Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Approximate optimal influence over an agent through an uncertain interaction dynamic $\ensuremath{^{\ensuremath{\alpha}}}$

Patryk Deptula^{a,*}, Zachary I. Bell^b, Federico M. Zegers^b, Ryan A. Licitra^c, 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

^c Lockheed Martin Missiles and Fire Control, Orlando, FL, 32819

ARTICLE INFO

Article history: Received 15 February 2019 Received in revised form 4 May 2021 Accepted 24 July 2021 Available online xxxx

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.

© 2021 Elsevier Ltd. All rights reserved.

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 reachavoid games. Game theory is concerned with the analysis of strategies that players can take based on particular conditions (von Neumann & Morgenstern, 1980). 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., Chen, Zha, Peng, & Gu, 2016; Chen, Zhou, & Tomlin, 2017; Chung, Hollinger, & Isler, 2011; Garcia, Casbeer, & Pachter, 2017, 2018; Isaacs, 1967, 1999; Kachroo, Shedied, Bay, & Vanlandingham, 2001; Khalafi & Toroghi, 2011; Kumkov, Le Ménec, & Patsko, 2017; Ramana & Kothari, 2017; Vidal, Shakernia, Kim, Shim, & Sastry, 2002; Yan, Jiang, Di, Jiang,

E-mail addresses: pdeptula@draper.com (P. Deptula), bellz121@ufl.edu (Z.I. Bell), fredzeg@ufl.edu (F.M. Zegers), ryan.a.licitra@lmco.com (R.A. Licitra), wdixon@ufl.edu (W.E. Dixon).

https://doi.org/10.1016/j.automatica.2021.109913 0005-1098/© 2021 Elsevier Ltd. All rights reserved. & Hao, 2018 and references therein). Several different approaches have been considered in pursuit-evasion games. For instance, works such as Chen et al. (2016) consider multi-player pursuitevasion capture conditions, and cooperative control strategies are calculated for pursuing agents to capture evading agents. Results such as Ramana and Kothari (2017) develop escape strategies for evading agents using mathematical frameworks based on Apollonius circles. Results such as Yan et al. (2018) 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 Chen et al. (2017) and Huang, Ding, Zhang, and Tomlin (2015) 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 (HII) 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





automatica



tica

 $[\]stackrel{\circ}{\sim}$ The material in this paper was partially presented at the 57th IEEE Conference on Decision and Control, December 17–19, 2018, Miami Beach, Florida, USA. This paper was recommended for publication in revised form by Associate Editor Zhihua Qu under the direction of Editor Daniel Liberzon.

Corresponding author.

problems, called herding, focus on directing uncontrolled agents to a goal location and have been investigated in results such as Bacon and Olgac (2012), Gadre (2001), Kachroo et al. (2001), Licitra, Bell, Doucette, and Dixon (2018), Licitra, Hutcheson, Doucette, and Dixon (2017), Lu (2006), Pierson and Schwager (2018), Shedied (2002). 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 roaming 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 Bacon and Olgac (2012), Gadre (2001), Kachroo et al. (2001), Licitra et al. (2018, 2017), Lu (2006), Pierson and Schwager (2018), Shedied (2002) to leverage such behaviors in controlling autonomous systems. Motivated by such results, the authors in Licitra et al. (2017) and Licitra et al. (2018) approach the indirect regulation (also known as indirect herding) problem via a switched-systems approach, where the influencing agent switches between target agents. In Licitra et al. (2017), a robust sliding mode approach is used to compensate for worst-case uncertainties of the target agent dynamics. Compared to Licitra et al. (2017), the result in Licitra et al. (2018) 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., Parikh, Kamalapurkar, & Dixon, 2019). In Pierson and Schwager (2018) and Bacon and Olgac (2012), 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 Chen et al. (2016), Ramana and Kothari (2017), Yan et al. (2018) generally consider point mass systems and know the form of the agent dynamics, while results such as Bacon and Olgac (2012), Licitra et al. (2018, 2017), Pierson and Schwager (2018) 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 Kachroo et al. (2001), Lu (2006), Shedied (2002) compute optimal policies for pursuing agents regulating multiple evading agents with known point mass dynamics. Specifically, the result in Lu (2006) uses the Sparse Nonlinear Optimizer (SNOPT) algorithm in Gill, Murray, and Saunders (2005), Gill, Wong, Murray, and Saunders (2015) to compute numerical solutions offline. The works in Kachroo et al. (2001), Shedied (2002) 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 Kachroo et al. (2001), Lu (2006), Shedied (2002) 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 neuralnetworks (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 (Kamalapurkar, Walters, Rosenfeld, & Dixon, 2018;

Kiumarsi, Vamvoudakis, Modares, & Lewis, 2017: Vamvoudakis, Modares, Kiumarsi, & Lewis, 2017; Zhang, Liu, Luo, & Wang, 2013). 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 Kamalapurkar, Walters, and Dixon (2016), Vamvoudakis et al. (2017), Wen, Ge, and Tu (2018) 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 (Deptula, Rosenfeld, Kamalapurkar, & Dixon, 2018; Kamalapurkar, Rosenfeld & Dixon, 2016; Rosenfeld, Kamalapurkar, & Dixon, 2019) 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., Kamalapurkar, Klotz, Walters, & Dixon, 2018; Modares, Lewis, Kang, & Davoudi, 2018; Sun, Liu, & Ye, 2017; Vamvoudakis & Hespanha, 2018; Vamvoudakis et al., 2017; Wang, Liu, Mu, & Ma, 2016; Zhang, Cui, & Luo, 2013). However, all of these results solve the multi-player problem by generating controllers and directly controlling each agent. Exceptions include the innovative work in Wen et al. (2018) and Wang, Liu, Liu, Li, and Wang (2017). Specifically, an ADPbased backstepping approach is developed in Wen et al. (2018) and Wang et al. (2017) for a class of known strict-feedback nonlinear systems containing a one-dimensional input. In Wang et al. (2017), for each individual step of the backstepping approach, a virtual control is obtained using the Sontag formula (Lin & Sontag, 1991) which is equivalent to the optimal control. While in Wen et al. (2018), 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 Wen et al. (2018) and Wang et al. (2017) 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 (Kamalapurkar, Walters, Rosenfeld et al., 2018) 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 Licitra et al. (2017) and Licitra et al. (2018), the drift dynamics of both agents are assumed to be unknown and an approximately optimal control strategy is developed. Compared to results such as Chen et al. (2016), Ramana and Kothari (2017), Yan et al. (2018), 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 Kachroo et al. (2001), Lu

(2006), Shedied (2002), the policy for the influencing agent is calculated online and does not use one-step discrete actions, and while the results in Kachroo et al. (2001), Lu (2006), Shedied (2002) 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 Deptula, Licitra, and Dixon (2018), this work includes redefined error signals. In addition, compared to the preliminary results in Deptula, Bell et al. (2018) 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 timederivative 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. Fig. 1 illustrates the actor-critic-identifier architecture.

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.



Fig. 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.

The partial derivative of k with respect to the state x is denoted by $\nabla k(x, y, ...)$, the transpose of a matrix or vector is denoted by $(\cdot)^T$, and the trace of a square matrix is denoted as tr (\cdot) . The vectorization operator of a matrix $A = [a_1, a_2, ..., a_m] \in \mathbb{R}^{n \times m}$ is denoted by vec $(A) \triangleq [a_1^T, a_2^T, ..., a_m^T]^T$, 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 $U[a, b] \mathbf{1}_{n \times m}$ 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 the 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 (Deptula, Bell et al., 2018; Licitra et al., 2018, 2017). 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

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

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

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 Chen et al. (2016, 2017), Huang et al. (2015), Licitra et al. (2018, 2017), Ramana and Kothari (2017), Yan et al. (2018).

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) stems from behavior where the roaming agent tries to flee from pursuing the agent; hence its dynamics are also dependent on the pursuer agent's state. The form of the dynamics in (2) is 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 $\overline{\alpha}_1, \overline{\alpha}_2 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ 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 Licitra et al. (2018, 2017), 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 $g^+ \triangleq (g^Tg)^{-1}g^T$ (Kamalapurkar, Dinh, Bhasin, & Dixon, 2015).

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 (Kamalapurkar, Andrews, Walters,& Dixon, 2017; Kamalapurkar et al., 2015; Kamalapurkar, Klotz et al., 2018).

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

$$e_{\eta}(t) \triangleq \eta(t) - \eta_{d}(t), \qquad (4)$$

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

$$e_{d}\left(t\right) \triangleq \eta_{d}\left(t\right) - z_{g} - k_{d}e_{z}\left(t\right),\tag{5}$$

where $k_d \in \mathbb{R}$ is positive constant control gain, which is generally selected as $k_d \ge 1$. The error signals in (4) and (5) have been modified from the preliminary result in Deptula, Bell et al. (2018), and

resemble those of backstepping approaches such as in Licitra et al. (2018); however, compared to Licitra et al. (2018), 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

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

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

$$\dot{e}_{\eta}(t) = h(z(t), \eta(t)) + g(\eta(t)) \mu_{\eta}(t) + g(\eta(t)) u_{d}(t) - \mu_{d}(t),$$
(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

$$u_d(t) \triangleq g\left(\eta_d(t)\right)^+ \left(\mu_d(t) - h\left(z\left(t\right), \eta_d\left(t\right)\right)\right).$$
(8)

Substituting (8) into (7) yields the following closed-loop system $\dot{e}_n(t) = h(z(t), \eta(t)) + g(\eta(t)) \mu_n(t) - g(\eta(t))g^+(\eta_d(t))$ $h(z(t), \eta_d(t)) + (g(\eta(t))g^+(\eta_d(t)) - I_n)\mu_d(t)$. 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 $x(t) \triangleq \left[e_z^T(t), e_d^T(t)\right]$ $e_{\eta}^{T}(t)$ ^T and $x_{d}(t) \triangleq [e_{z}^{T}(t), e_{d}^{T}(t), \mathbf{0}_{1 \times n}]^{T}$ denote the concate-nated error state and desired concatenated state, respectively. In addition, we define the mappings s_1, s_2 : $\mathbb{R}^{3n} \rightarrow \mathbb{R}^n$ as $s_1(x(t)) \triangleq e_z(t) + z_g$, and $s_2(x(t)) \triangleq e_\eta(t) + e_d(t) + k_d e_z(t) + k_d e_z(t)$ 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 $||f(s_1(x(t)), s_2(x(t)))|| \leq$ $\overline{\alpha}_1\left(\left\|\left(k_d-1\right)e_z\left(t\right)+e_\eta\left(t\right)+e_d\left(t\right)\right\|\right) + \overline{\alpha}_2\left(\left\|e_z\left(t\right)\right\|\right). \text{ Hence, if }$ $e_{z}(t), e_{\eta}(t), e_{d}(t) \to 0$ then $||f(s_{1}(x(t)), s_{2}(x(t)))|| \le \overline{\alpha}_{1}(0) +$ $\overline{\alpha}_2(0)$.

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

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

where $\mu(t) \triangleq \begin{bmatrix} \mu_{\eta}^{T}(t) & \mu_{d}^{T}(t) \end{bmatrix}^{T} \in \mathbb{R}^{m}$ 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))

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

$$\begin{bmatrix} 0_{n \times m_{\eta}}, & 0_{n \times n}, \\ 0_{n \times m_{\eta}}, & I_{n}, \\ g(s_{2}(x(t))), & G_{sd}(x(t)), \end{bmatrix} \text{ where } F_{sd}(x(t)) \triangleq g(s_{2}(x(t)))$$

$$g^{+}(s_{2}(x_{d}(t))) \cdot h(s_{1}(x(t)), s_{2}(x_{d}(t))), \text{ and } G_{sd}(x(t)) \triangleq$$

 $g(s_2(x_1(t))) = g(s_2(x_1(t)))^+ - I_n.$

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.

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 $z(t) \rightarrow z_g$, $\eta(t) \rightarrow \eta_d(t)$, and $\eta_d(t) \rightarrow z_g$ by the use of (3)–(5). This implies that $\eta(t) \rightarrow z_g$.

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

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

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

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

In (11), $Q : \mathbb{R}^{3n} \to \mathbb{R}_{\geq 0}$ is a user-defined continuous positivedefinite (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 $R = \text{diag} \{R_{\eta}, R_d\}$ 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 $P(x) = e^{\frac{1}{2a}(p_x(x))^2} - 1$, $P(x) = \alpha (p_x(x))^2$, $P(x) = \alpha \ln (\cosh(p_x(x))^2)$, where $p_x(x) \triangleq \max \{0, ||(1 - k_a)e_z - e_\eta - e_d||^2 - r_a(x)^2\}$, and $\alpha \in \mathbb{R}_{>0}$. Piecewise continuous smooth functions that saturate at a constant may also be used.

Definition 1 (*Wen et al., 2018*). Let $\Omega \subseteq \mathbb{R}^n$ be a set containing the origin x = 0. A control policy $\mu(x(t))$ is said to be admissible with respect to (9) in Ω , i.e., $\mu(x(t)) \in U(\Omega) \subset \mathbb{R}^m$, if $\mu(x(t))$ is continuous in Ω with $\mu(0) = 0_{m \times 1}$, $\mu(x(t))$ stabilizes (9) in Ω , and $V(x(t)) \triangleq J(x(t), \mu(x(t)))$ is finite.

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

$$V^*(\mathbf{x}(t)) = \inf_{\mu(\tau) \in U \mid \tau \in \mathbb{R}_{\ge t}} \int_t^\infty r(\mathbf{x}(\tau), \mu(\tau)) d\tau.$$
(12)

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

$$0 = \nabla V^* (x(t)) \left(F(x(t)) + G(x(t)) \mu^* (x(t)) \right) + r \left(x(t), \mu^* (x(t)) \right),$$
(13)

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

$$\mu^*(\mathbf{x}(t)) = -\frac{1}{2} R^{-1} G(\mathbf{x}(t))^T \left(\nabla V^*(\mathbf{x}(t)) \right)^T.$$
(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 Fig. 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 $k_df(z(t), \eta(t))$ and $h(z(t), \eta(t))$. Since these functions are unknown, we approximately minimize the cost function in (10) while simultaneously learning these functions as shown in Fig. 1. Various methods could be employed to learn the functions (cf., Basu Roy, Bhasin, & Kar, 2019; Cho, Shin, Kim, & Tsourdos, 2017; Kamalapurkar, Reish, Chowdhary, & Dixon, 2017; Parikh et al., 2019; Roy, Bhasin, & Kar, 2018). The following is based on the ICL strategy in Parikh et al. (2019). Using the universal function approximation property of single layer NNs (Farrell & Polycarpou, 2006; Lewis, Selmic, & Campos, 2002; Sadegh, 1993), (1) and (2) can be represented as

$$\check{\mathbf{x}}(t) = S(\mathbf{x}(t))\theta + \varepsilon(\mathbf{x}(t)) + \check{G}(\mathbf{x}(t), u(t)), \qquad (15)$$

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

Assumption 3. There exist $\overline{\theta}$, \overline{S} , $\overline{\varepsilon} \in \mathbb{R}_{>0}$ such that $\|\theta\| \leq \overline{\theta}$, $\sup_{x \in \chi} \|S(x)\| \leq \overline{S}$, and $\sup_{x \in \chi} \|\varepsilon(x)\| \leq \overline{\varepsilon}$ in a compact set $\chi \subseteq \mathbb{R}^{3n}$ (Kamalapurkar, Walters, Dixon, 2016; Lewis et al., 2002).

Based on the ICL strategy in Parikh et al. (2019), let $\Delta t_{\theta} \in \mathbb{R}_{>0}$ denote an integration time-window, where the integral of (15) at time $t_i \in [\Delta t_{\theta}, t]$ can be represented as $\check{x}(t_i) - \check{x}(t_i - \Delta t_{\theta}) =$ $S_i\theta + \mathcal{E}_i + \mathcal{G}_i$ such that $S_i = S(t_i) \triangleq \int_{t_i - \Delta t_{\theta}}^{t_i} S(x(\tau)) d\tau$, $\mathcal{E}_i = \mathcal{E}(t_i) \triangleq$ $\int_{t_i - \Delta t_{\theta}}^{t_i} \varepsilon(x(\tau)) d\tau$, and $\mathcal{G}_i = \mathcal{G}(t_i) \triangleq \int_{t_i - \Delta t_{\theta}}^{t_i} \check{G}(x(\tau), u(\tau)) d\tau$. A least-squares based parameter estimate update law is designed as

$$\dot{\hat{\theta}}(t) = k_{\theta} \Gamma_{\theta}(t) \sum_{i=1}^{M} S^{T}(t_{i}) \left(\check{x}(t_{i}) - \check{x}(t_{i} - \Delta t_{\theta}) \right)$$

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

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.

P. Deptula, Z.I. Bell, F.M. Zegers et al.

$$-\mathcal{G}(t_i) - \mathcal{S}(t_i)\hat{\theta}\Big),\tag{16}$$

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

where k_{θ} , $\beta_{\theta} \in \mathbb{R}_{>0}$ 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, $M \ge \frac{p}{2}$ is selected to ensure enough data is gathered to facilitate learning, i.e., $\sum_{i=1}^{M} S^{T}(t_{i}) S(t_{i})$ 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 $\lambda_1 I_p \leq \sum_{i=1}^M S_i^T S_i$, $\forall t \geq T_1$ (Parikh et al., 2019).

Provided $\lambda_{\min} \{ \Gamma_{\theta}^{-1}(t_0) \} > 0$, 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 loannou and Sun (1996, Corollary 4.3.2), where $\underline{\Gamma}_{\theta}, \overline{\Gamma}_{\theta} \in \mathbb{R}_{>0}$. Let $Z_{\theta}(t) = \operatorname{vec}(\tilde{\theta}(t))$ 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

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

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

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

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

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

Completing the squares, using the bounds on (18), and invoking the Comparison Lemma (Khalil, 2002, Lemma 3.4) yields

$$V_{\theta} \left(Z_{\theta} \left(t \right), t \right) \leq V_{\theta} \left(Z_{\theta} \left(t_{0} \right), t_{0} \right) e^{-\lambda_{\theta} \frac{k_{\theta} \epsilon_{\theta} 1 \underline{L}_{\theta}}{2} \left(t - t_{0} \right)} + \left(1 - e^{-\frac{k_{\theta} c_{\theta} 1 \underline{L}_{\theta}}{2} \left(t - t_{0} \right)} \right) \frac{2 \upsilon_{1}^{2}}{\underline{\Gamma}_{\theta} c_{\theta}^{2}},$$

$$(21)$$

for all $t \in [t_0, T_1)$. Then, $||Z_{\theta}(t)||^2 \leq \frac{\overline{\Gamma}_{\theta}}{\underline{\Gamma}_{\theta}} \left(||Z_{\theta}(t_0)||^2 + \frac{4v_1^2}{c_{\theta_1}^2} \right)$ follows for all $t \in \mathbb{R}_{\geq t_0}$. After $\sum_{i=1}^{M} S_i^T S_i$ becomes full rank, (16), (17), and Assumption 4 are used in the time-derivative of (18) to yield

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

for all $t \ge T_1$. Using the Comparison Lemma (Khalil, 2002, Lemma 3.4), $\forall t \ge T_1$

$$V_{\theta} \left(Z_{\theta} \left(t \right), t \right) \leq V_{\theta} \left(Z_{\theta} \left(T_{1} \right), T_{1} \right) e^{-\lambda_{\theta} \left(t - T_{1} \right)} + \left(1 - e^{-\lambda_{\theta} \left(t - T_{1} \right)} \right) \frac{c_{B}}{2 \underline{\Gamma}_{\theta}}.$$

$$(23)$$

From (21), $V_{\theta}(Z_{\theta}(T_1), T_1) \leq V_{\theta}(Z_{\theta}(t_0), t_0) e^{-\lambda_{\theta} \frac{k_{\theta}c_{\theta_1} \underline{\Gamma}_{\theta}}{2}(t-t_0)} + \frac{2v_1^2}{\underline{\Gamma}_{\theta}c_{\theta_1}^2}$ 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 $\sum_{i=1}^{M} S_i^T S_i$ 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., $\frac{2v_1^2}{\underline{\Gamma}_{\theta}c_{\theta_1}^2}$), because c_B depends on c_{θ_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 Kamalapurkar, Reish et al. (2017) and Chowdhary and Johnson (2011), 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 ε (x (t)) in (15); however, if an exact basis is known in the neural network, then ε (x (t)) = $0_{2\times n}$. Therefore, the bound in (19) can be reduced to $||Z_{\theta}(t)|| \le c_{\Gamma} \sqrt{c_{M}} e^{-\lambda_{\theta}(t-T_{1})}$, which produces convergence to the origin. In addition, results such as Bhasin, Kamalapurkar, Dinh, and Dixon (2013), Bhasin, Kamalapurkar, Johnson, Vamvoudakis, Lewis, and Dixon (2013), Patre, MacKunis, Kaiser, and Dixon (2008) 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 (Deptula, Rosenfeld et al., 2018; Kamalapurkar, Rosenfeld et al., 2016; Rosenfeld et al., 2019). 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 $c(x) \in (\overline{B_r(x)})^L$, the value function in (12) can be represented as

$$V^{*}(y) = W(x)^{T} \sigma(y, c(x)) + \epsilon_{v}(x, y), \qquad (24)$$

where $y \triangleq \left[y_{e_2}^T, y_{e_d}^T, y_{e_\eta}^T\right]^T \in \overline{B_r(x)}$ represents a composite state vector in the neighborhood of x (Deptula, Rosenfeld et al., 2018; Kamalapurkar, Rosenfeld et al., 2016; Rosenfeld et al., 2019), and the states y_{e_2} , 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

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

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

$$\begin{split} \delta\left(y, x, \hat{\theta}, \hat{W}_{c}, \hat{W}_{a}\right) &= Q\left(y\right) + P\left(y\right) + \Psi\left(\hat{\mu}\left(y, x, \hat{W}_{a}\right)\right) \\ &+ \nabla \hat{V}\left(y, x, \hat{W}_{c}\right)\left(F_{1}\left(y, \hat{\theta}\right) - F_{2}\left(y, \hat{\theta}\right)\right) \\ &+ \nabla \hat{V}\left(y, x, \hat{W}_{c}\right)\left(G\left(y\right)\hat{\mu}\left(y, x, \hat{W}_{a}\right)\right), \end{split} (27) \\ \end{split}$$
where $F_{1}\left(y, \hat{\theta}\right) \triangleq \left[\left(\frac{1}{k_{d}}\hat{\theta}_{z}^{T}S_{z}\left(y\right)\right)^{T}, \left(-\hat{\theta}_{z}^{T}S_{z}\left(y\right)\right)^{T}, \left(\hat{\theta}_{\eta}^{T}S_{\eta}\left(y\right)\right)^{T}\right]_{T}^{T}, \\ F_{2}\left(y, \hat{\theta}\right) \triangleq \left[\mathbf{0}_{1 \times 2n}, \left(g\left(y_{\eta}\right)g^{+}\left(y_{\eta_{d}}\right)\hat{\theta}_{\eta}^{T}S_{\eta}\left(\left[y_{e_{z}}^{T}, y_{e_{d}}^{T}\mathbf{0}_{n \times 1}^{T}\right]^{T}\right)\right)^{T}\right]^{T}. \end{split}$

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 $\hat{u}_{d}\left(y, x, \hat{\theta}, \hat{W}_{a}\right) \triangleq g^{+}\left(y_{n_{d}}\right)\left(\hat{\mu}_{d}\left(y, x, \hat{W}_{a}\right) - S_{\eta}\left(\left[y_{e_{z}}^{T}, y_{e_{d}}^{T}, 0_{n\times1}^{T}\right]^{T}\right)\hat{\theta}_{\eta}\right)$, 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

$$\delta_{t}(t) \triangleq \delta\left(x(t), x(t), \hat{\theta}(t), \hat{W}_{c}(t), \hat{W}_{a}(t)\right),$$
(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., Kamalapurkar, Andrews et al., 2017; Kamalapurkar et al., 2015; Wang et al., 2017, 2016). In contrast to injecting a probing signal, learning via simulation of experience is performed by extrapolating the BE to unexplored states in $B_{r}(x(t))$. Moreover, sets of functions $\{x_{i} : \mathbb{R}^{3n} \times \mathbb{R}_{\geq t_{0}} \to \mathbb{R}^{3n}\}_{i=1}^{N}$ 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 $x_i(x(t), t) = x(t) + U[a, b] \mathbf{1}_n$, where $a_k, b_k, a, b \in \mathbb{R}$ are constants). The specified approach is a design variable; however, results such as Deptula, Rosenfeld et al. (2018), Kamalapurkar, Rosenfeld et al. (2016), Walters, Kamalapurkar, and Dixon (2015) 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

$$\dot{\hat{W}}_{c}(t) \triangleq -\Gamma_{c}(t) \left(k_{c1} \frac{\omega(t)}{\rho^{2}(t)} \delta_{t}(t) + \frac{k_{c2}}{N} \sum_{i=1}^{N} \frac{\omega_{i}(t)}{\rho_{i}^{2}(t)} \delta_{ti}(t)\right), \quad (29)$$

$$\dot{\Gamma}_{c}(t) \triangleq \beta_{c} \Gamma_{c}(t) - \Gamma_{c}(t) k_{c1} \frac{\omega(t) \omega^{T}(t)}{\rho^{2}(t)} \Gamma_{c}(t) - \Gamma_{c}(t) \frac{k_{c2}}{N} \sum_{i=1}^{N} \frac{\omega_{i}(t) \omega_{i}^{T}(t)}{\rho_{i}^{2}(t)} \Gamma_{c}(t), \qquad (30)$$

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, $\omega(t) = \nabla \sigma(x(t), c(x(t))) \left(F_1(x(t), \hat{\theta}(t)) - F_2(x(t), \hat{\theta}(t)) + G(x(t)) \hat{\mu}(x(t), x(t), \hat{\psi}(t)) \right)$, and $\omega_i(t) = \nabla \sigma_i \left(F_{1i} - F_{2i} + G_i \hat{\mu}_i \right)$, with $\nabla \sigma_i \triangleq \nabla \sigma$ $(x_i(x(t), t), c(x(t))), \ F_{1i} \triangleq F_1 \left(x_i(x(t), t), \hat{\theta}(t) \right), \ F_{2i} \triangleq$ $F_2 \left(x_i(x(t), t), \hat{\theta}(t) \right), \ G_i \triangleq G(x_i(x(t), t)), \ and \ \hat{\mu}_i \triangleq \hat{\mu} \left(x_i(x(t), t), x(t), \hat{\psi}_a(t) \right).$

Using Assumption 5 along with $\lambda_{\min} \{\Gamma_c^{-1}(t_0)\} > 0$, a similar argument to loannou and Sun (1996, 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 (Kamalapurkar, Rosenfeld et al., 2016).

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

$$\begin{aligned} \dot{\hat{W}}_{a}(t) &= -K_{a}k_{a1}\left(\hat{W}_{a}(t) - \hat{W}_{c}(t)\right) - K_{a}k_{a2}\hat{W}_{a}(t) \\ &+ K_{a}\frac{k_{c1}}{4}G_{\sigma}^{T}(t)\hat{W}_{a}(t)\frac{\omega^{T}(t)}{\rho^{2}(t)}\hat{W}_{c}(t) \\ &+ K_{a}\frac{k_{c2}}{4N}\sum_{i=1}^{N}G_{\sigma i}^{T}(t)\hat{W}_{a}(t)\frac{\omega_{i}^{T}(t)}{\rho_{i}^{2}(t)}\hat{W}_{c}(t), \end{aligned}$$
(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 Deptula, Bell et al. (2018), Deptula, Rosenfeld et al. (2018), Kamalapurkar, Rosenfeld et al. (2016), Kamalapurkar, Walters, Rosenfeld et al. (2018), 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

$$\begin{split} \underline{c}_{1}I_{L} &\leq \inf_{t \in \mathbb{R} \geq t_{0}} \frac{1}{N} \sum_{i=1}^{N} \frac{\omega_{i}\left(t\right) \omega_{i}^{T}\left(t\right)}{\rho_{i}^{2}\left(t\right)}, \\ \underline{c}_{2}I_{L} &\leq \int_{t}^{t+T_{2}} \left(\frac{1}{N} \sum_{i=1}^{N} \frac{\omega_{i}\left(\tau\right) \omega_{i}^{T}\left(\tau\right)}{\rho_{i}^{2}\left(\tau\right)}\right) d\tau, \ \forall t \in \mathbb{R}_{\geq t_{0}}, \\ \underline{c}_{3}I_{L} &\leq \int_{t}^{t+T_{2}} \left(\frac{\omega\left(\tau\right) \omega^{T}\left(\tau\right)}{\rho^{2}\left(\tau\right)}\right) d\tau, \ \forall t \in \mathbb{R}_{\geq t_{0}}, \\ \text{where } T_{i} \text{ and at least one of the constants } c_{i}, c_{i} \text{ or } c_{i} \end{cases}$$

where T_2 and at least one of the constants \underline{c}_1 , \underline{c}_2 , or \underline{c}_3 is strictly positive (Kamalapurkar, Rosenfeld et al., 2016).

Remark 6. As stated in Deptula, Rosenfeld et al. (2018), Kamalapurkar, Rosenfeld et al. (2016), Kamalapurkar, Walters, Rosenfeld et al. (2018), \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 $\underline{c}_1 >$ 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, δ_t (t) and δ_{ti} (t), are

$$\delta_{t} = -\omega^{T} \tilde{W}_{c} + \frac{1}{4} \tilde{W}_{a}^{T} G_{\sigma} \tilde{W}_{a} - W^{T} \nabla \sigma \left(\tilde{F}_{1} - \tilde{F}_{2} \right) + \Delta \left(x \right),$$

$$\delta_{ti} = -\omega_{i}^{T} \tilde{W}_{c} + \frac{1}{4} \tilde{W}_{a}^{T} G_{\sigma i} \tilde{W}_{a} - W^{T} \nabla \sigma_{i} \left(\tilde{F}_{1i} - \tilde{F}_{2i} \right) + \Delta_{i}, \qquad (32)$$

where $\tilde{F}_1 \triangleq F_1(x, \tilde{\theta})$, $\tilde{F}_2 \triangleq F_2(x, \tilde{\theta})$, $\tilde{F}_{1i} \triangleq F_1(x_i, \tilde{\theta})$, $\tilde{F}_{2i} \triangleq F_2(x_i, \tilde{\theta})$, 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 $Z_L \triangleq \begin{bmatrix} x^T, & \tilde{W}_c^T, & \tilde{W}_a^T, & Z_\theta^T \end{bmatrix}^T$ 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

$$V_{L}(Z_{L},t) \triangleq V^{*}(x) + \frac{1}{2}\tilde{W}_{c}^{T}\Gamma_{c}^{-1}(t)\tilde{W}_{c} + \frac{1}{2}\tilde{W}_{a}^{T}K_{a}^{-1}\tilde{W}_{a} + V_{\theta}(Z_{\theta},t), \qquad (33)$$

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

$$\underline{v}_{l}\left(\|Z_{L}\|\right) \leq V_{L}\left(Z_{L},t\right) \leq \overline{v}_{l}\left(\|Z_{L}\|\right) \tag{34}$$

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, $\lambda_{min} \{H\} > 0$, and

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

where
$$H \triangleq \begin{bmatrix} \left(\frac{k_{a1}+k_{a2}}{4}-\varphi_a\right) & -\frac{\varphi_{ac}}{2} & \mathbf{0} \\ -\frac{\varphi_{ac}}{2} & \left(\frac{\beta_c}{\Gamma_c}+k_{c2}\varepsilon_1\right) & -\frac{\varphi_{c\theta}}{2} \\ \mathbf{0} & -\frac{\varphi_{c\theta}}{2} & \frac{k_{b}c_{\theta}}{8} \end{bmatrix}$$
, and κ, φ_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

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

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

$$-\frac{1}{2}\tilde{W}_{c}^{T}\left(\Gamma_{c}^{-1}\dot{\Gamma}_{c}\Gamma_{c}^{-1}\right)\tilde{W}_{c}+\tilde{W}_{a}^{T}K_{a}^{-1}\left(\dot{W}-\dot{\hat{W}}_{a}\right).$$

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

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

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 $\dot{V}_L \leq -\kappa \|Z_L\|^2 - \kappa \|Z_L\|^2 + \iota - Z_v^T H Z_v$, where $Z_v \triangleq \left[\|\tilde{W}_a\|, \|\tilde{W}_c\|, \|Z_\theta\| \right]^T$. Provided $\lambda_{\min} \{H\} > 0$ is met, then for all $Z \in B_{\zeta}$

$$\dot{V}_{L} \leq -\kappa \|Z_{L}\|^{2}, \ \forall \|Z_{L}\| \geq \sqrt{\frac{\iota}{\kappa}} > 0.$$
(37)

Using (34), (35), and (37), Khalil (2002, Theorem 4.18) is invoked to conclude that all trajectories $Z_L(t)$ that satisfy $||Z_L(t_0)|| \leq \overline{v_l}^{-1}(\underline{v_l}^{-}(\zeta))$ 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 $W(x) \in \mathcal{L}_{\infty}$. 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 , $(z - \eta) \in \mathcal{L}_{\infty}$. Finally, since μ , $\tilde{\theta}$, 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., $\sqrt{\frac{L}{\kappa}}$). Moreover, the smaller the value of $\sqrt{\frac{L}{\kappa}}$, the smaller the ultimate bound. Hence, provided the sufficient conditions can be selected such that $\sqrt{\frac{L}{\kappa}} \rightarrow 0$, then the ultimate bound converges to the origin Khalil (2002, Theorem 4.18).

Remark 8. The sufficient condition $\lambda_{\min} \{H\} > 0$ 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 $x_i (x(t), t)$ 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 $f(z(t), \eta(t)) = (Ae(t) + Be_z(t)) \exp(-\frac{1}{2}e(t)^T e(t))$, where $e(t) = z(t) - \eta(t)$, and, without a loss of generality, $h(z(t), \eta(t)) = 0_{2 \times 1}$, $g(\eta(t)) = I_2$, respectively. Based on these dynamics, Assumptions 1 and 2 are satisfied since $||f(z(t), \eta(t))|| \le ||A|| ||e(t)|| + ||B|| ||e_z(t)||$ and I_2 is full-rank. The unknown parameters to be 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 $S_n(x(t)) =$ e(t) and $S_{z}(x(t)) = \exp(-\frac{1}{2}e(t)^{T}e(t)) \times [e^{T}(t), e^{T}_{z}(t)]^{T}$. 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,

P. Deptula, Z.I. Bell, F.M. Zegers et al.

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
$Q(x) = x^{T}Q_{x}x, P(x) = \left(\max\left\{0, \left\ (1-k_{d})e_{z}(t) - e_{\eta}(t) - e_{d}(t)\right\ ^{2} - r_{a}^{2}\right\}\right)^{2},$
$Q_x = \text{diag} \{20, 20, 5, 5, 5, 5\}, R = 5I_4, k_d = 1.4, r_a = 0.15.$
Coins and nonemators for ADD undate laws

Gains and parameters for ADP update laws

 $k_{c1} = 0.9, \ k_{c2} = 0.02, \ k_{a1} = 0.25, \ k_{a2} = 0.005, \ \gamma_1 = 0.75,$ $\beta_c = 0.001, \ K_a = I_7, \ k_{\theta} = 0.5, \ \beta_{\theta} = 2, \ M = 100, \ N = 10.$



Fig. 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 Fig. 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 Fig. 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 Fig. 2c, $u_1(t)$ is represented by the squares and $u_2(t)$ is represented by lue squares and $u_2(t)$ is represented by red diamonds.

where S_{η} and S_z satisfy Assumption 3. 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 $\sigma_i(x(t), c_i(x(t))) = x^T(t)(x(t) + 0.7v(x(t))d_i), v(x(t)) = \frac{0.7x^T(t)x(t)}{(1+x^T(t)x(t))}$, and d_i are the vertices of a 6-simplex (Kamalapurkar, Rosenfeld et al., 2016; Rosenfeld et al., 2019; Walters et al., 2015). 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

Figs. 2–5 demonstrate that the influencing agent regulates the roaming agent to the goal location z_g . Fig. 2a shows that the concatenated state x(t) converges to the origin. Hence, both the system identification basis and value function weights remain bounded. Fig. 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 Fig. 2c, remains bounded and converges once the agent's reach the goal location. Figs. 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. Fig. 3c shows





Fig. 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 minimum eigenvalue of the regression matrix.



Fig. 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.



Fig. 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.

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. Fig. 4a shows that the system parameter estimates converge to the true values, while Fig. 4b shows when the history stack becomes positive definite, which validates Assumption 4. The positions of the roaming and influencing agents are shown in Fig. 5. The roaming agent is not independently motivated to go to the desired location; hence, in Fig. 5, the



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

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 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 Fig. 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 Monajjemi (2015) 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 120 Hz. A video of a typical run of this experiment is available at Deptula, Bell, Zegers, Licitra, and Dixon (2019).

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 $\sigma_i(x(t), c_i(x(t))) = \exp\left(\frac{x^T(t)c(x(t))}{\|x(0)\|^2}\right)$, $c(x(t)) = (x(t) + \|x(0)\| v(x(t)) d_i)$, $v(x(t)) = \frac{0.05x^T(t)x(t)}{x^T(t)}$ and d_t are the vertices of a 6 simpley. To perform

 $\frac{0.05x^{i}(t)x(t)}{(\|x(0)\|^{2}+x^{T}(t)x(t))}$, and d_{i} are the vertices of a 6-simplex. To perform

Table 2

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

BE extrapolation, ten trajectories $x_i(x(t), t)$ are selected at random from a uniform distribution over a $v(x(t)) \times v(x(t))$ 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.

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 rootmean 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.

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 Figs. 7 and 8, respectively. Figs. 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 Fig. 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, Fig. 8 shows similar metrics as in Fig. 7. Specifically, the norms of the concatenated state x(t)and regulation error $e_z(t)$ for experiment nine are displayed in Figs. 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. Fig. 8c shows the phase-space portrait for both agents. Compared to Fig. 7c, Fig. 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 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	diag {20, 10, 50, 25}	diag {1, 10, 1, 10, 1, 1}	26.25	4.0
2	diag {20, 5, 50, 25}	diag {1, 20, 1, 20, 1, 1}	25.0	7.33
3	diag {20, 5, 50, 25}	diag {1, 30, 1, 30, 1, 1}	25.0	10.67
4	diag {25, 10, 55, 30}	diag {1, 30, 1, 30, 1, 1}	30.0	10.67
5	diag {30, 20, 65, 40}	diag {1, 30, 1, 30, 1, 1}	38.75	10.67
6	diag {40, 30, 75, 50}	diag {1, 30, 1, 30, 1, 1}	48.75	10.67
7	diag {40, 30, 80, 55}	diag {1.5, 30, 1.5, 30, 1, 1}	51.25	10.83
8	diag {40, 30, 100, 75}	diag {1.5, 30, 1.5, 30, 1, 1}	61.25	10.83
9	diag {40, 30, 120, 100}	diag {1.5, 30, 1.5, 30, 1, 1}	72.5	10.83
10	diag {60, 40, 150, 125}	diag {1.5, 30, 1.5, 30, 1, 1}	93.75	10.83

Table 4

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

Experiment	Concatenated State Initial Norm	Regulation Error Initial Norm	Concatenated State Total RMS	Regulation Error Total RMS	Applied Input Total RMS	Total Cost	TTC
	x(0) (m)	$\ e_{z}(0)\ $ (m)	$ x _{RMS}$ (m)	$\ e_z\ _{RMS}$ (m)	$\ u\ _{RMS}$ $\left(\frac{\mathrm{m}}{\mathrm{sec}}\right)$	$(\times 10^{3})$	(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



Fig. 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 Fig. 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.

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



Fig. 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 Fig. 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 Fig. 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.

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 infinitehorizon 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 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.

Acknowledgments

This research is supported in part by NSF, USA award 1509516, Office of Naval Research, USA Grant N00014-13-1-0151, and AFOSR, USA 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, Draper, or Lockheed Martin Missiles and Fire Control.

Appendix. Auxiliary terms

To facilitate the analysis in Section 4, $\kappa \in \mathbb{R}_{>0}$ is defined as $\kappa \triangleq \min\left\{\frac{q}{2}, \frac{k_{0}c_{01}}{16}, \frac{k_{c2}\varepsilon}{8}, \frac{(k_{a1}+k_{a2})}{8}\right\}$, where the constants $\varphi_{a}, \varphi_{ac}, \varphi_{c\theta} \in \mathbb{R}_{>0}$ are defined as $\varphi_{a} \triangleq \frac{3\sqrt{3}(k_{c1}+k_{c2})}{64} \frac{\|\overline{C}_{\sigma}\|\|W\|}{\sqrt{r_{1}}} + \frac{\|\overline{\nabla W}_{\sigma} \nabla \sigma^{T}\|}{2\lambda_{\min}(k_{a})}, \varphi_{ac} \triangleq k_{a1} + \frac{3\sqrt{3}(k_{c1}+k_{c2})}{64} \frac{\|\overline{C}_{\sigma}\|\|W\|}{\sqrt{r_{1}}} + \frac{\|\overline{\nabla W}\|\|\overline{C}_{R}\|\|\nabla \sigma^{T}\|}{2\sum_{min}(k_{a})}$, and $\varphi_{c\theta} \triangleq \frac{3\sqrt{3}(k_{c1}+k_{c2})\left(\|\overline{W}\|\|\nabla \sigma\|\|S\|\left(1+\frac{1}{k_{d}}+\|\overline{g}\|\|g^{T}\|\right)\right)}{16\sqrt{r_{1}}}$. Furthermore, the constant $\iota \in \mathbb{R}_{>0}$ is defined as $\iota \triangleq \frac{1}{2} \|\overline{\nabla V^{*}}\|\|\overline{C}_{R}\|\|\nabla W^{T}\sigma + \nabla\epsilon^{T}\| + \frac{(\iota_{a1}+\iota_{a2})^{2}}{(k_{a1}+k_{a2})} + \frac{\iota_{c2}^{2}}{k_{c2}\varepsilon} + \frac{k_{\theta}\upsilon_{1}^{2}}{c_{\theta1}}$, where $\iota_{c} \triangleq \frac{\|\overline{\nabla W}\|\left(1+\frac{1}{k_{d}}+\|\overline{g}\|\|g^{T}\|\right)(\overline{\theta}S+\varepsilon)}{\sum_{k=1}^{C}} + \frac{\|\overline{\nabla W}\|\|\overline{C}_{R}\|\|\nabla\sigma^{T}W\|}{2\lambda_{\min}(K_{a})}$, $\iota_{a1} \triangleq \frac{\|\overline{\nabla W}\|\left(1+\frac{1}{k_{d}}+\|\overline{g}\|\|g^{T}\|\right)(\overline{\theta}S+\varepsilon)}{\lambda_{\min}(K_{a})} + \frac{\|\overline{\nabla W}\|\|\overline{C}_{R}\|\|\nabla\sigma^{T}W\|}{2\lambda_{\min}(K_{a})}$, and $\iota_{a2} \triangleq k_{a2}\|\overline{W}\| + \frac{3\sqrt{3}(k_{c1}+k_{c2})}{64}\frac{\|\overline{C}_{\sigma}\|\|W\|^{2}}{\sqrt{r_{1}}} + \frac{\|\overline{\nabla W}\|\|\overline{C}_{R}\|\|\nabla\sigma^{T}W\|}{2\lambda_{\min}(K_{a})}$.

References

- Bacon, M., & Olgac, N. (2012). Swarm herding using a region holding sliding mode controller. Journal of Vibration and Control, 18(7), 1056–1066.
- Basu Roy, S., Bhasin, S., & Kar, I. N. (2019). Composite adaptive control of uncertain euler-lagrange systems with parameter convergence without pe condition. *Asian Journal of Control.*
- Bhasin, S., Kamalapurkar, R., Dinh, H. T., & Dixon, W. (2013). Robust identification-based state derivative estimation for nonlinear systems. *IEEE Transactions on Automatic Control*, 58(1), 187–192.
- Bhasin, S., Kamalapurkar, R., Johnson, M., Vamvoudakis, K. G., Lewis, F. L., & Dixon, W. E. (2013). A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems. *Automatica*, 49(1), 89–92.
- Chen, J., Zha, W., Peng, Z., & Gu, D. (2016). Multi-player pursuit–evasion games with one superior evader. *Automatica*, 71, 24–32.
- Chen, M., Zhou, Z., & Tomlin, C. J. (2017). Multiplayer reach-avoid games via pairwise outcomes. *IEEE Transactions on Automatic Control*, 62(3), 1451–1457.
- Cho, N., Shin, H.-S., Kim, Y., & Tsourdos, A. (2017). Composite model reference adaptive control with parameter convergence under finite excitation. *IEEE Transactions on Automatic Control*, 63(3), 811–818.
- Chowdhary, G., & Johnson, E. (2011). A singular value maximizing data recording algorithm for concurrent learning. InProc. Am. Control Conf. (pp. 3547–3552). Chung, T. H., Hollinger, G. A., & Isler, V. (2011). Search and pursuit-evasion in
- mobile robotics. Autonomous Robots, 31(4), 299. Deptula, P., Bell, Z. I., Zegers, F., Licitra, R., & Dixon, W. E. (2018). Single agent
- indirect herding via approximate dynamic programming. In *Proc. IEEE conf. decis. control*(pp. 7136–7141).
- Deptula, P., Bell, Z. I., Zegers, F. M., Licitra, R. A., & Dixon, W. E. (2019). Approximate optimal influence over an agent through an uncertain interaction dynamic experiment. https://youtu.be/JeK4jTDulmo.

- Deptula, P., Rosenfeld, J., Kamalapurkar, R., & Dixon, W. E. (2018). Approximate dynamic programming: Combining regional and local state following approximations. *IEEE Transactions on Neural Network Learning System*, 29(6), 2154–2166.
- Farrell, J. A., & Polycarpou, M. M. (2006). Polycarpou, adaptive approximation based control: Unifying neural, fuzzy and traditional adaptive approximation approaches. In Ser. Adaptive and learning systems for signal processing, communications and control series, Vol.48. John Wiley & Sons.
- Gadre, A. S. (2001). Learning strategies in multi-agent systems-applications to the herding problem (M.S. thesis), Blacksburg, VA, USA: Dept. Elect. Comput. Eng. Virginia Tech.
- Garcia, E., Casbeer, D. W., & Pachter, M. (2017). Active target defence differential game: fast defender case. *IET Control Theory & Applications*, 11(17), 2985–2993.
- Garcia, E., Casbeer, D. W., & Pachter, M. (2018). Design and analysis of statefeedback optimal strategies for the differential game of active defense. *IEEE Transactions on Automation Contol.*
- Gill, P. E., Murray, W., & Saunders, M. A. (2005). Snopt: An sqp algorithm for large-scale constrained optimization. SIAM Review, 47(1), 99–131.
- Gill, P. E., Wong, E., Murray, W., & Saunders, M. A. (2015). User's guide for snopt version 7: Software for large-scale nonlinear programming. San Diego, La Jolla, CA: Department of Mathematics, University of California, 92093-0112. https://ccom.ucsd.edu/optimizers/static/pdfs/snopt7-7.pdf.
- Huang, H., Ding, J., Zhang, W., & Tomlin, C. J. (2015). Automation-assisted capture-the-flag: A differential game approach. *IEEE Transactions on Control Systems Technology*, 23(3), 1014–1028.
- Ioannou, P., & Sun, J. (1996). Robust adaptive control. Prentice Hall.
- Isaacs, R. (1967). Differential games. John Wiley.
- Isaacs, R. (1999). Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization. In *Ser. dover books on mathematics*. Dover Publications.
- Kachroo, P., Shedied, S. A., Bay, J. S., & Vanlandingham, H. (2001). Dynamic programming solution for a class of pursuit evasion problems: the herding problem. *IEEE Transactions on Systems, Man and Cybernetics*, 31(1), 35–41.
- Kamalapurkar, R., Andrews, L., Walters, P., & Dixon, W. E. (2017). Model-based reinforcement learning for infinite-horizon approximate optimal tracking. *IEEE Transactions on Neural Network Learning System*, 28(3), 753–758.
- Kamalapurkar, R., Dinh, H., Bhasin, S., & Dixon, W. E. (2015). Approximate optimal trajectory tracking for continuous-time nonlinear systems. *Automatica*, 51, 40–48.
- Kamalapurkar, R., Klotz, J. R., Walters, P., & Dixon, W. E. (2018). Model-based reinforcement learning for differential graphical games. *IEEE Transactions on Control Network System*, 5(1), 423–433.
- Kamalapurkar, R., Reish, B., Chowdhary, G., & Dixon, W. E. (2017). Concurrent learning for parameter estimation using dynamic state-derivative estimators. *IEEE Transactions on Automatic Control*, 62(7), 3594–3601.
- Kamalapurkar, R., Rosenfeld, J., & Dixon, W. E. (2016). Efficient model-based reinforcement learning for approximate online optimal control. *Automatica*, 74, 247–258.
- Kamalapurkar, R., Walters, P., & Dixon, W. E. (2016). Model-based reinforcement learning for approximate optimal regulation. *Automatica*, 64, 94–104.
- Kamalapurkar, R., Walters, P. S., Rosenfeld, J. A., & Dixon, W. E. (2018). Reinforcement learning for optimal feedback control: A Lyapunov-based approach. Springer.
- Khalafi, A. D., & Toroghi, M. R. (2011). Capture zone in the herding pursuit evasion games. Applied Mathematical Sciences, 5(39), 1935–1945.
- Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
- Kiumarsi, B., Vamvoudakis, K. G., Modares, H., & Lewis, F. L. (2017). Optimal and autonomous control using reinforcement learning: A survey. *IEEE Transactions on Neural Network Learning System*.
- Kumkov, S. S., Le Ménec, S., & Patsko, V. S. (2017). Zero-sum pursuit-evasion differential games with many objects: survey of publications. *Dynamic Games* and Applications, 7(4), 609–633.
- Lewis, F. L., Selmic, R., & Campos, J. (2002). Neuro-fuzzy control of industrial systems with actuator nonlinearities. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.
- Licitra, R., Bell, Z. I., Doucette, E., & Dixon, W. E. (2018). Single agent indirect herding of multiple targets: A switched adaptive control approach. *IEEE Control System Letter*, 2(1), 127–132.
- Licitra, R., Hutcheson, Z., Doucette, E., & Dixon, W. E. (2017). Single agent herding of n-agents: A switched systems approach. In *IFAC World Congress* (pp. 14374–14379.
- Lin, Y., & Sontag, E. D. (1991). A universal formula for stabilization with bounded controls. Systems & Control Letters, 16(6), 393–397.
- Lu, Z. (2006). Cooperative optimal path planning for herding problems (Ph.D. dissertation), Texas A & M University.
- Modares, H., Lewis, F. L., Kang, W., & Davoudi, A. (2018). Optimal synchronization of heterogeneous nonlinear systems with unknown dynamics. *IEEE Transactions on Automatic Control*, 63(1), 117–131.

- Monajjemi, M. (2015). bebop_autonomy library, http://bebop-autonomy. readthedocs.io.
- von Neumann, J., & Morgenstern, O. (1980). Theory of games and economic behavior. Princeton University Press.
- Parikh, A., Kamalapurkar, R., & Dixon, W. E. (2019). Integral concurrent learning: Adaptive control with parameter convergence using finite excitation. International Journal of Adaptive Control and Signal Processing, 33(12), 1775–1787.
- Patre, P. M., MacKunis, W., Kaiser, K., & Dixon, W. E. (2008). Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure. *IEEE Transactions on Automatic Control*, 53(9), 2180–2185.
- Pierson, A., & Schwager, M. (2018). Controlling noncooperative herds with robotic herders. IEEE Transactions on Robotics and Automation, 34(2), 517–525.
- Ramana, M. V., & Kothari, M. (2017). Pursuit-evasion games of high speed evader. Journal of Intelligent and Robotic Systems: Theory and Applications, 85(2), 293–306.
- Rosenfeld, J. A., Kamalapurkar, R., & Dixon, W. E. (2019). The state following (staf) approximation method. *IEEE Transactions on Neural Network Learning System*, 30(6), 1716–1730.
- Roy, S. B., Bhasin, S., & Kar, I. N. (2018). Combined mrac for unknown mimo lti systems with parameter convergence. *IEEE Transactions on Automatic Control*, 63(1), 283–290.
- Sadegh, N. (1993). A perceptron network for functional identification and control of nonlinear systems. *IEEE Transactions on Neural Networks*, 4(6), 982–988.
- Shedied, S. A. (2002). Optimal control for a two player dynamic pursuit evasion game; the herding problem (Ph.D. dissertation), Virginia Polytechnique Institute.
- Sun, J., Liu, C., & Ye, Q. (2017). Robust differential game guidance laws design for uncertain interceptor-target engagement via adaptive dynamic programming. *International Journal of Control*, 90(5), 990–1004.
- Vamvoudakis, K. G., & Hespanha, J. P. (2018). Cooperative q-learning for rejection of persistent adversarial inputs in networked linear quadratic systems. *IEEE Transactions on Automatic Control*, 63(4), 1018–1031.
- Vamvoudakis, K. G., Modares, H., Kiumarsi, B., & Lewis, F. L. (2017). Game theorybased control system algorithms with real-time reinforcement learning: How to solve multiplayer games online. *IEEE Control System*, 37(1), 33–52.
- Vidal, R., Shakernia, O., Kim, H., Shim, D., & Sastry, S. (2002). Probabilistic pursuitevasion games: theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18(5), 662–669.
- Walters, P., Kamalapurkar, R., & Dixon, W. E. (2015). Approximate optimal online continuous-time path-planner with static obstacle avoidance. In *Proc. IEEE Conf. Decis. Control*(pp. 650–655).
- Wang, Z., Liu, X., Liu, K., Li, S., & Wang, H. (2017). Backstepping-based Lyapunov function construction using approximate dynamic programming and sum of square techniques. *IEEE Transactions on Cybernetics*, 47(10), 3393–3403.
- Wang, D., Liu, D., Mu, C., & Ma, H. (2016). Decentralized guaranteed cost control of interconnected systems with uncertainties: a learning-based optimal control strategy. *Neurocomputing*, 214, 297–306.
- Wen, G., Ge, S. S., & Tu, F. (2018). Optimized backstepping for tracking control of strict-feedback systems. IEEE Transactions on Neural Network Learning System.
- Yan, F., Jiang, J., Di, K., Jiang, Y., & Hao, Z. (2018). Multiagent pursuit-evasion problem with the pursuers moving at uncertain speeds. *Journal of Intelligent* and Robotic Systems: Theory and Applications, 1–17.
- Zhang, H., Cui, L., & Luo, Y. (2013). Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network adp. IEEE Transactions on Cybernetics, 43(1), 206–216.
- Zhang, H., Liu, D., Luo, Y., & Wang, D. (2013). Adaptive dynamic programming for control algorithms and stability. In Ser. communications and control engineering. London: Springer-Verlag.



Patryk Deptula received his Ph.D. from the University of Florida in 2019. He is currently a Senior Robotics Researcher at Draper and his research interests include data-based learning, analysis, and prediction; autonomous systems and robotics; data fusion; bio-mechanics and human-machine tearning.



Zachary I. Bell received his Ph.D. from the University of Florida in 2019 and is a researcher for the Air Force Research Lab. His research focuses on cooperative guidance and control, computer vision, adaptive control, and reinforcement learning.



Federico M. Zegers received B.S. degrees in mechanical engineering and mathematics and the M.S. degree in mechanical engineering from the University of Florida in 2016 and 2019, respectively. Afterwards, Dr. Zegers received his Ph.D. degree under the supervision of Dr. Warren Dixon at the University of Florida in 2021. Federico's research interests include Lyapunov-based nonlinear and adaptive control, multi-agent systems, switched/hybrid systems theory, and robotics.



Ryan A. Licitra is a senior systems engineer with Lockheed Martin Missiles and Fire Control interested in research topics related to guidance, navigation, and control.



Warren E. Dixon received his Ph.D. in 2000 from the Department of Electrical and Computer Engineering from Clemson University. He was selected as a Eugene P. Wigner Fellow and worked as a staff researcher at Oak Ridge National Laboratory. In 2004, he joined the University of Florida in the Mechanical and Aerospace Engineering Department, where he is the Department Chair and currently holds the Newton C. Ebaugh Professorship. His main research interest is the development and application of Lyapunov-based control techniques for uncertain nonlinear systems. His work has been

acknowledged by various career and best paper awards, and he attained ASME and IEEE Fellow for contributions to adaptive control of uncertain nonlinear systems. He is an associate editor for Automatica and IEEE Control Systems.