \in \mathbb{R}^{3n}$ with radius $r \in \mathbb{R}_{>0}$. Using state-following centers, $c : \chi \to \chi^L$, centered around $x \in \chi$ such that $c(x) \in \left(\overline{B_r(x)}\right)^L$, the value function in (12) can be represented as

$$V^*(y) = W(x)^T \sigma(y, c(x)) + \epsilon_v(x, y), \tag{24}$$

where $y \triangleq \left[y_{e_z}^T, y_{e_d}^T, y_{e_\eta}^T\right]^T \in \overline{B_r(x)}$ represents a composite state vector in the neighborhood of x [32–34], and the states y_{e_z}, y_{e_d}, and y_{e_η} represent states in the neighborhoods of e_z, e_d, and e_η, respectively (i.e., $y_{e_z} \in \overline{B_r(e_z)}$, $y_{e_d} \in \overline{B_r(e_d)}$, and $y_{e_z} \in \overline{B_r(e_\eta)}$). In (24), $W : \chi \to \mathbb{R}^L$ is a vector of continuously differentiable ideal StaF weight functions, $\sigma : \chi \times \chi^L \to \mathbb{R}^L$ is a bounded vector of continuously differentiable nonlinear kernels, and $\epsilon_v : \chi \times \chi \to \mathbb{R}$ is a continuously differentiable function approximation error.

Since the ideal StaF weight $W(x)$ and function approximation error are unknown in (24), an approximate value function $\hat{V} : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \times \mathbb{R}^L \to \mathbb{R}$ is expressed as

$$\hat{V}\left(y, x, \hat{W}_c\right) = \hat{W}_c^T \sigma(y, c(x)), \tag{25}$$

and an approximate policy $\hat{\mu} : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \times \mathbb{R}^L \to \mathbb{R}^m$ is expressed as

$$\hat{\mu}\left(y, x, \hat{W}_a\right) = -\frac{1}{2} R^{-1} G(x)^T \nabla\sigma(y, c(x))^T \hat{W}_a, \tag{26}$$

where $\hat{W}_c, \hat{W}_a \in \mathbb{R}^L$ denote the critic and actor weight estimates, respectively. Substituting (25) and (26) along with the estimate $\hat{\theta}$ into (13) results in the BE $\delta : \mathbb{R}^{3n} \times \mathbb{R}^{3n} \times \mathbb{R}^L \times \mathbb{R}^L \times \mathbb{R}^{p \times n} \to \mathbb{R}$ given by

$$\begin{aligned}\delta\left(y, x, \hat{\theta}, \hat{W}_c, \hat{W}_a\right) &= Q(y) + P(y) + \Psi\left(\hat{\mu}\left(y, x, \hat{W}_a\right)\right)\\ &+\nabla\hat{V}\left(y, x, \hat{W}_c\right)\left(F_1\left(y, \hat{\theta}\right) - F_2\left(y, \hat{\theta}\right)\right)\\ &+\nabla\hat{V}\left(y, x, \hat{W}_c\right)\left(G(y)\,\hat{\mu}\left(y, x, \hat{W}_a\right)\right),\end{aligned} \tag{27}$$

where $F_1\left(y, \hat{\theta}\right) \triangleq \left[\left(\frac{1}{k_d}\hat{\theta}_z^T S_z(y)\right)^T, \left(-\hat{\theta}_z^T S_z(y)\right)^T, \left(\hat{\theta}_\eta^T S_\eta(y)\right)^T\right]^T$, $F_2\left(y, \hat{\theta}\right) \triangleq \left[0_{1\times 2n}, \left(g(y_\eta)\, g^+(y_{\eta_d})\, \hat{\theta}_\eta^T S_\eta\left(\left[y_{e_z}^T, y_{e_d}^T 0_{n\times 1}^T\right]^T\right)\right)^T\right]^T$. The controller for the influencing agent is $\hat{u}\left(y, x, \hat{\theta}, \hat{W}_a\right) \triangleq \hat{\mu}_\eta\left(y, x, \hat{W}_a\right) + \hat{u}_d\left(y, x, \hat{\theta}, \hat{W}_a\right)$, where $\hat{u}_d\left(y, x, \hat{\theta}, \hat{W}_a\right) \triangleq g^+(y_{n_d})\left(\hat{\mu}_d\left(y, x, \hat{W}_a\right) - S_\eta\left(\left[y_{e_z}^T,\ y_{e_d}^T,\ 0_{n\times 1}^T\right]^T\right)\hat{\theta}_\eta\right)$, and the approximate optimal terms $\hat{\mu}_\eta\left(y, x, \hat{W}_a\right)$ and $\hat{\mu}_d\left(y, x, \hat{W}_a\right)$ come from $\hat{\mu}\left(y, x, \hat{W}_a\right) \triangleq \left[\hat{\mu}_\eta^T\left(y, x, \hat{W}_a\right), \hat{\mu}_d^T\left(y, x, \hat{W}_a\right)\right]^T$ given in (26).

3.3 Online Learning

At each time instance $t \in \mathbb{R}_{\geq t_0}$, the BE in (27) is evaluated at the current state (i.e., $y = x(t)$), and the current parameter estimate $\hat{\theta}(t)$, $\hat{W}_c(t)$, and $\hat{W}_a(t)$, resulting in the instantaneous BE and influencing agent control policy given as

$$\delta_t(t) \triangleq \delta\left(x(t), x(t), \hat{\theta}(t), \hat{W}_c(t), \hat{W}_a(t)\right), \tag{28}$$

and $u(t) \triangleq \hat{u}\left(x(t), x(t), \hat{\theta}(t), \hat{W}_a(t)\right)$, respectively. However, if only the BE, given by $\delta_t(t)$, is used to update the estimate $\hat{W}_c$, then an exciting probing signal would need to be injected into the input $\hat{\mu}(t)$ (cf., [37,41,44,45]). In contrast to injecting a probing signal, learning via simulation of experience is performed by extrapolating the BE to unexplored states in $\overline{B_r(x(t))}$. Moreover, sets of functions $\left\{x_i : \mathbb{R}^{3n} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}^{3n}\right\}_{i=1}^N$ are selected by the critic such that $x_i(x(t), t) \in \overline{B_r(x(t))}$. Then, extrapolated versions of the BE and total input are evaluated at $y = x_i(x(t), t)$ as $\delta_{ti}(t) \triangleq \delta\left(x_i(x(t), t), x(t), \hat{\theta}(t), \hat{W}_c(t), \hat{W}_a(t)\right)$ and $u_i(t) \triangleq \hat{u}\left(x_i(x(t), t), x(t), \hat{\theta}(t), \hat{W}_a(t)\right)$, respectively.

Remark 5 *Many different approaches can be utilized to generate extrapolated states. For instance, the states can be selected to follow an oscillatory trajectory which lies within $\overline{B_r(x(t))}$ or they can be selected from a random distribution at each time instance (i.e, $x_i(x(t), t) = x(t) + \sum_{k=1}^{Q}(a_k \sin(b_k t + c_k) + d_k \cos(e_k + h_k))$ or $x_i(x(t), t) = x(t) + U[a, b]\,1_n$, where $a_k, b_k, a, b \in \mathbb{R}$ are constants). The specified approach is a design variable; however, results such as [32, 33, 59] have shown that selecting extrapolated states from a random distribution centered about the current states is sufficient.*

The critic aims to find a set of weights that minimize the BE; hence, the critic is updated according to

$$\dot{\hat{W}}_c(t) \triangleq -\Gamma_c(t)\left(k_{c1}\frac{\omega(t)}{\rho^2(t)}\delta_t(t)+\frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i(t)}{\rho_i^2(t)}\delta_{ti}(t)\right), \tag{29}$$

$$\begin{aligned}\dot{\Gamma}_c(t) \triangleq{}& \beta_c\Gamma_c(t) - \Gamma_c(t)\,k_{c1}\frac{\omega(t)\,\omega^T(t)}{\rho^2(t)}\Gamma_c(t)\\ &- \Gamma_c(t)\,\frac{k_{c2}}{N}\sum_{i=1}^{N}\frac{\omega_i(t)\,\omega_i^T(t)}{\rho_i^2(t)}\Gamma_c(t),\end{aligned} \tag{30}$$

where $\rho(t) \triangleq 1+\gamma_1\omega^T(t)\,\omega(t)$, $\rho_i(t) \triangleq 1+\gamma_1\omega_i^T(t)\,\omega_i(t)$, $k_{c1}, k_{c2}, \gamma_1, \beta_c \in \mathbb{R}_{>0}$ are learning gains, $\omega(t) = \nabla\sigma(x(t), c(x(t)))\Big(F_1(x(t), \hat{\theta}(t)) - F_2(x(t), \hat{\theta}(t)) + G(x(t))\,\hat{\mu}(x(t), x(t), \hat{W}_a(t))\Big)$, and $\omega_i(t) = \nabla\sigma_i(F_{1i} - F_{2i} + G_i\hat{\mu}_i)$, with $\nabla\sigma_i \triangleq \nabla\sigma(x_i(x(t), t), c(x(t)))$, $F_{1i} \triangleq F_1\left(x_i(x(t), t), \hat{\theta}(t)\right)$, $F_{2i} \triangleq F_2\left(x_i(x(t), t), \hat{\theta}(t)\right)$, $G_i \triangleq G(x_i(x(t), t))$, and $\hat{\mu}_i \triangleq \hat{\mu}\left(x_i(x(t), t), x(t), \hat{W}_a(t)\right)$.

Using Assumption 5 along with $\lambda_{\min}\left\{\Gamma_c^{-1}(t_0)\right\} > 0$, a similar argument to [53, Corollary 4.3.2] can be used to show that $\underline{\Gamma}_c I_L \leq \Gamma_c(t) \leq \overline{\Gamma}_c I_L$, where $\underline{\Gamma}_c$ and $\overline{\Gamma}_c$ are positive bounds [32].

The actor weight estimate is updated to follow the critic weight estimate as

$$\begin{aligned}\dot{\hat{W}}_a(t) \triangleq{}& -K_a k_{a1}\left(\hat{W}_a(t) - \hat{W}_c(t)\right) - K_a k_{a2}\hat{W}_a(t)\\ &+ K_a\frac{k_{c1}}{4}G_\sigma^T(t)\,\hat{W}_a(t)\,\frac{\omega^T(t)}{\rho^2(t)}\hat{W}_c(t)\\ &+ K_a\frac{k_{c2}}{4N}\sum_{i=1}^{N}G_{\sigma i}^T(t)\,\hat{W}_a(t)\,\frac{\omega_i^T(t)}{\rho_i^2(t)}\hat{W}_c(t),\end{aligned} \tag{31}$$

where $G_\sigma(t) \triangleq \nabla\sigma(x(t), c(x(t)))\,G_R(x(t)) \cdot \nabla\sigma^T(x(t), c(x(t)))$, $G_{\sigma i}(t) \triangleq \nabla\sigma_i G_i R^{-1} G_i^T \nabla\sigma_i^T$, $G_R(x(t)) \triangleq G(x(t))\,R^{-1}G^T(x(t))$, $k_{a1}, k_{a2} \in \mathbb{R}_{\geq 0}$ are learning gains, and $K_a \in \mathbb{R}^{L\times L}$ is a positive-definite symmetric matrix.

To facilitate learning in this paper, as in [26, 32, 33, 43], off-policy trajectories are selected, which can contain excitation signals to achieve a virtual excitation. Hence, the states $x(t)$ and $x_i(x(t), t)$ are assumed to satisfy the following assumption.

Assumption 5 *There exist constants $T_2, \underline{c}_1, \underline{c}_2, \underline{c}_3 \in \mathbb{R}_{\geq 0}$ such that*

$$\begin{aligned}\underline{c}_1 I_L &\leq \inf_{t\in\mathbb{R}_{\geq t_0}}\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_i(t)\,\omega_i^T(t)}{\rho_i^2(t)},\\ \underline{c}_2 I_L &\leq \int_t^{t+T_2}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\omega_i(\tau)\,\omega_i^T(\tau)}{\rho_i^2(\tau)}\right)d\tau,\ \forall t\in\mathbb{R}_{\geq t_0},\\ \underline{c}_3 I_L &\leq \int_t^{t+T_2}\left(\frac{\omega(\tau)\,\omega^T(\tau)}{\rho^2(\tau)}\right)d\tau,\ \forall t\in\mathbb{R}_{\geq t_0},\end{aligned}$$

where T_2 and at least one of the constants $\underline{c}_1$, $\underline{c}_2$, or $\underline{c}_3$ is strictly positive [32].

Remark 6 *As stated in [26, 32, 33], $\underline{c}_1$ can be made strictly positive by sampling sufficient data, i.e., selecting $N \gg L$, while $\underline{c}_2$ can be made strictly positive using virtual excitation, i.e., by sampling extrapolated trajectories at a high frequency. In general, $\underline{c}_3$ is strictly positive provided the system itself is PE. The extrapolated trajectories x_i are design variables; hence, they can be selected such that such that $\underline{c}_1 >$ or $\underline{c}_2 > 0$ since only one of the constants needs to be strictly positive.*

4 Stability Analysis

Following the development of the update laws used to learn the optimal solution in Section 3, as discussed in the roadmap in Section 1, this section provides the analysis used to show stability of the overall system. To facilitate the following stability analysis, let $B_\zeta \subset \mathbb{R}^{3n+np+2L}$ denote a closed ball of radius $\zeta \in \mathbb{R}_{>0}$ centered at the origin. By defining the critic and actor weight estimate errors as $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$, respectively, the BEs, $\delta_t(t)$ and $\delta_{ti}(t)$, are

$$\begin{aligned}\delta_t &= -\omega^T\tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_\sigma\tilde{W}_a - W^T\nabla\sigma\left(\tilde{F}_1 - \tilde{F}_2\right)\\ &\quad + \Delta(x),\\ \delta_{ti} &= -\omega_i^T\tilde{W}_c + \frac{1}{4}\tilde{W}_a^T G_{\sigma i}\tilde{W}_a - W^T\nabla\sigma_i\left(\tilde{F}_{1i} - \tilde{F}_{2i}\right)\\ &\quad + \Delta_i,\end{aligned} \tag{32}$$

where $\tilde{F}_1 \triangleq F_1\left(x, \tilde{\theta}\right)$, $\tilde{F}_2 \triangleq F_2\left(x, \tilde{\theta}\right)$, $\tilde{F}_{1i} \triangleq F_1\left(x_i, \tilde{\theta}\right)$, $\tilde{F}_{2i} \triangleq F_2\left(x_i, \tilde{\theta}\right)$, and $\Delta_i \triangleq \Delta(x_i)$. In (32), the functions $\Delta, \Delta_i : \mathbb{R}^n \to \mathbb{R}$ are uniformly bounded over a compact set χ such that $\overline{\|\Delta\|}$ and $\overline{\|\Delta_i\|}$ decrease with decreasing $\|\epsilon_v\|$, $\|\varepsilon\|$, and $\|W\|$.

Let $Z_L \triangleq \left[x^T,\ \tilde{W}_c^T,\ \tilde{W}_a^T,\ Z_\theta^T\right]^T$ denote the concatenated state vector, and let $V_L : \mathbb{R}^{3n+2L+np} \times \mathbb{R}_{\geq t_0} \to \mathbb{R}$

denote a candidate Lyapunov functional defined as

$$V_L\left(Z_L,t\right) \triangleq V^*\left(x\right) + \frac{1}{2}\tilde{W}_c^T\Gamma_c^{-1}\left(t\right)\tilde{W}_c + \frac{1}{2}\tilde{W}_a^T K_a^{-1}\tilde{W}_a + V_\theta\left(Z_\theta,t\right), \tag{33}$$

which, for class $\mathcal{K}$ functions $\underline{v}_l, \overline{v}_l : \mathbb{R} \to \mathbb{R}_{\geq 0}$, can be bounded as

$$\underline{v}_l\left(\left\|Z_L\right\|\right) \leq V_L\left(Z_L,t\right) \leq \overline{v}_l\left(\left\|Z_L\right\|\right) \tag{34}$$

for all $t \in \mathbb{R}_{\geq t_0}$ and $Z_L \in \mathbb{R}^{3n+2L+np}$.

Theorem 2 shows the overall stability of the system. Using the candidate Lyapunov functional in (33), along with the update laws in (29)-(31), and (20) from Theorem 1, the system states are shown to be bounded. The BE expressions in (32) are used to show that the estimation errors $\tilde{W}_c$ and $\tilde{W}_a$ remain bounded along with the concatenated state x.

Theorem 2 *Provided Assumptions 2-5 are satisfied,* $\lambda_{\min}\left\{H\right\} > 0$, *and*

$$\sqrt{\frac{\iota}{\kappa}} \leq \underline{v}_l^{-1}\left(\overline{v}_l\left(\zeta\right)\right), \tag{35}$$

where $H \triangleq \begin{bmatrix} \left(\frac{k_{a1}+k_{a2}}{4} - \varphi_a\right) & -\frac{\varphi_{ac}}{2} & 0 \\ -\frac{\varphi_{ac}}{2} & \frac{\left(\frac{\beta_c}{\overline{\Gamma}_c} + k_{c2}\underline{c}_1\right)}{8} & -\frac{\varphi_{c\theta}}{2} \\ 0 & -\frac{\varphi_{c\theta}}{2} & \frac{k_\theta c_{\theta 1}}{8} \end{bmatrix}$, *and* $\kappa, \varphi_a, \varphi_{ac}, \varphi_{c\theta}, \iota \in \mathbb{R}_{>0}$ *are defined in the Appendix, then the system errors defined in* Z_L *are bounded in the sense that*

$$\limsup_{t\to\infty}\left\|Z_L\left(t\right)\right\| \leq \underline{v}_l^{-1}\left(\overline{v}_l\left(\sqrt{\frac{\iota}{\kappa}}\right)\right). \tag{36}$$

PROOF. Taking the time-derivative of (33) along the system trajectory, and using the fact that $\dot{V}^*\left(x,t\right) = \nabla V^*\left(F\left(x\right) + G\left(x\right)\mu\right)$, results in

$$\begin{aligned}\dot{V}_L &= \nabla V^*\left(F + G\mu\right) + \dot{V}_\theta\left(Z_\theta,t\right) \\ &\quad + \tilde{W}_c^T\Gamma_c^{-1}\left(\dot{W} - \dot{\hat{W}}_c\right) - \frac{1}{2}\tilde{W}_c^T\left(\Gamma_c^{-1}\dot{\Gamma}_c\Gamma_c^{-1}\right)\tilde{W}_c \\ &\quad + \tilde{W}_a^T K_a^{-1}\left(\dot{W} - \dot{\hat{W}}_a\right).\end{aligned}$$

Substituting (13) and using $\dot{W} = \nabla W\left(x\right)\left(F\left(x\right) + G\left(x\right)\mu\right)$ yields

$$\begin{aligned}\dot{V}_L &= -r\left(x,\mu^*\left(x\right)\right) - \nabla V^* G\mu^* - \frac{1}{2}\tilde{W}_c^T\Gamma_c^{-1}\dot{\Gamma}_c\Gamma_c^{-1}\tilde{W}_c \\ &\quad + \tilde{W}_c^T\Gamma_c^{-1}\left(\nabla W\left(F + G\mu\right) - \dot{\hat{W}}_c\right) + \nabla V^* G\mu \\ &\quad + \tilde{W}_a^T K_a^{-1}\left(\nabla W\left(F + G\mu\right) - \dot{\hat{W}}_a\right) + \dot{V}_\theta\left(Z_\theta,t\right).\end{aligned}$$

Using (29) and (30), then substituting in (11), (20), (31), and (32), and using $\hat{W}_a = W - \tilde{W}_a$, $\hat{W}_c = W - \tilde{W}_c$, bounding, and completing the squares yields $\dot{V}_L \leq -\kappa\left\|Z_L\right\|^2 - \kappa\left\|Z_L\right\|^2 + \iota - Z_v^T H Z_v$, where $Z_v \triangleq \left[\left\|\tilde{W}_a\right\|, \left\|\tilde{W}_c\right\|, \left\|Z_\theta\right\|\right]^T$. Provided $\lambda_{\min}\left\{H\right\} > 0$ is met, then for all $Z \in B_\zeta$

$$\dot{V}_L \leq -\kappa\left\|Z_L\right\|^2, \; \forall\left\|Z_L\right\| \geq \sqrt{\frac{\iota}{\kappa}} > 0. \tag{37}$$

Using (34), (35), and (37), [54, Theorem 4.18] is invoked to conclude that all trajectories $Z_L\left(t\right)$ that satisfy $\left\|Z_L\left(t_0\right)\right\| \leq \overline{v}_l^{-1}\left(\underline{v}_l^{-}\left(\zeta\right)\right)$ remain bounded for all $t \in \mathbb{R}_{\geq t_0}$ and satisfy (36). Since $Z_L \in \mathcal{L}_\infty$, it follows that x, $\tilde{W}_c$, $\tilde{W}_a$, $\tilde{\theta} \in \mathcal{L}_\infty$, and therefore, $\mu \in \mathcal{L}_\infty$. Furthermore, since $x \in \mathcal{L}_\infty$ and W is a continuous function of x, then $W\left(x\right) \in \mathcal{L}_\infty$. Moreover, since $x \in \mathcal{L}_\infty$, it follows that e_d, e_η, $e_z \in \mathcal{L}_\infty$. Using (3)-(5), $z \in \mathcal{L}_\infty$ and $\eta_d \in \mathcal{L}_\infty$; hence, η, $\left(z - \eta\right) \in \mathcal{L}_\infty$. Finally, since $\mu, \tilde{\theta}, g^+$, $\eta_d \in \mathcal{L}_\infty$, then $u_d, \hat{\theta} \in \mathcal{L}_\infty$ and $u \in \mathcal{L}_\infty$. ■

Remark 7 *The bound in (36) implies that the concatenated state vector* Z_L *is bounded by a strictly increasing function (i.e., class* $\mathcal{K}$ *function) based on the residual bounds and sufficient conditions (i.e.,* $\sqrt{\frac{\iota}{\kappa}}$*). Moreover, the smaller the value of* $\sqrt{\frac{\iota}{\kappa}}$*, the smaller the ultimate bound. Hence, provided the sufficient conditions can be selected such that* $\sqrt{\frac{\iota}{\kappa}} \to 0$ *, then the ultimate bound converges to the origin [54, Theorem 4.18].*

Remark 8 *The sufficient condition* $\lambda_{\min}\left\{H\right\} > 0$ *can be satisfied by increasing the gains* k_{a2} *and* γ_1*, and selecting* K_a *and* R *with large minimum eigenvalues. In addition, increasing the number of neurons and number of sample points for the system identification, i.e.,* $p_z \gg n$*,* $p_\eta \gg n$*, and* $M \gg p$*, and also selecting extrapolation points* $x_i\left(x\left(t\right),t\right)$ *so that* $\underline{c}_1$ *is large will also help ensure the sufficient condition is satisfied.*

5 Simulation

Following the roadmap in Section 1, this section provides a simulation example that demonstrates the performance of the developed method. Specifically, a two-dimensional simulation is performed for the roaming agent in (1) and the influencing agent in (2) with $f\left(z\left(t\right),\eta\left(t\right)\right) = \left(Ae\left(t\right) + Be_z\left(t\right)\right)\exp\left(-\frac{1}{2}e\left(t\right)^T e\left(t\right)\right)$, where $e\left(t\right) = z\left(t\right) - \eta\left(t\right)$, and, without a loss of generality, $h\left(z\left(t\right),\eta\left(t\right)\right) = 0_{2\times 1}$, $g\left(\eta\left(t\right)\right) = I_2$, respectively. Based on these dynamics, Assumptions 1 and 2 are satisfied since $\left\|f\left(z\left(t\right),\eta\left(t\right)\right)\right\| \leq \left\|A\right\|\left\|e\left(t\right)\right\| + \left\|B\right\|\left\|e_z\left(t\right)\right\|$ and I_2 is full-rank. The unknown parameters to be

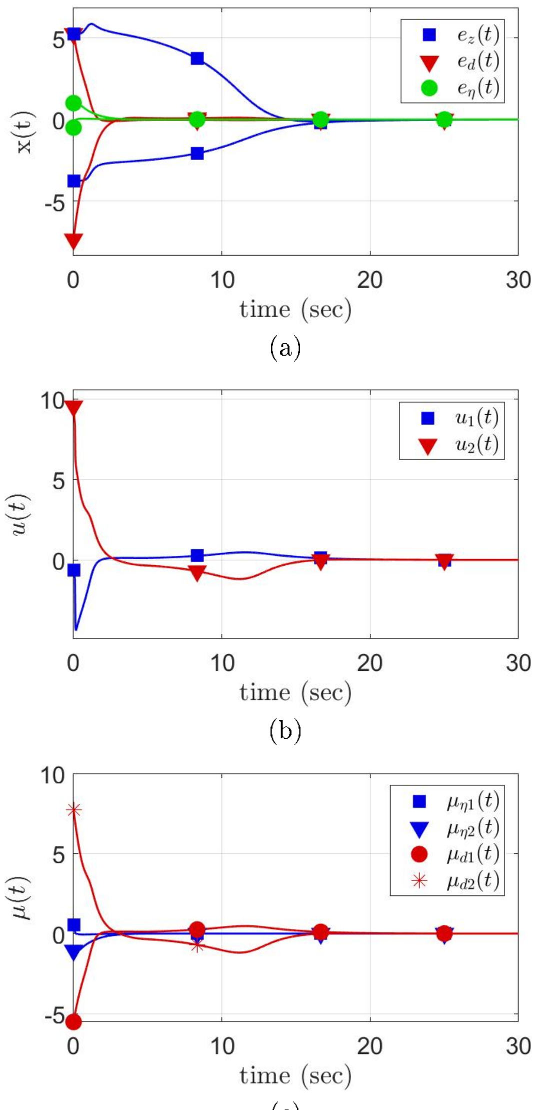
Table 1
Initial conditions and parameters selected for the simulation.

Initial conditions at $t_0 = 0$
$z(0) = [-3.75,\ 4.75]^T$, $\eta(0) = [-0.5, 0.5]^T$, $\eta_d(0) = [0, -0.5]^T$, $z_g = [0, -0.5]^T$, $\hat{W}_c(0) = 1 \times 1_{7\times 1}$, $\hat{W}_a(0) = 0.75 \times 1_{7\times 1}$, $\Gamma_c(0) = 2I_7$, $\hat{\theta}(0) = U[-1, 1] \times 1_{6\times 2}$, $\Gamma_\theta(0) = 100I_6$.
Penalizing parameters
$Q(x) = x^T Q_x x$, $P(x) = \left(\max\left\{0, \left\|(1-k_d)\, e_z(t) - e_\eta(t) - e_d(t)\right\|^2 - {r_a}^2\right\}\right)^2$, $Q_x = \operatorname{diag}\{20, 20, 5, 5, 5, 5\}$, $R = 5I_4$, $k_d = 1.4$, $r_a = 0.15$.
Gains and parameters for ADP update laws
$k_{c1} = 0.9$, $k_{c2} = 0.02$, $k_{a1} = 0.25$, $k_{a2} = 0.005$, $\gamma_1 = 0.75$, $\beta_c = 0.001$, $K_a = I_7$, $k_\theta = 0.5$, $\beta_\theta = 2$, $M = 100$, $N = 10$.

identified by the ICL update law in (16) and (17) are $\theta_\eta \triangleq 0_{2\times 2}$, $\theta_z = [k_d A, B]^T$, where $A = \begin{bmatrix} 1 & -0.6 \\ 0.5 & 1.5 \end{bmatrix}$ and $B = \begin{bmatrix} 0.05 & 0 \\ 0.25 & 0.1 \end{bmatrix}$. Hence, the ideal weights are bounded as specified in Assumption 3. For parameter identification, the basis functions are selected as $S_\eta(x(t)) = e(t)$ and $S_z(x(t)) = \exp\left(-\frac{1}{2} e(t)^T e(t)\right) \times \left[e^T(t),\ e_z^T(t)\right]^T$. In this simulation, the state is assumed to lie inside a compact set $\chi \subseteq \mathbb{R}^{3n}$, which is bounded, resulting in a bounded basis, where S_η and S_z satisfy Assumption 3. The StaF basis is selected as $\sigma(x, c(x)) = m\left[\sigma_1(x, c_1(x)), \ldots, \sigma_7(x, c_7(x))\right]^T$, where $\sigma_i(x(t), c_i(x(t))) = x^T(t)(x(t) + 0.7\nu(x(t))\, d_i)$, $\nu(x(t)) = \frac{0.7x^T(t)x(t)}{(1+x^T(t)x(t))}$, and d_i are the vertices of a 6-simplex [32, 34, 59]. To perform BE extrapolation, ten trajectories $x_i(x(t), t)$ are selected at random from a uniform distribution over a $[-1, 1] \times [-1, 1]$ square centered at the current state $x(t)$. The selected initial conditions and parameters are provided in Table 1.

5.1 *Discussion*

Figures 2-5 demonstrate that the influencing agent regulates the roaming agent to the goal location z_g. Figure 2a shows that the concatenated state $x(t)$ converges to the origin. Hence, both the system identification basis and value function weights remain bounded. Figure 2b shows that the input mismatch $\mu_\eta(t) = u(t) - u_d(t)$ converges faster than $\mu_d(t)$; hence, the influencing agent is using the desired input which is based on regulating the roaming agent to the desired location. The influencing agent's applied input $u(t) = \mu_\eta(t) + u_d(t)$, shown in Figure 2c, remains bounded and converges once the agent's reach the goal location. Figures 3a-3b show that the critic and actor weight estimates remain bounded; however, because the optimal StaF weights are unknown, the estimates cannot be compared to their ideal values. Figure 3c shows Assumption 5 is satisfied since the minimum eigenvalue of the BE regressor is positive. Since the roaming and influencing agents are modeled using linearly-parameterizable dynamics with an exactly known basis, the parameter estimates can be compared to the true values. Figure 4a shows that the system parameter estimates converge to the true values, while Figure 4b shows when the history stack becomes positive definite, which validates Assumption 4. The positions of the roaming and influencing agents are shown in Figure 5. The roaming agent is not independently mo-

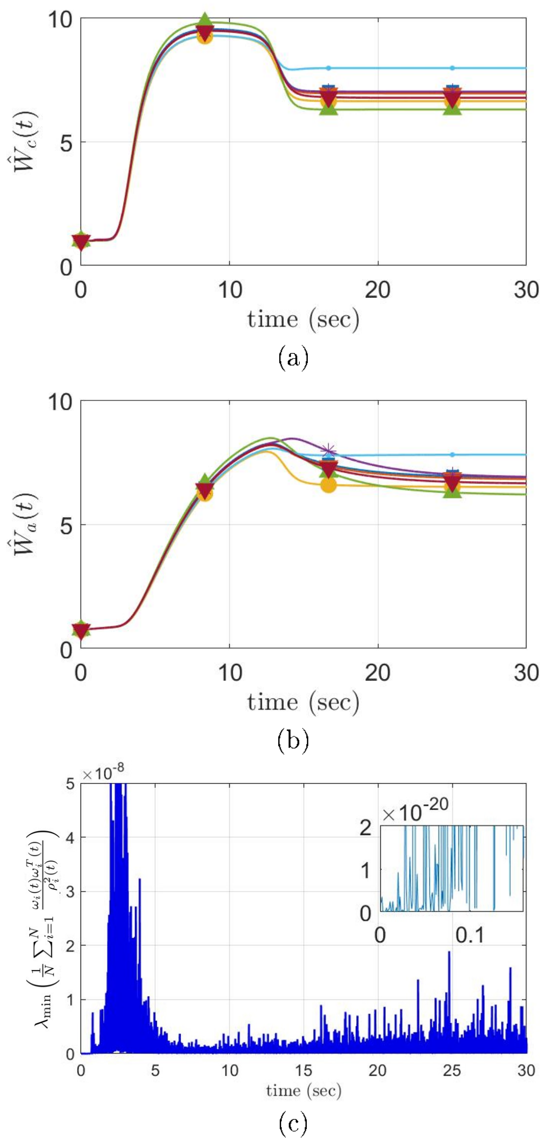


(a)

(b)

(c)

Figure 2. Sub-Figure (a) depicts the concatenated state $x(t)$, (b) depicts approximate optimal input $\mu(t)$, and (c) depicts applied influencing agent input $u(t)$. In Figure 2a, $e_z(t)$ is represented by blue squares, $e_d(t)$ is represented by red diamonds, and $e_\eta(t)$ is represented by green circles. In Figure 2b, the blue squares and diamonds represent $\mu_{\eta 1}(t)$ and $\mu_{\eta 2}(t)$, respectively, while the red circles and asterisks represent $\mu_{d1}(t)$ and $\mu_{d2}(t)$, respectively. In Figure 2c, $u_1(t)$ is represented by blue squares and $u_2(t)$ is represented by red diamonds.



(a)

(b)

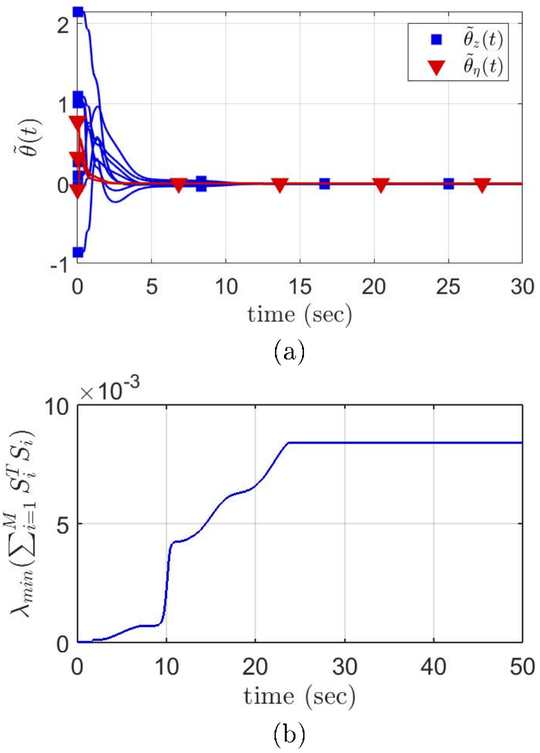
(c)

Figure 3. Sub-Figure (a) depicts the critic StaF weight estimates, and Sub-Figure (b) depicts the actor StaF weight estimates, both of which remain bounded. Sub-Figure (c) depicts the the minimum eigenvalue of the regression matrix.

tivated to go to the desired location; hence, in Figure 5, the roaming agent initially diverges from the goal location. As the influencing agent approaches the roaming agent, the roaming agent starts moving away. Motivated to regulate the roaming agent to the goal location, the influencing agent begins to regulate the roaming agent toward z_g. $[0, -0.5]^T$.

6 Experiment

The simulation results in Section 5 show the development works on an ideal system, where all assumptions could be verified. Following the roadmap in Section 1, this section provides experimental results to illustrate



(a)

(b)

Figure 4. Sub-Figure (a) depicts the system identification errors $\tilde{\theta}(t)$, which converge to the origin, while Sub-Figure (b) depicts the minimum eigenvalue of the regression matrix used for system identification.

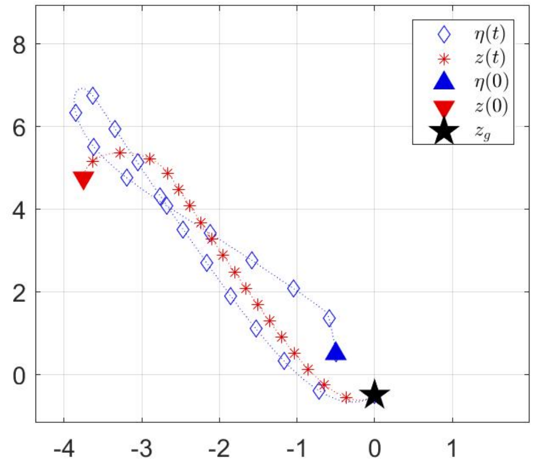


Figure 5. Positions of the influencing and roaming agents. The influencing agent (blue diamonds) intercepts and regulates the roaming agent (red asterisks) to the goal location (black star). The initial condition of the influencing agent is given by the blue triangle, and the initial condition for the roaming agent is given by the red triangle.

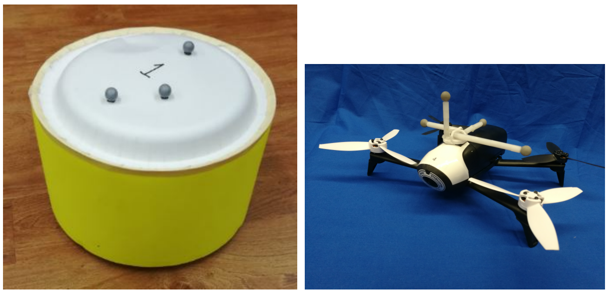


Figure 6. The unactuated paper platform (left) representing the roaming agent, and the Parrot Bebop 2.0 quadcopter (right) representing the influencing agent.

the performance of the developed approach. A series of ten experiments were conducted, where a different combination of state penalty, Q, and input penalty, R, weights were used to produce different performance characteristics. A Parrot Bebop 2 quadcopter was used as the influencing agent and an unactuated paper platform, shown in Figure 6, was used as the roaming agent. The unactuated paper platform was constructed from a paper plate, top and bottom, fastened to a colored poster board. The turbulent air caused by the quadcopter propellers produce a repulsing force, which causes the nearby roaming agent to slide away. In addition, to control the quadcopter, two-dimensional velocity commands were generated by the developed controller. In general, since the actual dynamics are unknown, it is difficult to show that Assumption 1 is satisfied. However, since the unactuated paper platform only moves when the quadcopter gets closer to the platform, it can be said that the overall interaction dynamics depend on the distance between the two agents.

A NaturalPoint, Inc. OptiTrack motion capture system is used to measure the pose of the quadcopter and paper platform. For this experimental setup, a ground station, equipped with the Robotic Operating System (ROS) Kinetic framework and the *bebop_*autonomy *package* developed by [60] running on Ubuntu 16.04, receives the pose from the motion capture system, calculates the policies as velocity commands, and broadcasts the commands to the quadcopter at 120Hz. A video of a typical run of this experiment is available at [61].

The influencing agent was implemented using dynamics such that $h(\eta(t), z(t)) = 0_{2\times 1}$; hence the dynamics did not need to be estimated. To identify the interaction dynamics in (1) , $p_z = 4$ Gaussian radial basis functions were selected. Each center of the basis was located in a quadrant around the influencing agent, where the standard deviation is selected as $\sqrt{0.5}$ m. Using this representation, the influencing agent estimated the repulsion effects it had on the roaming agent. To approximate the value function, the StaF basis is selected as $\sigma(x, c(x)) = [\sigma_1(x, c_1(x)), \ldots, \sigma_7(x, c_7(x))]^T$, where $\sigma_i(x(t), c_i(x(t))) = \exp\left(\frac{x^T(t)c(x(t))}{\|x(0)\|^2}\right)$, $c(x(t)) = (x(t) + \|x(0)\| \nu(x(t)) d_i)$, $\nu(x(t)) = \frac{0.05x^T(t)x(t}{(\|x(0)\|^2 + x^T(t)x(t))}$, and d_i are the vertices of a 6-simplex. To perform BE extrapolation, ten trajectories $x_i(x(t), t)$ are selected at random from a uniform distribution over a $\nu(x(t)) \times \nu(x(t))$ square centered at the current state $x(t)$. The goal of the experiment is to indirectly regulate the roaming agent to a neighborhood of radius $r_{goal} = 0.5$ m of the desired location $z_g = [-2, 0]^T$ m. The selected initial conditions and parameters are provided in Table 2.

Table 2
Initial conditions and parameters selected for the experiments.

Conditions at $t_0 = 0$
$\hat{W}_c(0) = 0.2 \times 1_{5\times 1}$, $\hat{W}_a(0) = 0.1 \times 1_{5\times 1}$, $\Gamma_c(0) = 0.01 I_5$, $\hat{\theta}(0) = U[-0.1, 0.1] \times 1_{4\times 2}$, $\Gamma_\theta(0) = 0.1 I_4$.
Penalizing parameters
$Q = x^T Q_x x$, $P(x) = 0$, $k_d = 1.15$.
Gains and parameters for ADP update laws
$k_{c1} = 0.1$, $k_{c2} = 0.9$, $k_{a1} = 0.9$, $k_{a2} = 0.1$, $\gamma_1 = 0.5$, $\beta_c = 0.001$, $K_a = I_5$, $N = 10$, $\beta_\theta = 0.1$, $M = 50$.

Table 3
State and input penalty weights for each experiment.

Experiment	R	Q_x $(\times 10^2)$	$\lambda_{avg}\{R\}$	$\lambda_{avg}\{Q\}$ $(\times 10^2)$
1	$\mathrm{diag}\{20, 10, 50, 25\}$	$\mathrm{diag}\{1, 10, 1, 10, 1, 1\}$	26.25	4.0
2	$\mathrm{diag}\{20, 5, 50, 25\}$	$\mathrm{diag}\{1, 20, 1, 20, 1, 1\}$	25.0	7.33
3	$\mathrm{diag}\{20, 5, 50, 25\}$	$\mathrm{diag}\{1, 30, 1, 30, 1, 1\}$	25.0	10.67
4	$\mathrm{diag}\{25, 10, 55, 30\}$	$\mathrm{diag}\{1, 30, 1, 30, 1, 1\}$	30.0	10.67
5	$\mathrm{diag}\{30, 20, 65, 40\}$	$\mathrm{diag}\{1, 30, 1, 30, 1, 1\}$	38.75	10.67
6	$\mathrm{diag}\{40, 30, 75, 50\}$	$\mathrm{diag}\{1, 30, 1, 30, 1, 1\}$	48.75	10.67
7	$\mathrm{diag}\{40, 30, 80, 55\}$	$\mathrm{diag}\{1.5, 30, 1.5, 30, 1, 1\}$	51.25	10.83
8	$\mathrm{diag}\{40, 30, 100, 75\}$	$\mathrm{diag}\{1.5, 30, 1.5, 30, 1, 1\}$	61.25	10.83
9	$\mathrm{diag}\{40, 30, 120, 100\}$	$\mathrm{diag}\{1.5, 30, 1.5, 30, 1, 1\}$	72.5	10.83
10	$\mathrm{diag}\{60, 40, 150, 125\}$	$\mathrm{diag}\{1.5, 30, 1.5, 30, 1, 1\}$	93.75	10.83

A survey of ten experiments was performed, where different combinations of penalty weights for the state and policy, Q_x and R (shown in Table 3), respectively, are used, while other parameters remained constant between experiments. Norms of the initial concatenated state and regulation error; the total root-mean square (RMS) values of the norms of the concatenated state x, regulation error e_z, and applied input u; the total cost; and time-to-completion (TTC) are calculated and tabulated in Table 4.

Table 4
The results for the survey of ten experiments with varying state and input penalty weights.

Experiment	Concatenated State	Regulation Error	Concatenated State	Regulation Error	Applied Input	Total Cost	TTC
	Initial Norm	Initial Norm	Total RMS	Total RMS	Total RMS		
	$\|x(0)\|$ (m)	$\|e_z(0)\|$ (m)	$\|x\|_{RMS}$ (m)	$\|e_z\|_{RMS}$ (m)	$\|u\|_{RMS}$ $\left(\frac{m}{sec}\right)$	$(\times 10^3)$	(sec)
1	3.973	3.861	3.084	3.043	0.284	20.51	20.60
2	4.026	3.801	3.178	3.102	0.332	26.34	19.07
3	4.403	4.252	3.529	3.476	0.336	26.28	19.31
4	4.363	4.293	3.584	3.561	0.309	30.12	21.35
5	4.455	4.343	3.540	3.499	0.314	28.97	19.19
6	4.367	4.307	3.551	3.531	0.255	39.39	24.87
7	4.285	4.223	3.398	3.368	0.284	35.23	18.34
8	4.372	4.197	3.442	3.379	0.296	49.22	21.39
9	3.973	3.798	3.100	3.027	0.273	37.73	17.95
10	3.930	3.544	2.980	2.844	0.302	66.57	19.49

6.1 Discussion

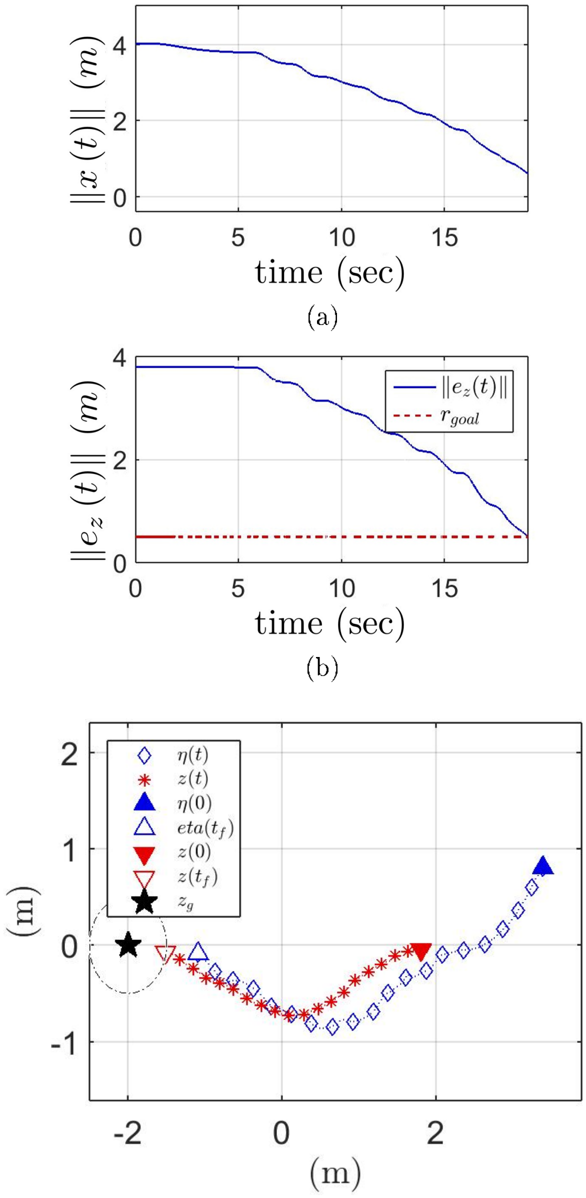
Experiment results are provided in Table 4. To display part of the experimental trials, two runs (experiments two and nine), containing different penalty weights and different trajectories, were selected to show the performance of the developed strategy. The concatenated state norm, $\|x(t)\|$, the regulation error norm, $\|e_z(t)\|$, and the phase-space portrait for experiments two and nine are shown in Figures 7 and 8, respectively. Figures 7a and 7b show the norms of the concatenated state and regulation error for experiment two, respectively, which decrease until the roaming agent is regulated to a neighborhood of the goal location (i.e, $\|e_z(t)\| \leq r_{goal}$). The trajectories of the influencing and roaming agents are shown in Figure 7c. Specifically, the influencing agent moves toward the roaming agent to guide it toward the goal location z_g. As the influencing agent approached the roaming agent, the roaming agent moves in the direction of the goal. However, as the roaming agent begins to drift in a wrong direction, the influencing agent adjusts its trajectory to regulate the roaming agent back in the direction of the goal location.

To show the performance of the agents under different state and input penalty weights, Figure 8 shows similar metrics as in Figure 7. Specifically, the norms of the concatenated state $x(t)$ and regulation error $e_z(t)$ for experiment nine are displayed in Figures 8a and 8, respectively, which show that the total state and regulation error decrease for experiment nine as the roaming agent is regulated to a neighborhood of the goal. Figure 8c shows the phase-space portrait for both agents. Compared to Figure 7c, Figure 8c shows that different state and input penalty weights affect how the agents will interact. Moreover, in experiment nine, the influencing agent still achieves the objective of regulating the roaming agent to a neighborhood of the z_g.

Table 3 shows the selected state and input penalty weights for each experiment, while the results are shown in Table 4. Specifically, Table 4 shows the effect of the system penalty weights and initial setup on the system performance, including: the concatenated state, regulation error, and applied input total RMS values; total cost; and TTC. Moreover, Table 4 shows that when the state penalty weights are kept constant, but the input penalty weights are increased, the total RMS values for the applied input decrease. But, when the state penalty weight is increased, the total RMS values of the concatenated state and regulation error decrease. Moreover, as the penalty weights are increased, the total cost is increased because the influencing agent's actions and the states of both agents are being penalized more. Finally, due to the complex environment, the roaming agent was affected by factors like varying friction. Hence, the TTC was greatly affected between experiments.

7 Conclusion

An indirect regulation problem is investigated for a roaming agent being directed by an influencing agent via interaction dynamics. To estimate the uncertainties in the roaming and influencing agent dynamics, a data-based estimator, which relaxes the PE condition, is used. The problem is posed as an infinite-horizon optimal control problem and a local StaF-based ADP method is used to approximate the optimal value function and controller. Uniformly ultimately bounded convergence is

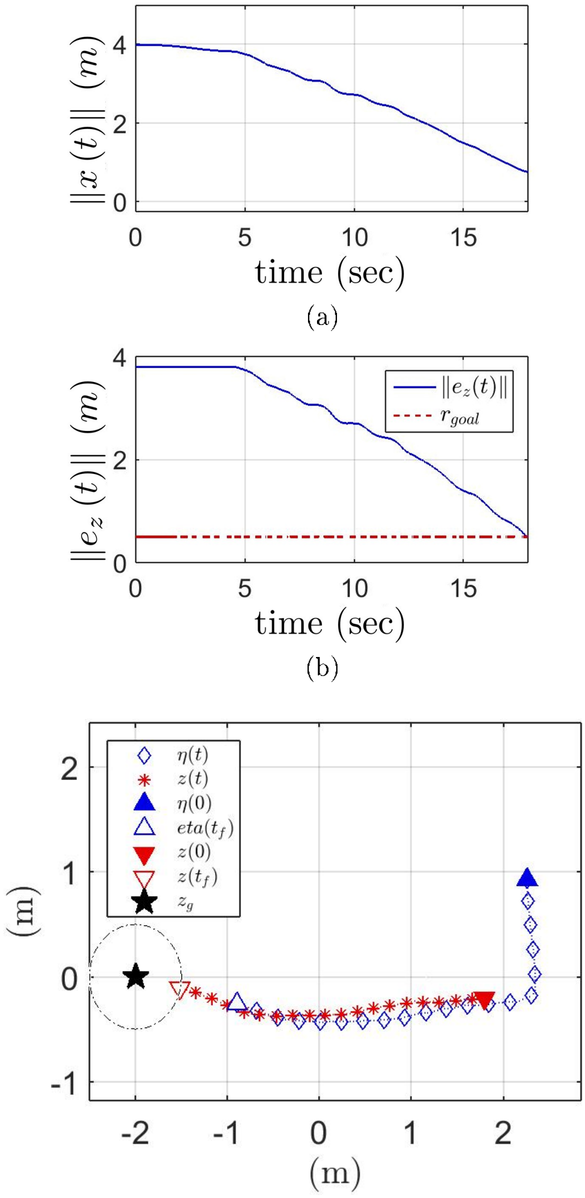


(a)

(b)

(c)

Figure 7. Sub-Figure (a) depicts the concatenated state norm, $\|x(t)\|$, and (b) depicts the regulation error norm, $\|e_z(t)\|$, where both decrease until the roaming agent is within r_{goal}. In Figure 7b, the blue solid line represents $\|e_z(t)\|$ while the red dashed line represents the neighborhood of the goal denoted by r_{goal}. Sub-Figure (c) depicts the phase-space portrait, which shows the+ trajectories of the roaming and influencing agents. The influencing agent (blue diamonds) regulates the roaming agent (red asterisks) to a neighborhood ($r_{goal} = 0.5$ m) of goal location (black star). The initial influencing agent condition is given by the blue triangle and the initial roaming agent condition is given by the red triangle.



(a)

(b)

(c)

Figure 8. Sub-Figure (a) depicts the concatenated state norm, $\|x(t)\|$, (b) depicts the regulation error norm, $\|e_z(t)\|$, and (c) depicts the phase-space portrait for experiment nine. In Figure 8b, the blue solid line represents $\|e_z(t)\|$ while the red dashed line represents the neighborhood of the goal denoted by r_{goal}. In Figure 8c, the influencing agent (blue diamonds) regulates the roaming agent (red asterisks) to a neighborhood ($r_{goal} = 0.5$ m) of goal location (black star). The initial influencing agent condition is given by the blue triangle and the initial roaming agent condition is given by the red triangle.

shown via a Lyapunov stability analysis for the closed-loop error system. Simulation results in addition to experimental results for two-state influencing and roaming agents are included, which illustrate the performance of the developed method. Future efforts will focus on posing the problem as a differential game and also considering multiple roaming agents.

References

[1] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, 1980.

[2] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*, ser. Dover Books on Mathematics. Dover Publications, 1999.

[3] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Autonomous robots*, vol. 31, no. 4, p. 299, 2011.

[4] S. S. Kumkov, S. Le Ménec, and V. S. Patsko, "Zero-sum pursuit-evasion differential games with many objects: survey of publications," *Dyn. Games Appl.*, vol. 7, no. 4, pp. 609–633, 2017.

[5] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Trans. Robot. and Autom.*, vol. 18, no. 5, pp. 662–669, Oct. 2002.

[6] A. D. Khalafi and M. R. Toroghi, "Capture zone in the herding pursuit evasion games," *Appl. Math. Sci.*, vol. 5, no. 39, pp. 1935–1945, 2011.

[7] P. Kachroo, S. A. Shedied, J. S. Bay, and H. Vanlandingham, "Dynamic programming solution for a class of pursuit evasion problems: the herding problem," *IEEE Trans. Syst. Man Cybern.*, vol. 31, no. 1, pp. 35–41, Feb. 2001.

[8] M. Chen, Z. Zhou, and C. J. Tomlin, "Multiplayer reach-avoid games via pairwise outcomes," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1451–1457, 2017.

[9] J. Chen, W. Zha, Z. Peng, and D. Gu, "Multi-player pursuit–evasion games with one superior evader," *Automatica*, vol. 71, pp. 24–32, 2016.

[10] M. V. Ramana and M. Kothari, "Pursuit-evasion games of high speed evader," *J. Intell. Rob. Syst.*, vol. 85, no. 2, pp. 293–306, 2017.

[11] F. Yan, J. Jiang, K. Di, Y. Jiang, and Z. Hao, "Multiagent pursuit-evasion problem with the pursuers moving at uncertain speeds," *J. Intell. Rob. Syst.*, pp. 1–17, 2018.

[12] R. Isaacs, *Differential Games*. John Wiley, 1967.

[13] E. Garcia, D. W. Casbeer, and M. Pachter, "Active target defence differential game: fast defender case," *IET Control Theory Appl.*, vol. 11, no. 17, pp. 2985–2993, 2017.

[14] ——, "Design and analysis of state-feedback optimal strategies for the differential game of active defense," *IEEE Trans Autom. Contol*, 2018.

[15] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "Automation-assisted capture-the-flag: A differential game approach," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1014–1028, 2015.

[16] A. S. Gadre, "Learning strategies in multi-agent systems-applications to the herding problem," *M.S. thesis, Dept. Elect. Comput. Eng., Virginia Tech, Blacksburg, VA, USA*, 2001.

[17] Z. Lu, "Cooperative optimal path planning for herding problems," Ph.D. dissertation, Texas A&M University, 2006.

[18] S. A. Shedied, "Optimal control for a two player dynamic pursuit evasion game; the herding problem," Ph.D. dissertation, Virginia Polytechnique Institute, 2002.

[19] R. Licitra, Z. Hutcheson, E. Doucette, and W. E. Dixon, "Single agent herding of n-agents: A switched systems approach," in *IFAC World Congr.*, 2017, pp. 14 374–14 379.

[20] 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, January 2018.

[21] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 517–525, 2018.

[22] M. Bacon and N. Olgac, "Swarm herding using a region holding sliding mode controller," *J. Vib. Control*, vol. 18, no. 7, pp. 1056–1066, 2012.

[23] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int J Adapt Control Signal Process*, to appear.

[24] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM REV.*, vol. 47, no. 1, pp. 99–131, 2005.

[25] P. E. Gill, E. Wong, W. Murray, and M. A. Saunders, "User's guide for snopt version 7: Software for large-scale nonlinear programming," https://ccom.ucsd.edu/ optimizers/static/pdfs/snopt7-7.pdf, 2015, department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112.

[26] R. Kamalapurkar, P. S. Walters, J. A. Rosenfeld, and W. E. Dixon, *Reinforcement learning for optimal feedback control: A Lyapunov-based approach*. Springer, 2018.

[27] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Syst.*, vol. 37, no. 1, pp. 33–52, 2017.

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

[29] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, 2017.

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

[31] G. Wen, S. S. Ge, and F. Tu, "Optimized backstepping for tracking control of strict-feedback systems," *IEEE Trans. Neural Netw. Learn. Syst.*, 2018.

[32] 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.

[33] 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, June 2018.

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

[35] K. G. Vamvoudakis and J. P. Hespanha, "Cooperative q-learning for rejection of persistent adversarial inputs in networked linear quadratic systems," *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 1018–1031, 2018.

[36] H. Modares, F. L. Lewis, W. Kang, and A. Davoudi, "Optimal synchronization of heterogeneous nonlinear systems with unknown dynamics," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 117–131, 2018.

[37] D. Wang, D. Liu, C. Mu, and H. Ma, "Decentralized guaranteed cost control of interconnected systems with uncertainties: a learning-based optimal control strategy," *Neurocomputing*, vol. 214, pp. 297–306, 2016.

[38] R. Kamalapurkar, J. R. Klotz, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for differential graphical games," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 423–433, 2018.

[39] 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, 2013.

[40] J. Sun, C. Liu, and Q. Ye, "Robust differential game guidance laws design for uncertain interceptor-target engagement via adaptive dynamic programming," *Int. J. Control*, vol. 90, no. 5, pp. 990–1004, 2017.

[41] Z. Wang, X. Liu, K. Liu, S. Li, and H. Wang, "Backstepping-based Lyapunov function construction using approximate dynamic programming and sum of square techniques," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3393–3403, 2017.

[42] Y. Lin and E. D. Sontag, "A universal formula for stabilization with bounded controls," *Systems & Control Letters*, vol. 16, no. 6, pp. 393–397, 1991.

[43] P. Deptula, Z. I. Bell, F. Zegers, R. Licitra, and W. E. Dixon, "Single agent indirect herding via approximate dynamic programming," in *Proc. IEEE Conf. Decis. Control*, Dec. 2018, pp. 7136–7141.

[44] 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.

[45] 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, 2017.

[46] S. B. Roy, S. Bhasin, and I. N. Kar, "Combined mrac for unknown mimo lti systems with parameter convergence," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 283–290, Jan. 2018.

[47] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, "Concurrent learning for parameter estimation using dynamic state-derivative estimators," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3594–3601, July 2017.

[48] S. Basu Roy, S. Bhasin, and I. N. Kar, "Composite adaptive control of uncertain euler-lagrange systems with parameter convergence without pe condition," *Asian J. Control*, 2019.

[49] N. Cho, H.-S. Shin, Y. Kim, and A. Tsourdos, "Composite model reference adaptive control with parameter convergence under finite excitation," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 811–818, Mar. 2017.

[50] J. A. Farrell and M. M. Polycarpou, *Adaptive approximation based control: Unifying neural, fuzzy and traditional adaptive approximation approaches*, ser. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. John Wiley & Sons, 2006, vol. 48.

[51] F. L. Lewis, R. Selmic, and J. Campos, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities.* Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.

[52] N. Sadegh, "A perceptron network for functional identification and control of nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 4, no. 6, pp. 982–988, 1993.

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

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

[55] G. Chowdhary and E. Johnson, "A singular value maximizing data recording algorithm for concurrent learning," in *Proc. Am. Control Conf.*, 2011, pp. 3547–3552.

[56] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, 2008.

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

[58] S. Bhasin, R. Kamalapurkar, H. T. Dinh, and W. Dixon, "Robust identification-based state derivative estimation for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 187–192, Jan. 2013.

[59] 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.

[60] "bebop_autonomy library," http://bebop-autonomy.readthedocs.io.

[61] P. Deptula, Z. I. Bell, F. M. Zegers, R. A. Licitra, and W. E. Dixon, "Approximate optimal influence over an agent through an uncertain interaction dynamic experiment," https://youtu.be/JeK4jTDuImo, Feb. 2019.

8 Auxiliary Terms

To facilitate the analysis in Section 4, $\kappa \in \mathbb{R}_{>0}$ is defined as $\kappa \triangleq \min\left\{\frac{\underline{q}}{2}, \frac{k_\theta c_{\theta 1}}{16}, \frac{k_{c2}\underline{c}}{8}, \frac{(k_{a1}+k_{a2})}{8}\right\}$, where the constants $\varphi_a, \varphi_{ac}, \varphi_{c\theta} \in \mathbb{R}_{>0}$ are defined as $\varphi_a \triangleq \frac{3\sqrt{3}(k_{c1}+k_{c2})}{64}\frac{\overline{\|G_\sigma\|\|W\|}}{\sqrt{\gamma_1}} + \frac{\overline{\|\nabla W G_R \nabla\sigma^T\|}}{2\lambda_{\min}\{K_a\}}$, $\varphi_{ac} \triangleq k_{a1} + \frac{3\sqrt{3}(k_{c1}+k_{c2})}{64}\frac{\overline{\|G_\sigma\|\|W\|}}{\sqrt{\gamma_1}} + \frac{\overline{\|\nabla W\|\|G_R\|\|\nabla\sigma^T\|}}{2\underline{\Gamma}_c}$, and $\varphi_{c\theta} \triangleq \frac{3\sqrt{3}(k_{c1}+k_{c2})\left(\overline{\|W\|\|\nabla\sigma\|\|S\|}\left(1+\frac{1}{k_d}+\overline{\|g\|\|g^+\|}\right)\right)}{16\sqrt{\gamma_1}}$. Furthermore, the constant $\iota \in \mathbb{R}_{>0}$ is defined as $\iota \triangleq \frac{1}{2}\overline{\|\nabla V^*\|\|G_R\|\|\nabla W^T\sigma + \nabla\epsilon^T\|} + \frac{(\iota_{a1}+\iota_{a2})^2}{(k_{a1}+k_{a2})} + \frac{\iota_c^2}{k_{c2}\underline{c}} + \frac{k_\theta \upsilon_1^2}{c_{\theta 1}}$, where $\iota_c \triangleq \frac{\overline{\|\nabla W\|}\left(1+\frac{1}{k_d}+\overline{\|g\|\|g^+\|}\right)\left(\overline{\theta S}+\varepsilon\right)}{\underline{\Gamma}_c} + \frac{\overline{\|\nabla W\|\|G_R\|\|\nabla\sigma^T W\|}}{2\underline{\Gamma}_c} + \frac{3\sqrt{3}(k_{c1}+k_{c2})\overline{\|\Delta\|}}{16\sqrt{\gamma_1}}$, $\iota_{a1} \triangleq$

$\frac{\overline{\|\nabla W\|}\left(1+\frac{1}{k_d}+\overline{\|g\|}\overline{\|g^+\|}\right)\left(\overline{\theta S}+\overline{\varepsilon}\right)}{\lambda_{\min}\{K_a\}} + \frac{\overline{\|\nabla W\|}\overline{\|G_R\|}\overline{\|\nabla\sigma^T W\|}}{2\lambda_{\min}\{K_a\}}$, and $\iota_{a2} \triangleq k_{a2}\overline{\|W\|} + \frac{3\sqrt{3}(k_{c1}+k_{c2})}{64}\frac{\overline{\|G_\sigma\|}\overline{\|W\|}^2}{\sqrt{\gamma_1}} + \frac{\overline{\|\nabla V^*\|}\overline{\|G_R\|}\overline{\|\nabla\sigma\|}}{2}$.