



AFRL-AFOSR-VA-TR-2022-0183

A Switched Systems Approach for Navigation and Control with Intermittent Feedback

**Dixon, Warren
UNIVERSITY OF FLORIDA
1523 UNION RD RM 207
GAINESVILLE, FL, 32611
USA**

**05/24/2022
Final Technical Report**

<p>DISTRIBUTION A: Distribution approved for public release.</p>

Air Force Research Laboratory
Air Force Office of Scientific Research
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE 20220524		2. REPORT TYPE Final		3. DATES COVERED <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none;">START DATE 20180201</td> <td style="width: 50%; border: none;">END DATE 20220131</td> </tr> </table>		START DATE 20180201	END DATE 20220131
START DATE 20180201	END DATE 20220131						
4. TITLE AND SUBTITLE A Switched Systems Approach for Navigation and Control with Intermittent Feedback							
5a. CONTRACT NUMBER		5b. GRANT NUMBER FA9550-18-1-0109		5c. PROGRAM ELEMENT NUMBER 61102F			
5d. PROJECT NUMBER		5e. TASK NUMBER		5f. WORK UNIT NUMBER			
6. AUTHOR(S) Warren Dixon							
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) UNIVERSITY OF FLORIDA 1523 UNION RD RM 207 GAINESVILLE, FL 32611 USA				8. PERFORMING ORGANIZATION REPORT NUMBER			
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/AFOSR RTA2		11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-AFOSR-VA-TR-2022-0183		
12. DISTRIBUTION/AVAILABILITY STATEMENT A Distribution Unlimited: PB Public Release							
13. SUPPLEMENTARY NOTES							
14. ABSTRACT Imaging systems provide unique sensing capabilities for estimation and control. Image feedback typically involves tracking the evolution of image features in time/space to determine the relative motion of an autonomous system or tracked targets. The resulting image dynamics are nonlinear and include uncertainties inherent to the Euclidean-space to imagespace mapping. Moreover, such feedback is intermittent due to the inevitable loss of features because of the limited camera field-of-view or feature occlusion. The intermittent nature of image feedback can lead to degradation or instabilities in developed state estimates or controllers. Efforts in this project seek to generalize the typical assumption of continuous feature point observation from a single imaging source for estimation and control problems. To develop estimation and control solutions that use image-based feedback, fundamental problems associated with analyzing the stability of uncertain switched nonlinear systems must be addressed. The focus of Aim 1 is the development of switched systems tools using Lyapunov-based methods to enable arbitrary switching between different image sources (i.e., synthetic persistence where switching is between stable subsystems) using asymptotic adaptive update laws. In Aim 2, a novel data-based integral concurrent learning (ICL) method is proposed to yield exponential learning to facilitate the development of dwell-time							
15. SUBJECT TERMS							
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU		18. NUMBER OF PAGES 100		
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U					
19a. NAME OF RESPONSIBLE PERSON FREDERICK LEVE				19b. PHONE NUMBER (Include area code) 696-9730			

From: [Air Force Office of Scientific Research](#)
To: [Technical Reports](#)
Subject: [URL Verdict: Neutral][Non-DoD Source] Deliverable Received: FA9550-18-1-0109
Date: Friday, April 29, 2022 3:13:09 PM



OMA Team,
A new Final Performance has been submitted for your attention. Award
Number: FA9550-18-1-0109

Supporting
Report: https://afosr.gov1.qualtrics.com/WRQualtricsSurveyEngine/File.php?F=F_2sT6z8XIe3RnSIR

- Date Stamp: 15:11:37 , 4/29/2022
- Title: A Switched Systems Approach for Navigation and Control with Intermittent Feedback
- DISTRIBUTION: Yes- Approved for Public Release (Distro A)
- Principal Investigator: Warren Dixon
 - PI Email: wdixon@ufl.edu
- Program Officer: Frederick Leve
- Report Type: Final Performance
- Reporting Period
 - Start Date: 01/01/2021
 - End Date: 04/29/2022
- Award Number: FA9550-18-1-0109
- Report Due Date: 04/29/2022

ABSTRACT:

Imaging systems provide unique sensing capabilities for estimation and control. Image feedback typically involves tracking the evolution of image features in time/space to determine the relative motion of an autonomous system or tracked targets. The resulting image dynamics are nonlinear and include uncertainties inherent to the Euclidean-space to image-space mapping. Moreover, such feedback is intermittent due to the inevitable loss of features because of the limited camera field-of-view or feature occlusion. The intermittent nature of image feedback can lead to degradation or instabilities in developed state estimates or controllers. Efforts in this project seek to generalize the typical assumption of continuous feature point observation from a single imaging source for estimation and control problems. To develop estimation and control solutions that use image-based feedback, fundamental problems associated with analyzing the stability of uncertain switched nonlinear systems must be addressed. The focus of Aim 1 is the development of switched systems tools using Lyapunov-based methods to enable arbitrary switching between different image sources (i.e., synthetic persistence where switching is between stable subsystems) using asymptotic adaptive update laws. In Aim 2, a novel data-based integral concurrent learning (ICL) method is proposed to yield exponential learning to facilitate the development of dwell-time

conditions for switching between stable (when features are visible) and unstable (when persistence is lost) subsystems. Efforts in Aim 3 focus on the novel idea of developing dwell-time dependent controllers/path planners that can purposefully exploit the flexibility permitted by not having to maintain continuous views of a target/reference frame. Outcomes of this project include new design/analysis tools that involve adaptation for switched nonlinear systems that involve intermittent feedback. This outcome enables the development of image-based estimation and path planning/control methods that don't require persistent viewing of image features by a single image source.

A Switched Systems Approach for Navigation and Control with Intermittent Feedback

Final Report

Air Force Office of Scientific Research Award No: FA9550-18-1-0109

PI: Warren Dixon, wdixon@ufl.edu

1 Project Overview

Imaging systems provide unique sensing capabilities for estimation and control. Image feedback typically involves tracking the evolution of image features in time/space to determine the relative motion of an autonomous system or tracked targets. The resulting image dynamics are nonlinear and include uncertainties inherent to the Euclidean-space to image-space mapping. Moreover, such feedback is intermittent due to the inevitable loss of features because of the limited camera field-of-view or feature occlusion. The intermittent nature of image feedback can lead to degradation or instabilities in developed state estimates or controllers. Efforts in this project seek to generalize the typical assumption of continuous feature point observation from a single imaging source for estimation and control problems. To develop estimation and control solutions that use image-based feedback, fundamental problems associated with analyzing the stability of uncertain switched nonlinear systems must be addressed. The focus of Aim 1 is the development of switched systems tools using Lyapunov-based methods to enable arbitrary switching between different image sources (i.e., synthetic persistence where switching is between stable subsystems) using asymptotic adaptive update laws. In Aim 2, a novel data-based integral concurrent learning (ICL) method is proposed to yield exponential learning to facilitate the development of dwell-time conditions for switching between stable (when features are visible) and unstable (when persistence is lost) subsystems. Efforts in Aim 3 focus on the novel idea of developing dwell-time dependent controllers/path planners that can purposefully exploit the flexibility permitted by not having to maintain continuous views of a target/reference frame. Outcomes of this project include new design/analysis tools that involve adaptation for switched nonlinear systems that involve intermittent feedback. This outcome enables the development of image-based estimation and path planning/control methods that don't require persistent viewing of image features by a single image source.

The following sections provide additional details for the project. Section 2 provides a review of the proposed specific aims. Section 3 provides additional details on the progress and results obtained, and Section 4 describes further impacts of the developed work.

2 Specific Aims

This project proposed to develop fundamental breakthroughs in control methods for autonomous systems with intermittent feedback by investigating three specific aims. These aims, as described in the proposal, are provided below. As described in the subsequent sections, during the no-cost extension period of FY 2021, progress was made in Aim 2 and Aim 3, including generalizations to other application domains and networks of agents.

Aim 1: Arbitrarily Switched Image-based Estimation with Synthetic Persistence

Current visual servo control and image-based state estimation algorithms assume that image feedback is continuously available from a single source. This aim generalizes the persistent staring assumption to the assumption of synthetic persistence, where images of the target are available from a plurality of intermittent sources. Specifically, this aim considers image-based estimation for target tracking and navigation applications, where the state estimate is obtained by switching between multiple imaging/sensing sources. The fundamental barrier to achieve such an objective is the ability to arbitrarily switch between different uncertain and nonlinear image dynamics, and in this work, an adaptive solution is sought to compensate for the uncertainty. Efforts to achieve the objective in this aim include:

- Explore the use of the Lyapunov-based (e.g., extended LaSalle-Yoshizawa) theorems to analyze switching between asymptotically stable subsystems in a way that preserves asymptotic stability of the overall system.

- Develop new theorems that can be used to examine stability for a system in which arbitrary switching occurs between asymptotically stable nonlinear subsystems.

Aim 2: Dwell-based Switched Image-based Estimation with Intermittent Measurements

This aim is focused on the development of image-based state estimates in the presence of intermittent image-feedback resulting from tracked image features leaving the field-of-view or becoming occluded (i.e., loss of persistence). When image features are not visible, image-based state estimates become unstable. However, if the duration of time when the features are visible or not visible meets sufficient dwell-time conditions, then estimator stability can still be established. Known convergence/divergence rates (i.e., exponential rates) are required to develop such dwell-time conditions. While adaptive control is an advantageous approach to compensate for uncertainty, these methods typically result in asymptotic convergence due to the negative semi-definite derivative of a Lyapunov function candidate. The fundamental barrier to achieve the goals of this aim is the ability to develop adaptive state estimation while switching between stable and unstable modes of the dynamics. The objectives of this aim include:

- Develop switched systems methods that yield dwell-time conditions to ensure stability of image-based state estimates in the presence of intermittent feedback.
- Develop ICL methods that exponentially learn target motion models that can predict the location of image features when they are not visible.
- Develop ICL-based predictive switched systems methods that yield exponential state estimation in the presence of intermittent image feedback.

Aim 3: Dwell-time Dependent Control/Path Planning with Intermittent Measurements

This aim is focused on the development of image-based control/path planning methods in the presence of intermittent image-feedback. Current image-based regulation/tracking methods assume that the target or set of reference features remain visible throughout task execution. The continuous observation assumption places significant path constraints on the vehicle trajectory. This aim seeks to enable more advantageous vehicle trajectories that do not require features to remain visible (i.e., planned switching into unstable dynamics). That is, the planned path can deliberately maneuver in a way that will lose visibility of features, provided dwell-time conditions are satisfied. The fundamental barrier to achieve the goals of this aim is the ability to develop dwell-time dependent path planning/control methods that purposefully switch between stable and unstable modes of the uncertain dynamics. The objectives of this aim include:

- Develop visual servo control/path planning methods (e.g., homography based visual servo control) using adaptive ICL methods while maintaining exponential tracking control.
- Develop ICL-based, dwell-time-dependent, path planning methods that exploit switching (between stable and unstable uncertain nonlinear dynamics) to relax the assumption of continuously visible features.

3 2021 Accomplishments

The fundamental problem common to each of the specific aims in this project is how to perform switched systems control design and analysis for uncertain nonlinear systems using adaptive control methods. Deliverables target the development of new state estimation and control methods in the presence of parametric uncertainty and intermittent feedback. Progress was made in Aim 2 and Aim 3, including generalizations and applications. As described in Section 3.1, these efforts resulted in various research products including: journal and peer reviewed conference proceedings, a workshop, and invited seminars. A description of the approach taken and accomplishments for this reporting cycle are provided in Section 3.2. Copies of resulting publications are also provided.

3.1 Products

Outcomes of the project resulted in research advancements in Aim 2 and Aim 3. These outcomes were documented through peer reviewed journal and conference proceedings, including results that were primarily

motivated by this project [1–3] and additional results that leveraged underlying fundamental breakthroughs for different applications: adaptive functional electrical stimulation controllers that switch between different skeletal muscle groups [4], deep learning for switched systems [1, 5, 6], networks of agents [7–10], and other general theoretical results [11, 12]. As an invited seminar speaker, the PI delivered a talk entitled “Intermittent Image-based Feedback” at the Conference on Control Technology and Applications (CCTA) Workshop entitled *The Confluence of Vision and Control*, on August 08, 2021. He gave an invited seminar entitled “Assured Autonomy: Uncertainty, Optimality, and Data Intermittency” for the Office of the Under Secretary of Defense (Research and Engineering), DoD Basic Research Forum on February 25, 2021. He also gave a keynote talk and a plenary talk, both entitled “The Intermittent Joy of Intermittent Feedback,” at the International Symposium on Test Automation and Instrumentation in Xining, China (presented virtually) in Dec. 2020, and the 16th International Conference on Control, Automation, Robotics and Vision (ICARCV 2020) in Shenzhen, China (presented virtually) in Dec. 2020.

3.2 Results

Results Inspired by Aim 2: In many applications, the state (e.g., position and orientation) of an autonomous agent and its local environment (e.g., relative positions of objects in the surrounding environment) must be determined from sensor data. With imaging systems, this problem is well known as simultaneous localization and mapping (SLAM). In [2], we developed exponentially converging observers that use a camera to estimate the Euclidean distance to features on a stationary object in the camera field-of-view (FOV) while also estimating the Euclidean trajectory of the camera tracking the object. In our previous work in [13], we demonstrated exponential convergence of depth estimates using CL; however, these results rely on the assumption that the features are on a plane. Our efforts in Aim 2, focused on the use of ICL to remove the need for measuring the state derivative and to address the inherent singularity when one of the coordinates becomes zero (i.e., the so-called depth to the target). Specifically, previous results assume a positive depth constraint where the distance from the focal point of the camera to the target along the axis perpendicular to the image plane remains positive. The positive depth constraint is satisfied if the features remain in the camera FOV; however, the constraint can be violated for some camera rotations that cause the feature to leave the FOV. Since our previous work in this project opens the possibility for features to intermittently leave the FOV, we developed a new formulation of the error system in [2]. In [2], we exploit alternative image geometry insights to express the error system with a more general distance measure that only becomes zero when the target and camera are coincident; thereby, avoiding the positive depth constraint. While this result also requires the features to remain in the FOV (which ensures that the positive depth constraint is satisfied), eliminating the positive depth constraint eliminates a barrier for future development that would allow intermittent viewing of the features. An issue with the new image geometry based error system is that it still contains the unmeasurable distance to the target. However, in [2] we explain how the unmeasurable state can be related to an unknown constant to enable the use of ICL. Regardless of the system identification method used, there is a delay before sufficient excitation occurs to identify the parameters. Therefore, our preliminary result in [14] can exhibit an arbitrarily long delay before determining the feature Euclidean coordinates. In [2], we modify the developed learning strategy to include gradient terms that enable transient learning until sufficient data has been collected for the ICL terms. This result highlighted the need for faster learning and learning methods that provide better function approximation.

Motivated by the need for improved learning capabilities, efforts during the no-cost extension also focused on deep neural networks (DNNs) to potentially improve function approximation performance. Given recent advances in DNNs, various researchers have explored the use of DNN-based control architectures. However, such offline methods pose limitations since training typically requires large amounts of data, and the resulting feedforward terms are implemented as an open-loop approximator based on the offline training. In contrast to offline training, NN weight update laws derived from Lyapunov-based stability analysis methods have been developed to adjust the NN weights in real-time as an adaptive closed-loop feedforward term. Although NN-based adaptive architectures are well-established, these methods only apply to NNs with a single hidden-layer. The complex nature of DNNs being nested nonlinear parameterizations of inner-layer activation functions, weights, and bias terms presents challenges that preclude development of real-time adaptation laws with Lyapunov-based methods.

We developed two different approaches to address issues related to deep learning for real-time autonomous

systems. Specifically, in [6] we developed general constraints on the inner-layer adaptation laws to update the inner-layer weight estimates in real-time. A Lyapunov-based analysis is used to develop a continuous adaptation law to estimate the output-layer weights. Our result provides a first insight into the development of Lyapunov-based adaptive update laws for both the inner-layer DNN weights as well as the output layer weights. Inspired by modular adaptive control designs from traditional adaptive control literature, general constraints on the inner-layer DNN weight update laws are developed that enable modular design and selection of update laws. The developed DNN-based modular adaptive architecture allows more flexibility when selecting inner-layer DNN weight update laws. In arbitrary width and depth DNNs, there may be hundreds or thousands of inner-layer weights. Simultaneously updating all the inner-layer weights online may be computationally intractable in real-time or undesired. Hence, the developed method provides a switched framework that provides design guidelines that can be used in future research efforts to guide inner-layer weight adaptive update laws. In doing so, the inner-layer weight update laws may be arbitrarily switched on and off to allocate computational resources while updating the desired weights. Additionally, inner-layer weights may dropout, or be selectively turned off to prevent over-fitting and improve overall function approximation performance.

Motivated by the insights from the modular deep learning design in [6], we also developed the first result with Lyapunov-based real-time weight adaptation laws for each layer of a DNN in [5]. To address the challenges posed by nested nonlinear parameterizations of the inner-layer DNN weights, we develop a recursive representation of the inner-layer DNN structure to facilitate the analysis. Then, a Taylor’s first order approximation of the uncertainty is recursively derived. The update laws are derived from a Lyapunov-based stability analysis in which the first-order terms are canceled by the weight update law-based terms. The remaining terms in the Lyapunov-based analysis are eliminated using a robust control approach. The adaptation laws developed in [5] depend on gradients of activation functions. The adaptation laws contain discontinuities if an activation function with a discontinuous gradient is used in the DNN architecture.

Our advancements in deep learning in [5, 6] were then incorporated in an intermittent image-based observation problem in [15] as a transition to AFRL. Specifically, our result in [15] is motivated by tracking a mobile target with a camera, where occlusions prevent continuous observations of the target. To this end, we developed a deep motion model network (DMMN) framework. The DMMN framework is an estimator where the output layer weights are updated online while the target is in the agent’s FOV. Simultaneously, motivated by our result in [16], a replay buffer of target poses is collected online while the target is in the agent’s FOV. When a sufficient number of target poses are collected, batch updates to the DMMN basis are performed to improve the estimate of the DMMN basis. After a series of batch updates, the prediction of the target’s velocity switches to the new basis. Then, when the target leaves the agent’s FOV, the DMMN is used as a predictor to estimate the target’s pose and velocity while the target remains outside the agent’s FOV. A Lyapunov-based dwell-time analysis is performed to determine the maximum dwell-time condition that ensures the estimation error of the target’s pose does not grow beyond the size of the camera’s FOV while the target is outside the FOV. While it is not possible to ensure these conditions are satisfied (e.g., since a target is generally not cooperative), this dwell-time condition can be used to determine when the prediction error of the pose has grown too large and gives constraints to subsequently search for the target, improving the chances of finding the target. Experimental results in an AFRL facility (RW Autonomy Laboratory at the UF REEF) demonstrated the performance of the proposed DMMN framework using a quadrotor with a downward facing camera tracking a mobile ground vehicle that intermittently leaves the agent’s FOV.

Results Inspired by Aim 3: As described previously, Aim 3 focuses on the development of image-based control/path planning methods in the presence of intermittent image-feedback, because state feedback is a critical component in designing path planning methods used for guidance, navigation, and control of autonomous vehicles. Factors such as task definition, operating environment, sensor modality, and adversarial effects may result in intermittent state feedback, inhibiting a system’s ability to achieve its task. Autonomous systems often operate in environments where state feedback may not be available, such as in anti-access and area denial environments. In these environments, it is often required that an agent track a path, despite interruptions in state feedback. As a result, we invented a class of problems that we termed Relay-Explorer problems, where a switched system approach is used to account for intermittent state feedback (e.g., due to feature tracking or limited camera FOV). Past work on these problems established a framework for developing dwell-time conditions for stable tracking using these methods. However, existing work only applies to a limited class of reference paths and feedback region geometries. Our result in [3] advances a

topologically inspired method for guaranteeing re-acquisition of feedback for nearly arbitrary geometries in arbitrary dimensions, all while relaxing the dwell-time conditions and retaining the uniformly ultimately bounded stability result from preceding work. Numerical experiments in the plane demonstrate an increase of hundreds of percentage points—even for fairly generic geometries—in the tracking error the agent could afford, using the proposed method, without sacrificing stability.

The result in [3] specifically focused our development on encoding the sensor availability within the path-planning approach. When we considered this problem more generally, we were inspired to examine related network control problems, which trade sensor intermittency for communication intermittency. Network communication is often expressed in terms of distance-based constraints: two agents may communicate if they are sufficiently close. While network connectivity is a necessary component for achieving objectives for a multi-agent system (MAS), often assumptions are introduced about the network remaining connected or that continuous communication along edges is always possible. Our result in [10] explores the rendezvous problem for a MAS with distance-limited, intermittent communication and sensing. Unlike previous works that provide specific event-triggered controllers (ETCs), we provide a framework that characterizes a family of distributed ETCs leveraging non-singular edge-potentials to achieve approximate rendezvous while maintaining the initial distance limited graph. The framework excludes the possibility of Zeno behavior and accommodates the development of self-triggered controllers. The combination of continuous and impulsive dynamics results in a hybrid system, where the closed-loop dynamics of the MAS are presented and analyzed using hybrid differential inclusions. The approximate rendezvous problem is recast into a set stabilization problem and sufficient conditions of the rendezvous set are obtained through a Lyapunov-based analysis.

Because continuous communication between neighbors in a network places a constraint on the physical distance between agent pairs, control algorithms constructed on infinitely often jointly connected graphs, which allow for the intermittent relaxation of physical distance constraints, have grown in interest. Akin to the idea of intermittently connected graphs, clustered MAS have also been an active area of research since they impart high-level task division among sub-groups of the whole MAS. While cluster control strategies grant each component of a MAS autonomy, these same techniques can be leveraged to obtain system-wide consensus while reducing the communication between clusters. In [7], the consensus problem for a clustered MAS is investigated. Given a MAS, the agents are organized into disjoint clusters, where each cluster forms a connected network. The agents that are within the same cluster can communicate continuously with their neighbors. Between some cluster pairs, there exists an inter-cluster that enables the relay of information between the two clusters. Agents that have neighbors in clusters different from their own can communicate intermittently and asynchronously with their different-cluster neighbors. The goal of each agent in the MAS is to converge to an agreement value by sharing local information of a continuous-time homogeneous process. Note that the intermittent communication events of the continuous-time process are inherently hybrid. Therefore, we designed a unique coupling between a static consensus protocol and a hybrid consensus protocol in [7]. The closed-loop network model is presented using a hybrid systems framework. The consensus problem is then recast into a set stability problem and sufficient conditions of the consensus set are presented through leveraging a Lyapunov-based stability analysis. A clustered MAS composed of 16 agents that is partitioned into 3 clusters and 2 inter-clusters is used to illustrate the results through numerical simulations and shown to achieve consensus using the proposed control strategy.

Motivated by intermittent communication challenges, ETC enables the manipulation of continuous-time dynamical systems under intermittent state feedback. The ETC design can have significant ramifications on the overall system performance. Typically, agents are required to continuously monitor their trigger condition while each neighbor continuously monitors a neighbor’s communication. Self-triggered control (STC) provides a more efficient triggering method that leverages the system model to predict when to monitor/communicate. Moreover, an STC strategy can also be developed that eliminates the need for an agent’s neighbors to continuously monitor for information requests. While ETC and STC strategies provide numerous benefits, critical communication timing conditions introduce potential vulnerabilities. Specifically, since the trigger conditions are based on feedback from multiple agents, erroneous feedback can lead to undesired outcomes. Distributed ETCs and STCs are developed for approximate leader-follower consensus with robustness to adversarial Byzantine agents for a class of homogeneous MAS. We developed a strategy in [8] for each agent to detect Byzantine agent behaviors within their neighbor set and then selectively disregard their transmission. Selectively removing Byzantine agents results in time-varying discontinuous changes to the network topology. Nonsmooth dynamics also result from the use of ETC and STC strategies

and triggering condition estimators that enable intermittent communication. Nonsmooth Lyapunov methods are used to prove approximate consensus of the MAS consisting of the remaining cooperative agents. Simulations validate the result and outline the tradeoff between communication and performance. In [17], we also developed a distributed ETC for formation control and leader tracking (FCLT) with robustness to adversarial Byzantine agents. Assuming each agent can accurately measure the state of a neighbor whenever the neighbor broadcasts its state, a reputation-based strategy is developed for each agent to detect Byzantine agent behaviors within their neighbor set and then selectively disregard Byzantine state information. Selectively ignoring Byzantine agents results in a time-varying graph topology. Nonsmooth dynamics also result from intermittent communication due to an ETC strategy, which facilitates the efficient use of resources. Nonsmooth Lyapunov methods are used to prove stability and FCLT of the MAS consisting of the remaining cooperative agents.

4 2021 Impacts

As described in greater detail in the previous section and in the included publications, the development in this project provides new mathematical tools for designing controllers and performing stability analysis for continuous time systems that experience discrete changes when feedback is available or not available, with a particular focus on imaging systems, where feature tracking is often interrupted by occlusions or limited sensor field-of-view. A specific focus included understanding how to design controllers and perform analysis when the switching occurred between uncertain dynamic systems. During this no-cost extension, we were able to transition some of the outcomes to a testbed managed by AFRL/RW and experimental results from this transition are included in [15].

The mathematical development that resulted from this project was also shown to have an impact in other disciplines such as network science where connectivity among agents can yield intermittent instances of feedback. It was also extended to rehabilitation applications for people with movement disorders. Specifically, results were developed that use functional electrical stimulation of human skeletal muscle to yield coordinated limb motions, which required switching between different muscle groups to activate at specific times, where each muscle group includes a different uncertain nonlinear dynamical system.

As described in the participants section of this report, a number of PhD students were supported by this project. Federico Zegers PhD, is currently employed as an AFRL research staff member at Eglin AFB, Max Greene PhD is now working at Aurora, and Runhan Sun PhD is about to sign a contract with Mathworks. General mathematical development in this project was included in two courses at the University of Florida, which were taught as distance learning courses. The courses are entitled Nonlinear Control and Adaptive Control. Otherwise, there were no impacts on physical, institutional, and information resources that form infrastructure. The developed products involved general mathematical development that could have broader implications beyond the scope of this project; however, specific efforts were not made to yield impacts beyond science and technology described in previous sections.

5 Changes

This reporting cycle covered an approved no-cost extension. There were no other changes in approach, delays, expenditures, or other. The no-cost extension provided an opportunity to complete several publications, transition outcomes to AFRL, and to generalize developed approaches.

References

- [1] M. Greene, Z. Bell, S. Nivison, J. How, and W. E. Dixon, “Cooperative model-based reinforcement learning for approximate optimal tracking,” in *Proc. Am. Control Conf.*, 2021, pp. 1973–1978.
- [2] Z. I. Bell, P. Deptula, E. A. Doucette, J. W. Curtis, and W. E. Dixon, “Simultaneous estimation of euclidean distances to a stationary object’s features and the euclidean trajectory of a monocular camera,” *IEEE Trans. Autom. Control*, vol. 66, no. 9, pp. 4252–4258, 2021.

- [3] S. Edwards, D. Le, D. Guralnik, and W. E. Dixon, "A topologically inspired path-following method with intermittent state feedback," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4449–4456, 2021.
- [4] C. Cousin, P. Deptula, C. Rouse, and W. E. Dixon, "A switched Lyapunov-passivity approach to motorized FES cycling using adaptive admittance control," *IEEE Trans. Control Syst. Tech.*, vol. 30, no. 2, pp. 740–754, 2022.
- [5] O. Patil, D. Le, M. Greene, and W. E. Dixon, "Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network," *IEEE Control Syst. Lett.*, vol. 6, pp. 1855–1860, 2022.
- [6] D. Le, M. Greene, W. Makumi, and W. E. Dixon, "Real-time modular deep neural network-based adaptive control of nonlinear systems," *IEEE Control Syst. Lett.*, vol. 6, pp. 476–481, 2021.
- [7] F. Zegers, S. Phillips, and W. E. Dixon, "Consensus over clustered networks with asynchronous inter-cluster communication," in *Proc. Am. Control Conf.*, 2021, pp. 4249–4254.
- [8] F. Zegers, P. Deptula, J. Shea, and W. E. Dixon, "Event/self-triggered approximate leader-follower consensus with resilience to byzantine adversaries," *IEEE Trans. Autom. Control*, vol. 67, no. 3, pp. 1356–1370, Mar. 2022.
- [9] F. Zegers, M. Hale, J. M. Shea, and W. E. Dixon, "Reputation-based event-triggered formation control and leader tracking with resilience to byzantine adversaries," in *Proc. Am. Control Conf.*, 2020, pp. 761–766.
- [10] F. M. Zegers, D. P. Guralnik, and W. E. Dixon, "Event/self-triggered multi-agent system rendezvous with graph maintenance," in *Proc. IEEE Conf. Decis. Control*, 2021.
- [11] P. Deptula, H.-Y. Chen, R. Licitra, J. Rosenfeld, and W. E. Dixon, "Approximate optimal motion planning to avoid unknown moving avoidance regions," *IEEE Trans. Robotics*, vol. 36, no. 2, pp. 414–430, 2020.
- [12] O. Patil, A. Isaly, B. Xian, and W. E. Dixon, "Exponential stability with rise controllers," *IEEE Control Syst. Lett.*, vol. 6, pp. 1592–1597, 2022.
- [13] A. Parikh, R. Kamalapurkar, H.-Y. Chen, and W. E. Dixon, "Homography based visual servo control with scene reconstruction," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 6972–6977.
- [14] Z. I. Bell, H.-Y. Chen, A. Parikh, and W. E. Dixon, "Single scene and path reconstruction with a monocular camera using integral concurrent learning," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 3670–3675.
- [15] Z. Bell, R. Sun, K. Volle, P. Ganesh, S. Nivison, and W. E. Dixon, "Mobile ground target tracking subject to intermittent measurements by a mobile aerial camera using attention deep neural networks," *IEEE Control Syst. Lett.*, submitted.
- [16] R. Sun, M. Greene, D. Le, Z. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Syst. Lett.*, vol. 6, pp. 193–198, 2021.
- [17] F. Zegers, M. Hale, J. Shea, and W. E. Dixon, "Event-triggered formation control and leader tracking with resilience to byzantine adversaries: A reputation-based approach," *IEEE Trans. Netw. Syst.*, vol. 8, no. 3, pp. 1417–1429, 2021.

Event/Self-Triggered Approximate Leader-Follower Consensus With Resilience to Byzantine Adversaries

Federico M. Zegers¹, Patryk Deptula¹, John M. Shea¹, and Warren E. Dixon¹, *Fellow, IEEE*

Abstract—Distributed event- and self-triggered controllers are developed for approximate leader-follower consensus with robustness to adversarial Byzantine agents for a class of homogeneous multi-agent systems (MASs). A strategy is developed for each agent to detect Byzantine agent behaviors within their neighbor set and then selectively disregard their transmission. Selectively removing Byzantine agents results in time-varying discontinuous changes to the network topology. Nonsmooth dynamics also result from the use of event/self-triggered strategies and triggering condition estimators that enable intermittent communication. Nonsmooth Lyapunov methods are used to prove approximate consensus of the MAS consisting of the remaining cooperative agents. Simulations are included to validate the result and to outline the tradeoff between communication and performance.

Index Terms—Decentralized control, fault tolerant control, multi-agent systems, networked control systems.

I. INTRODUCTION

APPROXIMATE leader-follower consensus along with several other consensus variants have been extensively studied since they have a wide range of applications in multi-agent systems (MASs) and distributed computing (see [1]–[9]). The approximate leader-follower consensus objective is to position the follower agents of an MAS so they converge into a

forward-invariant neighborhood of the leader's position. Most consensus results consider continuous-time dynamical MASs and assume continuous communication and sensing. However, physical constraints on a network, like communication bandwidth, data packet losses, and delays, may inhibit continuous communication.

Motivated by intermittent communication challenges, event-triggered control (ETC) enables the manipulation of continuous-time dynamical systems under intermittent state feedback (see [10]–[15]). ETC opportunistically selects when to update the system to efficiently perform a task [16]. With respect to an MAS, interagent communication occurs at times dictated by an event-trigger, which is derived from the need to preserve system stability, and if desired, a performance criterion [17]. For example, the authors in [10] investigate event-triggered pinning control for the synchronization of complex networks of nonlinear dynamical systems. A multi-agent formation control problem is investigated in [18] with ETC updates and additive disturbances, where agents only communicate by exchanging information via a cloud repository. In [12], a decentralized controller is developed that uses ETC scheduling to enable leader-follower consensus under fixed and switching communication topologies.

The event-trigger design can have significant ramifications on the overall system performance. Typically, agents are required to continuously monitor their trigger condition while each neighbor continuously monitors a neighbor's communication. Self-triggered control (STC) provides a more efficient triggering method that leverages the system model to predict when to monitor/communicate (see [19], [20]). Moreover, an STC strategy can also be developed that eliminates the need for an agent's neighbors to continuously monitor for information requests.¹ While ETC and STC strategies provide numerous benefits, critical communication timing conditions introduce potential vulnerabilities. Specifically, since the trigger conditions are based on feedback from multiple agents, erroneous feedback can lead to undesired outcomes.

Assuming knowledge of the number of adversarial agents, the authors in [22] study event-triggered secure cooperative control of linear MASs under denial of service (DoS) attacks, which are

Manuscript received October 7, 2020; accepted March 23, 2021. Date of publication April 21, 2021; date of current version February 28, 2022. This work was supported in part by A Task Order Contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, in part by the Office of Naval Research under Grant N00014-13-1-0151, in part by the NEEC Award N00174-18-1-0003, in part by the AFOSR Award FA9550-18-1-0109, and in part by the AFOSR Award FA9550-19-1-0169. Recommended by Associate Editor R. M. Jungers. (Corresponding author: Federico Zegers.)

Federico M. Zegers and Patryk Deptula are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: fredzeg@ufl.edu; pdeptula@ufl.edu).

John M. Shea is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: jshea@ece.ufl.edu).

Warren E. Dixon is with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250 USA, and also with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: wdixon@ufl.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TAC.2021.3074849>.

Digital Object Identifier 10.1109/TAC.2021.3074849

¹Eliminating the need for continuous monitoring allows the potential for power savings [21].

defined as interruptions of communication on the control channels carried out by an intelligent adversary. Static networks with cooperative and malicious nodes have been extensively studied within the computer science literature [23], [24], resulting in iterative consensus algorithms for distributed computing applications that are robust to components subject to faults. Different research communities are beginning to extend these architectures to MAS applications consisting of mobile agents, which requires the nontrivial study of motion planning to preserve algebraic graph properties such as connectivity. For example, the authors in [25] address the problem of resilient in-network consensus in the presence of Byzantine nodes [23], where resilience is designed for worst-case security breaches and omnipotent malicious nodes. The weighted-mean-subsequence-reduced (W-MSR) algorithm enables each node to receive state information from its neighbors, sort the states, and neglect at most F extreme states relative to the node's state. While asymptotic consensus is obtained using the W-MSR algorithm, the result requires an upper bound of the maximum number of Byzantine agents. Such an approach requires at least $2F + 1$ reliable neighbors, which is difficult to scale with increasing node number. Moreover, information is shared with all agents regardless of their cooperative/Byzantine status, which can impact secure MAS consensus. The result in [26] also enables resilient consensus assuming the maximum number of malicious agents is known. The result in [27] achieves resilient approximate consensus by removing extreme values similar to the W-MSR algorithm while enabling asynchronous communication through a self-triggered strategy given a known upper bound for the maximum number of Byzantine agents. The result in [28] achieves consensus tracking of an arbitrary piecewise continuous step function for a network of agents using the W-MSR algorithm provided the network is strongly $(2F + 1)$ -robust. While malicious agent identification is not required, MAS consensus is limited to at most F malicious agents. Nonetheless, ETC and STC strategies that enable secure MAS consensus while detecting and mitigating against Byzantine adversaries require further investigation.

In graph theoretic terms, antagonistic interactions can be modeled by replacing the standard communication graph, characterized by nonnegative weights, with a signed graph displaying both positive and negative weights (see [29], [30]). Positive directed/undirected paths correspond to cooperative interactions with agents, while negative directed/undirected paths describe interactions with antagonistic agents. The work in [31] develops the concept of bipartite consensus among agents with antagonistic interaction, where bipartite consensus or agreed dissensus is defined as when all agents in the MAS converge to a state that is the same in magnitude but not in sign, effectively enabling each team to reach their own consensus state. By addressing the classical example of homogeneous agents modeled as simple scalar integrators, the authors in [31] prove that if the signed, weighted, and connected communication graph describing the agents' interactions is structurally balanced, then the agents reach bipartite consensus. If the interactions are antagonistic, but not structurally balanced, the only agreement that can be achieved among the agents is the trivial one, where all the agents' states converge to zero.

The authors in [32] extend the results in [31] to an MAS consisting of N agents with linear time-invariant (LTI) dynamics and establish conditions to ensure consensus and bipartite consensus under the assumption that the agent interactions can be described by a weighted, signed, connected, and structurally balanced graph. Moreover, bipartite consensus can always be reached under the assumption that the agent dynamics are stabilizable. However, consensus to a common state for the two antagonistic groups can be achieved only under more restrictive requirements on the Laplacian associated with the communication graph and on the agents' description. In particular, consensus may be achieved only if there is some equilibrium between the two groups, both in terms of cardinality and in terms of the weights of the conflicting interactions among agents. The result in [33] considers the bipartite consensus problem for an MAS composed of agents with LTI dynamics and input saturation over directed and unbalanced networks.

The works in [31]–[33] share the common goal of enabling both teams in the antagonistic network to reach their own respective consensus, which assumes both consensus states can coexist. However, signed graphs can also be extended to applications, where the goal of one team is to reach consensus while the goal of the opposition is to prevent the other team from reaching consensus. Such a scenario works under the assumption that each team has a goal and must cooperatively work together to reach their goal. However, it is not possible for both teams to reach their goal simultaneously. Hence, if one team is to conspire against the other in a cooperative manner, then signed graphs can be leveraged to model these interactions. However, if one group is organized while the other is not, a traditional graph is sufficient to model the cooperative interactions of the organized team.

In this article, the approximate leader-follower consensus problem for cooperative agents in the presence of Byzantine adversaries is investigated, where approximate leader-follower consensus is achieved when the state of the followers is driven into a forward invariant neighborhood of the state of the leader. Since only the cooperative followers must collaborate to reach leader-follower consensus and the Byzantine adversaries are assumed to operate independently (i.e., do not work together against the cooperative followers), a traditional unsigned graph is used to model the network topology of the followers. Moreover, this article presents event- and self-triggered strategies for the approximate leader-follower consensus problem while providing robustness to Byzantine adversaries. Unlike methods such as W-MSR, that need to know an upper bound on the number of Byzantine agents *a priori* and then reject some amount of outlier information, the work in this article uses a Lyapunov-based detection strategy to identify Byzantine actors online. Motivated by the potential for different ways in which an agent can be compromised (e.g., different combinations of sensing, actuation, or computation) and different potential responses to such behavior as described in Section V-C, we segregate Byzantine agents into different categories. Based on the agent category, the resulting graph can become directed, time-varying, and unbalanced. Despite these complications, the result achieves approximate leader-follower consensus, between the leader and cooperative followers, in a distributed manner, where

each agent's control input is computed from local neighbor interactions. Moreover, in the absence of Byzantine adversaries, this work can recover the result in and is a generalization of [12], which did not consider Byzantine agents. The stability of the event- and self-triggered control strategies are examined through a nonsmooth Lyapunov analysis. This work is a generalization of our precursory result in [34]. Specifically, this article provides broader context, additional mathematical development for the stability analysis, a more detailed fault detection method, and simulation results investigating the tradeoff between the event- and self-triggered strategies with respect to power savings, communication bandwidth, and performance.

II. PRELIMINARIES

A. Notation

Let \mathbb{R} and \mathbb{Z} denote the set of real numbers and integers, respectively, where $\mathbb{R}_{\geq 0} \triangleq [0, \infty)$, $\mathbb{R}_{> 0} \triangleq (0, \infty)$, $\mathbb{Z}_{\geq 0} \triangleq \mathbb{R}_{\geq 0} \cap \mathbb{Z}$, and $\mathbb{Z}_{> 0} \triangleq \mathbb{R}_{> 0} \cap \mathbb{Z}$. Let $A \in \mathbb{R}^{p \times q}$ be a real-valued $p \times q$ matrix, where $p, q \in \mathbb{Z}_{> 0}$. If $p = q$ and A has real eigenvalues, then the maximum and minimum eigenvalues of A are denoted by $\lambda_{\max}(A) \in \mathbb{R}$ and $\lambda_{\min}(A) \in \mathbb{R}$, respectively. If $p \neq q$, then the maximum singular value of A is denoted by $S_{\max}(A) \in \mathbb{R}_{\geq 0}$. The $p \times q$ zero matrix and the $p \times 1$ zero vector are denoted by $0_{p \times q}$ and 0_p , respectively. Let $1_p \in \mathbb{R}^p$ denote a column vector with all entries being 1. The $p \times p$ identity matrix is denoted by I_p . The Euclidean norm of a vector $r \in \mathbb{R}^p$ is denoted by $\|r\| \triangleq \sqrt{r^T r}$. The Kronecker product of $A \in \mathbb{R}^{p \times q}$ and $B \in \mathbb{R}^{u \times v}$ is denoted by $(A \otimes B) \in \mathbb{R}^{pu \times qv}$. The symbols \wedge , \vee , and \neg denote logical AND, OR, and NOT, respectively. Let $2^{\mathcal{S}}$ denote the power set of the set \mathcal{S} .

B. Algebraic Graph Properties

Let $\mathcal{G}(t) \triangleq (\mathcal{V}, \mathcal{E}(t), \mathcal{A}(t))$ be a time-varying, weighted, and undirected graph with node set $\mathcal{V} \triangleq \{1, 2, \dots, N\}$ for some $N \in \mathbb{Z}_{> 0}$, undirected edge mapping $\mathcal{E} : [0, \infty) \rightarrow 2^{\mathcal{V} \times \mathcal{V}}$, and weighted adjacency mapping $\mathcal{A} : [0, \infty) \rightarrow \mathbb{R}_{\geq 0}^{N \times N}$, where $\mathcal{A}(t) \triangleq [a_{ij}(t)]$ and $a_{ij} : [0, \infty) \rightarrow \{0, 1\}$. Note that $a_{ij}(t) = 1$ implies node i can receive information from node j , and $a_{ij}(t) = 0$ implies node i cannot receive information from node j , where only binary weights are considered. Within the context of this article, no self-loops are considered, and therefore, $a_{ii}(t) \triangleq 0$ for all $i \in \mathcal{V}$ and $t \geq 0$. In general, $a_{ij}(t) \neq a_{ji}(t)$, but equality is possible. An undirected edge is defined as an ordered pair (j, i) , where $(j, i) \in \mathcal{E}(t)$ if and only if $(i, j) \in \mathcal{E}(t)$. Note that $(j, i) \in \mathcal{E}(t)$ implies node j can send information to node i . An undirected path is a sequence of edges in $\mathcal{E}(t)$. An undirected graph is called connected if and only if there exists an undirected path between any two distinct nodes. The time-varying neighbor set of node i is defined by $\mathcal{N}_i : [0, \infty) \rightarrow 2^{\mathcal{V}}$, where $\mathcal{N}_i(t) \triangleq \{j \in \mathcal{V} : (j, i) \in \mathcal{E}(t), j \neq i\}$.

The diagonal degree mapping $\Delta : [0, \infty) \rightarrow \mathbb{R}_{\geq 0}^{N \times N}$ of the undirected graph $\mathcal{G}(t)$ is defined by $\Delta(t) \triangleq [\Delta_{ij}(t)]$, where for all $i \neq j$ and $t \geq 0$, $\Delta_{ij}(t) \triangleq 0$ and $\Delta_{ii}(t) \triangleq \sum_{j \in \mathcal{V}} a_{ij}(t)$. The graph Laplacian $L : [0, \infty) \rightarrow \mathbb{R}^{N \times N}$ of the undirected graph $\mathcal{G}(t)$ is defined by $L(t) \triangleq \Delta(t) - \mathcal{A}(t)$. Consider a single node,

indexed by 0, along with the mapping $D : [0, \infty) \rightarrow \mathbb{R}_{\geq 0}^{N \times N}$, such that $D \triangleq [d_{ij}]$ and $d_{ij} : [0, \infty) \rightarrow \{0, 1\}$. For all $i \neq j$ and $t \geq 0$, $d_{ij}(t) \triangleq 0$. Let $d_i(t) \triangleq d_{ii}(t)$, where $d_i(t) = 1$ if node 0 can send information to node i , and $d_i(t) = 0$ otherwise. Hence, the weighted connectivity matrix encoding all communication links between all nodes in $\mathcal{V} \cup \{0\}$ is given by the mapping $H : [0, \infty) \rightarrow \mathbb{R}^{N \times N}$, where $H(t) \triangleq L(t) + D(t)$.

Remark 1: While $H(t)$ is a time-varying matrix, the set $\{H(t)\}$ is finite since there are a finite number of configurations in which the agents of the MAS can be connected.

III. AGENT DYNAMICS AND NETWORK TOPOLOGY

Consider a homogeneous MAS consisting of a single leader indexed by 0 and a set of $N \in \mathbb{Z}_{> 0}$ follower agents indexed by \mathcal{V} . The linear time-invariant model of agent $i \in \mathcal{V} \cup \{0\}$ is

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad (1)$$

where $x_i : [0, \infty) \rightarrow \mathbb{R}^m$ denotes the position, $\dot{x}_i : [0, \infty) \rightarrow \mathbb{R}^m$ denotes the velocity, $A \in \mathbb{R}^{m \times m}$ denotes the known constant state matrix, $B \in \mathbb{R}^{m \times n}$ denotes the known full-row rank control effectiveness matrix, and $u_i : [0, \infty) \rightarrow \mathbb{R}^n$ denotes the control input for agent i . Within this article, the followers can be categorized as either Byzantine or cooperative. Let the mapping $\mathcal{B} : [0, \infty) \rightarrow 2^{\mathcal{V}}$ define the time-varying set of Byzantine agents and the mapping $\mathcal{C} : [0, \infty) \rightarrow 2^{\mathcal{V}}$ define the time-varying set of cooperative agents, where $\mathcal{B}(t) \cap \mathcal{C}(t) = \emptyset$ and $\mathcal{B}(t) \cup \mathcal{C}(t) = \mathcal{V}$ for all $t \geq 0$. The following assumptions are made to facilitate the subsequent analysis.

Assumption 1: Each agent is capable of measuring its own position for all $t \geq 0$.

Assumption 2: The pair (A, B) is stabilizable.

Assumption 3: The control and state of the leader are continuous and bounded, i.e., there exists $M_0, \bar{M}_0 \in \mathbb{R}_{> 0}$, such that $\|u_0(t)\| \leq M_0$ and $\|x_0(t)\| \leq \bar{M}_0$ for all $t \geq 0$.

The flow of information between the followers of the MAS is modeled by a time-varying, weighted, and undirected graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), \mathcal{A}(t))$. Within this article, $(j, i) \in \mathcal{E}(t)$ if and only if $\|x_i(t) - x_j(t)\| \leq R_{\text{com}}$, where $R_{\text{com}} \in \mathbb{R}_{> 0}$ denotes the communication radius of each agent $i \in \mathcal{V} \cup \{0\}$. It then follows that $\mathcal{N}_i(t) = \{j \in \mathcal{V} \setminus \{i\} : \|x_i(t) - x_j(t)\| \leq R_{\text{com}}\}$. Let $\mathcal{E}_C(t)$ denote the undirected edge set and $\mathcal{A}_C(t)$ denote the weighted adjacency matrix associated with all cooperative followers in $\mathcal{C}(t)$. The sub-MAS consisting of only the cooperative followers is modeled by the time-varying, weighted, and undirected graph $\mathcal{G}_C(t) \triangleq (\mathcal{C}(t), \mathcal{E}_C(t), \mathcal{A}_C(t))$ and is referred to as the cooperative MAS (CMAS).

Assumption 4: The leader is a cooperative agent for all $t \geq 0$.

Assumption 5: The graph $\mathcal{G}_C(t)$ is connected for all $t \geq 0$, and $d_i(t) = 1$ for some $i \in \mathcal{C}(t)$ for all $t \geq 0$.²

IV. OBJECTIVES

The objective is to design distributed event- and self-triggered controllers for each cooperative agent governed by (1) to

²Future works will develop network connectivity maintenance methods for the CMAS. Potential inroads to such results include [35].

achieve approximate leader-follower consensus while identifying Byzantine adversaries and disregarding their disruptive inputs. Resilience to Byzantine adversaries is achieved by providing each follower the ability to detect Byzantine agents in their neighbor set and delete existing edges between themselves and all Byzantine neighbors. The result ensures each cooperative agent coordinates its motion based only on information from cooperative neighbors yielding $\mathcal{G}_C(t)$. To quantify the consensus objective, let $e_{1,i} : [0, \infty) \rightarrow \mathbb{R}^m$ be defined as

$$e_{1,i}(t) \triangleq x_i(t) - x_0(t). \quad (2)$$

Approximate leader-follower consensus is achieved when $e_{1,i}(t)$ is uniformly ultimately bounded (UUB) for all cooperative followers $i \in \mathcal{V}$. Since the behavior of Byzantine agents cannot be guaranteed, the objective can only be satisfied by the cooperative agents. The use of ETC/STC methods also motivates the development of an observer to provide state estimates. The state estimation error of follower i is defined by $e_{2,i} : [0, \infty) \rightarrow \mathbb{R}^m$, where

$$e_{2,i}(t) \triangleq \hat{x}_i(t) - x_i(t) \quad (3)$$

such that $\hat{x}_i : [0, \infty) \rightarrow \mathbb{R}^m$ denotes the state estimate of x_i for each $i \in \mathcal{V}$.

In Section V, the Byzantine adversary model, detection strategy, and mitigation protocol are presented. A general detection method enables each follower to distinguish between cooperative and Byzantine neighbors, and one such approach is introduced in Section V-B. Section V-C then presents a method enabling each follower to disregard data transmissions from Byzantine neighbors to satisfy the objective. In Section VI-A, an event-triggered strategy that satisfies the objective is developed, where Section VI-B provides a stability analysis for the controller and observer presented in Section VI-A. Section VI-C completes the ETC development by excluding the possibility of Zeno behavior for the proposed event-trigger. Section VII provides an STC extension, and Section VIII illustrates and compares the performance of the ETC and STC methods.

V. AGENT MODELS, DETECTION, AND MITIGATION

A detection method is presented in this section that provides distributed detection, where each agent formulates an inequality-based test to determine if an agent is Byzantine. Since (sufficiently disturbing) Byzantine actions violate the inequality, the detection method is considered instantaneous (or finite-time over the duration between communication events) in the sense that the inequality condition is interrogated at each communication event. Example strategies that enable finite-time Byzantine agent detection include [36]–[39]. Note that the methods in [36]–[38] are not directly applicable to the results in this article since these techniques utilize observers that require continuous communication. Furthermore, while [39] provides an alternative detection strategy, like our detector, it also does not guarantee detection of Byzantine adversaries that have complete knowledge of the detection strategy.

A. Agent Definitions

In this article, a Byzantine agent is defined as a noncompliant follower (cf. [23], [40]). Since noncompliance covers a broad scope of behaviors, we narrow our focus to two types of Byzantine agents, namely, Type I and Type II. A Type I Byzantine agent is defined as a follower that executes the intended controller but communicates false state information about itself to its neighbors. A Type II Byzantine agent is defined as a follower that executes a controller that is different from the intended controller or executes the intended controller under the influence of faulty hardware, while communicating true or no state information about itself to its neighbors. Consequently, a Type I Byzantine agent remains within the communication range of the CMAS since a Type I Byzantine agent executes the intended controller. In contrast, a Type II Byzantine agent may potentially leave the communication range of the CMAS. Note that nonresponsive communication is a characteristic of Type II Byzantine behavior and can occur due to a follower leaving the CMAS, radio failure, or malicious intent. A cooperative agent is defined as a follower that successfully executes the intended controller and provides true state information about itself to all its neighbors.

B. Agent Models and Detection

The Byzantine agent detection problem is similar to fault detection since a Byzantine agent can elicit undesirable behavior in an MAS. Several methods can enable Byzantine agent detection, e.g., performance-based fault detection and model-based fault detection [41], [42]. Such detection strategies are threshold-based methods, which compare an error metric to some user-defined threshold. There are multiple ways to determine a threshold, e.g., through a statistical analysis of data generated by a simulation/experimental study or an analysis-based derivation. Yet, no threshold strategy is perfect, and we cannot guarantee Byzantine agent detection for all instances and all types of Byzantine behavior. Hence, Byzantine agent detection is an open problem that requires further investigation.

Let $\{t_k^i\}_{k=0}^\infty \subset \mathbb{R}_{\geq 0}$ be an increasing sequence of event-times determined by a subsequently defined event-trigger (see Theorem 1). Note that t_k^i denotes the k^{th} instance follower i broadcasts its state to its neighbors. Suppose follower i broadcasts its state information to all followers $j \in \mathcal{N}_i(t)$ at time t_k^i , where $x_i(t_k^i)$ denotes the true state of follower i at time t_k^i , and $x_{i,j}(t_k^i) \in \mathbb{R}^m$ denotes the state information that is broadcast from follower i to follower j at time t_k^i . Given the ETC strategy, the state estimate of follower i is reset to the broadcast state of follower i as defined in (11), where $\hat{x}_i(t_k^i) = x_{i,j}(t_k^i)$ for each $j \in \mathcal{N}_i(t)$. Moreover, let $\hat{x}_i^-(t_k^i)$ denote the state estimate of follower i the moment before being reset to $x_{i,j}(t_k^i)$, where the mismatch between the state estimate of follower i before and after the reset at time t_k^i with respect to follower j is defined as

$$\bar{e}_{i,j}(t_k^i) \triangleq \hat{x}_i^-(t_k^i) - x_{i,j}(t_k^i). \quad (4)$$

Similarly, let

$$e_{2,i}^-(t_k^i) \triangleq \hat{x}_i^-(t_k^i) - x_i^-(t_k^i) \quad (5)$$

denote the state estimation error of follower i the moment before being reset at time t_k^i . In a cooperative setting, follower i broadcasts its true state to all of its neighbors at time t_k^i , i.e., $x_i(t_k^i) = x_{i,j}(t_k^i)$ for all $j \in \mathcal{N}_i(t)$, where it can be shown that $e_{2,i}^-(t_k^i) = \bar{e}_{i,j}(t_k^i)$. Let $\Psi_{i,k}$ denote an upper bound³ for $\|e_{2,i}^-(t_k^i)\|$, where

$$\|e_{2,i}^-(t_k^i)\| \leq \Psi_{i,k}. \quad (6)$$

Since $\|e_{2,i}^-(t_k^i)\| = \|\bar{e}_{i,j}(t_k^i)\|$ and $\|e_{2,i}^-(t_k^i)\| \leq \Psi_{i,k}$, $\|\bar{e}_{i,j}(t_k^i)\| \leq \Psi_{i,k}$. However, within a contested environment, it is possible for $x_i(t_k^i)$ to differ from $x_{i,j}(t_k^i)$, i.e., follower i can provide misinformation about its state. Therefore, the Byzantine agent detector that follower j uses to determine the status of follower $i \in \mathcal{N}_j(t)$ at time t_k^i is⁴

$$\Xi_{i,k} \triangleq \|\bar{e}_{i,j}(t_k^i)\| - \Psi_{i,k}, \quad (7)$$

where follower i is a cooperative neighbor of follower j at time t_k^i if $\Xi_{i,k} \leq 0$, and follower i is a Byzantine neighbor of follower j at time t_k^i if $\Xi_{i,k} > 0$. The estimated state of follower i before the reset is compared to the potential state estimate update of follower i after the reset in (4). Therefore, this strategy enables instantaneous detection of Byzantine adversaries that provide sufficiently disturbing state information.

The following atomic propositions are presented to precisely model the behavior of Type I Byzantine agents, Type II Byzantine agents, and cooperative agents within this article. Let $\mathcal{D}_{k,i}$ define the statement $\Xi_{i,k} \leq 0$, $\mathcal{X}_{i,j}$ define the statement $x_i(t) = x_{i,j}(t)$, and $\mathcal{T}_{k,i}$ define the statement $t_k^i \leq t_{k-1}^i + \Delta_i$, where $\Delta_i \in \mathbb{R}_{>0}$ is a constant parameter that is defined in Remark 6. Hence, $\mathcal{D}_{k,i}$, $\mathcal{X}_{i,j}$, and $\mathcal{T}_{k,i}$ are each either true or false.

Observe that $\mathcal{D}_{k,i}$, $\mathcal{X}_{i,j}$, and $\mathcal{T}_{k,i}$ encode acceptable agent motion, honest state reporting, and punctual state reporting, respectively, for follower i at time t_k^i . With respect to follower j and for each broadcast time t_k^i , follower $i \in \mathcal{N}_j(t)$ is

$$\begin{cases} \text{cooperative,} & \mathcal{D}_{k,i} \wedge \mathcal{X}_{i,j} \wedge \mathcal{T}_{k,i} \\ \text{Type I,} & \neg(\mathcal{D}_{k,i} \vee \mathcal{X}_{i,j}) \wedge \mathcal{T}_{k,i} \\ \text{Type II,} & \neg(\mathcal{D}_{k,i} \wedge \mathcal{T}_{k,i}) \wedge \mathcal{X}_{i,j}. \end{cases} \quad (8)$$

Hence, given the agent models in (8), the set of cooperative neighbors of follower j is given by

$$\mathcal{C}_j(t) \triangleq \{i \in \mathcal{N}_j(t) : \forall t_k^i \leq t \mathcal{D}_{k,i} \wedge \mathcal{T}_{k,i}\},$$

and the set of Byzantine neighbors of follower j is given by

$$\mathcal{B}_j(t) \triangleq \{i \in \mathcal{N}_j(t) : \exists t_k^i \leq t \neg(\mathcal{D}_{k,i} \wedge \mathcal{T}_{k,i})\}.$$

By convention, if follower i does not provide follower j with state information within a predetermined time period, i.e., $t_k^i > t_{k-1}^i + \Delta_i$, then follower j categorizes follower i as Byzantine, i.e., $i \in \mathcal{B}_j(t)$. Efforts in this article focus only on detectable Type I and Type II behaviors, and additional efforts are required

to generalize the development to broader classes of adversarial behaviors.

C. Mitigation

Since the objective is to achieve approximate leader-follower consensus by the cooperative followers, and both the detector in (7) and the communication timing condition, i.e., $t_k^i \leq t_{k-1}^i + \Delta_i$, allows each follower to identify their cooperative and Byzantine neighbors, the edge weights can be intermittently updated according to the status of each neighbor. Hence, for all $t \in [t_k^j, t_{k+1}^j)$, the piecewise constant edge weight $a_{ij}(t)$ is defined by

$$a_{ij}(t) \triangleq \begin{cases} 1, & j \in \mathcal{C}_i(t) \\ 0, & j \in \mathcal{B}_i(t). \end{cases} \quad (9)$$

From (9), the edge weight $a_{ij}(t) = 1$ for all $t \in [t_k^j, t_{k+1}^j)$ if follower j is a cooperative neighbor of follower i at time t_k^j , and $a_{ij}(t) = 0$ for all $t \in [t_k^j, t_{k+1}^j)$ if follower j is a Byzantine neighbor of follower i at time t_k^j .

Although the proposed detection and mitigation strategy enables each cooperative agent to insulate itself from Byzantine neighbors, the choice remains to allow each cooperative agent to communicate state information about itself to Byzantine neighbors or not. Communication could enable a Type I Byzantine adversary to be regulated to a desired location for remediation. However, communicating with a compromised agent could endanger security. Without loss of generality, the subsequent work enables communication from cooperative agents to their Byzantine neighbors.

Assumption 6: For all $j \in \mathcal{B}_i(t)$ and each $i \in \mathcal{V}$, follower i communicates state information about itself to follower j .⁵

Remark 2: While a threshold-based detection strategy is used to categorize the neighbors of follower i as either cooperative or Byzantine for each $i \in \mathcal{V}$, any other detection method that enables the construction of (9) can be implemented along with the observer in (10)–(11), controller in (12)–(13), and event-trigger in (24) to achieve the objective.

VI. STATE ESTIMATION AND EVENT-TRIGGERED CONTROL

A. Control and Observer Development

Given the use of an event-triggered strategy, the state estimate of follower $j \in \mathcal{V}$ is generated by the following observer:

$$\dot{\hat{x}}_j(t) \triangleq A\hat{x}_j(t), \quad t \in [t_k^j, t_{k+1}^j) \quad (10)$$

$$\hat{x}_j(t_k^j) \triangleq x_{j,i}(t_k^j), \quad (11)$$

which is synchronized among all followers $i \in \mathcal{N}_j(t) \cup \{j\}$. Note that for each $j \in \mathcal{V}$, self-communication does not occur and $\hat{x}_j(t_k^j) \triangleq x_j(t_k^j)$. Moreover, recall that $x_{j,i}(t_k^j) = x_j(t_k^j)$ provided follower j is cooperative. The solution to (10) over $[t_k^j, t_{k+1}^j)$ is $\hat{x}_j(t) = e^{A(t-t_k^j)}\hat{x}_j(t_k^j)$. Hence, accurate estimation

³See Remark 5 (for ETC) or Remark 8 (for STC) for specific examples of $\Psi_{i,k}$.

⁴ $\Psi_{i,k}$ in (6) represents a threshold that is used for Byzantine agent detection. As with any threshold detection method, if the adversary knows this threshold, then it can inject small perturbations below the threshold to yield some effect. These small perturbations can be modeled as a bounded disturbance, which leads to a larger UUB bound given the proposed controller. An open problem for all such detection strategies is to determine the most sensitive threshold that balances detection with false positives, especially in the presence of noise.

⁵The leader will also communicate state information about itself to its neighbors.

of the state of follower j requires an accurate initial condition, i.e., correct state estimation is ensured provided $\hat{x}_j(t_k^j) = x_j(t_k^j)$. Therefore, in this article, if $j \in \mathcal{B}_i(t_k^j)$, then $j \in \mathcal{B}_i(t)$ for all $t \geq t_k^j$. The Byzantine designation is permanent since follower i does not have an accurate state of the follower in question with which to propagate the estimate forward and compare to the corresponding sampled state. There are various methods to allow the inclusion of a rehabilitated follower into the cooperative follower set, e.g., trusted third party information can be used to reset the observer or a reputation algorithm such as in [43] can potentially be used as an alternative to adjust the graph edge weights. These potential extensions merit further investigation and are beyond the scope of this article.

Assumption 7: All followers are cooperative agents at time $t = 0$.

Assumption 7 enables the detection of Byzantine followers after the initial time $t = 0$. Like most fault or change detection methods, a baseline condition (i.e., at $t = 0$) is first required for comparison (cf. [41], [42]). However, Byzantine follower detection can be accomplished using the threshold-based detector in (7) at the initial time provided all followers know the initial position of their neighbors. Motivated by [12], the controller for follower $i \in \mathcal{V}$ is

$$u_i(t) \triangleq Kz_i(t) + Ke_{2,i}(t) \quad (12)$$

$$z_i(t) \triangleq \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (\hat{x}_j(t) - \hat{x}_i(t)) + d_i(t) (x_0(t) - \hat{x}_i(t)), \quad (13)$$

where $z_i : [0, \infty) \rightarrow \mathbb{R}^m$ is the estimate-based consensus control effort. The gain matrix $K \in \mathbb{R}^{n \times m}$ in (12) is designed as $K \triangleq B^T P$, where $P \in \mathbb{R}^{m \times m}$ is the symmetric and positive definite solution to the algebraic Riccati equation (ARE) given by

$$A^T P + PA - \lambda_{\min}(H_{\min}) 2PBB^T P + kI_m = 0_{m \times m}. \quad (14)$$

Note that $\lambda_{\min}(H_{\min}) \triangleq \min\{\lambda_{\min}(H_{\text{sym}}(t))\} \in \mathbb{R}_{>0}$, such that $H_{\text{sym}}(t) \triangleq \frac{1}{2}(H(t) + H(t)^T) \in \mathbb{R}^{N \times N}$ and $k \geq k_1 + \frac{\rho^2}{\delta}$, where $\rho > 2\sqrt{N}M_0 S_{\max}(PB) \in \mathbb{R}_{>0}$, $k_1 \triangleq k_2 + k_3$, and $k_2, k_3, \delta \in \mathbb{R}_{>0}$ are user-defined parameters. Rather than use a traditional sample-and-hold event-triggered consensus control law such as in [20], the combined use of (10)–(13) enable each follower i to continuously compute a control input that evolves according to the leader's drift dynamics [44].

By using (1)–(3), (12), and (13), the time-derivative of (2) can be expressed as

$$\begin{aligned} \dot{e}_{1,i}(t) &= Ae_{1,i}(t) - BKd_i(t)e_{1,i}(t) - BKd_i(t)e_{2,i}(t) \\ &\quad + BK \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (e_{1,j}(t) - e_{1,i}(t)) \\ &\quad + BK \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (e_{2,j}(t) - e_{2,i}(t)) \\ &\quad + BK e_{2,i}(t) - Bu_0(t). \end{aligned} \quad (15)$$

Similarly, using (1)–(3), (10), (12), and (13), the weak time-derivative⁶ of (3) can be expressed as

$$\begin{aligned} \dot{e}_{2,i}(t) &= Ae_{2,i}(t) + BKd_i(t)e_{2,i}(t) + BKd_i(t)e_{1,i}(t) \\ &\quad - BK \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (e_{2,j}(t) - e_{2,i}(t)) \\ &\quad - BK \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (e_{1,j}(t) - e_{1,i}(t)) \\ &\quad - BK e_{2,i}(t). \end{aligned} \quad (16)$$

The stacked forms of (2) and (3) are defined as

$$e_1(t) \triangleq [e_{1,1}^T(t), e_{1,2}^T(t), \dots, e_{1,N}^T(t)]^T \in \mathbb{R}^{mN}, \quad (17)$$

$$e_2(t) \triangleq [e_{2,1}^T(t), e_{2,2}^T(t), \dots, e_{2,N}^T(t)]^T \in \mathbb{R}^{mN}, \quad (18)$$

respectively. Substituting (15) and (16) into the time-derivative of (17) and (18), respectively, and compactly expressing the results with the Kronecker product yields

$$\begin{aligned} \dot{e}_1(t) &= (I_N \otimes A) e_1(t) + ((I_N - H(t)) \otimes BK) e_2(t) \\ &\quad - (H(t) \otimes BK) e_1(t) - (1_N \otimes Bu_0(t)), \end{aligned} \quad (19)$$

$$\begin{aligned} \dot{e}_2(t) &= (I_N \otimes (A - BK)) e_2(t) + (H(t) \otimes BK) e_2(t) \\ &\quad + (H(t) \otimes BK) e_1(t), \end{aligned} \quad (20)$$

respectively. The stacked form of (13) is defined by $z(t) \triangleq [z_1^T(t), z_2^T(t), \dots, z_N^T(t)]^T \in \mathbb{R}^{mN}$. Substituting $\hat{x}(t) \triangleq [\hat{x}_1^T(t), \hat{x}_2^T(t), \dots, \hat{x}_N^T(t)]^T \in \mathbb{R}^{mN}$ and (13) for all $i \in \mathcal{V}$ into $z(t)$ yields

$$z(t) = - (H(t) \otimes I_m) \hat{x}(t) + (D(t) \otimes I_m) (1_N \otimes x_0(t)). \quad (21)$$

Moreover, substituting (2), (3), and (13) for all $i \in \mathcal{V}$ into $z(t)$ yields

$$z(t) = - (H(t) \otimes I_m) e_2(t) - (H(t) \otimes I_m) e_1(t). \quad (22)$$

Since the objective is achieved when $e_{1,i}(t)$ is UUB for all cooperative followers $i \in \mathcal{V}$, it is sufficient to show that $e_1(t)$ is UUB. However, since $e_1(t)$ may contain error signals belonging to Byzantine followers, where the behavior of Byzantine followers cannot be controlled, the objective can only be guaranteed for the cooperative followers. Hence, if follower j is categorized as a Byzantine agent at t_k^j for some $k \in \mathbb{Z}_{\geq 0}$, then $e_{1,j}(t) \triangleq 0_m$ and $e_{2,j}(t) \triangleq 0_m$ for all $t \geq t_k^j$.

B. Stability Analysis

To facilitate the subsequent stability analysis, let

$$\gamma \triangleq \max \{ \|(I_N - H(t)) \otimes 2PBB^T P\| \} \in \mathbb{R}_{>0},$$

$$\phi_1 \triangleq k_2 - \frac{\kappa(2k_3 + \gamma)}{2} \in \mathbb{R}_{>0},$$

$$\phi_2 \triangleq k_3 + \frac{2k_3 + \gamma}{2\kappa} \in \mathbb{R}_{>0},$$

$$\phi_3 \triangleq \frac{k_3}{\max \{ \|H(t) \otimes I_m\|^2 \}} \in \mathbb{R}_{>0},$$

⁶Weak time-derivative refers to the existence of the time-derivative for almost all time.

where $\kappa \in \mathbb{R}_{>0}$ is a user-defined parameter used in Young's inequality. Moreover, let $\bar{\delta} \triangleq \delta + \theta \in \mathbb{R}_{>0}$, where $\delta \in \mathbb{R}_{>0}$ is a user-defined parameter used to compensate for the effect of the leader's control input, and $\theta \in \mathbb{R}_{>0}$ is a user-defined parameter used to exclude Zeno behavior. Based on the definition of $\bar{\delta}$, we define additional constants, to facilitate the analysis, as

$$\beta_1 \triangleq \sqrt{\frac{\lambda_{\max}(I_N \otimes P) \bar{\delta}}{\lambda_{\min}(I_N \otimes P) \phi_1}} \in \mathbb{R}_{>0},$$

$$\beta_2 \triangleq \sqrt{\frac{V_1(e_1(0))}{\lambda_{\min}(I_N \otimes P)}} \in \mathbb{R}_{\geq 0},$$

$$\beta_3 \triangleq \frac{\phi_1}{2\lambda_{\max}(I_N \otimes P)} \in \mathbb{R}_{>0}.$$

Given Assumption 2, the ARE in (14) has a positive definite solution P provided $\lambda_{\min}(H_{\min}) > 0$ [12], [44]. The following lemma shows $\lambda_{\min}(H_{\min}) > 0$.

Lemma 1: Suppose Assumptions 4–6 are satisfied. If all cooperative followers detect their Byzantine neighbors and employ (9), then $\lambda_{\min}(H_{\min}) > 0$.

Proof: See Appendix A. \square

Theorem 1: The edge weight policy in (9), state observer in (10) and (11), and controller in (12) and (13) ensure the leader-follower error $e_1(t)$ is globally UBB as

$$\|e_1(t)\| \leq \beta_1 + \beta_2 e^{-\beta_3 t} \quad (23)$$

provided state feedback is available as dictated by the event-trigger in

$$t_{k+1}^i \triangleq \inf \left\{ t > t_k^i : \phi_2 \|e_{2,i}(t)\|^2 \geq \phi_3 \|z_i(t)\|^2 + \frac{\theta}{N} \right\} \quad (24)$$

for all $i \in \mathcal{V}$, Assumptions 1–7 are satisfied, and the following sufficient user-defined parameter conditions are selected as follows: Use Algorithm 1 to determine P , and select $\kappa > 0$, $\delta > 0$, $\theta > 0$, $\rho > 2\sqrt{N}M_0S_{\max}(PB)$, $k_3 > 0$, $k_2 > \frac{\kappa(2k_3+\gamma)}{2}$, and $k \geq k_1 + \frac{\rho^2}{\delta}$.

Proof: Consider the following candidate Lyapunov function $V_1 : \mathbb{R}^{mN} \rightarrow \mathbb{R}_{\geq 0}$ defined as:

$$V_1(e_1(t)) \triangleq e_1^T(t) (I_N \otimes P) e_1(t). \quad (25)$$

By the Rayleigh quotient,

$$\begin{aligned} \lambda_{\min}(I_N \otimes P) \|e_1(t)\|^2 &\leq V_1(e_1(t)) \\ &\leq \lambda_{\max}(I_N \otimes P) \|e_1(t)\|^2. \end{aligned} \quad (26)$$

Suppose $g : [0, \infty) \rightarrow \mathbb{R}^{mN}$ is a Filippov solution to the differential inclusion $\dot{g}(t) \in \bar{K}[h](g(t))$, where $g(t) = e_1(t)$, $\bar{K}[\cdot]$ is defined as in [45], and $h : \mathbb{R}^{mN} \rightarrow \mathbb{R}^{mN}$ is defined as $h(g(t)) = \dot{e}_1(t)$. The time-derivative of V_1 exists almost everywhere (a.e.), i.e., for almost all $t \in [0, \infty)$, and

$$\dot{V}_1(g(t)) \stackrel{a.e.}{\in} \dot{\tilde{V}}_1(g(t)), \quad (27)$$

where $\dot{\tilde{V}}_1(g(t))$ is the generalized time-derivative of V_1 along the Filippov trajectories of $\dot{g}(t) = h(g(t))$. By [46, eq. 13], $\dot{\tilde{V}}_1(g(t)) \triangleq \bigcap_{\xi \in \partial V_1(g(t))} \xi^T [\bar{K}[h]^T(g(t)), 1]^T$, where $\partial V_1(g(t))$ denotes the Clarke generalized gradient of

$V_1(g(t))$. Since $V_1(g(t))$ is continuously differentiable in $g(t)$, $\partial V_1(g(t)) = \{\nabla V_1(g(t))\}$, where ∇ denotes the gradient operator. The generalized time-derivative of (25) is

$$\dot{\tilde{V}}_1(g(t)) \subseteq 2e_1^T(t) (I_N \otimes P) \bar{K}[h](g(t)). \quad (28)$$

Using the calculus of $\bar{K}[\cdot]$ from [45] along with (28) and simplifying the substitution of (19) and $K = B^T P$ into the generalized time-derivative of (25) yields

$$\begin{aligned} \dot{\tilde{V}}_1(g(t)) &\subseteq \{e_1^T(t) (I_N \otimes (A^T P + PA)) e_1(t) \\ &\quad - e_1^T(t) \bar{K}[H(t) \otimes 2PBB^T P] e_1(t) \\ &\quad + e_1^T(t) \bar{K}[(I_N - H(t)) \otimes 2PBB^T P] e_2(t)] \\ &\quad - \{e_1^T(t) (1_N \otimes 2PBu_0(t))\}. \end{aligned} \quad (29)$$

Let $M \triangleq H(t) \otimes 2PBB^T P \in \mathbb{R}^{mN \times mN}$, $M_{\text{sym}} \triangleq \frac{1}{2}(M + M^T) \in \mathbb{R}^{mN \times mN}$, and $M_{\text{skew}} \triangleq \frac{1}{2}(M - M^T) \in \mathbb{R}^{mN \times mN}$, where $M = M_{\text{sym}} + M_{\text{skew}}$. Since M_{skew} is a skew symmetric matrix, we see that $e_1^T(t)(H(t) \otimes 2PBB^T P)e_1(t) = e_1^T(t)M_{\text{sym}}e_1(t)$. It follows from the definition of M_{sym} that $e_1^T(t)M_{\text{sym}}e_1(t) = e_1^T(t)(H_{\text{sym}}(t) \otimes 2PBB^T P)e_1(t)$. Hence, $e_1^T(t)(H(t) \otimes 2PBB^T P)e_1(t) = e_1^T(t)(H_{\text{sym}}(t) \otimes 2PBB^T P)e_1(t)$. Since $H_{\text{sym}}(t)$ is a real, symmetric matrix, we then see that $H_{\text{sym}}(t)$ is diagonalizable, where there exists an orthogonal eigenvector matrix $T(t) \in \mathbb{R}^{N \times N}$ and eigenvalue matrix $\Lambda(t) \in \mathbb{R}^{N \times N}$, such that $H_{\text{sym}}(t) = T(t)\Lambda(t)T^T(t) \in \mathbb{R}^{N \times N}$. Hence, the eigendecomposition of $H_{\text{sym}}(t)$, the ARE, and (27) enable (29) to yield

$$\begin{aligned} \dot{V}_1(e_1(t)) &\stackrel{a.e.}{\leq} -ke_1^T(t)e_1(t) - e_1^T(t) (1_N \otimes 2PBu_0(t)) \\ &\quad + e_1^T(t) ((I_N - H(t)) \otimes 2PBB^T P) e_2(t). \end{aligned} \quad (30)$$

Using Assumption 3 and selecting $k \geq k_1 + \frac{\rho^2}{\delta}$, equation (30) can be upper bounded by

$$\dot{V}_1(e_1(t)) \stackrel{a.e.}{\leq} -k_1 \|e_1(t)\|^2 + \delta + \gamma \|e_2(t)\| \|e_1(t)\|. \quad (31)$$

Using (22), equation (31) can be further upper bounded by

$$\begin{aligned} \dot{V}_1(e_1(t)) &\stackrel{a.e.}{\leq} -\phi_1 \|e_1(t)\|^2 + \bar{\delta} + \sum_{i \in \mathcal{V}} \left[\phi_2 \|e_{2,i}(t)\|^2 \right. \\ &\quad \left. - \phi_3 \|z_i(t)\|^2 - \frac{\theta}{N} \right]. \end{aligned} \quad (32)$$

By selecting $k_3 > 0$ and $k_2 > \frac{\kappa(2k_3+\gamma)}{2}$, $\phi_1 > 0$, $\phi_2 > 0$, and $\phi_3 > 0$. Based on (32), the event-trigger for each follower $i \in \mathcal{V}$ is given by (24). Hence, provided state feedback is available according to (24), it follows that:

$$\dot{V}_1(e_1(t)) \stackrel{a.e.}{\leq} -\phi_1 \|e_1(t)\|^2 + \bar{\delta} \quad (33)$$

for all $t \geq 0$. Substituting (26) into (33) yields

$$\dot{V}_1(e_1(t)) \stackrel{a.e.}{\leq} -\frac{\phi_1}{\lambda_{\max}(I_N \otimes P)} V_1(e_1(t)) + \bar{\delta}. \quad (34)$$

Since the set of discontinuities as given by $\bigcup_{k \in \mathbb{Z}_{\geq 0}} \bigcup_{i \in \mathcal{V}} \{t_k^i\}$ is countable, $\dot{V}_1(e_1(t))$ and $V_1(e_1(t))$ are Lebesgue integrable over $\mathbb{R}_{\geq 0}$. The result in (23) then follows from (34) [47, Th. 2.5.1. Part V]. Observe that the constant β_1 can be made small, resulting in

a small steady-state error for $e_1(t)$. Since $e_1(t) \in \mathcal{L}_\infty$ by (23), $e_{1,i}(t) \in \mathcal{L}_\infty$ for all $i \in \mathcal{C}(t)$. From Assumption 3, $x_0(t) \in \mathcal{L}_\infty$. By (2) and $x_0(t) \in \mathcal{L}_\infty$, $x_i(t) \in \mathcal{L}_\infty$ for each $i \in \mathcal{C}(t)$. Given (10) and (11), we see that $\hat{x}_i(t) = e^{A(t-t_k^i)} \hat{x}_i(t_k^i)$ over $t \in [t_k^i, t_{k+1}^i)$, where $\hat{x}_i(t_k^i) = x_i(t_k^i)$ for all $k \in \mathbb{Z}_{\geq 0}$. Therefore, $\hat{x}_i(t) \in \mathcal{L}_\infty$ for each $i \in \mathcal{C}(t)$, which then implies $e_{2,i}(t) \in \mathcal{L}_\infty$ by (3) for each $i \in \mathcal{C}(t)$. Hence, $u_i(t) \in \mathcal{L}_\infty$ for each $i \in \mathcal{C}(t)$. ■

Remark 3: There are two reasons why UUB stability is obtained rather than asymptotic stability. In (30), since

$$\begin{aligned} & -e_1^T(t) (1_N \otimes 2PBu_0(t)) \\ & \leq \|e_1(t)\| \|1_N \otimes 2PBu_0(t)\|, \end{aligned}$$

where $\|1_N \otimes 2PBu_0(t)\| \leq c$ for some $c \in \mathbb{R}_{>0}$ by Assumption 3, $-(1_N \otimes 2PBu_0(t)) e_1(t) \leq c \|e_1(t)\|$. Since the controller in (12)–(13) does not employ a sliding mode term or one of its variants, $c \|e_1(t)\|$ can only be compensated with high gain, which results in a residual $\delta > 0$. Note that it is not clear how to incorporate a sliding mode term into the controller in (12)–(13) since doing so would require $\text{sgn}(e_{1,i}(t))$ and $e_{1,i}(t)$ is not measurable by all followers. Next, the event-triggered strategy requires exclusion from Zeno behavior, which is accomplished by injecting $\theta > 0$ into (32), as shown in the proof of Theorem 2. Hence, the $\theta > 0$ term is combined with the residual $\delta > 0$ resulting in $\bar{\delta} = \delta + \theta$, and hence, UUB stability.

C. Exclusion of Zeno Behavior

Theorem 2: For each follower $i \in \mathcal{V}$, the difference between consecutive broadcast times generated by the event-trigger of follower i in (24) is uniformly lower bounded by

$$t_{k+1}^i - t_k^i \geq \frac{1}{\|A - BK\|} \ln \left(\frac{\|A - BK\|}{\|BK\| \bar{z}_i} \sqrt{\frac{\theta}{N\phi_2}} + 1 \right) \quad (35)$$

for all $k \in \mathbb{Z}_{\geq 0}$.

Proof: Let $t \in [t_k^i, \infty)$. Substituting (1), (10), and (12) into the time-derivative of (3) yields $\dot{e}_{2,i}(t) = (A - BK)e_{2,i}(t) - BKz_i(t)$. Since $\|x_0(t)\| \leq \bar{M}_0$ by Assumption 3 and $\hat{x} \in \mathcal{L}_\infty$ by the proof of Theorem 1, equation (21) implies the existence of $\bar{z}_i \in \mathbb{R}_{>0}$, such that $\|z_i(t)\| \leq \bar{z}_i$ for all $t \in \mathbb{R}_{\geq 0}$. It then follows that:

$$\|\dot{e}_{2,i}(t)\| \leq \|A - BK\| \|e_{2,i}(t)\| + \|BK\| \bar{z}_i. \quad (36)$$

Let $v_i : [t_k^i, \infty) \rightarrow \mathbb{R}_{\geq 0}$ satisfy $\dot{v}_i(t) = \|A - BK\| v_i(t) + \|BK\| \bar{z}_i$ with initial condition $v_i(t_k^i) = \|e_{2,i}(t_k^i)\|$. Then, $v_i(t_k^i) = 0$ and

$$v_i(t) = \frac{\|BK\| \bar{z}_i}{\|A - BK\|} \left(e^{\|A - BK\|(t-t_k^i)} - 1 \right). \quad (37)$$

Since $\frac{d}{dt} \|e_{2,i}(t)\| \stackrel{a.e.}{\leq} \|\dot{e}_{2,i}(t)\|$, equation (36) implies $\frac{d}{dt} \|e_{2,i}(t)\| \stackrel{a.e.}{\leq} \|A - BK\| \|e_{2,i}(t)\| + \|BK\| \bar{z}_i$, where $\|e_{2,i}(t)\| \leq v_i(t)$ for all $t \in [t_k^i, \infty)$. Since $\|e_{2,i}(t)\| \leq v_i(t)$ and $\|z_i(t)\| \geq 0$, equation (24) implies (35), where $\frac{1}{\|A - BK\|} \ln \left(\frac{\|A - BK\|}{\|BK\| \bar{z}_i} \sqrt{\frac{\theta}{N\phi_2}} + 1 \right) > 0$ since $\frac{\|A - BK\|}{\|BK\| \bar{z}_i} \sqrt{\frac{\theta}{N\phi_2}} > 0$. ■

Remark 4: Since the event-trigger in (24) is free from Zeno behavior by the proof of Theorem 2, no follower continuously broadcasts state information about itself to its neighbors. Moreover, the difference between consecutive event-times can be made arbitrarily large by selecting a large θ . Since $\beta_1 = \sqrt{\frac{\lambda_{\max}(I_N \otimes P) \bar{\delta}}{\lambda_{\min}(I_N \otimes P) \phi_1}}$, where $\bar{\delta} = \delta + \theta$, selecting a large θ forces β_1 to be large as well. Hence, there is a tradeoff between the size of the neighborhood containing the cooperative followers and leader once approximate consensus is achieved and the amount of communication.

Remark 5: Define (37) over $[t_{k-1}^i, t_k^i)$, and observe that since $\|e_{2,i}(t)\| \leq v_i(t)$ over $[t_{k-1}^i, t_k^i)$, $\|e_{2,i}(t_k^i)\| = 0$, and $v_i(t)$ in (37) is strictly increasing, it follows that $\|e_{2,i}(t)\|$ defined in (5) satisfies the inequality $\|e_{2,i}(t)\| \leq v_i(t_k^i)$. Therefore, $v_i(t_k^i)$ is a candidate for $\Psi_{i,k}$.

Remark 6: If the event-trigger condition in (24) is satisfied for all $t \geq 0$, then

$$\phi_2 \|e_{2,i}(t)\|^2 \leq \phi_3 \|z_i(t)\|^2 + \frac{\theta}{N}.$$

Since $\|z_i(t)\| \leq \bar{z}_i$ for all $t \in \mathbb{R}_{\geq 0}$, substituting $\|z_i(t)\| \leq \bar{z}_i$ into $\phi_2 \|e_{2,i}(t)\|^2 \leq \phi_3 \|z_i(t)\|^2 + \frac{\theta}{N}$ yields

$$\|e_{2,i}(t)\| \leq \sqrt{\frac{\phi_3 \bar{z}_i^2}{\phi_2} + \frac{\theta}{N\phi_2}} \in \mathbb{R}_{>0}.$$

Since $\|e_{2,i}(t)\| \leq v_i(t)$ for all $t \in [t_{k-1}^i, t_k^i)$, $v_i(t_k^i) > 0$, and $\|e_{2,i}(t_k^i)\| = 0$, $\|e_{2,i}(t)\| \leq v_i(t)$ for all $t \in [t_{k-1}^i, t_k^i]$. Since $v_i(t)$ will reach $\sqrt{\frac{\phi_3 \bar{z}_i^2}{\phi_2} + \frac{\theta}{N\phi_2}}$ before or at the same time as $\|e_{2,i}(t)\|$, we then see that $\frac{\|BK\| \bar{z}_i}{\|A - BK\|} (e^{\|A - BK\|(t_k^i - t_{k-1}^i)} - 1) \leq \sqrt{\frac{\phi_3 \bar{z}_i^2}{\phi_2} + \frac{\theta}{N\phi_2}}$ implies $t_k^i \leq t_{k-1}^i + \Delta_{i,\min}$, where

$$\Delta_{i,\min} \triangleq \frac{1}{\|A - BK\|} \ln \left(\frac{\|A - BK\|}{\|BK\| \bar{z}_i} \sqrt{\frac{\phi_3 \bar{z}_i^2}{\phi_2} + \frac{\theta}{N\phi_2}} + 1 \right) \quad (38)$$

and $\Delta_{i,\min} \in \mathbb{R}_{>0}$. Hence, $\|e_{2,i}(t)\| \leq \sqrt{\frac{\phi_3 \bar{z}_i^2}{\phi_2} + \frac{\theta}{N\phi_2}}$ for all $t \geq 0$ provided $t_k^i \leq t_{k-1}^i + \Delta_i$ for each $k \in \mathbb{Z}_{>0}$, where Δ_i is a user-defined parameter to be selected, such that $\Delta_i \geq \Delta_{i,\min}$. Note that an analytical upper bound for Δ_i requires the derivation of a nonzero lower bound for $\|e_{2,i}(t)\|$, which is not obvious.

Algorithm 1 presents a method for parameter selection.

VII. SELF-TRIGGERED CONTROL

When the trigger condition in (24) is true, follower i will broadcast its state to each follower $j \in \mathcal{N}_i(t)$ to reset (3) and ensure (33). Such an ETC strategy requires follower i to continuously monitor (24) and for each follower $j \in \mathcal{N}_i(t)$ to continuously sense for follower i 's broadcast. An STC strategy is developed in this section, where follower i determines and reports to its neighbors the future time when its own trigger condition will become true, eliminating the need for followers to continuously monitor for a neighbor's broadcast.

Based on (32), stability is preserved when $\phi_2 \|e_{2,i}(t)\|^2 - \phi_3 \|z_i(t)\|^2 - \frac{\theta}{N} \leq 0$ for each $i \in \mathcal{V}$. Since $\phi_3 > 0$ and

Algorithm 1: Parameter Selection Protocol.

```

1: Select  $\delta, \kappa, \theta, k, k_3 \in \mathbb{R}_{>0}$ .
2: Compute  $\bar{\delta} = \delta + \theta$ .
3: Compute  $\lambda_{\min}(H_{\min}) = \min\{\lambda_{\min}(H_{\text{sym}}(t))\}$ .
4: while true do
5:   Compute  $P$  from (14).
6:   Select  $\rho > 2\sqrt{N}M_0S_{\max}(PB)$ .
7:   Compute
      $\gamma = \max\{\|(I_N - H(t)) \otimes 2PB B^T P\|\}$ .
8:   Select  $k_2 > \frac{\kappa(2k_3 + \gamma)}{2}$ .
9:   Compute  $k_1 = k_2 + k_3$ .
10:  if  $k_1 + \frac{\rho^2}{\delta} \leq k$  then
11:    break
12:  else
13:     $k = k_1 + \frac{\rho^2}{\delta}$ .
14:  end if
15: end while
16: Compute  $K = B^T P$ .
17: Compute  $\phi_1 = k_2 - \frac{\kappa(2k_3 + \gamma)}{2}$ .
18: Compute  $\phi_2 = k_3 + \frac{2k_3 + \gamma}{2\kappa}$ .
19: Compute  $\phi_3 = \frac{k_3}{\max\{\|H(t) \otimes I_m\|^2\}}$ .

```

$\|z_i(t)\| \geq 0$, $\phi_3\|z_i(t)\|^2 \geq 0$, where stability is preserved provided $\phi_2\|e_{2,i}(t)\|^2 - \frac{\theta}{N} \leq 0$ for each $i \in \mathcal{V}$. While triggering based on $\phi_2\|e_{2,i}(t)\|^2 - \frac{\theta}{N} \geq 0$ results in more conservative event-times for follower i than when triggering based on $\phi_2\|e_{2,i}(t)\|^2 - \phi_3\|z_i(t)\|^2 - \frac{\theta}{N} \geq 0$, the former results in a simpler condition from which to develop a self-trigger. Let t_k^i mark the k^{th} instance when $\phi_2\|e_{2,i}(t)\|^2 - \frac{\theta}{N} \geq 0$. Hence, an event for follower i occurs at t_k^i provided $\phi_2\|e_{2,i}(t_k^i)\|^2 - \frac{\theta}{N} \geq 0$. Note that for $t \in [t_k^i, t_{k+1}^i)$, $\phi_2\|e_{2,i}(t)\|^2 - \frac{\theta}{N} \leq 0$ since $\|e_{2,i}(t_k^i)\| = 0$ and $\|e_{2,i}(t)\|$ may increase otherwise.

Substituting (1) and (10) into the time-derivative of (3) yields

$$\dot{e}_{2,i}(t) = Ae_{2,i}(t) - Bu_i(t). \quad (39)$$

The evolution of (3) is governed by (39), where the solution to (39) is not available since $u_i(t)$ is unknown *a priori*. Therefore, follower i cannot determine its own event-times. Let $\check{e}_{2,i} : [0, \infty) \rightarrow \mathbb{R}_{\geq 0}$ denote an estimate of $\|e_{2,i}(t)\|$, and let $\{\hat{t}_k^i\}_{k=1}^\infty \subset \mathbb{R}_{\geq 0}$ be an increasing sequence of estimated event-times determined by a subsequently developed self-trigger for follower i . The estimate $\check{e}_{2,i}(t)$ is designed, such that

$$\phi_2\|e_{2,i}(t)\|^2 - \frac{\theta}{N} \leq \phi_2\check{e}_{2,i}^2(t) - \frac{\theta}{N} \quad (40)$$

holds for all $t \in [t_k^i, t_{k+1}^i)$. Hence, executing the consensus protocol based on the estimated event-times originating from a self-trigger using $\check{e}_{2,i}(t)$ for all $i \in \mathcal{V}$ ensures the stability of the MAS. Based on the subsequent stability analysis, for each $[t_k^i, t_{k+1}^i)$, the estimate $\check{e}_{2,i}(t)$ is designed as

$$\check{e}_{2,i}(t) \triangleq \xi_i \left(e^{S_{\max}(A)(t-t_k^i)} - 1 \right), \quad (41)$$

$$\xi_i \triangleq \frac{S_{\max}(B)M_i}{S_{\max}(A)} \in \mathbb{R}_{>0}. \quad (42)$$

In (42), $M_i \in \mathbb{R}_{>0}$ is a known upper bound for $\|u_i(t)\|$, which exists given the proof of Theorem 1. Furthermore, Lemma 2 provides an upper bound for $\|u_i(t)\|$.

Lemma 2: If the conditions in Theorem 1 are satisfied, then for each $i \in \mathcal{C}(t)$ and all $t \geq 0$

$$\begin{aligned} \|u_i(t)\| &\leq S_{\max}(B^T P) \max\{\|H(t) \otimes I_m\|\} \sqrt{\frac{\theta}{\phi_2}} \\ &\quad + S_{\max}(B^T P) \max\{\|H(t) \otimes I_m\|\} (\beta_1 + \beta_2) \\ &\quad + S_{\max}(B^T P) \sqrt{\frac{\theta}{N\phi_2}}. \end{aligned} \quad (43)$$

Proof: See Appendix B.

Theorem 3: The estimate given by (41) and (42) satisfies (40) for all $t \in [t_k^i, t_{k+1}^i)$, where $\hat{t}_{k+1}^i \leq t_{k+1}^i$, such that \hat{t}_{k+1}^i originates from the self-trigger given by

$$\hat{t}_{k+1}^i \triangleq \inf \left\{ t > \hat{t}_k^i : \phi_2 \check{e}_{2,i}(t)^2 \geq \frac{\theta}{N} \right\}. \quad (44)$$

Proof: To satisfy (40), it is equivalent to show that $\check{e}_{2,i}(t) \geq \|e_{2,i}(t)\|$ for all $t \in [t_k^i, t_{k+1}^i)$. Let $t \in [t_k^i, t_{k+1}^i)$. A Lyapunov-like function $V_{2,i} : \mathbb{R}^m \rightarrow \mathbb{R}_{\geq 0}$ is defined as

$$V_{2,i}(e_{2,i}(t)) \triangleq \frac{1}{2} e_{2,i}^T(t) e_{2,i}(t), \quad (45)$$

which is continuously differentiable over $t \in [t_k^i, t_{k+1}^i)$. Substituting (39) into the time-derivative of (45) results in

$$\begin{aligned} \dot{V}_{2,i}(e_{2,i}(t)) &\leq S_{\max}(A) \|e_{2,i}(t)\|^2 \\ &\quad + S_{\max}(B) \|e_{2,i}(t)\| \|u_i(t)\|. \end{aligned} \quad (46)$$

Based on the proof of Theorem 1, $u_i(t) \in \mathcal{L}_\infty$, where $\|u_i(t)\| \leq M_i$, such that M_i is a known bounding constant. Hence, equation (46) can be upper bounded by

$$\begin{aligned} \dot{V}_{2,i}(e_{2,i}(t)) &\leq 2S_{\max}(A)V_{2,i}(e_{2,i}(t)) \\ &\quad + S_{\max}(B)M_i \sqrt{2V_{2,i}(e_{2,i}(t))}. \end{aligned} \quad (47)$$

Since follower i broadcasts its state at t_k^i , $V_{2,i}(e_{2,i}(t_k^i)) = 0$. Invoking the Comparison lemma in [48, Lemma 3.4] on (47) yields

$$V_{2,i}(e_{2,i}(t)) \leq \left(\frac{\sqrt{2}\xi_i \left(e^{S_{\max}(A)(t-t_k^i)} - 1 \right)}{2} \right)^2. \quad (48)$$

Substituting (45) into (48) yields $\|e_{2,i}(t)\| \leq \xi_i (e^{S_{\max}(A)(t-t_k^i)} - 1) = \check{e}_{2,i}(t)$. Hence, equation (40) holds for all $t \in [t_k^i, t_{k+1}^i)$, where the conditions in (24) and (44) imply $\hat{t}_{k+1}^i \leq t_{k+1}^i$.⁷ ■

Remark 7: The self-trigger condition in (44) is free from Zeno behavior by a similar argument provided in the proof of Theorem 2.

Remark 8: Define (41) over $[t_{k-1}^i, t_k^i)$, and observe that since $\|e_{2,i}(t)\| \leq \check{e}_{2,i}(t)$ over $[t_{k-1}^i, t_k^i)$, $\|e_{2,i}(t_k^i)\| = 0$, and $\check{e}_{2,i}(t)$ in (41) is strictly increasing, it follows that $\|e_{2,i}^-(t)\|$ defined in (5) satisfies the inequality $\|e_{2,i}^-(t)\| \leq \check{e}_{2,i}(t_k^i)$. Therefore, $\check{e}_{2,i}(t_k^i)$ is a candidate for $\Psi_{i,k}$.

⁷While (41) is initialized at t_k^i in the development, implementation of the STC strategy requires communication at \hat{t}_k^i .

VIII. SIMULATION STUDY

A simulation study is included to demonstrate and compare the performances of the developed approaches. The simulated MAS consists of five follower agents and a single leader agent. The initial positions of each agent, which are equivalent for all simulations, are $x_0(0) = [6, 2]^T$ m, $x_1(0) = [12, 2.5]^T$ m, $x_2(0) = [12, 2]^T$ m, $x_3(0) = [12, 1.5]^T$ m, $x_4(0) = [13, 2.25]^T$ m, and $x_5(0) = [13, 1.75]^T$ m. The known state and control effectiveness matrices used in all simulations are given by

$$A \triangleq \begin{bmatrix} 0.05 & 0 \\ 0 & 0 \end{bmatrix}, \quad B \triangleq \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}.$$

The known desired trajectory $x_d : [0, \infty) \rightarrow \mathbb{R}^2$ of the leader is

$$x_d(t) \triangleq [5 \cos(0.2\pi t), 5 \sin(0.4\pi t)]^T, \quad (49)$$

while the leader's trajectory tracking error $e_0 : [0, \infty) \rightarrow \mathbb{R}^2$ is

$$e_0(t) \triangleq x_d(t) - x_0(t). \quad (50)$$

The leader's tracking error in (50) can be globally exponentially regulated using the following controller:

$$u_0(t) \triangleq B^{-1}(\dot{x}_d(t) - Ax_0(t) + k_0 e_0(t)), \quad (51)$$

where $k_0 > 0$.

Lemma 3: The controller of the leader provided in (51) ensures (50) is globally exponentially regulated and $x_0, u_0 \in \mathcal{L}_\infty$ provided the desired trajectory satisfies $x_d, \dot{x}_d \in \mathcal{L}_\infty$, Assumption 1 is satisfied, the right pseudo inverse of the control effectiveness matrix, i.e., B , exists, and k_0 is selected, such that $k_0 > 0$.

Proof: See Appendix C.

All simulations are 12-s long and use an integration time-step of 1.00×10^{-5} s. Additionally, all ETC and STC simulations used the following parameters, which originate from Algorithm 1: $k_0 = 3s^{-1}$, $\delta = 3 \times 10^7$, $\kappa = 1.00 \times 10^{-2}$, $\rho = 4.87 \times 10^5$, $\gamma = 1.54 \times 10^5$, $k_1 = 2.78 \times 10^3$, $k_2 = 1.78 \times 10^3$, $k_3 = 1.00 \times 10^3$, $k = 1.07 \times 10^4$, $\theta = 1.00 \times 10^3 \text{ m}^2 \cdot \text{s}^{-1}$, $\phi_1 = 1.00 \times 10^3 \text{ s}^{-1}$, $\phi_2 = 7.80 \times 10^6 \text{ s}^{-1}$, $\phi_3 = 37.87 \text{ s}^{-1}$, $M_1 = 800 \text{ m}$, $M_2 = 800 \text{ m}$, $M_3 = 800 \text{ m}$, $M_4 = 800 \text{ m}$, and $M_5 = 800 \text{ m}$.

A. Benchmark Simulation

As a benchmark, the MAS is first simulated by using an event-triggered approach, where all followers are designed as cooperative agents for the entire simulation. The network used in the benchmark simulation is depicted by the left topology in Fig. 1, and Figs. 2 and 3 display the results.

Fig. 2 displays a planar view of the MAS trajectories for the ETC method with cooperative agents. Fig. 3 presents the norm of the tracking errors of the followers and leader as quantified by (2) and (50), respectively. In Fig. 3, the followers connected to the leader experience smaller tracking errors than the followers that are not connected to the leader. The maximum steady-state tracking errors of Followers 1–5 are 0.41, 0.46, 0.68, 0.59, and 0.72 m, respectively. Moreover, the maximum steady-state velocities of Followers 1–5 are 7.38, 7.34, 7.27, 7.39, and 7.22 m/s, respectively. The time instances a follower sent information to a neighbor were measured throughout the simulation. The minimum time difference between consecutive communication

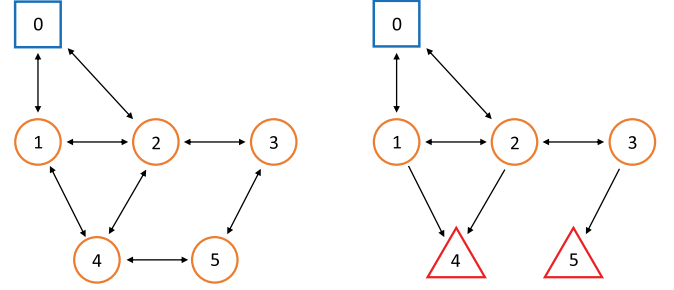


Fig. 1. Illustration of the network topologies used in simulations. The network on the left consists only of cooperative followers while the network on the right consists of both cooperative and Byzantine followers. The blue square denotes the leader agent, the orange circles denote the cooperative followers, and the red triangles denote the Byzantine followers.

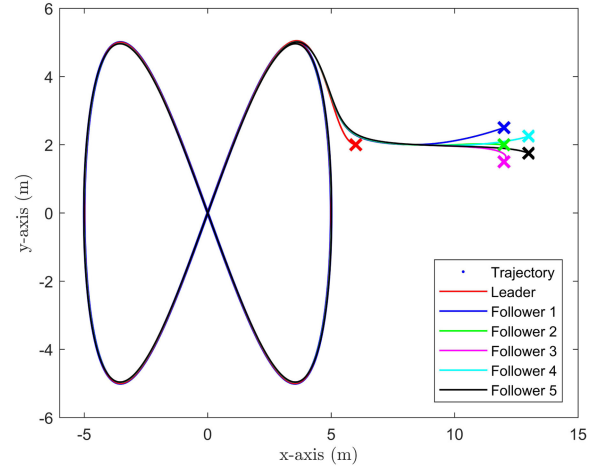


Fig. 2. Planar trajectories of the MAS using an event-triggered approach. The x's denote the starting position of each agent.

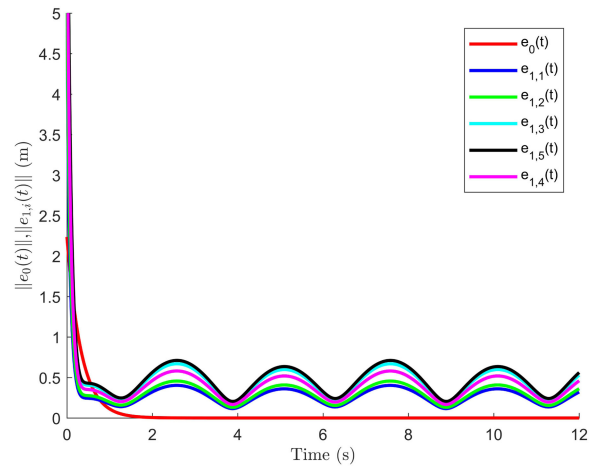


Fig. 3. Norm of tracking errors for the MAS using the ETC approach. All followers are cooperative agents.

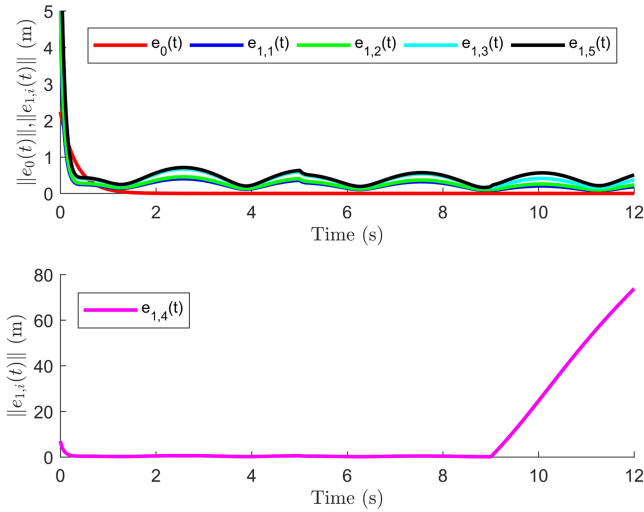


Fig. 4. Norm of tracking errors for the MAS using the ETC approach. Follower 4 is a Type II Byzantine agent, and Follower 5 is a Type I Byzantine agent.

instances for all followers was 6.00×10^{-5} s, which implies that all followers must be equipped with radios capable of broadcasting at approximately 16.67 kHz.

B. ETC Simulation With Byzantine Adversaries

The next simulation is similar to the benchmark, except two originally cooperative followers are converted into Byzantine agents. Specifically, Follower 4 is converted into a Type II Byzantine agent at time $t = 9$ s, and Follower 5 is converted into a Type I Byzantine agent at time $t = 5$ s. For $t \geq 9$ s, Follower 4 executes the controller $u_4(t) = [50, -50]^T$, and for $t \geq 5$ s, Follower 5 communicates state information about itself to its neighbors according to $x_{5,i}(t) = e^{100I_2(t-5)}x_5(t)$, where $x_{5,i}(t)$ denotes the position of Follower 5 within the global coordinate frame at time $t \geq 5$ that is communicated to neighbor i . The objective of the remaining cooperative followers is to identify any potential Byzantine neighbors, remove the Byzantine agent's influence from their controllers, and synchronize their trajectories to the leader's trajectory. Successful execution of this protocol will transform the communication topology from the left network to the right network in Fig. 1.

Fig. 4 demonstrates that the cooperative followers satisfied the objective, where Followers 1 and 2 each detected the Type II Byzantine agent at $t = 9.04$ s and Follower 3 detected the Type I Byzantine agent at $t = 5.02$ s. As seen in Fig. 4, the followers connected to the leader experience smaller tracking errors than followers that are not connected to the leader. The maximum steady-state tracking errors of Followers 1–3 are 0.41, 0.46, and 0.68 m, respectively. Moreover, the maximum steady-state velocities of Followers 1–3 are 7.58, 7.75, 11.57 m/s, respectively. The minimum time difference between consecutive communication instances for all followers was 6.00×10^{-5} s. Followers 4 and 5 are Byzantine adversaries, where their behaviors cannot be guaranteed. Therefore, their steady-state tracking error and maximum steady-state velocity are not included. When compared to

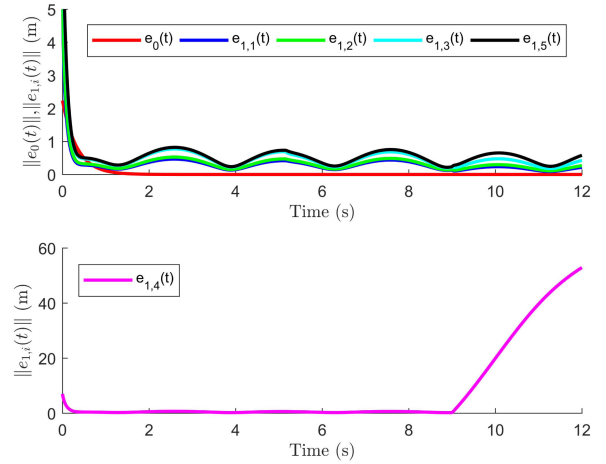


Fig. 5. Norm of tracking errors for the MAS using the STC approach. Follower 4 is a Type II Byzantine agent, and Follower 5 is a Type I Byzantine agent.

the benchmark simulation results, the cooperative followers in this simulation obtain the same steady-state tracking errors and similar maximum steady-state velocities. Therefore, the ETC method obtained similar performance to the benchmark result.

C. STC Simulation With Byzantine Adversaries

This simulation is identical to the one in Section VIII-B, except the STC method from Section VII is used instead. Fig. 5 demonstrates that the cooperative followers satisfied the objective, where Followers 1 and 2 detected the Type II Byzantine agent at $t = 9.001$ s and Follower 3 detected the Type I Byzantine agent at $t = 5.001$ s. As depicted in Fig. 5, the followers connected to the leader experience smaller tracking errors than followers that are not connected to the leader. The maximum steady-state tracking errors of Followers 1–3 are 0.40, 0.46, and 0.67 m, respectively. Moreover, the maximum steady-state velocities of Followers 1–3 are 7.25, 7.31, 7.70 m/s, respectively. The minimum time difference between consecutive communication instances for all followers was 2.00×10^{-5} s.

When compared to the ETC results with Byzantine agents, the cooperative followers in the STC simulation obtain similar steady-state tracking errors and maximum steady-state velocities. While the same θ parameter was used between simulations, further investigation of the effect θ has on communication is needed for the ETC and STC approaches. The results of such a study are provided in the following subsection.

D. Communication Frequency Versus Performance Study

In this section, six simulations are performed for the ETC and STC strategies under the same parameters as the previous simulations, except θ is varied to investigate the tradeoff between communication frequency and steady-state consensus errors. Furthermore, an additional reference simulation is performed to enable comparison between the ETC and STC results. The reference simulation is performed under the same parameters as

TABLE I
ETC COMMUNICATION-PERFORMANCE SUMMARY

Simulation	1	2	3	4	5	6
$\theta (m^2 \cdot s^{-1})$	1.00×10^3	1.00×10^4	1.00×10^5	1.00×10^6	1.00×10^7	1.00×10^8
$\beta_{1,m} (m)$	0.68	0.69	0.78	0.81	0.86	1.15
Min. Comm. Time (s)	6.00×10^{-5}	1.00×10^{-4}	2.60×10^{-4}	2.90×10^{-4}	8.30×10^{-4}	6.70×10^{-3}
Comm. Max. Energy (J)	1.45×10^3	467.04	148.96	54.17	53.63	28.89
Follower 1: Comm. Fraction	0.17	0.06	0.02	0.01	0.007	0.004
Follower 2: Comm. Fraction	0.27	0.09	0.03	0.01	0.009	0.003
Follower 3: Comm. Fraction	0.19	0.06	0.02	0.01	0.007	0.004
Follower 4: Comm. Fraction	0.25	0.08	0.03	0.01	0.009	0.005
Follower 5: Comm. Fraction	0.17	0.06	0.02	0.01	0.006	0.002*

TABLE II
STC COMMUNICATION-PERFORMANCE SUMMARY

Simulation	1	2	3	4	5	6
$\theta (m^2 \cdot s^{-1})$	1.00×10^3	1.00×10^4	1.00×10^5	1.00×10^6	1.00×10^7	1.00×10^8
$\beta_{1,m} (m)$	0.67	0.67	0.67	0.67	0.69	0.74
Min. Comm. Time (s)	2.00×10^{-5}	6.00×10^{-5}	2.10×10^{-4}	6.60×10^{-4}	2.10×10^{-3}	3.20×10^{-3}
Comm. Max. Energy (J)	7.49×10^4	2.37×10^4	7.50×10^3	2.37×10^3	749.90	237.22
Follower 1: Comm. Fraction	8.28	2.62	0.83	0.26	0.08	0.03
Follower 2: Comm. Fraction	13.02	4.12	1.30	0.41	0.13	0.04
Follower 3: Comm. Fraction	9.47	2.99	0.95	0.30	0.09	0.03
Follower 4: Comm. Fraction	11.83	3.74	1.18	0.37	0.12	0.04
Follower 5: Comm. Fraction	8.28	2.62	0.83	0.26	0.08	0.03

the previous simulations except all agents communicate at the same fixed rate of 10 kHz. A communication rate of 10 kHz for a 12-s simulation results in 1.20×10^5 reference event-times, i.e., reference communication instances. Tables I and II summarize the results of the ETC and STC simulations, respectively, where $\beta_{1,m}$ denotes the maximum steady-state tracking error between Followers 1–3, Min. Comm. Time denotes the minimum time difference between consecutive communication events over all followers, Comm. Fraction represents the amount of communication performed by an agent as determined by

$$\text{Comm. Fraction} \triangleq \frac{\text{Number of Event Times}}{\text{Number of Ref. Event Times}}, \quad (52)$$

and Comm. Max. Energy denotes the maximum amount of energy used by all followers to monitor and transmit data. According to [49], the energy $J_i : [0, \infty) \rightarrow \mathbb{R}_{\geq 0}$ consumed by follower i at time t to monitor and transmit data under an ETC approach is given by $J_i(t) \triangleq \sum_{k \in \Gamma_i(t)} (c_1 + p_2 c_2) + c_3 t$, where $\Gamma_i(t) \triangleq \{k : t_k^i \leq t\}$, c_1 describes the cost associated to the packet overhead transmission, c_2 describes the cost per transmitted scalar, c_3 describes the cost of continuous monitoring, and p_2 denotes the number of transmitted scalars. Since

the STC method does not require continuous monitoring of the trigger condition, where monitoring is done at the same time as transmission, the monitoring cost is negligible when compared to the ETC approach. Therefore, the energy consumption function for follower i using STC is $J_i(t) \triangleq \sum_{k \in \Gamma_i(t)} (c_1 + p_2 c_2)$. The parameters c_1 through c_3 denote fixed energy costs consumed per update. Since each follower transmits its state at the current event-time under the ETC approach, $p_2 = 2$ given $x_i(t) \in \mathbb{R}^2$. For STC, each follower transmits its state and future event-time at the current event-time. Hence, $p_2 = 3$ since $x_i(t) \in \mathbb{R}^2$ and $t_k^i \in \mathbb{R}_{\geq 0}$. The parameters used in the energy consumption functions are $c_1 = 38.4$ mJ, $c_2 = 3.2$ mJ, and $c_3 = 60$ mW, which are approximations obtained from power consumption values for a MicaZ using a ZigBee for wireless communication [49].

The * next to the Comm. Fraction of Follower 5 in Table I indicates that Follower 5 was not able to detect Follower 4 as a Byzantine neighbor. As seen in Tables I and II, frequent communication results in high-energy costs. Moreover, frequent communication leads to better tracking performance, where the cooperative followers can track the leader more closely.

Tables I and II also indicate that the STC approach yields better performance than the ETC strategy relative to the same value of θ because of the more frequent communication by STC than ETC. However, ETC can yield comparable performance to STC while using less energy to communicate.

IX. CONCLUSION

In this work, the approximate leader-follower consensus problem in the presence of Byzantine adversaries for a homogeneous MAS is examined. Distributed event- and self-triggered controllers are developed along with a Lyapunov-based detection method that enables followers to discern between cooperative and Byzantine neighbors. The controllers can remove the influence from Byzantine agents by altering the interaction topology and enabling consensus for all cooperative followers. Moreover, a time-based estimate for each follower's trigger condition is developed, which allows each follower to estimate the future time when state information from its neighbors will be required. The STC approach alleviates the continuous monitoring requirement of ETC and enables intermittent communication and monitoring. Future efforts could focus on generalizing the result to more abstract network topologies, developing more capable and sensitive Byzantine detection and trigger condition estimation methods, and relaxing Assumption 5 by developing a controller capable of ensuring network connectivity maintenance in the presence of Byzantine adversaries. Moreover, uncertain agent dynamics can be considered, the impact of which leads to faster divergence rates between the estimated and true follower position, and more frequent triggering and communication. Simulation results demonstrate that both ETC and STC methods enable approximate leader-follower consensus while identifying and mitigating against Byzantine adversaries. Results also show that increased communication leads to better tracking of the leader for both ETC and STC. Moreover, both methods can provide identical tracking performance, but depending on the choice of parameters, one method can provide communication energy savings over the other.

APPENDIX

A. Proof of Lemma 1

Proof: Assume the hypothesis of Lemma 1, and fix $t_1 \geq 0$. By Assumptions 5 and 6, $H(t_1)$ is a diagonally dominant matrix, where each row of $H(t_1)$ is nonzero. By Assumptions 4 and 5, $H(t_1)$ contains a strictly diagonally dominant row, i.e., $|H_{ii}(t_1)| > \sum_{j \in \mathcal{V}, j \neq i} |H_{ij}(t_1)|$ for some $i \in \mathcal{V}$. Claim: If $H(t_1)$ is a diagonally dominant matrix with a strictly diagonally dominant row, then $H_{\text{sym}}(t_1)$ is a symmetric, diagonally dominant matrix with a strictly diagonally dominant row. Suppose the claim is true. Therefore, $H_{\text{sym}}(t_1)$ is a symmetric, diagonally dominant matrix with a strictly diagonally dominant row. Since $I_N + |H_{\text{sym}}(t_1)|$ is a symmetric, strictly diagonally dominant matrix, $I_N + |H_{\text{sym}}(t_1)|$ is positive definite by the Gershgorin disk theorem in [9, Th. 3.9.]. Let $\{\lambda_i\}_{i=1}^N \subset \mathbb{R}_{>0}$ denote the eigenvalues of $I_N + |H_{\text{sym}}(t_1)|$. Since $\lambda_i \in \mathbb{R}_{>0}$ for all $i \in \mathcal{V}$, and the eigenvalues of $(I_N + |H_{\text{sym}}(t_1)|)^{N-1}$

are $\{\lambda_i^{N-1}\}_{i=1}^N$, $\lambda_i^{N-1} \in \mathbb{R}_{>0}$ for each $i \in \mathcal{V}$. Since $(I_N + |H_{\text{sym}}(t_1)|)^{N-1}$ is a symmetric matrix with positive eigenvalues, $(I_N + |H_{\text{sym}}(t_1)|)^{N-1}$ is positive definite. Hence, $H_{\text{sym}}(t_1)$ is irreducible by [50, Theorem 6.2.24.]. Since $H_{\text{sym}}(t_1)$ is an irreducible, diagonally dominant matrix with a strictly diagonally dominant row, $H_{\text{sym}}(t_1)$ is irreducibly diagonally dominant by [50, Definition 6.2.25.]. Since $H_{\text{sym}}(t_1)$ is irreducibly diagonally dominant, $H_{\text{sym}}(t_1)$ is nonsingular [50, Corollary 6.2.27.], i.e., $\lambda_{\min}(H_{\text{sym}}(t_1)) > 0$. Since t_1 is arbitrary, $\lambda_{\min}(H_{\text{sym}}(t)) > 0$ for all $t \geq 0$. Since $\{\lambda_{\min}(H_{\text{sym}}(t)) : \forall t \geq 0\}$ is a finite set, $\lambda_{\min}(H_{\text{min}}) \triangleq \min\{\lambda_{\min}(H_{\text{sym}}(t))\} \in \mathbb{R}_{>0}$ is well defined.

Proof of Claim: Let $H(t_1)$ be a diagonally dominant matrix, i.e., row diagonally dominant, where $|H_{ii}(t_1)| \geq \sum_{j \neq i} |H_{ij}(t_1)|$ for all $i \in \mathcal{V}$. Then, $H(t_1)^T$ is a column diagonally dominant matrix, where $|H_{ii}(t_1)| \geq \sum_{j \neq i} |H_{ji}(t_1)|$ for all $i \in \mathcal{V}$. Recall that $H_{\text{sym}}(t_1) = \frac{1}{2}(H(t_1) + H(t_1)^T)$. Then, for fixed $i \in \mathcal{V}$, it follows that:

$$\begin{aligned} \sum_{j \neq i} |H_{\text{sym},ij}| &= \sum_{j \neq i} \left| \frac{1}{2} (H_{ij}(t_1) + H_{ji}(t_1)) \right| \\ &= \frac{1}{2} \sum_{j \neq i} |H_{ij}(t_1) + H_{ji}(t_1)| \\ &\leq \frac{1}{2} \sum_{j \neq i} (|H_{ij}(t_1)| + |H_{ji}(t_1)|) \\ &\leq |H_{ii}(t_1)| \\ &= |H_{\text{sym},ii}(t_1)|. \end{aligned}$$

Since $\sum_{j \neq i} |H_{\text{sym},ij}| \leq |H_{\text{sym},ii}(t)|$ for each $i \in \mathcal{V}$, $H_{\text{sym}}(t_1)$ is diagonally dominant. The existence of a strictly diagonally dominant row/column follows by a similar argument. ■

B. Proof of Lemma 2

Proof: Let $t \geq 0$. Given (12) and $K = B^T P$, it follows that

$$\|u_i(t)\| \leq S_{\max}(B^T P) \|z_i(t)\| + S_{\max}(B^T P) \|e_{2,i}(t)\|. \quad (53)$$

Provided the self-trigger in (44) is satisfied for all $t \geq 0$, $\phi_2 \|e_{2,i}(t)\|^2 - \frac{\theta}{N} \leq 0$ for all $t \geq 0$ by Theorem 3, where

$$\|e_{2,i}(t)\| \leq \sqrt{\frac{\theta}{N\phi_2}}. \quad (54)$$

By (22), it follows that:

$$\|z(t)\| \leq \max\{\|H(t) \otimes I_m\|\} (\|e_2(t)\| + \|e_1(t)\|). \quad (55)$$

Since $\|e_1(t)\| \leq \beta_1 + \beta_2 e^{-\beta_3 t}$ for all $t \geq 0$ by Theorem 1

$$\|e_1(t)\| \leq \beta_1 + \beta_2. \quad (56)$$

Substituting (54) into $\|e_2(t)\|^2 = \sum_{i \in \mathcal{V}} \|e_{2,i}(t)\|^2$ implies

$$\|e_2(t)\| \leq \sqrt{\frac{\theta}{\phi_2}}. \quad (57)$$

Since $\|z_i(t)\| \leq \|z(t)\|$, (55) implies

$$\|z_i(t)\| \leq \max\{\|H(t) \otimes I_m\|\} (\|e_2(t)\| + \|e_1(t)\|). \quad (58)$$

Substituting (54), (56), (57), and (58) into (53) yields (43). ■

C. Proof of Lemma 3

Proof: Consider the candidate Lyapunov function $V_0 : \mathbb{R}^m \rightarrow \mathbb{R}_{\geq 0}$ defined as

$$V_0(e_0(t)) \triangleq \frac{1}{2} e_0^T(t) e_0(t). \quad (59)$$

Substituting (1) and (51) into the time-derivative of (50) yields

$$\dot{e}_0(t) = -k_0 e_0(t). \quad (60)$$

Substituting (60) into the time-derivative of (59) yields

$$\dot{V}_0(e_0(t)) = -k_0 e_0^T(t) e_0(t), \quad (61)$$

where substituting (59) into (61) yields

$$\dot{V}_0(e_0(t)) = -2k_0 V_0(e_0(t)). \quad (62)$$

Solving (62) yields $V_0(e_0(t)) = V_0(e_0(0))e^{-2k_0 t}$. Since $V_0(e_0(t))$ is radially unbounded with an unrestricted domain, equation (50) is globally exponentially regulated. Since $V_0(e_0(t))$ is positive definite and $\dot{V}_0(e_0(t))$ is negative definite, provided $k_0 > 0$, $V_0(e_0(t)) \in \mathcal{L}_\infty$. Since $V_0(e_0(t)) \in \mathcal{L}_\infty$, $e_0(t) \in \mathcal{L}_\infty$. Since $x_d(t) \in \mathcal{L}_\infty$, $x_0(t) \in \mathcal{L}_\infty$. Since $x_d(t) \in \mathcal{L}_\infty$, $\dot{x}_d(t) \in \mathcal{L}_\infty$, and $x_0(t) \in \mathcal{L}_\infty$, we see that $u_0(t) \in \mathcal{L}_\infty$. ■

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this article are those of the author(s) and do not necessarily reflect the views of sponsoring agencies.

REFERENCES

- [1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
- [2] C. Mohan, "Blockchains and databases: A new era in distributed computing," *Proc. Int. Conf. Data Eng.*, 2018, pp. 1739–1740.
- [3] N. Abaid and M. Porfiri, "Leader-follower consensus over numerosity-constrained random networks," *Automatica*, vol. 48, no. 8, pp. 1845–1851, 2012.
- [4] L. Ding, Q.-L. Han, and G. Guo, "Network-based leader-following consensus for distributed multi-agent systems," *Automatica*, vol. 49, no. 7, pp. 2281–2286, 2013.
- [5] X.-H. Wang and H.-B. Ji, "Leader-follower consensus for a class of nonlinear multi-agent systems," *Int. J. Control, Automat. Syst.*, vol. 10, pp. 27–35, 2012.
- [6] J. Hu and G. Feng, "Distributed tracking control of leader-follower multi-agent systems under noisy measurement," *Automatica*, vol. 46, no. 8, pp. 1382–1387, 2010.
- [7] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
- [8] Z. Kan, J. Shea, and W. E. Dixon, "Leader-follower containment control over directed random graphs," *Automatica*, vol. 66, pp. 56–62, 2016.
- [9] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
- [10] A. Adaldo *et al.*, "Event-triggered pinning control of switching networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 204–213, Jun. 2015.
- [11] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.
- [12] T. H. Cheng, Z. Kan, J. R. Klotz, J. M. Shea, and W. E. Dixon, "Event-triggered control of multi-agent systems for fixed and time-varying network topologies," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5365–5371, Oct. 2017.
- [13] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson, "Control of multi-agent systems with event-triggered cloud access," in *Proc. Eur. Control Conf.*, 2015, pp. 954–961.
- [14] X. Dong and G. Hu, "Time-varying formation control for general linear multi-agent systems with switching directed topologies," *Automatica*, vol. 73, pp. 47–55, 2016.
- [15] G. Wen, W. Yu, G. Hu, J. Cao, and X. Yu, "Pinning synchronization of directed networks with switching topologies: A multiple lyapunov functions approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3239–3250, Dec. 2015.
- [16] C. Nowzari and J. Cortés, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.
- [17] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.
- [18] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson, "Cloud-supported formation control of second-order multi-agent systems," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1563–1574, Dec. 2018.
- [19] W. Heemels, K. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE Conf. Decis. Control*, Dec. 2012, pp. 3270–3285.
- [20] X. Yi, K. Liu, D. V. Dimarogonas, and K. H. Johansson, "Dynamic event-triggered and self-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3300–3307, Aug. 2019.
- [21] Y. Fan, L. Liu, G. Feng, and Y. Wang, "Self-triggered consensus for multi-agent systems with zeno-free triggers," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2779–2784, Oct. 2015.
- [22] Z. Feng and G. Hu, "Secure cooperative event-triggered control of linear multiagent systems under DoS attacks," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 741–752, May 2020.
- [23] N. H. Vaidya, L. Tseng, and G. Liang, "Iterative approximate Byzantine consensus in arbitrary directed graphs," *Proc. ACM Symp. Prin. Distrib. Comput.*, 2012, pp. 365–374.
- [24] J. Hromkovič, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger, *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*. Berlin, Germany: Springer, 2005.
- [25] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.
- [26] S. M. Dibaji and H. Ishii, "Resilient consensus of second-order agent networks: Asynchronous update rules with delays," *Automatica*, vol. 81, pp. 123–132, 2017.
- [27] D. Senejohnny, S. Sundaram, C. De Persis, and P. Tesi, "Resilience against misbehaving nodes in self-triggered coordination networks," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 2848–2853.
- [28] J. Usevitch and D. Panagou, "Resilient leader-follower consensus to arbitrary reference values," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 1292–1298.
- [29] Y. Hou, J. Li, and Y. Pan, "On the Laplacian eigenvalues of signed graphs," *Linear Multilinear Algebra*, vol. 51, no. 1, pp. 21–30, 2003.
- [30] T. Zaslavsky, "Signed graphs," *Discrete Appl. Math.*, vol. 4, no. 1, pp. 47–74, 1982.
- [31] C. Altafini, "Consensus problems on networks with antagonistic interactions," *IEEE Trans. Autom. Control*, vol. 58, no. 4, pp. 935–946, Apr. 2013.
- [32] M. E. Valcher and P. Misra, "On the consensus and bipartite consensus in high-order multi-agent dynamical systems with antagonistic interactions," *Syst. Control Lett.*, vol. 66, pp. 94–103, 2014.
- [33] J. Qin, W. Fu, W. X. Zheng, and H. Gao, "On the bipartite consensus for generic linear multiagent systems with input saturation," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 1948–1958, Aug. 2017.
- [34] F. M. Zegers, P. Deptula, J. M. Shea, and W. E. Dixon, "Event-triggered approximate leader-follower consensus with resilience to Byzantine adversaries," in *Proc. IEEE Conf. Decis. Control*, Nice, France, Dec. 2019, pp. 6412–6417.
- [35] Z. Kan, A. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1827–1832, Jul. 2012.
- [36] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.
- [37] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

- [38] A. Barboni, H. Rezaee, F. Boem, and T. Parisini, "Detection of covert cyber-attacks in interconnected systems: A distributed model-based approach," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3728–3741, Sep. 2020.
- [39] A. Rahnema and P. J. Antsaklis, "Learning-based event-triggered control for synchronization of passive multi-agent systems under attack," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4170–4185, Oct. 2020.
- [40] H. Liu, M. Xu, Y. Wu, N. Zheng, Y. Chen, and M. Z. A. Bhuiyan, "Resilient bipartite consensus for multi-agent networks with antagonistic interactions," in *Proc. IEEE 17th Int. Conf. Trust Secur. Privacy Comp. Commun./12th IEEE Int. Conf. Big Data Sci. Eng.*, 2018, pp. 1262–1269.
- [41] E. Garoudja, F. Harrou, Y. Sun, K. Kara, A. Chouder, and S. Silvestre, "Statistical fault detection in photovoltaic systems," *Sol. Energy*, vol. 150, pp. 485–499, 2017.
- [42] L. Li, H. Luo, S. X. Ding, Y. Yang, and K. Peng, "Performance-based fault detection and fault-tolerant control for automatic control systems," *Automatica*, vol. 99, pp. 308–316, 2019.
- [43] J. R. Klotz, A. Parikh, T.-H. Cheng, and W. E. Dixon, "Decentralized synchronization of uncertain nonlinear systems with a reputation algorithm," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 434–445, Mar. 2018.
- [44] E. Garcia, Y. Cao, and D. W. Casbeer, "Decentralized event-triggered consensus with general linear dynamics," *Automatica*, vol. 50, no. 10, pp. 2633–2640, 2014.
- [45] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. CS-34, no. 1, pp. 73–82, Jan. 1987.
- [46] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, Sep. 1994.
- [47] V. I. Bogachev, *Measure Theory*. Berlin, Germany: Springer, 2007, vol. 1.
- [48] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [49] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Proc. IEEE Conf. Decis. Control*, 2008, pp. 435–440.
- [50] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.



Federico M. Zegers received the B.S. degree in mechanical engineering, the B.S. degree in mathematics, and the M.S. degree in mechanical engineering from the University of Florida, Gainesville, FL, USA, in 2016, 2016, and 2019, respectively, where he is currently working toward the Ph.D. degree under the supervision of Dr. Warren Dixon.

His research interests include Lyapunov-based nonlinear and adaptive control, multi-agent systems, switched/hybrid systems theory, and robotics.



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

His current research interests include, but are not limited to, learning-based and adaptive methods, estimation and control, sensor fusion,

multi-agent systems, human-machine interaction, vision-based navigation and control, biomedical systems, biomechanics, and robotics applied to a variety of fields.



John M. Shea received the B.S. degree in computer engineering, and the M.S. and Ph.D. degrees in electrical engineering from Clemson University, Clemson, SC, USA, in 1993, 1995, and 1998, respectively.

In 1999, he joined the Faculty of the Department of Electrical and Computer Engineering with the University of Florida, Gainesville, FL, USA, where he was promoted to Professor, in 2015. His research interests include wireless and military communications and networked au-

tonomous systems.

Dr. Shea was the recipient of the Technical Achievement Award for contributions to military communications from the IEEE Military Communications Conference (MILCOM), in 2012, the Ellersick Award from the IEEE Communications Society for the Best Paper in the Unclassified Program of MILCOM, in 2013 and 1996. He was the recipient of the Outstanding Young Alumni Award from the Clemson University College of Engineering and Science, in 2011. He was selected as a Finalist for the 2004 Eta Kappa Nu Outstanding Young Electrical Engineer. He has been an Editor for *IEEE Wireless Communications Magazine*, since 2010. He was an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, from 2008 to 2012 and was an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, from 2002 to 2007. He was the Technical Program Chair for the Unclassified Program of MILCOM, in 2010.








Warren E. Dixon (IEEE, Fellow) received the Ph.D. degree from Clemson University, Clemson, SC, USA, in 2000.

He worked as a Research Staff Member and Eugene P. Wigner Fellow with Oak Ridge National Laboratory (ORNL), Oak Ridge, TN, USA, until 2004, when he joined the Mechanical and Aerospace Engineering Department, with the University of Florida, Gainesville, FL, USA, where he is an Ebaugh Professor and Department Chair. His research interests include the

development and application of Lyapunov-based control techniques for uncertain nonlinear systems.

Dr. Dixon is an ASME and IEEE Fellow. He was the recipient of the 2015 and 2009 American Automatic Control Council (AACC) O. Hugo Schuck (Best Paper) Award, the 2013 Fred Ellersick Award for Best Overall MILCOM Paper, the 2011 American Society of Mechanical Engineers (ASME) Dynamics Systems and Control Division Outstanding Young Investigator Award, and the 2006 IEEE Robotics and Automation Society (RAS) Early Academic Career Award. He was the recipient of the Air Force Commander's Public Service Award (2016) for his contributions to the U.S. Air Force Science Advisory Board. He was/is an Associate Editor for *ASME Journal of Dynamic Systems, Measurement and Control*, *Automatica*, *IEEE CONTROL SYSTEMS*, *IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS: PART B CYBERNETICS*, and the *International Journal of Robust and Nonlinear Control*.

Simultaneous Estimation of Euclidean Distances to a Stationary Object's Features and the Euclidean Trajectory of a Monocular Camera

Zachary I. Bell , *Member, IEEE*, Patryk Deptula , *Member, IEEE*, Emily A. Doucette ,
J. Willard Curtis , *Member, IEEE*, and Warren E. Dixon , *Fellow, IEEE*

Abstract—Data-based, exponentially converging observers are developed for a monocular camera to estimate the Euclidean distance (and hence accurately scaled coordinates) to features on a stationary object and to estimate the Euclidean trajectory taken by the camera while tracking the object, without requiring the typical positive depth constraint. A Lyapunov-based stability analysis shows that the developed observers are exponentially converging without requiring persistence of excitation through the use of a data-based learning method. An experimental study is presented, which compares the developed Euclidean distance observer to previous observers demonstrating the effectiveness of this result.

Index Terms—Computer vision, nonlinear observers, simultaneous localization and mapping (SLAM), structure from motion (SfM), vision-based localization.

I. INTRODUCTION

In many applications, the state (e.g., position and orientation) of an autonomous agent and its local environment (e.g., relative positions of objects in the surrounding environment) must be determined from sensor data. This problem is well known as simultaneous localization and mapping (SLAM) (cf., [1]–[5]). Often, a global positioning system (GPS) is used to estimate the position; however, in many environments, GPS is unavailable (e.g., when agents operate in GPS denied or contested environments) motivating the use of only local sensing data (e.g., camera images, inertial measurement units, and wheel encoders) to estimate the position and model the surrounding environment.

Using cameras to reconstruct the surrounding environment (i.e., determine the Euclidean scale of objects in the environment) requires the assumption that object features are in the camera field of view (FOV) and may be extracted and tracked through a sequence of images. However, a significant challenge arises in determining the scale of objects in an image using a camera given the loss of depth information. Specifically, images of objects are 2-D projections of the 3-D environment. Approaches to reconstruct (i.e., estimate the structure) objects use multiple images of an object along with scale information (cf., [6],

[7]) or motion (cf., [8]–[23]), such as linear and angular velocities of the camera. The latter of these methods is referred to as structure from motion (SfM). Generally, the Euclidean scales of objects are not known; however, multiple calibrated cameras may be used to recover the scale (cf., [6], [7]). However, this approach does not work in all scenarios because some objects may have limited or no parallax between the camera images. In SfM approaches, the potential for limited parallax still exists; however, a camera may travel to generate enough parallax, which is generally not possible in stereo vision.

The SfM problem may be approached using online iterative methods (cf., [8]–[24]) and offline batch methods (cf., [6], [7], [25], and the references contained within). These offline approaches perform an optimization over an image sequence, but only show convergence for limited cases (cf., [26], [27]). Most online SfM approaches assume continuous measurements of objects by the camera or only update when a new image is received (cf., [8]–[23]); however, recent results have discussed approaches to handle objects temporarily leaving the camera FOV (cf., [28]–[30]). Many results apply the extended Kalman filter (EKF) to estimate depth (cf., [8], [10]–[12]); however, the EKF generally does not guarantee convergence and may fail in some applications [31], [32]. Compared to the EKF approach, techniques, such as [13], [15], [16], [18], [20], and [22], show asymptotic convergence of the structure estimation errors. Furthermore, results, such as [9], [14], [17], [19], and [21], show exponential convergence of the scale estimate assuming some form of a persistence of excitation (PE) condition or the more strict extended output Jacobian (EOJ) is satisfied. Specifically, the authors in [17] show exponential convergence assuming the PE condition is met and either the initial estimation error is small or the velocities are limited. Furthermore, the development in [19] yields exponential convergence assuming the observer satisfies the EOJ condition. In [21], an exponentially stable observer is developed that requires the motion along at least one axis to be nonzero, and the observer remains ultimately bounded if the PE assumption does not hold, whereas in [19], the observer becomes singular. Typically, SfM approaches require the motion (e.g., linear and angular velocities) to be known; however, the design in [22], extending an approach similar to Dani *et al.* [21], demonstrates a partial solution to the more challenging problem (i.e., compared to SfM) of structure and motion where not only are the feature Euclidean coordinates estimated, but also two of the linear velocities and the three angular velocities of the camera are estimated assuming PE and the linear velocity, and acceleration is measurable along one axis.

In this article, and our preliminary work in [23], exponentially converging observers are developed that use a camera to estimate the Euclidean distance to features on a stationary object in the camera FOV while also estimating the Euclidean trajectory of the camera tracking the object. Unlike previous methods, such as [9], [14], [17], [19], and [21], that assume a PE condition, the developed estimator only requires finite excitation. The finite excitation condition results from the use of concurrent learning (CL) (cf., [33]–[36]). The concept of

Manuscript received July 27, 2020; accepted September 30, 2020. Date of publication November 3, 2020; date of current version August 30, 2021. This work was supported in part by the National Science Foundation under Award 1509516, in part by a Task Order Contract With the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, and in part by AFOSR under Award FA9550-18-1-0109. Recommended by Associate Editor S. K. Spurgeon. (*Corresponding author: Zachary I. Bell.*)

Zachary I. Bell, Patryk Deptula, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: bellz121@ufl.edu; pdeptula@ufl.edu; wdixon@ufl.edu).

Emily A. Doucette and J. Willard Curtis are with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542 USA (e-mail: emily.doucette@eglin.af.mil; jwillardcurtis@gmail.com).

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

Digital Object Identifier 10.1109/TAC.2020.3035597

CL is to use recorded input and output data from system trajectories to identify uncertain constant parameters of the system in real time under the assumption that the system is sufficiently excited for a finite amount of time. This approach relaxes the PE assumption and can be monitored and verified online. The results in [24] demonstrate exponential convergence of depth estimates using CL; however, these results rely on the assumption that the features are on a plane. CL could be applied to achieve the result in this article; however, motivated by the performance improvement as discussed in [37] (especially for noise-prone image feedback), this article employs integral CL (ICL) (cf., [23], [37]–[39]). ICL removes the necessity to estimate the highest order derivative of the system required in traditional CL (cf., [23], [37]–[39]).

Although ICL removes the need for measuring the state derivative, it still requires the state to be measurable; yet, a unique challenge in this article is that the state depends on the unmeasurable distance to the target. Moreover, the traditional state used in results, such as [8]–[22], [24], includes an inherent singularity when one of the coordinates becomes zero (i.e., the so-called depth to the target). Specifically, previous results assume a positive depth constraint where the distance from the focal point of the camera to the target along the axis perpendicular to the image plane remains positive. The positive depth constraint is satisfied if the features remain in the camera FOV; however, the constraint can be violated for some camera rotations that cause the feature to leave the FOV. Since new results are being developed that allow features to intermittently leave the FOV (cf., [29], [30], [40], [41]), a new formulation of the error system is motivated.

In this article, we exploit alternative image geometry insights to express the error system with a more general distance measure that only becomes zero when the target and camera are coincident; thereby, avoiding the positive depth constraint. While this result also requires the features to remain in the FOV (which ensures that the positive depth constraint is satisfied), eliminating the positive depth constraint eliminates a barrier for future development that would allow intermittent viewing of the features. Although, the new image geometry based error system avoids the potential depth singularity, the resulting error system still contains the unmeasurable distance to the target. However, the development in Section III illustrates how the unmeasurable state can be related to an unknown constant to enable the use of ICL. Regardless of the system identification method used, there is a delay before sufficient excitation occurs to identify the parameters. Therefore, the preliminary result in [23] and the development in Section III exhibit an arbitrarily long delay before determining the feature Euclidean coordinates. In Section IV, we modify the developed learning strategy to include gradient terms that enable transient learning until sufficient data have been collected for the ICL terms.

To illustrate the performance of the developed observers, multiple experiments are presented, including a comparison of the observers in Sections III and IV with the results in [21] and an EKF. These results indicate that the EKF and result in [21] have improved transient performance over the result in Section III, before the ICL-based estimates converge. The EKF and result in [21] have similar transient response as the observer in Section IV before the ICL-based estimates converge. After the ICL-based estimates converge, the observers in Sections III and IV converge to steady state with improved performance over the EKF and observer in [21].

II. MOTION MODEL FOR STATIONARY FEATURES

The following definitions and assumptions are presented to aid in the development of the subsequent observers.

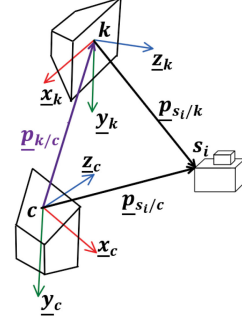


Fig. 1. Example geometry for tracking the position of the i th feature of s . Example shows the camera starts at the top left where the key image is taken and is traveling downward from the upper left to the lower left while tracking a stationary object on the right.

Definition 1: A key frame is defined as the camera frame at which features are first extracted from an image of an object.

The key frame, denoted by \mathcal{F}_k , has its origin at the principal point of that image, denoted by k , and basis $\{\underline{x}_k, \underline{y}_k, \underline{z}_k\}$. The frame at which the current image is taken, denoted by \mathcal{F}_c , has its origin at the principal point of the current image, denoted by c , and basis $\{\underline{x}_c, \underline{y}_c, \underline{z}_c\}$. This implies that \mathcal{F}_k is established to coincide with \mathcal{F}_c at time $t = 0$.

Assumption 1: There exists a stationary object s with features that can be detected and tracked provided they are within the FOV of the camera. Specifically, for all time $t \in \mathbb{R}_{\geq 0}$, a set of at least $p \in \mathbb{Z}_{\geq 4}$ trackable planar features or $p \in \mathbb{Z}_{\geq 5}$ nonplanar features are on s are in the camera's FOV.

Remark 1: It is assumed that the object remains in the FOV in this result; however, the subsequent development may be extended to not require this constraint. Additionally, using feature extraction techniques, such as [42], a set of features can be extracted from an image of a stationary object. These features can then be tracked while they are in the FOV of the camera using techniques, such as [43] and [44].

Assumption 2: The camera intrinsic matrix $A \in \mathbb{R}^{3 \times 3}$ is known and invertible [7].

Assumption 3: The camera linear and angular velocities, $\underline{v}_c(t), \underline{\omega}_c(t) \in \mathbb{R}^3$, are measurable and expressed in \mathcal{F}_c .

As shown in Fig. 1, the position of the i th feature on s , $s_i \in \mathbb{Z}_{>0} \quad \forall i = \{1, \dots, p\}$, can be described as

$$\underline{p}_{s_i/c}(t) = \underline{p}_{k/c}(t) + R_{k/c}(t) \underline{p}_{s_i/k} \quad (1)$$

where $\underline{p}_{k/c}(t) \in \mathbb{R}^3$ is the position of k with respect to c expressed in \mathcal{F}_c , $\underline{p}_{s_i/k}(t) \in \mathbb{R}^3$ is the position of feature s_i with respect to k expressed in \mathcal{F}_k , $R_{k/c}(t) \in \mathbb{R}^{3 \times 3}$ is the rotation matrix describing the orientation of \mathcal{F}_k with respect to \mathcal{F}_c , and $\underline{p}_{s_i/c}(t) \in \mathbb{R}^3$ is the position of feature s_i with respect to c expressed in \mathcal{F}_c . Rearranging (1) gives

$$\begin{bmatrix} \underline{u}_{s_i/c}(t) - \underline{u}_{k/c}(t) \\ d_{s_i/c}(t) \end{bmatrix} \begin{bmatrix} d_{s_i/c}(t) \\ d_{k/c}(t) \end{bmatrix} = R_{k/c}(t) \underline{u}_{s_i/k} d_{s_i/k} \quad (2)$$

where $d_{s_i/c}(t) \in \mathbb{R}_{>0}$ and $\underline{u}_{s_i/c}(t) \in \mathbb{R}^3$ are the distance and unit vector of feature s_i with respect to c expressed in \mathcal{F}_c , respectively; $d_{k/c}(t) \in \mathbb{R}_{>0}$ and $\underline{u}_{k/c}(t) \in \mathbb{R}^3$ are the distance and unit vector of k with respect to c expressed in \mathcal{F}_c , respectively; and $d_{s_i/k} \in \mathbb{R}_{>0}$ and $\underline{u}_{s_i/k} \in \mathbb{R}^3$ are the distance and unit vector of feature s_i with respect to k expressed in \mathcal{F}_k , respectively.

Assumption 4: The origins k and c are not coincident for all time $t > 0$, implying $d_{k/c}(t) > 0$. Additionally, the motion of the camera is

not parallel to the position of a feature, $\|\underline{u}_{k/c}(t) - \underline{u}_{s_i/c}(t)\| > 0$, for all time $t \geq 0$.

Under Assumptions 1–4, the rotation $R_{k/c}(t)$ and unit vector $\underline{u}_{k/c}(t)$ can be determined from a general set of stationary features, using existing techniques, such as planar homography decomposition or essential decomposition.¹ Additionally, $\underline{u}_{s_i/k}$ and $\underline{u}_{s_i/c}(t)$ can always be determined from $\underline{u}_{s_i/k} = \frac{A^{-1}p_{s_i/k}}{\|A^{-1}p_{s_i/k}\|}$ and $\underline{u}_{s_i/c}(t) = \frac{A^{-1}p_{s_i/c}(t)}{\|A^{-1}p_{s_i/c}(t)\|}$, where $p_{s_i/k}, p_{s_i/c}(t) \in \mathbb{R}^3$ are the homogeneous pixel coordinates of feature s_i in \mathcal{F}_k and \mathcal{F}_c , respectively. Let $H_{s_i}(t) \triangleq [\underline{u}_{s_i/c}(t) \quad \underline{u}_{k/c}(t)]$. While $d_{k/c}(t) > 0$, (2) is invertible such that

$$\begin{bmatrix} d_{s_i/c}(t) \\ d_{k/c}(t) \end{bmatrix} = \psi_{s_i}(t) d_{s_i/k} \quad (3)$$

where $\psi_{s_i}(t) \triangleq (H_{s_i}^T(t)H_{s_i}(t))^{-1}H_{s_i}^T(t)R_{k/c}(t)\underline{u}_{s_i/k}$ is invertible and measurable under Assumptions 1–4. Furthermore, given \mathcal{F}_k and s are stationary, the time derivatives of the unknown distances are measurable as

$$\frac{d}{dt}(d_{s_i/c}(t)) = \eta_{s_i,1}(t) \quad (4)$$

$$\frac{d}{dt}(d_{k/c}(t)) = \eta_{s_i,2}(t) \quad (5)$$

and

$$\frac{d}{dt}(d_{s_i/k}) = 0 \quad (6)$$

where $\eta_{s_i,1}(t)$ and $\eta_{s_i,2}(t)$ are the first and second elements of $\eta_{s_i}(t) \triangleq -[\frac{\underline{u}_{s_i/c}(t)}{\underline{u}_{k/c}(t)}]^T \underline{v}_c(t)$, respectively.

III. INTEGRAL CL OBSERVER UPDATE LAWS FOR EUCLIDEAN DISTANCES

Motivated by the developments in [23] and [38], an ICL update law is implemented to estimate the constant unknown distances $d_{s_i/k}$ by integrating (4) and (5) over a time window $\varsigma \in \mathbb{R}_{>0}$ yielding

$$\begin{bmatrix} d_{s_i/c}(t) \\ d_{k/c}(t) \end{bmatrix} - \begin{bmatrix} d_{s_i/c}(t-\varsigma) \\ d_{k/c}(t-\varsigma) \end{bmatrix} = \int_{t-\varsigma}^t \eta_{s_i}(\iota) d\iota, \quad t > \varsigma$$

where ς may be constant in size or change over time. While $\int_{t-\varsigma}^t \eta_{s_i}(\iota) d\iota$ is a known quantity, $[\frac{d_{s_i/c}(t)}{d_{k/c}(t)}]$ and $[\frac{d_{s_i/c}(t-\varsigma)}{d_{k/c}(t-\varsigma)}]$ are unknown; however, the relationship in (3) may be utilized at the current time t and the previous time $t-\varsigma$ yielding

$$\mathcal{Y}_{s_i}(t) d_{s_i/k} = \mathcal{U}_{s_i}(t) \quad (7)$$

where $\mathcal{Y}_{s_i}(t) \triangleq \begin{cases} 0_{2 \times 1}, & t \leq \varsigma, \\ (\psi_{s_i}(t) - \psi_{s_i}(t-\varsigma)), & t > \varsigma, \end{cases}$ and $\mathcal{U}_{s_i}(t) \triangleq \begin{cases} 0_{2 \times 1}, & t \leq \varsigma, \\ \int_{t-\varsigma}^t \eta_{s_i}(\iota) d\iota, & t > \varsigma. \end{cases}$ The dynamics in (7) demonstrate that CL may be used to estimate the constant distances $d_{s_i/k}$ to the features on s . Specifically, multiplying both sides of (7) by $\mathcal{Y}_{s_i}^T(t)$ yields

$$\mathcal{Y}_{s_i}^T(t) \mathcal{Y}_{s_i}(t) d_{s_i/k} = \mathcal{Y}_{s_i}^T(t) \mathcal{U}_{s_i}(t). \quad (8)$$

In general, $\mathcal{Y}_{s_i}(t)$ will not have full column rank (e.g., when the camera is stationary) implying $\mathcal{Y}_{s_i}^T(t) \mathcal{Y}_{s_i}(t) \geq 0$. However, the equality in (8)

may be evaluated at instances in time and summed together (i.e., history stacks) as

$$\Sigma_{\mathcal{Y}_{s_i}} d_{s_i/k} = \Sigma \mathcal{U}_{s_i} \quad (9)$$

where $\Sigma_{\mathcal{Y}_{s_i}} \triangleq \sum_{h_i=1}^N \mathcal{Y}_{s_i}^T(t_{h_i}) \mathcal{Y}_{s_i}(t_{h_i})$, $\Sigma \mathcal{U}_{s_i} \triangleq \sum_{h_i=1}^N \mathcal{Y}_{s_i}^T(t_{h_i}) \mathcal{U}_{s_i}(t_{h_i})$, $t_{h_i} \in (\varsigma, t)$, and $N \in \mathbb{Z}_{>1}$. A method for selecting data is subsequently described in Remark 4.

Assumption 5: The camera has sufficiently rich motion so that there exists finite constants $\tau_{s_i} \in \mathbb{R}_{>\varsigma}$, $\lambda_\tau \in \mathbb{R}_{>0}$ such that for all time $t \geq \tau_{s_i}$, $\lambda_{\min}\{\Sigma_{\mathcal{Y}_{s_i}}\} > \lambda_\tau$, where $\lambda_{\min}\{\cdot\}$ and $\lambda_{\max}\{\cdot\}$ are the minimum and maximum eigenvalues of $\{\cdot\}$, respectively.² This assumption is an observability condition for the subsequent development that is similar to other image-based observers (cf., [9], [14], [17], [19], [21]); however, if this condition is not satisfied, the observer remains bounded, as demonstrated in the subsequent analysis.

Assumption 5 can be verified online and is heuristically easy to satisfy because it only requires a finite collection of sufficiently exciting $\mathcal{Y}_{s_i}(t)$ and $\mathcal{U}_{s_i}(t)$ to yield $\lambda_{\min}\{\Sigma_{\mathcal{Y}_{s_i}}\} > \lambda_\tau$. The time τ_{s_i} is unknown; however, it can be determined online by checking the minimum eigenvalue of $\Sigma_{\mathcal{Y}_{s_i}}$. After τ_{s_i} , $\lambda_{\min}\{\Sigma_{\mathcal{Y}_{s_i}}\} > \lambda_\tau$ implies that a constant unknown distance $d_{s_i/k}$ can be determined from (9) as

$$d_{s_i/k} = \mathcal{X}_{s_i}, \quad t \geq \tau_{s_i} \quad (10)$$

where $\mathcal{X}_{s_i} \triangleq \begin{cases} 0, & t < \tau_{s_i} \\ \Sigma_{\mathcal{Y}_{s_i}}^{-1} \Sigma \mathcal{U}_{s_i}, & t \geq \tau_{s_i} \end{cases}$. When $t \geq \tau_{s_i}$, (10) can be substituted into (3) to yield

$$d_{s_i/c}(t) = \nu_{s_i,1}(t), \quad t \geq \tau_{s_i} \quad (11)$$

and

$$d_{k/c}(t) = \nu_{s_i,2}(t), \quad t \geq \tau_{s_i} \quad (12)$$

where $\nu_{s_i,1}(t)$ and $\nu_{s_i,2}(t)$ are the first and second elements of $\nu_{s_i}(t) \triangleq \psi_{s_i}(t) \mathcal{X}_{s_i}$, respectively.

The estimation errors, $\tilde{d}_{s_i/c}(t), \tilde{d}_{k/c}(t), \tilde{d}_{s_i/k}(t) \in \mathbb{R}$, are quantified as

$$\tilde{d}_{s_i/c}(t) \triangleq d_{s_i/c}(t) - \hat{d}_{s_i/c}(t) \quad (13)$$

$$\tilde{d}_{k/c}(t) \triangleq d_{k/c}(t) - \hat{d}_{k/c}(t) \quad (14)$$

and

$$\tilde{d}_{s_i/k}(t) \triangleq d_{s_i/k} - \hat{d}_{s_i/k}(t) \quad (15)$$

where $\hat{d}_{s_i/c}(t), \hat{d}_{k/c}(t), \hat{d}_{s_i/k}(t) \in \mathbb{R}$ are the estimates with initial conditions $\hat{d}_{s_i/c}^0, \hat{d}_{k/c}^0, \hat{d}_{s_i/k}^0 \in \mathbb{R}$ selected based on the tracking objective (e.g., user knowledge of the application or related sensors, such as an altimeter, when viewing ground targets from an airborne camera). Motivated by the subsequent stability analysis, the implementable observer update laws for the estimates are designed using (10)–(12) as

$$\frac{d}{dt}(\hat{d}_{s_i/c}(t)) \triangleq \begin{cases} \eta_{s_i,1}(t), & t < \tau_{s_i} \\ \eta_{s_i,1}(t) + k_1(\nu_{s_i,1}(t) - \hat{d}_{s_i/c}(t)), & t \geq \tau_{s_i} \end{cases} \quad (16)$$

$$\frac{d}{dt}(\hat{d}_{k/c}(t)) \triangleq \begin{cases} \eta_{s_i,2}(t), & t < \tau_{s_i} \\ \eta_{s_i,2}(t) + k_2(\nu_{s_i,2}(t) - \hat{d}_{k/c}(t)), & t \geq \tau_{s_i} \end{cases} \quad (17)$$

¹ See [6], [7], and [45] for examples on calculating the rotation and normalized translation from planar and nonplanar features; however, nonplanar methods will require more than four features to be tracked.

² See [46] and [47] for some examples of methods for selecting data to satisfy the assumption.

and

$$\frac{d}{dt}(\hat{d}_{s_i/k}(t)) \triangleq \begin{cases} 0, & t < \tau_{s_i} \\ k_3(\mathcal{X}_{s_i} - \hat{d}_{s_i/k}(t)), & t \geq \tau_{s_i} \end{cases} \quad (18)$$

where $k_1, k_2, k_3 \in \mathbb{R}_{>0}$ are constants. Taking the time derivative of (13)–(15), and substituting (10)–(15), (4)–(6), and (16)–(18) yields

$$\frac{d}{dt}(\tilde{d}_{s_i/c}(t)) = \begin{cases} 0, & t < \tau_{s_i} \\ -k_1\tilde{d}_{s_i/c}(t), & t \geq \tau_{s_i} \end{cases} \quad (19)$$

$$\frac{d}{dt}(\tilde{d}_{k/c}(t)) = \begin{cases} 0, & t < \tau_{s_i} \\ -k_2\tilde{d}_{k/c}(t), & t \geq \tau_{s_i} \end{cases} \quad (20)$$

and

$$\frac{d}{dt}(\tilde{d}_{s_i/k}(t)) = \begin{cases} 0, & t < \tau_{s_i} \\ -k_3\tilde{d}_{s_i/k}(t), & t \geq \tau_{s_i} \end{cases} \quad (21)$$

implying for all time $t \geq \tau_{s_i}$, the estimation error derivatives are negative definite functions of the estimation errors. The forms of the update laws in (16)–(18) are implementable and used in practice, whereas the forms of the time derivative of the estimation errors in (19)–(21) are analytical and provided to facilitate the subsequent analysis.

IV. EXTENDED OBSERVER UPDATE LAW FOR EUCLIDEAN DISTANCE TO FEATURES FROM CAMERA

The subsequent analysis demonstrates that (13) and (19) will remain bounded while $t < \tau_{s_i}$. However, after sufficient data are gathered, for all $t \geq \tau_{s_i}$, (13) will be shown to decay exponentially. The delay required to get sufficient excitation may reduce transient performance (i.e., the error is not guaranteed to reduce until after time $t \geq \tau_{s_i}$), which is a disadvantage compared to previous approaches, such as [21], which improve estimation errors by estimating optical flow. Motivated by the optical flow estimator form of the inverse depth estimator in [21], the time rate of change of $\underline{u}_{s_i/c}(t)$ is approximated and used to provide additional information to the estimator in (16), which will improve transient performance while sufficient excitation has not occurred.

Taking the time derivative of $\underline{u}_{s_i/c}(t)$ yields

$$\begin{aligned} \frac{d}{dt}(\underline{u}_{s_i/c}(t)) &= -\underline{\omega}_c^\times(t)\underline{u}_{s_i/c}(t) \\ &+ \frac{1}{d_{s_i/c}(t)}(\underline{u}_{s_i/c}(t)\underline{u}_{s_i/c}^T(t) - I_{3 \times 3})\underline{v}_c(t) \end{aligned}$$

and

$$\xi_{s_i}^T(t)\xi_{s_i}(t)d_{s_i/c}(t) = \xi_{s_i}^T(t)\rho_{s_i}(t) \quad (22)$$

$$\text{where } \xi_{s_i}(t) \triangleq \left(\frac{d}{dt}(\underline{u}_{s_i/c}(t)) + \underline{\omega}_c^\times(t)\underline{u}_{s_i/c}(t) \right), \quad \rho_{s_i}(t) \triangleq \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix},$$

and $I_{3 \times 3} \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. To aid in the subsequent analysis, let

$\mu_{s_i}(t) \triangleq \eta_{s_i,1}(t) + k_\xi(\xi_{s_i}^T(t)\rho_{s_i}(t) - \xi_{s_i}^T(t)\xi_{s_i}(t)\hat{d}_{s_i/c}(t))$, then an extended version of the estimator in (16) is designed as

$$\frac{d}{dt}(\hat{d}_{s_i/c}(t)) \triangleq \begin{cases} \mu_{s_i}(t), & t < \tau_{s_i} \\ \mu_{s_i}(t) + k_1(\nu_{s_i,1}(t) - \hat{d}_{s_i/c}(t)), & t \geq \tau_{s_i} \end{cases} \quad (23)$$

where $k_\xi \in \mathbb{R}_{>0}$. Using (13) and (22) in (23), then simplifying yields

$$\frac{d}{dt}(\hat{d}_{s_i/c}(t)) = \begin{cases} \eta_{s_i,1}(t) + k_\xi\Xi_{s_i}(t)\tilde{d}_{s_i/c}(t), & t < \tau_{s_i} \\ \eta_{s_i,1}(t) + (k_1 + k_\xi\Xi_{s_i}(t))\tilde{d}_{s_i/c}(t), & t \geq \tau_{s_i} \end{cases} \quad (24)$$

TABLE I
RMS DEPTH ERROR AND POSITION ERROR IN METERS
OVER 15 EXPERIMENTS

Experiment	Estimator 1	Estimator 2	Estimator 3	Estimator 4	Trajectory
1	70.499	55.525	58.325	55.435	0.017
2	66.358	42.065	52.827	42.838	0.026
3	80.466	64.701	67.616	63.106	0.037
4	76.250	59.583	65.593	59.565	0.032
5	79.419	63.244	70.798	65.674	0.032
6	82.611	69.300	71.996	68.558	0.018
7	65.971	48.192	57.287	46.392	0.020
8	73.864	60.002	64.290	59.858	0.027
9	77.699	61.468	69.227	60.708	0.020
10	72.053	55.916	63.679	55.771	0.023
11	75.472	59.757	59.916	60.798	0.016
12	77.571	63.623	66.451	63.686	0.022
13	83.663	67.817	72.995	68.274	0.036
14	74.030	58.764	66.941	59.245	0.030
15	77.742	56.067	64.983	56.612	0.029
Mean	75.578	59.068	64.862	59.141	0.026
Standard Deviation	5.086	6.819	5.522	6.938	0.007

TABLE II
RMS DEPTH ERROR AND POSITION ERROR IN METERS OVER 15
EXPERIMENTS BEFORE LEARNING

Experiment	Estimator 1	Estimator 2	Estimator 3	Estimator 4	Trajectory
1	133.393	105.884	109.492	104.984	0.007
2	137.855	87.811	107.410	88.252	0.005
3	137.133	110.694	114.093	108.139	0.005
4	134.047	105.321	113.742	104.732	0.005
5	138.285	110.761	120.448	112.189	0.005
6	137.122	115.571	118.032	113.951	0.005
7	126.254	92.370	103.992	87.503	0.004
8	128.662	104.859	109.866	103.421	0.004
9	136.862	108.817	119.151	106.569	0.005
10	128.454	100.033	110.136	98.836	0.004
11	137.677	109.201	107.660	110.506	0.006
12	132.034	108.494	111.989	108.105	0.005
13	139.927	113.816	119.680	112.477	0.009
14	131.922	104.987	114.095	101.154	0.005
15	142.222	103.213	116.301	103.489	0.006
Mean	134.790	105.455	113.073	104.287	0.005
Standard Deviation	4.440	7.212	4.825	7.645	0.001

where $\Xi_{s_i}(t) \triangleq \xi_{s_i}^T(t)\xi_{s_i}(t)$. Substituting (24) into the time derivative of (13) yields

$$\frac{d}{dt}(\tilde{d}_{s_i/c}(t)) = \begin{cases} -k_\xi\Xi_{s_i}(t)\tilde{d}_{s_i/c}(t), & t < \tau_{s_i} \\ -(k_1 + k_\xi\Xi_{s_i}(t))\tilde{d}_{s_i/c}(t), & t \geq \tau_{s_i}. \end{cases} \quad (25)$$

Remark 2: Under Assumption 5, $\Xi_{s_i}(t) \geq 0$ given $\|\underline{v}_c(t)\|$ may be zero for any period of time; however, for Assumption 5 to be satisfied, there will be times where $\Xi_{s_i}(t) > 0$. Specifically, there will exist a set of times $\mathcal{T}_{s_i} \subset \bigcup_{h_i=1}^N(t_{h_i} - \varsigma, t_{h_i})$ such that $\Xi_{s_i}(t) > 0 \quad \forall t \in \mathcal{T}_{s_i}$, where h_i and t_{h_i} are from (9), implying the design in (23) may improve transient performance under Assumption 5. The reasoning is that for Assumption 5 to hold, there must be a change in $\underline{u}_{s_i/c}(t)$ over the time intervals $\mathcal{T}_{s_i} \subset \bigcup_{h_i=1}^N(t_{h_i} - \varsigma, t_{h_i})$ implying $\|\xi_{s_i}(t)\| > 0$ since $\xi_{s_i}(t) \triangleq (\frac{d}{dt}(\underline{u}_{s_i/c}(t)) + \underline{\omega}_c^\times(t)\underline{u}_{s_i/c}(t))$. Since $\Xi_{s_i}(t) \triangleq \xi_{s_i}^T(t)\xi_{s_i}(t)$, $\Xi_{s_i}(t) > 0$ over those time intervals implying $\frac{d}{dt}(\tilde{d}_{s_i/c}(t)) = -k_\xi\Xi_{s_i}(t)\tilde{d}_{s_i/c}(t) < 0$, and hence, $\tilde{d}_{s_i/c}(t)$ will be decaying.

Remark 3: As shown in (10)–(12), the distance values are determined and may be used directly; however, Assumption 5 describes the period of time τ_{s_i} required to learn (10) where no information about the distances is available except through the dynamics. Additionally, feature tracking is noisy resulting in noisy estimates of the rotation $R_{k/c}(t)$ and unit vector $\underline{u}_{k/c}(t)$ implying the history stack will have noisy data. The estimators in (16)–(18) combine the dynamics with feedback resulting in improved estimates and robustness to noise. Specifically, during the learning period, $t < \tau_{s_i}$, the estimators ensure the error is bounded as subsequently shown in the analysis. After the learning period, $t \geq \tau_{s_i}$, the estimators use error feedback to converge to the value by effectively filtering the measurements [e.g., from (16), $\frac{d}{dt}(\hat{d}_{s_i/c}(t)) \triangleq \eta_{s_i,1}(t) + k_1(\nu_{s_i,1}(t) - \hat{d}_{s_i/c}(t))$ after $t \geq \tau_{s_i}$ implying a control over how much a noisy $\nu_{s_i,1}(t)$ can affect the state estimate through k_1). Additionally, using the extended estimator in (23) further improves transient performance as subsequently shown in Fig. 4 where the estimator has practically converged when $t = \max\{\tau_{s_i}\}$, the time all the distances have been learned. Furthermore, the results in Tables I – III show that (23) outperforms the other methods

TABLE III
RMS DEPTH ERROR AND POSITION ERROR IN METERS OVER 15
EXPERIMENTS AFTER LEARNING

Experiment	Estimator 1	Estimator 2	Estimator 3	Estimator 4	Trajectory
1	13.374	3.979	12.838	8.567	0.017
2	10.655	1.868	13.894	8.166	0.029
3	12.513	2.657	14.246	10.123	0.045
4	12.517	1.180	15.656	7.629	0.038
5	13.431	1.903	20.777	17.759	0.038
6	13.359	2.414	16.889	7.562	0.022
7	10.403	3.502	22.177	10.390	0.024
8	11.632	3.804	17.099	11.610	0.033
9	12.661	1.683	19.959	9.847	0.024
10	11.162	1.861	20.035	9.457	0.028
11	10.239	2.828	13.483	8.119	0.019
12	9.996	1.576	12.845	7.833	0.027
13	13.093	3.261	19.967	16.608	0.044
14	11.044	3.090	24.677	21.283	0.036
15	13.885	2.590	18.542	8.494	0.034
Mean	11.998	2.547	17.539	10.897	0.031
Standard Deviation	1.284	0.831	3.567	4.080	0.008

in terms of rms error. An approach that only uses (10)–(12) would imply no information is available until enough data are collected, which will have higher rms error and no control over the rate of learning.

V. STABILITY ANALYSIS

Given the observer in (23) is an extension of (16), the resulting stability analysis of (16) is identical to Theorem 1 and is excluded. Let $\eta(t) \triangleq [\tilde{d}_{s_i/c}(t) \ \tilde{d}_{k/c}(t) \ \tilde{d}_{s_i/k}(t)]^T$ and $V(\eta(t)) : \mathbb{R}^3 \rightarrow \mathbb{R}$ be a candidate Lyapunov function defined as

$$V(\eta(t)) \triangleq \frac{1}{2} \eta^T(t) \eta(t). \quad (26)$$

Theorem 1: The observer update laws defined in (17), (18), and (23) ensure the estimation errors in $\eta(t)$ are bounded and globally exponentially stable in the sense that

$$\|\eta(t)\| \leq \|\eta(0)\| \exp(\beta \tau_{s_i}) \exp(-\beta t). \quad (27)$$

Proof: Taking the time derivative of (26), then substituting the error derivatives in (20), (21), and (25), simplifying, then upper bounding yields

$$\frac{d}{dt} (V(\eta(t))) \leq \begin{cases} 0, & t < \tau_{s_i} \\ -2\beta V(\eta(t)), & t \geq \tau_{s_i} \end{cases} \quad (28)$$

where $\beta = \min\{k_1, k_2, k_3\}$. From (26) and (28), [48, Th. 8.4] can be invoked to conclude that $\|\eta(t)\|^2 \leq \|\eta(0)\|^2 \ \forall t \leq \tau_{s_i}$. From [48, Th. 4.10], $\|\eta(t)\|^2 \leq \|\eta(\tau_{s_i})\|^2 \exp(2\beta \tau_{s_i}) \exp(-2\beta t) \ \forall t \geq \tau_{s_i}$. Evaluating the first bound on $\|\eta(t)\|^2$ at $t = \tau_{s_i}$, then substituting into the second bound on $\|\eta(t)\|^2$ and taking the square root yields (27). ■

VI. EXPERIMENTAL RESULTS

Fifteen experiments are provided to demonstrate the performance of the developed observers. The performances of the developed observers in (16)–(18) and (23) were tested using the Eigen3, OpenCV, and ROS c++ libraries (cf., [49], [50], and [51], respectively). A Kobuki Turtlebot with a 1920×1080 monochrome iDS uEye camera, shown in Fig. 2, provides images at 30 Hz. Features were extracted from images of a checkerboard, shown in Fig. 2, with 8×6 corners (48 total features) where each square is $0.06 \text{ m} \times 0.06 \text{ m}$ in dimension. The linear and angular velocities of the camera were calculated using the Turtlebot wheel encoders and a gyroscope at 50 Hz. In addition, an Optitrack motion capture system operating at 120 Hz measured the pose of the camera and checkerboard, allowing for the position of each feature relative to the camera to be known for comparison. Image processing and estimators were multithreaded using four threads and executed simultaneously on a computer with an Intel i7 processor running at 3.4 GHz, ensuring the system ran at 30 Hz. The errors of the distance estimators in (16) and (23) are compared to the estimator in [21] and an EKF. Given the estimator in [21] and the EKF estimate the inverse



Fig. 2. Image shows the checkerboard, Kobuki Turtlebot, and iDS uEye camera used for experiments.

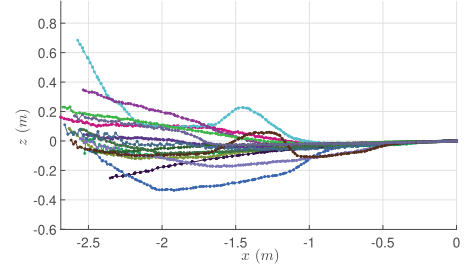


Fig. 3. Camera trajectories for each of the 15 experiments.

depth (i.e., $\frac{1}{z_{s_i/c}(t)}$, where $z_{s_i/c}(t)$ is the depth to feature s_i from c expressed in \mathcal{F}_c), whereas the estimators in (16) and (23) estimate the distance, the comparison of the four methods is shown by the depth, $z_{s_i/c}(t)$, which is the third element of $u_{s_i/c}(t) d_{s_i/c}$.

For each experiment, the Turtlebot started approximately 3 m away from the checkerboard, and various trajectories were taken, shown in Fig. 3, while maintaining the checkerboard in the FOV. In each experiment, the Turtlebot initially started at rest, and after traveling 2.5 m, the estimators were stopped to provide a large baseline. After the Turtlebot started its motion in the beginning of each experiment, the Turtlebot traveled without stopping until after the estimators were stopped in an effort to have the ideal conditions for estimation (i.e., continuous motion of the features in the camera FOV and continuous linear motion of the camera as is required for [21] and the EKF). The initial distances for the estimators in (16), (18), (23), the estimator in [21] and the EKF were initialized from a depth of 0.5 m. The estimator in (17) was initialized to 0 m. The gains for (16)–(18) and (23) were selected as $k_1 = k_2 = k_3 = 25$ and $k_\xi = 25k_1$, respectively. The maximum value for ς was 5 s. The 48 feature estimates were combined using a mean at each instance to update (17). The gain for the method in [21] was selected to be 100. The covariance matrices for the EKF were selected as $R = r \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for the measurement

covariance, $Q = r \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100000 \end{bmatrix}$ for the process covariance,

and $P(0) = r \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 150000 \end{bmatrix}$ for the initial state covariance, where

$r = 0.00001$. The gains and covariance matrices used were experimentally determined to obtain the performance shown in Figs. 3–5 and Tables I–III.

Remark 4: For a general system, the optimal approach to select good data and remove bad data for (9) (e.g., due to noise or parameter changes) remains an open problem and is often left to intuition about the system. Results in [47] demonstrated a purging method to remove bad data using two separate history stacks. The first stack was actively

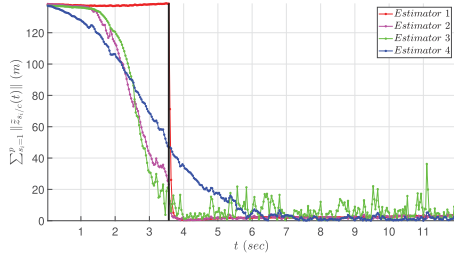


Fig. 4. Sum of the norm of each depth error across the 48 features (i.e., $\sum_{s_i=1}^{48} \|\tilde{z}_{s_i/c}(t)\|$) for experiment 11. Estimator 1 (red) refers to (16), estimator 2 (magenta) refers to (23), estimator 3 (green) refers to Dani *et al.* [21], and estimator 4 (blue) refers to the EKF. The black vertical line indicates the time when enough information was collected for learning.

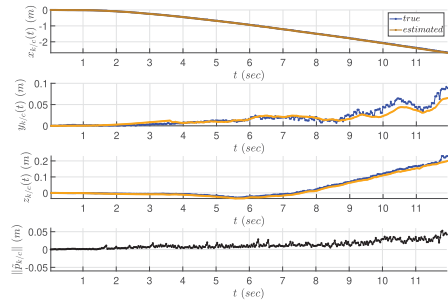


Fig. 5. Position error of the camera and the distance error using the estimator in (17) for experiment 11.

used by the estimator while the second stack collected data using a singular value maximization method. After a set of conditions was satisfied, the first stack was replaced by the second stack and the second stack was purged. After purging the second stack, new data were again collected using the singular value maximization method and the process repeated. For the experiments in this article, the selection of data was based on knowledge of approximate noise magnitudes in feature tracking and velocity measurements. Specifically, data were only selected if $\|\mathcal{Y}_{s_i}(t)\| \geq \epsilon_y$ and $\|\mathcal{U}_{s_i}(t)\| \geq \epsilon_u$ where $\epsilon_y, \epsilon_u \in \mathbb{R}_{>0}$ are values selected based on an empirical process. Because $\mathcal{Y}_{s_i}(t)$ is full column rank when $\|\mathcal{Y}_{s_i}(t)\| \geq \epsilon_y$ and $\|\mathcal{U}_{s_i}(t)\| \geq \epsilon_u$, the value of $d_{s_i/k}$ approximated by $\mathcal{Y}_{s_i}(t)$ and $\mathcal{U}_{s_i}(t)$ can be determined. Given $d_{s_i/k} > 0$, and some knowledge about reasonable values for the distances, values of $d_{s_i/k}$ can be determined and only $\mathcal{Y}_{s_i}(t)$ and $\mathcal{U}_{s_i}(t)$ values that had $d_{s_i/k}$ estimates falling in these bounds are saved to $\Sigma_{\mathcal{Y}_{s_i}}$ and $\Sigma_{\mathcal{U}_{s_i}}$. The values for ϵ_y and ϵ_u were $\epsilon_y = \epsilon_u = 0.1$ and the bounds on the distance were selected as 0.5 and 6 m.

A comparison of the example performance over time of the estimators is shown in Fig. 4 and Tables I–III, where before learning refers to $t < \max\{\tau_{s_i}\}$ and after learning refers to $t \geq \max\{\tau_{s_i}\}$. Fig. 4 shows the typical learning time where $\max\{\tau_{s_i}\} = 3.6$ s is shown as a black vertical line. Fig. 4 shows a comparison of the sum of the norm of each depth error across the 48 features (i.e., $\sum_{s_i=1}^{48} \|\tilde{z}_{s_i/c}(t)\|$) on the checkerboard for the estimators in (16), (23), [21], and the EKF, respectively. As shown in Fig. 4, the EKF estimator starts converging the fastest, but reaches steady state slower than the estimators in (16), (23), and [21]. However, after converging, the EKF has a similar error to the estimators in (16) and (23). Fig. 4 also shows that the estimator in (16) does not converge until sufficient learning occurs (at $t = 3.6$ s for experiment 11). The extension in Section IV shows an advantage of using current input–output data in the estimator, as shown by the

mean rms errors in Tables I and II. Specifically, the estimator in (16) is at a disadvantage to the other estimators before sufficient excitation has occurred, whereas the estimator in (23) starts converging to the true depths at a similar time frame as the estimator in [21]. The average rms error of (16) is more than 10 m greater than the other estimators over the entirety of each experiment, and more than 20 m greater before learning. However, Table III shows that the average error of (16) after learning is only 1 m greater than the EKF on average.

The extension in Section IV, specifically the design in (23), improves the error convergence of (16) such that the rms error is lower than the EKF on average. As shown in Table I, the average error over the entire experiment runtime was 59.068 m for (23) compared to 59.141 m for the EKF. After learning, the average rms error for the estimator in (23) was smaller (2.547 m) compared to the EKF (10.897 m). However, as shown in Table II, the rms error before learning was smaller for the EKF compared to (23), where the errors were 104.287 m for the EKF and 105.455 m for the estimator in (23). Additionally, Tables I–III show that the design in (23) has a smaller rms error than the design in [21] on average. Fig. 5 and Tables I–III show that the position error using (17) is small with an average rms error of 0.026 m over the entire run; 0.005 m before learning and 0.031 m after learning, which is approximately 1.2% error relative to trajectory length. The error increase after learning is a result of noise, which, as shown in Fig. 4 and Table III, causes the depth error to remain small but bounded at approximately 1.8% of the initial error. These experimental results demonstrate the ability of the observer in (16) to leverage both immediate information and learning to both converge quickly with low rms error and maintain a low rms error after converging.

VII. CONCLUSION

Novel observers using a single camera and structure from motion theory are developed to estimate the Euclidean distance to features on a stationary object and the Euclidean trajectory the camera takes while observing the object. Unlike previous results that estimate the inverse depth to features, the developed observer for estimating the Euclidean distance to features does not require the positive depth constraint. A Lyapunov-based stability analysis shows that the observer error is exponentially converging without requiring PE and instead only requires finite excitation through the use of ICL. An experimental comparison of the developed estimator to existing estimators shows that it achieves lower rms error when comparing feature depth estimates on average and the rms error of the position also remains low.

Future work may examine extending this result to include a solution for estimating the velocities of the camera and the structure of multiple objects while allowing those objects to leave the camera FOV over time either temporarily or permanently. The extended result would allow a camera to travel over larger distances and allow the camera to reconstruct a larger environment, which may be used in the development of a SLAM algorithm.

VII. ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this article are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

REFERENCES

- [1] G. Dubbelman and B. Browning, “COP-SLAM: Closed-form online pose-chain optimization for visual slam,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1194–1213, Oct. 2015.
- [2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

- [3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
- [4] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSI Trans. Comput. Vis. Appl.*, vol. 9, no. 1, 2017, Art. no. 16.
- [5] M. Karrer, P. Schmuck, and M. Chli, "CVI-SLAM—Collaborative visual-inertial SLAM," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2762–2769, Oct. 2018.
- [6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [7] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision*. Vienna, Austria: Springer, 2004.
- [8] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithm for estimating depth from image sequences," *Int. J. Comput. Vis.*, vol. 3, pp. 209–236, 1989.
- [9] M. Jankovic and B. Ghosh, "Visually guided ranging from observations points, lines and curves via an identifier based nonlinear observer," *Syst. Control Lett.*, vol. 25, no. 1, pp. 63–73, 1995.
- [10] S. Soatto, R. Frezza, and P. Perona, "Motion estimation via dynamic vision," *IEEE Trans. Automat. Control*, vol. 41, no. 3, pp. 393–413, Mar. 1996.
- [11] H. Kano, B. K. Ghosh, and H. Kanai, "Single camera based motion and shape estimation using extended Kalman filtering," *Math. Comput. Model.*, vol. 34, pp. 511–525, 2001.
- [12] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from motion causally integrated over time," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 523–535, Apr. 2002.
- [13] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, "Range identification for perspective vision systems," *IEEE Trans. Autom. Control*, vol. 48, no. 12, pp. 2232–2238, Dec. 2003.
- [14] X. Chen and H. Kano, "State observer for a class of nonlinear systems and its application to machine vision," *IEEE Trans. Autom. Control*, vol. 49, no. 11, pp. 2085–2091, Nov. 2004.
- [15] D. Karagiannis and A. Astolfi, "A new solution to the problem of range identification in perspective vision systems," *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 2074–2077, Dec. 2005.
- [16] D. Braganza, D. M. Dawson, and T. Hughes, "Euclidean position estimation of static features using a moving camera with known velocities," in *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, USA, Dec. 2007, pp. 2695–2700.
- [17] A. De Luca, G. Oriolo, and P. R. Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int. J. Robot. Res.*, vol. 27, no. 10, pp. 1093–1116, 2008.
- [18] G. Hu, D. Aiken, S. Gupta, and W. Dixon, "Lyapunov-based range identification for a paracatadioptric system," *IEEE Trans. Autom. Control*, vol. 53, no. 7, pp. 1775–1781, Aug. 2008.
- [19] F. Morbidi and D. Prattichizzo, "Range estimation from a moving camera: An immersion and invariance approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 2810–2815.
- [20] N. Zarrouati, E. Aldea, and P. Rouchon, "SO(3)-invariant asymptotic observers for dense depth field estimation based on visual data and known camera motion," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, Jun. 2012, pp. 4116–4123.
- [21] A. Dani, N. Fischer, Z. Kan, and W. E. Dixon, "Globally exponentially stable observer for vision-based range estimation," *Mechatronics*, vol. 22, no. 4, pp. 381–389, 2012.
- [22] A. Dani, N. Fischer, and W. E. Dixon, "Single camera structure and motion," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 241–246, Jan. 2012.
- [23] Z. I. Bell, H.-Y. Chen, A. Parikh, and W. E. Dixon, "Single scene and path reconstruction with a monocular camera using integral concurrent learning," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 3670–3675.
- [24] A. Parikh, R. Kamalapurkar, H.-Y. Chen, and W. E. Dixon, "Homography based visual servo control with scene reconstruction," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 6972–6977.
- [25] J. Oliensis, "A critique of structure-from-motion algorithms," *Comput. Vis. Image Underst.*, vol. 80, pp. 172–214, 2000.
- [26] J. Oliensis and R. Hartley, "Iterative extensions of the Strum/Triggs algorithm: Convergence and nonconvergence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2217–2233, Dec. 2007.
- [27] F. Kahl and R. Hartley, "Multiple-view geometry under the L_∞ -norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1603–1617, Sep. 2008.
- [28] A. Parikh, T.-H. Cheng, H.-Y. Chen, and W. E. Dixon, "A switched systems framework for guaranteed convergence of image-based observers with intermittent measurements," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 266–280, Apr. 2017.
- [29] A. Parikh, T.-H. Cheng, R. Licitra, and W. E. Dixon, "A switched systems approach to image-based localization of targets that temporarily leave the camera field of view," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 2149–2156, Nov. 2018.
- [30] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Target tracking in the presence of intermittent measurements via motion model learning," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 805–819, Jun. 2018.
- [31] M. Boutayeb, H. Rafaralahy, and M. Darouach, "Convergence analysis of the extended Kalman filter used as an observer for nonlinear deterministic discrete-time systems," *IEEE Trans. Autom. Control*, vol. 42, no. 4, pp. 581–586, Apr. 1997.
- [32] K. Reif and R. Unbehauen, "The extended Kalman filter as an exponential observer for nonlinear systems," *IEEE Trans. Signal Process.*, vol. 47, no. 8, pp. 2324–2328, Aug. 1999.
- [33] G. V. Chowdhary and E. N. Johnson, "Theory and flight-test validation of a concurrent-learning adaptive controller," *J. Guid. Control Dyn.*, vol. 34, no. 2, pp. 592–607, Mar. 2011.
- [34] G. Chowdhary, M. Mühlegg, J. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*, Q. Chu, B. Mulder, D. Choukroun, E.-J. van Kampen, C. de Visser, and G. Looye, Eds. Berlin, Germany: Springer, 2013, pp. 29–47.
- [35] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.
- [36] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.
- [37] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 12, pp. 1775–1787, Dec. 2019.
- [38] Z. Bell, P. Deptula, H.-Y. Chen, E. Doucette, and W. E. Dixon, "Velocity and path reconstruction of a moving object using a moving camera," in *Proc. Amer. Control Conf.*, 2018, pp. 5256–5261.
- [39] Z. Bell, J. Nezhadovitz, A. Parikh, E. Schwartz, and W. Dixon, "Global exponential tracking control for an autonomous surface vessel: An integral concurrent learning approach," *IEEE J. Ocean Eng.*, vol. 45, no. 2, pp. 362–370, Apr. 2020.
- [40] H.-Y. Chen, Z. I. Bell, P. Deptula, and W. E. Dixon, "A switched systems framework for path following with intermittent state feedback," *IEEE Control Syst. Lett.*, vol. 2, no. 4, pp. 749–754, Oct. 2018.
- [41] H.-Y. Chen, Z. Bell, P. Deptula, and W. E. Dixon, "A switched systems approach to path following with intermittent state feedback," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 725–733, Jun. 2019.
- [42] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.
- [43] J.-Y. Bouguet, "Pyramidal implementation of the affine Lucas Kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1–10, p. 4, 2001.
- [44] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.
- [45] K. Fathian, J. P. Ramirez-Paredes, E. A. Doucette, J. W. Curtis, and N. R. Gans, "QuEst: A quaternion-based approach for camera motion estimation from minimal feature points," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 857–864, Apr. 2018.
- [46] G. Chowdhary and E. Johnson, "A singular value maximizing data recording algorithm for concurrent learning," in *Proc. Amer. Control Conf.*, 2011, pp. 3547–3552.
- [47] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, "Concurrent learning for parameter estimation using dynamic state-derivative estimators," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3594–3601, Jul. 2017.
- [48] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [49] G. Guennebaud et al., "Eigen v3," 2010. [Online]. Available: <http://eigen.tuxfamily.org>
- [50] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
- [51] M. Quigley et al., "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, 2009, vol. 3, no. 3.2, p. 5.

Mobile Ground Target Tracking Subject to Intermittent Measurements by a Mobile Aerial Camera using Attention Deep Neural Networks

Zachary I. Bell, Runhan Sun, Kyle Volle, Prashant Ganesh, Scott A. Nivison, and Warren E. Dixon

Abstract—This paper presents a novel estimator and predictor framework for target tracking applications that estimates the pose of a mobile target that intermittently leaves the field-of-view (FOV) of a mobile agent's camera. Specifically, the framework uses an attention deep motion model network (DMMN) to estimate the dynamics of the target when the target is in the agent's FOV and uses the DMMN to predict the position, orientation, and velocity of the target when the target is outside the agent's FOV. A Lyapunov-based stability analysis is performed to determine the maximum dwell-time condition on target measurement availability, and experimental results are provided to demonstrate the performance of the proposed framework.

I. INTRODUCTION

Mobile target tracking tasks typically require mobile agents to use sensors such as cameras that have a limited field-of-view (FOV) which results in intermittent feedback of the target when it leaves the camera's FOV (i.e., the target is not visible to the agent). Intermittent measurements can occur for multiple reasons; for example, when the target is occluded by obstacles or other environmental factors. These obstacles and environmental factors may additionally require a mobile agent to purposely navigate away from a target, causing the target to leave the FOV. These conditions resulting in intermittent measurements which present numerous challenges to estimating the pose and velocity of the target by a mobile agent (cf., [1]–[6]).

Intermittency in measurements has traditionally been addressed by using probabilistic estimators. Various types of Kalman filters are widely used approaches that show local convergence of the mean (cf., [7]–[9]). Another approach is to use particle filters which show convergence as the number of samples approaches infinity (cf., [10]–[13]). In contrast, deterministic estimators typically assume boundedness of uncertainties and disturbances in the analysis, yielding uniformly ultimately bounded results (cf., [5] and [6]).

Runhan Sun, Kyle Volle, Prashant Ganesh, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611. E-mail: {runhansun, kvolle, prashant.ganesh, wdixon}@ufl.edu.

Zachary I. Bell and Scott A. Nivison are with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL 32542 USA. Email: {zachary.bell.10, scott.nivison}@us.af.mil.

This research is supported in part by A Task Order contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, AFOSR award numbers FA9550-18-1-0109, FA8651-20-F-1025 and 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 the sponsoring agency.

The recent work in [6] presented a novel method to handle intermittent target feedback when using multiple agents to track a single target. In [6], a centralized approach of modeling the motion of the target (i.e., the target's dynamics) is developed using a single layer neural network called the motion model network (MMN) with a fixed basis. The use of an MMN was motivated by the results in [5]; however, while [5] showed convergence over a finite number of cycles of losing and acquiring a target in an agent's FOV using an average dwell-time condition, there was no condition provided to ensure the estimation error didn't exceed the size of the agent's FOV. In practice, reacquiring a target in an agent's FOV can be challenging if the estimation error grows beyond the size of an agent's FOV. In contrast, [6] considered the size of the agent's FOV and developed minimum and maximum dwell-time conditions which dictate the minimum amount of time the target must be in the FOV and the maximum amount of time the target can leave the FOV to ensure the estimation error doesn't exceed the size of the FOV.

The approach in [6] develops dwell-time conditions based on feedback region sizes and was motivated by the results in [14] where dwell-times were determined for path following in feedback-denied environments. In [14], dwell-time conditions were developed to determine how long a mobile agent could remain in a feedback-denied region without the position estimation error growing beyond the size of the feedback region. However, target tracking introduces additional challenges in developing intermittent feedback dwell-time conditions since a target is typically not directly controlled by the mobile agent (i.e., a target is typically not cooperative) and the motion model of the target is unknown. Previous results have considered methods of indirect control of targets through unknown interaction models between targets and agents (cf., [15] and [16]), and developed approaches to learn and influence a target to follow a desired trajectory; however, a target is generally not guaranteed to be influenced by the relative pose between the target and the agent or that the agent can always follow the target. Following a target while relying on local feedback (i.e., feedback from local sensors such as a camera) to estimate the pose of the target introduces significant challenges over results that assume feedback is always globally available. Furthermore, the majority of these methods use a single layer MMN and only estimate the ideal output layer with a fixed basis which has been shown to reduce performance (cf., [17]).

In this paper, a novel deep MMN (DMMN) is proposed

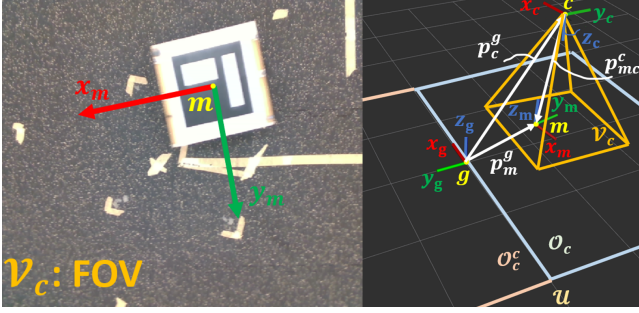


Figure 1: Kinematic relationship between cooperative and target agents.

to estimate the motion model of a mobile target that intermittently leaves an agent's FOV and provide dwell-time conditions for the availability of target measurements. Specifically, motivated by [17], the contribution of this paper is the development of a DMMN framework using a deep neural network (DNN) for the basis (i.e., the underlying dynamics features) in contrast to a fixed basis (cf., [6]). Furthermore, recent advances in attention DNNs (cf., [18]–[20]) are incorporated which have shown improvements in image-based pose estimation tasks (cf., [19]).

The DMMN framework is an estimator where the output layer weights are updated online while the target is in the agent's FOV. Simultaneously, motivated by [17], a replay buffer of target poses is collected online while the target is in the agent's FOV. When a sufficient number of target poses are collected, batch updates to the DMMN basis are performed to improve the estimate of the DMMN basis. After a series of batch updates, the prediction of the target's velocity switches to the new basis. Then, when the target leaves the agent's FOV, the DMMN is used as a predictor to estimate the target's pose and velocity while the target remains outside the agent's FOV.

A Lyapunov-based dwell-time analysis is performed to determine the maximum dwell-time condition that ensures the estimation error of the target's pose does not grow beyond the size of the camera's FOV while the target is outside the FOV. While it is not possible to ensure these conditions are satisfied (e.g., since a target is generally not cooperative), this dwell-time condition can be used to determine when the prediction error of the pose has grown too large and gives constraints to subsequently search for the target, improving the chances of finding the target. Experimental results are provided to demonstrate the performance of the proposed DMMN framework using a quadrotor with a downward facing camera tracking a mobile ground vehicle that intermittently leaves the agent's FOV.

II. SYSTEM DYNAMICS

As shown in the schematic in Figure 1, three coordinate frames are used to describe the tracking objective inside the tracking environment, denoted as $\mathcal{U} \subset \mathbb{R}^3$, where \mathcal{U} is convex and compact. Since tracking agents are often restricted by environmental factors (e.g., obstacles and battery life), the tracking agent is constrained within a specified operating region denoted as $\mathcal{O}_c \subset \mathcal{U}$, where $\mathcal{O}_c^c \triangleq \{p \in \mathcal{U} \mid p \notin \mathcal{O}_c\}$ describes the remaining space

outside of \mathcal{O}_c in \mathcal{U} . Let the Euclidean space of the target be represented as $\mathcal{M} \subset \mathcal{U}$ (i.e., all the 3D feature points on the target). Feedback of the target's state is only available when the target is in the tracking agent camera's FOV, $\mathcal{M} \subset \mathcal{V}_c$, where $\mathcal{V}_c \subset \mathcal{O}_c$ denotes the Euclidean space contained in the FOV as shown in Figure 1. The tracking agent's camera frame is represented as \mathcal{F}_c and has the origin at the principal point of the camera, denoted as c , with the basis $\{x_c, y_c, z_c\}$, where the z_c axis is along the viewing direction and co-linear with the optical axis, the y_c axis is along the image plane vertical, and the x_c axis is along the image plane horizontal. The frame \mathcal{F}_g denotes the inertial frame with an arbitrarily selected origin g with the basis $\{x_g, y_g, z_g\}$. The mobile target frame is represented as \mathcal{F}_m and has an origin located at an arbitrarily selected feature point on the target, denoted as m , with the basis $\{x_m, y_m, z_m\}$.

A. Target Dynamics

The objective of this work is to provide a motion model to estimate the pose (i.e., position and orientation) and velocity (i.e., linear and angular velocity) of the target using the tracking agent's camera, despite the target intermittently leaving the FOV. The pose of the tracking agent (i.e., the pose of \mathcal{F}_c with respect to \mathcal{F}_g), $\eta_c(t) \in \mathbb{R}^7$, is defined as $\eta_c(t) \triangleq [p_c^g(t)^\top \ q_c(t)^\top]^\top$, where $p_c^g(t) \in \mathbb{R}^3$ represents the position of \mathcal{F}_c with respect to \mathcal{F}_g expressed in \mathcal{F}_g , and $q_c(t) \in \mathbb{R}^4$ is the quaternion parameterization of $R_c(t) \in \mathbb{R}^{3 \times 3}$, the rotation matrix representing the orientation of \mathcal{F}_c with respect to \mathcal{F}_g . Rotation matrices can be represented using the quaternion parameterization, $q(t) \triangleq [q_0(t) \ q_v^\top(t)]^\top \in \mathcal{S}^4$ which has the standard basis $\{1, i, j, k\}$, where $\mathcal{S}^4 \triangleq \{x \in \mathbb{R}^4 \mid x^\top x = 1\}$, and $q_0(t) \in \mathbb{R}$ and $q_v(t) \in \mathbb{R}^3$ represent the scalar and vector components of $q(t)$, respectively.

Assumption 1. Measurements of the tracking agent's pose, $\eta_c(t)$, are always available using onboard sensors (e.g., an inertial navigation system).

The pose of the target (i.e., the pose of \mathcal{F}_m with respect to \mathcal{F}_g), $\eta_m(t) \in \mathbb{R}^7$, is defined as $\eta_m(t) \triangleq [p_m^g(t)^\top \ q_m(t)^\top]^\top$, where $p_m^g(t) \in \mathbb{R}^3$ represents the position of \mathcal{F}_m with respect to \mathcal{F}_g expressed in \mathcal{F}_g , $q_m(t) \in \mathbb{R}^4$ is the quaternion parameterization of $R_m(t) \in \mathbb{R}^{3 \times 3}$, the rotation matrix representing the orientation of \mathcal{F}_m with respect to \mathcal{F}_g . The pose of the target cannot be directly measured and instead the pose of the camera is used with the measurable relative pose of the target with respect to the camera (i.e., \mathcal{F}_m with respect to \mathcal{F}_c expressed in \mathcal{F}_c), $\eta_{mc}(t) \in \mathbb{R}^7$, is defined as $\eta_{mc}(t) \triangleq [p_{mc}^c(t)^\top \ q_{mc}(t)^\top]^\top$, where $p_{mc}^c(t) \in \mathbb{R}^3$ represents the position of \mathcal{F}_m with respect to \mathcal{F}_c expressed in \mathcal{F}_c , $q_{mc}(t) \in \mathbb{R}^4$ is the quaternion parameterization of $R_{mc}(t) \in \mathbb{R}^{3 \times 3}$, the rotation matrix representing the orientation of \mathcal{F}_m with respect to \mathcal{F}_c . Using $\eta_c(t)$ and $\eta_{mc}(t)$, the target position is described as illustrated in Figure 1 as

$$p_m^g(t) = p_c^g(t) + R_c(t)p_{mc}^c(t), \quad (1)$$

and the orientation of \mathcal{F}_m with respect to \mathcal{F}_g is

$$q_m(t) = q_c(t) \cdot q_{mc}(t), \quad (2)$$

where $q_c \cdot q_{mc} = \begin{bmatrix} q_{c0} & -q_{cv}^\top \\ q_{cv} & q_{c0}I_3 + q_{cv}^\times \end{bmatrix} q_{mc}$, and $q_{c0}(t) \in \mathbb{R}$ and $q_{cv}(t) \in \mathbb{R}^3$ are the scalar and vector components of $q_c(t)$, respectively.

The target's velocity in the target frame (i.e., velocity of \mathcal{F}_m expressed in \mathcal{F}_m), $\varphi_m(t) \in \mathbb{R}^6$, is represented as $\varphi_m(t) \triangleq [v_m(t)^\top \ \omega_m(t)^\top]^\top$, where $v_m(t), \omega_m(t) \in \mathbb{R}^3$ are the linear and angular velocity of \mathcal{F}_m expressed in \mathcal{F}_m , respectively. Using $\varphi_m(t)$, the time derivative of $\eta_m(t)$ yields the velocity of \mathcal{F}_m with respect to \mathcal{F}_g expressed in \mathcal{F}_g as

$$\dot{\eta}_m(t) = f_m(t) \varphi_m(t), \quad (3)$$

where $f_m(t) \triangleq \begin{bmatrix} R_m(t) & 0_{3 \times 3} \\ 0_{4 \times 3} & \frac{1}{2}B(q_m(t)) \end{bmatrix} \in \mathbb{R}^{7 \times 6}$, and $B(q_m(t)) \triangleq \begin{bmatrix} -q_{mv}^\top \\ q_{m0}I_3 + q_{mv}^\times \end{bmatrix} \in \mathbb{R}^{4 \times 3}$, with the pseudo-inverse property $B(q_m(t))^\top B(q_m(t)) = I_{3 \times 3}$ and $(\cdot)^\times : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ represents the skew operator, and $q_{m0}(t) \in \mathbb{R}$ and $q_{mv}(t) \in \mathbb{R}^3$ are the scalar and vector components of $q_m(t)$, respectively (cf., [5]).

Assumption 2. Measurements of $\eta_m(t)$ and $\dot{\eta}_m(t)$ are only available when the target is in the FOV (i.e., $\mathcal{M} \subset \mathcal{V}_c$).

III. ESTIMATION DESIGN

To achieve the objective, a method for estimating the pose of the target agent is developed that uses the target's pose as feedback while the target is in the FOV and predicts the pose and velocity of the target when the target intermittently leaves the FOV. Figure 2 illustrates the target tracking objective where the target enters the tracking agent camera's FOV (left image) and leaves the FOV (right image). To accomplish the tracking objective, an estimator and predictor are developed to estimate the target's pose $\eta_m(t)$. Specifically, a DMMN is used to estimate the target's velocity, $\varphi_m(t)$, when feedback of $\eta_m(t)$ is available (i.e., $\mathcal{M} \subset \mathcal{V}_c$). The DMMN is also used to propagate the pose estimates through time when feedback is unavailable (i.e., $\mathcal{M} \not\subset \mathcal{V}_c$) which occurs when the target is occluded or the target leaves the operating region, $\mathcal{M} \not\subset \mathcal{O}_c$. Let $\rho_c(t) \in \{a, u\}$ be a switching signal indicating if feedback of the target is available (i.e., $\rho_c(t) = a$ when $\mathcal{M} \subset \mathcal{V}_c$) or unavailable (i.e., $\rho_c(t) = u$ when $\mathcal{M} \not\subset \mathcal{V}_c$).

Assumption 3. The pose of the target $\eta_m(t) \in \Omega$ is bounded, where $\Omega \subset \mathbb{R}^7$ is a convex and compact set since $p_m^g(t) \in \mathcal{U}$ and $q_m(t) \in \mathcal{S}^4$ are convex and compact.

Assumption 4. The target's velocity, $\varphi_m(t)$, is described by a locally Lipschitz function of the target's pose, which is not explicitly time dependent. Specifically, $v_m(t) = \varphi_1(\eta_m(t))$ and $\omega_m(t) = \varphi_2(\eta_m(t))$, where $v_m(t)$ and $\omega_m(t)$ are bounded, $\sup_{\eta_m(t) \in \Omega} \{\|v_m(t)\|\} \leq \bar{v}_m \in \mathbb{R}_{>0}$ and $\sup_{\eta_m(t) \in \Omega} \{\|\omega_m(t)\|\} \leq \bar{\omega}_m \in \mathbb{R}_{>0}$, implying $\varphi_1, \varphi_2 \in \mathbb{R}^3$ are bounded (cf., [5]).

Assumption 4 guarantees there exists a function that can be approximated, using universal function approximators (e.g., neural networks), that describes $v_m(t)$ and $\omega_m(t)$ to an arbitrary level of accuracy via the Stone–Weierstrass

Theorem [21]. Furthermore, the Stone–Weierstrass Theorem only ensures the approximation is accurate over a closed interval. Thus, dependence on $\eta_m(t)$ is allowed since it is bounded via Assumption 3. Specifically, from Assumption 4, $\varphi_m(t) = [v_m(t)^\top \ \omega_m(t)^\top]^\top = [\varphi_1(\eta_m(t))^\top \ \varphi_2(\eta_m(t))^\top]^\top$ can be approximated using a DNN, that is, the approximation of the DMMN is

$$\varphi_m(t) = W^\top \sigma(\Phi(\eta_m(t))) + \varepsilon(\eta_m(t)), \quad (4)$$

where $W \in \mathbb{R}^{L \times 6}$ denotes the constant unknown bounded ideal output layer weight matrix, $\sigma \in \mathbb{R}^L$ denotes the known bounded activation functions corresponding to the output layer, $L \in \mathbb{Z}_{>0}$ denotes the user-defined number of neurons used in the output layer, $\varepsilon \in \mathbb{R}^6$ denotes the unknown bounded function reconstruction error, and $\Phi \in \mathbb{R}^L$ denotes a function that contains the inner layer ideal weights and activation functions of the DNN. Specifically,

$$\Phi(\eta_m(t)) \triangleq \phi_r \circ \phi_{r-1} \circ \dots \circ \phi_2 \circ \phi_1(\eta_m(t)), \quad (5)$$

where $\phi_l = \sigma_l(W_l^\top \phi_{l-1} + b_l)$, $l \in \{1, \dots, r\}$ with $r \in \mathbb{Z}_{\geq 1}$ denoting the user-defined number of inner layers of the DNN, $\phi_0 = \eta_m(t)$ is the input to DNN, $\sigma_l \in \mathbb{R}^{L_l}$ is the activation function for the l th inner layer, $W_l \in \mathbb{R}^{L_{l-1} \times L_l}$ denotes the ideal constant weight matrix for the l th inner layer, and $b_l \in \mathbb{R}^{L_l}$ denotes the ideal constant bias column matrix.

Assumption 5. There exist known constants $\bar{\varphi}_m, \underline{\sigma}, \bar{\sigma} \in \mathbb{R}_{>0}$, where $\sup_{\eta_m(t) \in \Omega} \{\|\varphi_m(\eta_m(t))\|\} \leq \bar{\varphi}_m$, $\inf_{\eta_m(t) \in \Omega} \{\sigma(\Phi(\eta_m(t)))\} \geq \underline{\sigma}$, and $\sup_{\eta_m(t) \in \Omega} \{\sigma(\Phi(\eta_m(t)))\} \leq \bar{\sigma}$, such that constants $\bar{W}, \bar{\varepsilon} \in \mathbb{R}_{>0}$ can be determined, with $\|W\| \leq \bar{W}$ and $\sup_{\eta_m(t) \in \Omega} \{\|\varepsilon(\eta_m(t))\|\} \leq \bar{\varepsilon}$, (cf., [5], [6], and [17]).

Substituting (4) into (3) yields

$$\dot{\eta}_m(t) = f_m(t) W^\top \sigma(\Phi(\eta_m(t))) + f_m(t) \varepsilon(\eta_m(t)). \quad (6)$$

Let $\tilde{\eta}_m(t) \in \mathbb{R}^7$ quantify the pose error as

$$\tilde{\eta}_m(t) \triangleq \eta_m(t) - \hat{\eta}_m(t), \quad (7)$$

where $\hat{\eta}_m(t) \in \mathbb{R}^7$ is the estimate of $\eta_m(t)$. Let $\tilde{W}(t) \in \mathbb{R}^{L \times 6}$ quantify the error in the estimate of the ideal output weights of the DMMN as

$$\tilde{W}(t) \triangleq W - \hat{W}(t), \quad (8)$$

where $\hat{W}(t) \in \mathbb{R}^{L \times 6}$ is the estimate of W .

Since the ideal weights of inner layers of the DNN contained in $\Phi(\eta_m(t))$ are unknown, let $\hat{\Phi}_k(\eta_m(t)) \in \mathbb{R}^L$, $k \in \{1, 2, \dots\}$, represent the k th iterative update to approximate $\Phi(\eta_m(t))$. Furthermore, let $T_k \in \mathbb{R}_{\geq 0}$ represent the time when $\hat{\Phi}_k(\eta_m(t))$ is used to approximate (4), since $\hat{\Phi}_k(\eta_m(t))$ is updated at a slower timescale using a subsequently defined loss function, (cf., [17]).

A. Target Pose Estimator

While $\rho_c(t) = a$ (i.e., the target is in the FOV and measurements of the target pose, $\eta_m(t)$, are available), an estimator is designed to update the pose estimate; however, a predictor must be used while $\rho_c(t) = u$ (i.e., the target is outside the FOV and measurements of the target pose, $\eta_m(t)$, are unavailable). Let $t_j^a \in \mathbb{R}_{\geq 0}$ represent the j th instance in time when $\rho_c(t) = a$ (i.e., the j th time the

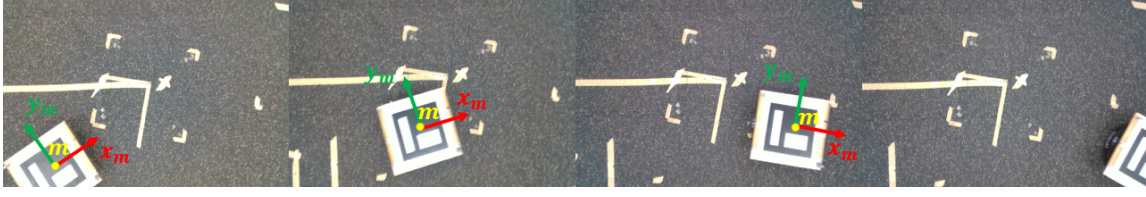


Figure 2: Generalized schematic of the target tracking objective.

target enters the FOV) and let $t_j^u \in \mathbb{R}_{>0}$ represent the j th instance in time when $\rho_c(t) = u$ (i.e., the j th time the target leaves the FOV), where $j \in \{1, 2, \dots\}$. Since the objective is to track and predict the target's trajectory, dwell-times are defined to quantify the amount of time the target is inside or outside of the FOV. Specifically, let $\Delta t_j^a \triangleq t_j^u - t_j^a$ and $\Delta t_j^u \triangleq t_{j+1}^a - t_j^u$ represent the j th amount of time $\rho_c(t) = a$ and $\rho_c(t) = u$, respectively.

Assumption 6. The target is in the FOV upon initialization (i.e., $\rho_c(0) = a$, $t_1^a = 0$, and $t_1^u < t_1^a$), (cf., [5] and [6]).

Based on the subsequent analysis, the pose estimate update law is designed as

$$\dot{\hat{\eta}}_m(t) = \begin{cases} f_m(t) \hat{\varphi}_m(\eta_m(t)) + k_\eta \tilde{\eta}_m(t) \\ + k_\varepsilon \text{SGN}(\tilde{\eta}_m(t)), & \rho_c(t) = a, \\ \text{proj} \left\{ \hat{f}_m(t) \hat{\varphi}_m(\hat{\eta}_m(t)) \right\}, & \rho_c(t) = u, \end{cases} \quad (9)$$

where $\text{SGN}(\tilde{\eta}_m(t)) = \begin{cases} 1, & \tilde{\eta}_m(t) > 0, \\ 0, & \tilde{\eta}_m(t) = 0, \\ -1, & \tilde{\eta}_m(t) < 0, \end{cases}$ is applied

element-wise since $\tilde{\eta}_m$ is a vector, $k_\eta, k_\varepsilon \in \mathbb{R}_{>0}^{7 \times 7}$ are constant control gains, $\text{proj}(\cdot)$ is a continuous projection operator defined in [22] with state and velocity bounds which are known under Assumptions 3 and 4, $\hat{\varphi}_m(\hat{\eta}_m(t)) \triangleq \hat{W}^\top(t) \sigma(\hat{\Phi}_k(\hat{\eta}_m(t)))$, and $\hat{f}_m(t) \triangleq \begin{bmatrix} \hat{R}_m(t) & 0_{3 \times 3} \\ 0_{4 \times 3} & \frac{1}{2} B(\hat{q}_m(t)) \end{bmatrix}$.

Remark 1. When the target leaves the FOV, the pose estimate is reset to the last measured pose before predicting the pose using (9) while $\rho_c(t) = u$. This reset reduces the prediction error while $\rho_c(t) = u$ and is only used at the first instances of $\rho_c(t) = u$ (i.e., at $t = t_j^u$, $\hat{\eta}_m(t_j^u) \mapsto \eta_m(t_j^u)$ implying $\|\tilde{\eta}_m(t_j^u)\| = 0$).

B. Weight Estimator

The weight estimator update laws are designed based on whether sufficient data has been collected on the trajectory of the target and whether the target is in the FOV.

1) *Output Weight Updates:* Motivated by the subsequent analysis, the output weight update law $\hat{W}(t)$ is designed

$$\text{vec}(\dot{\hat{W}}(t)) = \text{proj}(\mu(t), \text{vec}(\hat{W}(t))), \quad (10)$$

where $\text{vec}(\cdot)$ is the vectorization operator which stacks (\cdot) column-wise, and $\mu \in \mathbb{R}^{6L}$ is

$$\mu(t) = \begin{cases} \mu_k^a(t), & \rho_c(t) = a, \\ 0_{6L \times 1}, & \rho_c(t) = u, \end{cases} \quad (11)$$

$\mu_k^a(t) \triangleq \text{vec}(\Gamma \sigma(\hat{\Phi}_k(\eta_m(t))) \tilde{\eta}_m^\top(t) f_m(t))$, and $\Gamma \in \mathbb{R}_{>0}^{L \times L}$ is a positive definite, constant gain matrix.

2) *Inner Weight Updates:* The inner weights are updated periodically while $\rho_c(t) = a$ by saving tuples of data to a buffer $\mathcal{B}(t) = \{\eta_m(t_h), \dot{\eta}_m(t_h)\}_{h=1}^{b(t)}$ and when $b(t) > \bar{b}$, an optimization is performed to minimize the mean squared error of the velocity estimates. Specifically, the objective of the $k+1$ th batch optimization is to keep $\hat{W}(t)$ fixed and update the k th approximation of $\Phi(\eta_m(t))$ using the loss,

$$\mathcal{L}_{k+1}(t) = \frac{1}{b(t)} \sum_{h=1}^{b(t)} \left\| \dot{\eta}_m(t_h) - \hat{W}^\top(t) \sigma(\hat{\Phi}_k(\eta_m(t_h))) \right\|^2,$$

where Adam [23] is used to perform the optimization.

IV. ANALYSIS

The subsequent Lyapunov-based analysis provides conditions to ensure the tracking and weight estimation errors remain bounded despite the target intermittently leaving the FOV. The following provides a maximum dwell-time condition (i.e., maximum amount of time the target can leave the FOV) that ensures the target estimation error doesn't grow beyond a user-defined threshold.

Consider a stacked error $\xi(t) \triangleq \begin{bmatrix} \tilde{\eta}_m(t)^\top & \text{vec}(\tilde{W}(t))^\top \end{bmatrix}^\top \in \mathbb{R}^{7+6L}$ and a Lyapunov-based function candidate defined as

$$V(\xi(t)) \triangleq \frac{1}{2} \tilde{\eta}_m^\top(t) \tilde{\eta}_m(t) + \frac{1}{2} \text{tr}(\tilde{W}^\top(t) \Gamma^{-1} \tilde{W}(t)), \quad (12)$$

which is bounded as $\frac{\beta_\xi}{\beta_\xi} \|\xi(t)\|^2 \leq V(\xi(t)) \leq \frac{\beta_\xi}{\beta_\xi} \|\xi(t)\|^2$, $\frac{\beta_\xi}{\beta_\xi} \triangleq \frac{1}{2} \min\{1, \lambda_{\min}\{\Gamma^{-1}\}\}$, and $\frac{\beta_\xi}{\beta_\xi} \triangleq \frac{1}{2} \max\{1, \lambda_{\max}\{\Gamma^{-1}\}\}$, where λ_{\min} and λ_{\max} denote the minimum and maximum eigenvalue of $\{\cdot\}$, respectively.

Theorem 1. The tracking and weight error in $\xi(t)$ is uniformly ultimately bounded while $\rho_c(t) = a$, using the update laws in (9) and (10), in the sense that

$$\|\xi(t)\|^2 \leq \frac{\beta_\xi}{\beta_\xi} \|\xi(t_j^u)\|^2 \exp(-\beta_a(t - t_j^u)) + \frac{\delta_a}{\beta_a \beta_\xi}, \quad (13)$$

where $\beta_a \triangleq \frac{2\lambda_{\min}\{k_\eta\}}{\max\{1, \lambda_{\max}\{\Gamma^{-1}\}\}}$ and $\delta_a \triangleq 4\lambda_{\min}\{k_\eta\} \bar{\varphi}_m^2$.

Proof: Substituting (6), (9)-(11), and the time derivative of (7) and (8) into the time derivative of (12), for $\rho_c(t) = a$, using the bounds on (12), Assumptions 3-5, and simplifying yields

$$\dot{V}(\xi(t)) \leq -\beta_a V(\xi(t)) + \delta_a, \quad \rho_c(t) = a. \quad (14)$$

Applying the Comparison Lemma [24, Lemma 3.4] to (14), the bounds on (12), and simplifying yields (13). ■

Theorem 2. The tracking and weight error in $\xi(t)$ is bounded while $\rho_c(t) = u$, using the update laws in (9) and (10), in the sense that

$$\|\xi(t_{j+1}^a)\|^2 \leq \frac{\beta_\xi}{\beta_\xi} \|\xi(t_j^u)\|^2 + \frac{\Delta t_j^u \bar{\zeta}_u}{\beta_\xi}, \quad (15)$$

provided the following maximum dwell-time condition, $\Delta t^u \in \mathbb{R}_{>0}$, is satisfied

$$\overline{\Delta t^u} \leq \frac{\bar{\zeta}_u}{4\bar{\varphi}_m^2}, \quad (16)$$

where $\bar{\zeta}_u \in \mathbb{R}_{>0}$ is a user-defined threshold based on the size of the tracking agent's FOV.

Proof: Substituting (6), (9)-(11), and the time derivative of (7) and (8) into the time derivative of (12), for $\rho_c(t) = u$, using the bounds on (12), Assumptions 3-5, and simplifying yields

$$\dot{V}(\xi(t)) \leq \zeta_u(\|\tilde{\eta}_m(t)\|), \quad \rho_c(t) = u, \quad (17)$$

where $\zeta_u(\|\tilde{\eta}_m(t)\|) \triangleq 2\bar{\varphi}_m\|\tilde{\eta}_m(t)\|$. For the user-defined threshold on the error to hold, $\zeta_u(\|\tilde{\eta}_m(t)\|) \leq \bar{\zeta}_u \iff \|\tilde{\eta}_m(t)\| \leq \zeta_u^{-1}(\bar{\zeta}_u)$ resulting in the maximum dwell-time condition in (16). Using the maximum dwell-time constraint in (16), applying the Comparison Lemma [24, Lemma 3.4] to (17), the bounds on (12), and simplifying yields (15). ■

Remark 2. The design of $\bar{\zeta}_u$ can be considered an engineering parameter that is dependent on \mathcal{O}_c , \mathcal{V}_c , $\eta_{mc}(t)$, and assumptions made about the target (e.g., $\bar{\varphi}_m$).

V. EXPERIMENTS

The performance of the developed estimation and prediction framework is validated with a quadcopter equipped with a downward facing camera tracking a mobile ground target as shown in Figure 3. Figure 3 shows the mobile target tracks a 4×1 meter figure-8 trajectory using a motion capture system and is equipped with an ArUco marker [25] which is a placeholder for target recognition. The quadcopter is manually piloted using the REEF Estimator [26] for velocity and altitude control and when the ArUco marker is detected in the FOV, the estimator gets the target's pose $\eta_m(t)$ using motion capture; however, as shown in Figure 3, a table is used to occlude the target for approximately 5 seconds every time the target passes under the 1×1 meter portion in the center of the figure-8. Additionally, as shown in Figures 4 and 5, the target's ArUco marker is frequently not detected because image distortion from camera vibrations in-flight or the target leaves the FOV. It was assumed the altitude, roll, and pitch of the target were 0, so only the x-y position and yaw of the target were estimated.

The gains were selected as $k_\eta = 5$ and $\Gamma = I_{10 \times 10}$. The DMMN consisted of 3 ReLU layers with 10 basis each, a self-attention block, and a tanh layer at the output with 10 basis. The multi-layer perceptron (MLP) in Figures 4 and 5 show the DMMN without the self-attention block to compare the effectiveness of adding self-attention. The self-attention block looks for interaction between the features vectors and uses a non-linear mapping to combine them for prediction. Figure 4 shows the individual components of the tracking error over time starting after the first training cycle which occurred at around 90 seconds. Figure 5 shows the norm of the error over that time period. A total of 5 training cycles occurred over the 2 minutes shown in Figures 4 and 5, indicated by the black vertical lines. While feedback was available, pose measurements were stored in a buffer and once 500 measurements were saved, the DMMNs

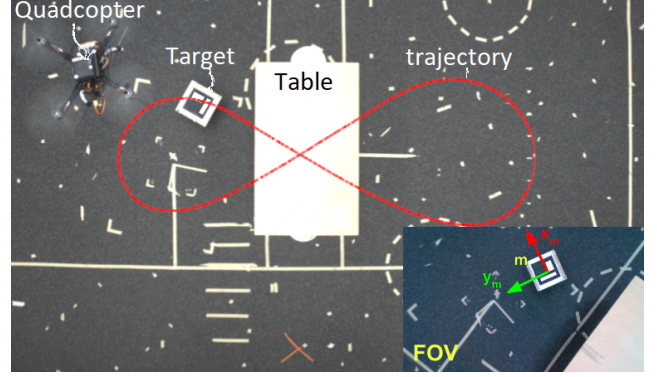


Figure 3: Overhead view of the quadcopter tracking the ground robot using a camera with the FOV embedded in the bottom right of the image. The target is occluded as it passes under the table in the center. The figure overlays the actual trajectory of the ground robot.

were trained as described in Section III-B2 for 75 epochs yielding a loss less than 10^{-4} . Once a training cycle was complete, half the measurements were randomly thrown out and the model was trained after collecting another 250 new measurements, saving the 250 old measurements for each training cycle. Over the 2 minutes shown in Figures 4 and 5, the mean squared error of the position was 0.59 meters with the self-attention block and 0.64 meters without the self-attention block implying the self-attention block improved the position estimates over frequent periods of time when feedback was unavailable.

For this set of experiments, the user-defined threshold was approximately 1 meter based on the FOV, and based on the maximum linear and angular velocity bounds of 0.5 m/s and 0.5 rad/s, respectively, and the maximum dwell-time was approximately 1 second based on (16) in the analysis; however, feedback of the target was frequently unavailable for longer than 1 second and typically the error didn't exceed the user-defined threshold until after 6 seconds of unavailable feedback implying the dwell-time was conservative. As shown by the gray regions in Figures 4 and 5, the norm of the error was on average approximately 0.1 meters after 1 second of unavailable feedback, 0.2 meters after 2 seconds, 0.5 meters after 4 seconds, and 1.1 meters after 6 seconds. This implies that while the error growth beyond the threshold was exponential as expected, the self-attention DMMN provided better than anticipated performance over periods of unavailable feedback based on the maximum dwell-time. Additionally, the self-attention DMMN performed well overall since the mean squared error of the estimate was 0.59 meters which is within the user-defined threshold of 1 meter.

VI. CONCLUSION

A novel estimation framework was presented that uses self-attention DNNs to estimate the pose and velocity of a mobile target that intermittently leaves the FOV of a mobile tracking agent equipped with a camera. A Lyapunov-based analysis was used to determine a maximum dwell-time condition on the availability of feedback to determine the maximum amount of time the target could leave the FOV

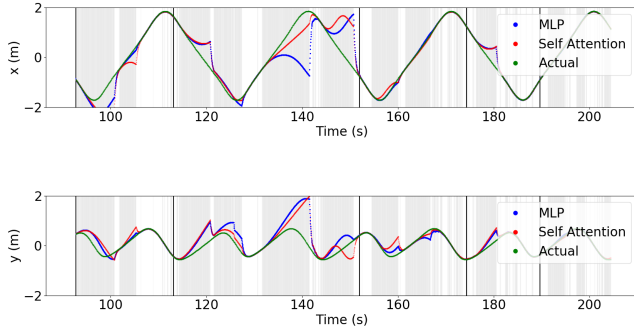


Figure 4: Estimated and actual x and y positions versus time obtained from the self-attention DMMN and the DMMN without attention (i.e., the MLP). Gray regions indicate when feedback is unavailable and the black vertical lines denote the time instances when training occurs.

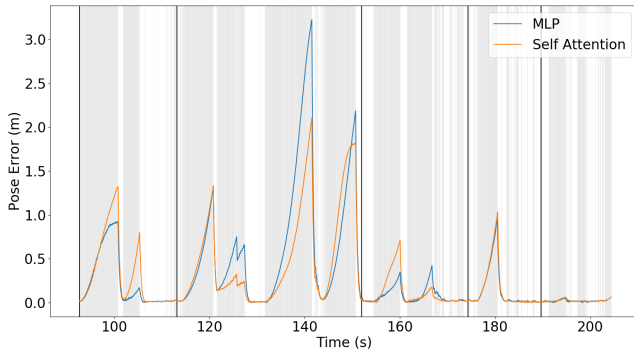


Figure 5: Pose error in the DMMN with self-attention and without self-attention, labeled MLP. The gray regions denote when feedback is unavailable and the black vertical lines denote the time instances when training occurs.

before the estimation error grew beyond a user-defined threshold. The presented experimental results demonstrated that the proposed self-attention DMMN performed better than expected by the Lyapunov-based analysis. Future work will examine methods to improve maximum dwell-time estimates and determine methods of extending this framework to developing consensus DMMNs with cooperative agents.

REFERENCES

- [1] S. Avidan and A. Shashua, "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 348–357, Apr. 2000.
- [2] J. Kaminski and M. Teicher, "A general framework for trajectory triangulation," *J. Math. Imag. Vis.*, vol. 21, no. 1, pp. 27–41, 2004.
- [3] A. Dani, N. Fischer, and W. E. Dixon, "Single camera structure and motion," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 241–246, Jan. 2012.
- [4] D. Chwa, A. Dani, and W. E. Dixon, "Range and motion estimation of a monocular camera using static and moving objects," *IEEE Trans. Control Syst. Tech.*, vol. 24, no. 4, pp. 1174–1183, July 2016.
- [5] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Target tracking in the presence of intermittent measurements via motion model learning," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 805–819, 2018.
- [6] C. Harris, Z. Bell, R. Sun, E. Doucette, J. W. Curtis, and W. E. Dixon, "Target tracking in the presence of intermittent measurements by a network of mobile cameras," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 5962–5967.
- [7] J. Sola, A. Monin, M. Devy, and T. Vidal-Calleja, "Fusing monocular information in multicamera SLAM," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 958–968, Oct. 2008.
- [8] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.
- [9] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1226–1243, Oct. 2013.
- [10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. Int. Joint Conf. Artif. Intell.*, 2003, pp. 1151–1156.
- [11] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellised particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
- [12] M. J. McCourt, J.-P. Ramirez-Paredes, E. A. Doucette, and J. W. Curtis, "A hybrid estimation algorithm for tracking an adversarial team," in *IEEE Int. Conf. on Sys., Man, and Cybern.*, 2017.
- [13] J.-P. Ramirez-Paredes, E. A. Doucette, J. W. Curtis, and N. R. Gans, "Distributed information-based guidance of multiple mobile sensors for urban target search," in *Autonomous Robots*, 2018.
- [14] H.-Y. Chen, Z. Bell, P. Deptula, and W. E. Dixon, "A switched systems approach to path following with intermittent state feedback," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 725–733, 2019.
- [15] R. Licitra, Z. Bell, and W. Dixon, "Single agent indirect herding of multiple targets with unknown dynamics," *IEEE Trans. Robotics*, vol. 35, no. 4, pp. 847–860, 2019.
- [16] P. Deptula, Z. Bell, F. Zegers, R. Licitra, and W. E. Dixon, "Approximate optimal influence over an agent through an uncertain interaction dynamic," *Automatica*, vol. 134, pp. 1–13, Dec. 2021.
- [17] R. Sun, M. Greene, D. Le, Z. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Syst. Lett.*, vol. 6, pp. 193–198, 2021.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advanc. Neural Inf. Proc. Sys.*, 2017, pp. 5998–6008.
- [19] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "Atloc: Attention guided camera localization," in *Proc. AAAI Conf. on Artif. Intell.*, vol. 34, no. 06, 2020, pp. 10 393–10 401.
- [20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [21] M. H. Stone, "The generalized weierstrass approximation theorem," *Math. Mag.*, vol. 21, no. 4, pp. 167–184, 1948.
- [22] Z. Cai, M. S. de Queiroz, and D. M. Dawson, "A sufficiently smooth projection operator," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 135–139, Jan. 2006.
- [23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [24] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [25] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and vision Computing*, vol. 76, pp. 38–47, 2018.
- [26] J. H. Ramos, P. Ganesh, W. Warke, K. Volle, and K. Brink, "Reef estimator: A simplified open source estimator and controller for multirotors," in *2019 IEEE National Aerospace and Electronics Conference (NAECON)*. IEEE, 2019, pp. 606–613.

Real-Time Modular Deep Neural Network-Based Adaptive Control of Nonlinear Systems

Duc M. Le^{ID}, Max L. Greene^{ID}, *Student Member, IEEE*, Wanjiku A. Makumi^{ID},
and Warren E. Dixon^{ID}, *Fellow, IEEE*

Abstract—A real-time deep neural network (DNN) adaptive control architecture is developed for uncertain control-affine nonlinear systems to track a time-varying desired trajectory. A Lyapunov-based analysis is used to develop adaptation laws for the output-layer weights and develop constraints for inner-layer weight adaptation laws. Unlike existing works in neural network and DNN-based control, the developed method establishes a framework to simultaneously update the weights of multiple layers for a DNN of arbitrary depth in real-time. The real-time controller and weight update laws enable the system to track a time-varying trajectory while compensating for unknown drift dynamics and parametric DNN uncertainties. A non-smooth Lyapunov-based analysis is used to guarantee semi-global asymptotic tracking. Comparative numerical simulation results are included to demonstrate the efficacy of the developed method.

Index Terms—Adaptive control, deep neural networks, Lyapunov methods, nonlinear control systems.

I. INTRODUCTION

NEURAL networks (NNs) have gained popularity due to their ability to approximate nonlinear functions. Conventional NNs can approximate functions to a prescribed accuracy [1] and [2]; however, recent evidence indicates deep neural networks (DNNs) exploit more complex learning features that can potentially improve function approximation performance [3]. Although DNNs may potentially approximate the nonlinear dynamics of a system more accurately, it is difficult to derive real-time adaptation laws for DNNs with multiple layers because the uncertain ideal weights are nested within a collection of nonlinear activation functions.

Manuscript received March 4, 2021; revised May 6, 2021; accepted May 10, 2021. Date of publication May 17, 2021; date of current version June 25, 2021. This work was supported in part by the Office of Naval Research under Grant N00014-13-1-0151; in part by NEEC under Award N00174-18-1-0003; in part by AFOSR under Award FA9550-18-1-0109 and Award FA9550-19-1-0169; in part by AFRL under Award FA8651-19-2-0009; and in part by NSF under Award 1762829. Recommended by Senior Editor C. Seatzu. (*Corresponding author: Duc M. Le.*)

The authors are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: ledan50@ufl.edu; maxgreene12@ufl.edu; makumiw@ufl.edu; wdixon@ufl.edu).

Digital Object Identifier 10.1109/LCSYS.2021.3081361

Results in [4]–[7] leverage Lyapunov-based analysis to develop multi-timescale DNN-based controllers containing real-time and offline iterative learning components. The output-layer weights of the DNN are adjusted online (i.e., in real-time) using NN-based adaptive control techniques. Concurrent to real-time execution, data is collected and DNN training algorithms, such as gradient descent and stochastic gradient descent (see [4], [5], [7] and [8, Ch. 8]), are used to iteratively update the inner-layer DNN weights. Since DNN learning algorithms are performed iteratively, the inner-layer weights are not updated continuously in real-time. The benefit of iterative learning is that the system performance improves with the quality of the DNN estimate. However, improving the quality of the DNN estimate may require a large training data set to capture the nonlinearities of the dynamics and significant computational resources to adjust the inner-layer weights. The DNN-based methods in [4]–[7] also raise questions regarding the inner-layer weights updates such as: when to collect data, what is the most efficient way to retrain the inner-layer weights, when should the inner-layer weights be updated in the implemented adaptation law, etc.

While such open questions are topics for further investigation, this letter investigates general characteristics and structures of inner-layer adaptation laws. Specifically, this letter develops general constraints on the inner-layer adaptation laws to update the inner-layer weight estimates in real-time. A Lyapunov-based analysis is used to develop a continuous adaptation law to estimate the output-layer weights. However, unlike previous methods, this letter provides a first insight into the development of Lyapunov-based adaptive update laws for both the inner-layer DNN weights as well as the output-layer weights. Inspired by modular adaptive control designs in [9]–[12], general constraints on the inner-layer DNN weight update laws are developed that enable modular design and selection of update laws. The developed DNN-based modular adaptive architecture allows more flexibility when selecting inner-layer DNN weight update laws.

In arbitrary width and depth DNNs, there may be hundreds or thousands of inner-layer weights. Simultaneously updating all the inner-layer weights online may be computationally intractable in real-time or undesired. Hence, the developed method provides a switched framework that provides design guidelines that can be used in future research efforts to guide

inner-layer weight adaptive update laws. In doing so, the inner-layer weight update laws may be arbitrarily switched on and off to allocate computational resources while updating the desired weights. Additionally, inner-layer weights may dropout, or be selectively turned off to prevent over-fitting and improve overall function approximation performance [13].

Results such as [6], [14], and [15] develop a robust sliding mode method to achieve asymptotic tracking with NN feedforward controllers. Like the aforementioned results, the developed method uses a sliding mode control term to yield asymptotic tracking in the presence of the NN reconstruction error, but also uses switched adaptation laws for the inner-layer weights. Hence, this letter leverages a nonsmooth Lyapunov-like analysis [16] to guarantee asymptotic tracking of a desired trajectory. Unlike existing works, the developed modular adaptive architecture incorporates the inner-layer weights of an arbitrarily deep DNN into a Lyapunov-based analysis to develop and characterize a general class of suitable inner-layer DNN adaptation laws. Moreover, sufficient conditions are developed to guarantee the tracking objective is achieved, despite the arbitrary number of inner-layers and switched update laws.

Notation: For a square matrix $A \in \mathbb{R}^{n \times n}$, the trace of A is denoted by $\text{tr}(A)$. Let $0_{n \times m}$ denote an $n \times m$ matrix of zeros. Let $\text{vec}(\cdot)$ denote the vectorization operator that transforms a matrix into a column vector, e.g., for a matrix $A \in \mathbb{R}^{n \times m}$, $\text{vec}(A) \in \mathbb{R}^{nm}$.

II. PROBLEM FORMULATION

A. System Dynamics

Consider a control-affine nonlinear dynamic system modeled as

$$\dot{x} = f(x) + g(x)u, \quad (1)$$

where $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ denotes the state, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes unknown, locally Lipschitz drift dynamics, $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ denotes the known control effectiveness matrix,¹ and $u : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ denotes the control input. To facilitate the subsequent control design, the following assumption is made on the control effectiveness matrix. The control effectiveness matrix $g(x)$ is assumed to be full-row rank for all $x \in \mathbb{R}^n$. The right pseudo inverse of $g(x)$ is denoted by $g^+ : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$, where $g^+(\cdot) \triangleq g^T(\cdot)(g(\cdot)g^T(\cdot))^{-1}$ is assumed to be bounded given a bounded argument.

B. Control Objective

The control objective is to track a user-defined time-varying trajectory $x_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ despite unknown system drift dynamics. The desired trajectory and its time derivative are assumed to be continuous and bounded, i.e., $x_d, \dot{x}_d \in \mathcal{L}_\infty$. To quantify the tracking objective, the tracking error $e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ is defined as

$$e \triangleq x - x_d. \quad (2)$$

¹While the developed method does not account for an uncertain control effectiveness matrix for simplicity and to better focus the result on the unique specific contributions, the method in [6] can be used with the developed method to approximate the uncertain control effectiveness matrix online.

III. CONTROL DESIGN

A. Feedforward DNN Estimate

NN-based adaptive control architectures are well-suited for uncertain or unstructured models, as in (1) where the drift dynamics $f(\cdot)$ are unknown. Using the universal function approximation property in [2], a DNN-based feedforward estimate of the drift dynamics is developed in this section. Let $\Omega \subset \mathbb{R}^n$ be a compact simply connected set and define $\mathbb{C}(\Omega)$ as the space where $f : \Omega \rightarrow \mathbb{R}^n$ is continuous. The universal function approximation property states there exist ideal weights and basis functions such that the drift dynamics $f(x) \in \mathbb{C}(\Omega)$ can be represented as

$$f(x) = W^{*T} \sigma^*(\Phi^*(x)) + \varepsilon(x), \quad (3)$$

where $W^* \in \mathbb{R}^{L \times n}$ denotes the unknown ideal output-layer weight matrix of the DNN, $\sigma^* : \mathbb{R}^p \rightarrow \mathbb{R}^L$ denotes the unknown vector of ideal activation functions corresponding to the output-layer of the DNN, $L \in \mathbb{Z}_{>0}$ denotes the user-defined number of neurons used in the output-layer, $\varepsilon : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes the unknown function reconstruction error, and $\Phi^* : \mathbb{R}^n \rightarrow \mathbb{R}^p$ denotes the inner-layers of the DNN containing unknown ideal weight matrices and activation functions. Specifically, the ideal inner DNN Φ^* can be expressed as

$$\Phi^*(x) \triangleq (V_k^{*T} \phi_k^* \circ V_{k-1}^{*T} \phi_{k-1}^* \circ \cdots \circ V_1^{*T} \phi_1^*)(V_0^{*T} x), \quad (4)$$

where $k \in \mathbb{Z}_{>0}$ denotes the user-defined number of inner-layers, $V_j^* \in \mathbb{R}^{L_j \times L_{j+1}}$ for all $j \in \{0, \dots, k\}$ denotes the j^{th} inner-layer ideal weight matrix, and $\phi_j^* : \mathbb{R}^{L_j} \rightarrow \mathbb{R}^{L_j}$ for all $j \in \{1, \dots, k\}$ denotes the j^{th} inner-layer vector of ideal activation functions, and the symbol \circ denotes function composition, e.g., $(g \circ h)(x) = g(h(x))$. The user-selected parameters $L_j \in \mathbb{Z}_{>0}$ for all $j \in \{1, \dots, k\}$ denote the number of neurons in the j^{th} inner-layer. Note that $L_0 = n$ and $L_{k+1} = p$.

Based on (3), the DNN feedforward estimate of the drift dynamics $\hat{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined as

$$\hat{f}(x) \triangleq \hat{W}^T \hat{\sigma}(\hat{\Phi}(x)), \quad (5)$$

where $\hat{W} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L \times n}$ denotes the output-layer weight matrix estimate, $\hat{\sigma} : \mathbb{R}^p \rightarrow \mathbb{R}^L$ denotes the user-selected vector of activation functions, and $\hat{\Phi} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ denotes the estimated inner-layers of the DNN. The inner-layer DNN estimate is defined as

$$\hat{\Phi}(x) \triangleq (\hat{V}_k^T \hat{\phi}_k \circ \hat{V}_{k-1}^T \hat{\phi}_{k-1} \circ \cdots \circ \hat{V}_1^T \hat{\phi}_1)(\hat{V}_0^T x), \quad (6)$$

where $\hat{V}_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_j \times L_{j+1}}$ for all $j \in \{0, \dots, k\}$ denotes the j^{th} inner-layer estimated weight matrix, and $\hat{\phi}_j : \mathbb{R}^{L_j} \rightarrow \mathbb{R}^{L_j}$ for all $j \in \{1, \dots, k\}$ denotes the j^{th} inner-layer vector of activation functions. The design of the update laws on the weight estimates \hat{W} and \hat{V}_j are subsequently defined. The weight estimate mismatch of the ideal output-layer weight $\tilde{W} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L \times n}$ and weight estimate mismatch of the ideal inner-layer weights $\tilde{V}_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_j \times L_{j+1}}$ for all $j \in \{0, \dots, k\}$ are defined as

$$\tilde{W} \triangleq W^* - \hat{W}, \quad (7)$$

$$\tilde{V}_j \triangleq V_j^* - \hat{V}_j. \quad (8)$$

It is assumed there exist known constants $\overline{W}^*, \overline{V}^*, \overline{\sigma}^*, \overline{\hat{\sigma}}, \overline{\varepsilon} \in \mathbb{R}_{\geq 0}$ that upper bound the unknown ideal weights W^* , unknown ideal weights V_j^* , unknown ideal bounded activation functions² $\sigma^*(\cdot)$, user-selected bounded activation functions $\hat{\sigma}(\cdot)$, and the function reconstruction error $\varepsilon(\cdot)$, respectively, as $\sup_{x \in \Omega} \|W^*\|_F \leq \overline{W}^*$, $\sup_{x \in \Omega, \forall j} \|V_j^*\|_F \leq \overline{V}^*$, $\sup_{x \in \Omega} \|\sigma^*(\cdot)\| \leq \overline{\sigma}^*$, $\sup_{x \in \Omega} \|\hat{\sigma}(\cdot)\| \leq \overline{\hat{\sigma}}$, and $\sup_{x \in \Omega} \|\varepsilon(\cdot)\| \leq \overline{\varepsilon}$ [1].

B. Control Development

Based on the subsequent stability analysis, the control input is designed as

$$u \triangleq g^+(x) \left(\dot{x}_d - k_1 e - k_s \text{sgn}(e) - \hat{f}(x) \right), \quad (9)$$

where $k_1, k_s \in \mathbb{R}_{>0}$ are user-defined control gains, and $\text{sgn}(\cdot)$ denotes the signum function. Based on the subsequent Lyapunov-based stability analysis, the output-layer weight estimate update law $\dot{W} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L \times n}$ is designed as

$$\dot{W} \triangleq \Gamma_W \hat{\sigma}(\hat{\Phi}(x)) e^T, \quad (10)$$

where $\Gamma_W \in \mathbb{R}^{L \times L}$ denotes a user-defined positive definite gain matrix used to adjust the learning rate of the output-layer weight matrix estimate.

Taking the time derivative of (2) and substituting in (1), (3), (5), and (9) yields the closed-loop error system

$$\begin{aligned} \dot{e} &= W^{*T} \sigma^*(\Phi^*(x)) + \varepsilon(x) - k_1 e - k_s \text{sgn}(e) \\ &\quad - \hat{W}^T \hat{\sigma}(\hat{\Phi}(x)). \end{aligned} \quad (11)$$

In [6], the inner-layer weights of the DNN were held constant and only updated discretely with data-driven learning algorithms. Common learning algorithms include gradient descent variants (see [4], [5], [7], and [8, Ch. 8]). These algorithms often use training data sets to update DNN weights through an optimization process in which the algorithms seek to minimize a cost function. However, DNN training algorithms often require large amounts of training data and high computational costs [3], making real-time execution intractable. Motivated to execute real-time learning and allow flexibility in user-selection of training algorithms while maintaining stability guarantees, we develop modular inner-layer DNN weight estimate update laws (see [17] for single-hidden-layer NNs).

For all $j \in \{0, \dots, k\}$, the j^{th} inner-layer weight update law $\dot{V}_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_j \times L_{j+1}}$ is designed as

$$\dot{V}_j \triangleq p_j(t) v_j(e, t) \mathbf{1}_{\{\hat{V}_j \leq \overline{V}_j\}}, \quad (12)$$

where $p_j : \mathbb{R}_{\geq 0} \rightarrow \{0, 1\}$ denotes a switching signal that indicates the active inner-layer weight estimate updates, $\mathbf{1}_{\{\cdot\}}$ is the indicator function, $\hat{V}_j, \overline{V}_j \in \mathbb{R}$ are user-defined constants where $\hat{V}_j \leq \overline{V}^*$, and $v_j : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_j \times L_{j+1}}$ denotes a user-defined function that satisfies

$$\|v_j\|_F \leq \rho(\|e\|) \|e\|, \quad (13)$$

²For some common activation functions, e.g., hyperbolic tangent functions, sigmoid functions, radial basis functions, $\sigma^* = \hat{\sigma} = L$.

where $\rho : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a positive, globally invertible, and non-decreasing function.

IV. STABILITY ANALYSIS

Theorem 1: Consider a system modeled by the dynamics in (1) with the initial condition $x(0) \in \Omega$. Then the control input in (9), output-layer weight adaptation law in (10), and the family of potential inner-layer weight adaptation laws that satisfy (12) ensure the closed-loop error system in (11) yields semi-global asymptotic tracking in the sense that $\lim_{t \rightarrow \infty} \|e(t)\| = 0$, provided the following sufficient gain condition is satisfied

$$k_s > \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) + \overline{\varepsilon} + 2\overline{V}^* (k+1) \rho \left(\sqrt{\frac{\overline{\alpha}}{\underline{\alpha}}} \|z(0)\| \right), \quad (14)$$

where $\underline{\alpha}, \overline{\alpha} \in \mathbb{R}_{\geq 0}$ are known constants.

Proof: Let $\mathcal{D} \subset \mathbb{R}^\Psi$ be a set containing $z = 0_{\Psi \times 1}$ and Ω , where $z : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^\Psi$ denotes a concatenated state defined as $z \triangleq [e^T, \text{vec}(\tilde{W})^T, \text{vec}(\tilde{V}_0)^T, \dots, \text{vec}(\tilde{V}_k)^T]^T$, and $\Psi \triangleq n(L+1) + \sum_{j=0}^k L_j L_{j+1}$ is defined for notional brevity. Consider the candidate Lyapunov function $V_L : \mathcal{D} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ defined as

$$V_L(z, t) \triangleq \frac{1}{2} e^T e + \frac{1}{2} \text{tr}(\tilde{W}^T \Gamma_W^{-1} \tilde{W}) + \frac{1}{2} \sum_{j=0}^k \text{tr}(\tilde{V}_j^T \tilde{V}_j), \quad (15)$$

which satisfies the inequality $\underline{\alpha} \|z\|^2 \leq V_L(z, t) \leq \overline{\alpha} \|z\|^2$, where $\underline{\alpha}, \overline{\alpha} \in \mathbb{R}_{\geq 0}$ are known constants. Let $\zeta : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^\Psi$ be a Filippov solution to the differential inclusion $\dot{\zeta} \in K[h](\zeta, t)$, where $\zeta(t) = z(t)$, the calculus of $K[\cdot]$ is used to compute Filippov's differential inclusion as defined in [18], and $h : \mathbb{R}^\Psi \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^\Psi$ is defined as $h(\zeta, t) \triangleq [\dot{e}^T, \text{vec}(\dot{\tilde{W}})^T, \text{vec}(\dot{\tilde{V}}_0)^T, \dots, \text{vec}(\dot{\tilde{V}}_k)^T]^T$. The generalized time-derivative of V_L along the Filippov trajectories of $\dot{\zeta} = h(\zeta, t)$ is defined by $\dot{V}_L(\zeta, t) \triangleq \bigcap_{\zeta \in \partial V_L(\zeta, t)} \zeta^T \begin{bmatrix} K[h](\zeta, t) \\ 1 \end{bmatrix}$, where $\partial V_L(\zeta, t)$ denotes Clarke's generalized gradient of $V_L(\zeta, t)$ [19, eq. 13]. Since $V_L(\zeta, t)$ is continuously differentiable in ζ , then $\partial V_L(\zeta, t) = \{\nabla V_L(\zeta, t)\}$, where ∇ denotes the gradient operator. Additionally, the time derivative of V_L exists almost everywhere (a.e.), i.e., $\dot{V}_L(\zeta, t) \stackrel{\text{a.e.}}{=} \dot{\tilde{V}}_L(\zeta, t)$ for almost all $t \in \mathbb{R}_{\geq 0}$.

Taking the generalized time derivative of (15), using the trace operator property,³ and substituting the closed-loop error system in (11), the output-layer adaptive update law in (10), and the inner-layer adaptive update laws in (12) yields

$$\begin{aligned} \dot{V}_L &\leq e^T \left(W^{*T} \sigma^*(\Phi^*(x)) + \varepsilon(x) - k_1 e - k_s K[\text{sgn}(e)] \right. \\ &\quad \left. - \hat{W}^T K[\hat{\sigma}(\hat{\Phi}(x))] - \tilde{W}^T K[\hat{\sigma}(\hat{\Phi}(x))] \right) \\ &\quad - \sum_{j=0}^k \text{tr} \left(K \left[\tilde{V}_j^T p_j(t) v_j(e, t) \mathbf{1}_{\{\hat{V}_j \leq \overline{V}_j\}} \right] \right). \end{aligned} \quad (16)$$

Adding and subtracting $e^T (W^{*T} K[\hat{\sigma}(\hat{\Phi}(x))])$ in (16) yields

$$\dot{V}_L \leq e^T W^{*T} \sigma^*(\Phi^*(x)) + e^T \varepsilon(x) - k_1 e^T e$$

³For real column vectors $a, b \in \mathbb{R}^n$, the trace of the outer product is equivalent to the inner product, i.e., $\text{tr}(ba^T) = a^T b$.

$$-k_s e^T K[\text{sgn}(e)] - e^T W^* K[\hat{\sigma}(\hat{\Phi}(x))] - \sum_{j=0}^k \text{tr} \left(K \left[\tilde{V}_j^T p_j(t) v_j(t) \mathbf{1}_{\{\hat{V}_j \leq \|\hat{V}_j\|_F \leq \bar{V}_j\}} \right] \right). \quad (17)$$

By the definition of the calculus $K[\cdot]$, $e^T K[\text{sgn}(e)] = \|e\|$. Using (13), (17) can be upper bounded as

$$\dot{V}_L \stackrel{\text{a.e.}}{\leq} -\|e\| \left(k_s - \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) - \bar{\varepsilon} - 2(k+1)\overline{V}^* \rho(\|e\|) \right) - k_1 \|e\|^2. \quad (18)$$

To ensure $k_s > \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) + \bar{\varepsilon} + 2(k+1)\overline{V}^* \rho(\|e\|)$, it is required that $\|e\| < \rho^{-1}(\frac{k_s - \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) - \bar{\varepsilon}}{2(k+1)\overline{V}^*})$, which implies $\|z\| < \rho^{-1}(\frac{k_s - \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) - \bar{\varepsilon}}{2(k+1)\overline{V}^*})$. The inequality in (18) can be upper bounded as

$$\dot{V}_L \stackrel{\text{a.e.}}{\leq} -k_1 \|e\|^2, \quad \forall z \in \mathcal{D}, \quad (19)$$

where $\mathcal{D} \triangleq \{z \in \mathbb{R}^\Psi : \|z\| < \rho^{-1}(\frac{k_s - \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) - \bar{\varepsilon}}{2(k+1)\overline{V}^*})\}$. Then using (15) and (17), V_L is positive definite and non-increasing, which implies $\|z(0)\| \leq \sqrt{\frac{V_L(0)}{\alpha}}$. Therefore, it is sufficient to show $\|z(0)\| < \sqrt{\frac{\alpha}{\alpha}} \rho^{-1}(\frac{k_s - \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) - \bar{\varepsilon}}{2(k+1)\overline{V}^*})$, which implies $\mathcal{S} \triangleq \{z \in \mathcal{D} : \sqrt{\frac{\alpha}{\alpha}} \rho^{-1}(\frac{k_s - \overline{W}^* (\overline{\sigma}^* + \overline{\hat{\sigma}}) - \bar{\varepsilon}}{2(k+1)\overline{V}^*})\}$ is the region where (19) holds, and yields the sufficient gain condition in (14).

From (15) and (19), $V_L \in \mathcal{L}_\infty$, which implies $z \in \mathcal{L}_\infty$, and hence, $e, \tilde{W} \in \mathcal{L}_\infty$. Using (2) and (7) implies $x \in \mathcal{L}_\infty$ and $\hat{W} \in \mathcal{L}_\infty$, respectively. Using (12) and (13), $e \in \mathcal{L}_\infty$ implies $v_j \in \mathcal{L}_\infty$, and by the use of the indicator function $\mathbf{1}_{\{\hat{V}_j \leq \|\hat{V}_j\|_F \leq \bar{V}_j\}}$, implies $\hat{V}_j \in \mathcal{L}_\infty$ for all $j \in \{0, \dots, k\}$. Using (6), the fact that $x, \hat{V}_j \in \mathcal{L}_\infty$ implies $\hat{\Phi} \in \mathcal{L}_\infty$. By design, $\hat{x}_d, \hat{\sigma} \in \mathcal{L}_\infty$. Using (9), the fact that $x, e, \hat{x}_d, \hat{W}, \hat{\sigma} \in \mathcal{L}_\infty$ implies $u \in \mathcal{L}_\infty$. Using (10), the fact that $\hat{\sigma}, e \in \mathcal{L}_\infty$ implies $\hat{W} \in \mathcal{L}_\infty$. By the LaSalle-Yoshizawa theorem extension for nonsmooth systems in [16] and [20], $k_1 \|e\|^2 \rightarrow 0$, which implies $\|e(t)\| \rightarrow 0$ as $t \rightarrow \infty$. ■

V. SIMULATION

To demonstrate the performance of the developed method, simulations are performed on the nonlinear system from [21], where $f(x) = [-x_1 + x_2, -\frac{1}{2}x_1 + \frac{1}{2}x_2(1 - (\cos(2x_1 + 2)^2))]^T$ and $g(x) = \text{diag}[5, 3]$ are used to model the drift dynamics and control effectiveness in (1). The desired trajectory is $x_d = [3 \cos(t), 5 \sin(t)]^T$. The initial condition is $x(0) = [3, 0]^T$. The controller gains are selected as $k_s = 0.5$ and $k = 7$.

To illustrate the modularity of the architecture, two simulation studies are conducted for 60 seconds, each with different inner-layer adaptation laws and structures implemented. During the entire 60 seconds, the output-layer update law in (10) is active in both studies. Due to the large number of weights in this system, only one inner-layer weight update

law is active at a time.⁴ However, the selection of switching signals that dictates when to update each inner-layer DNN may be arbitrarily selected. Computational resources may be allocated to update a subset of the inner-layer weights, or all inner-layers may be updated arbitrarily at a time. Additionally, inner-layer weights may dropout, or be selectively turned off to prevent over-fitting [13].

To reduce the computational load and show the flexibility in selection of the switching signals to update each inner-layer DNN weight estimate, in both simulation studies, the switching signal is arbitrarily designed as

$$p_j(t) = \begin{cases} 1, & t \in [10j, 10(j+1)], \\ 0, & \text{else,} \end{cases} \quad (20)$$

for all $j \in \{0, \dots, 5\}$. Based on the switching signals in (20), each inner-layer weight update law is active for 10 seconds during the duration of the simulation and is activated in consecutive order, i.e., \hat{V}_0 is active from 0 to 10 seconds, \hat{V}_1 is active from 10 to 20 seconds, etc. If the update law is not active (i.e., $p_j(t) = 0$), then its associated weights are not updated.

The DNN is composed of 6 layers and the hyperbolic tangent function is the activation function for each neuron. Layers 1-6 have 12, 10, 15, 15, 12, and 20 neurons, respectively. The outer-layer weight learning parameter is set to $\Gamma_W = 10 \cdot \mathbf{1}_{L \times L}$, where $\mathbf{1}_{n \times m}$ is an $n \times m$ matrix of ones. The bounds on the inner-layer weights are $\hat{V}_j = 10^{-6}$ and $\bar{V}_j = 250$ for all $j \in \{0, 1, \dots, 5\}$. The initial conditions for the output-layer weight estimate \tilde{W} and the inner-layer weight estimates \hat{V}_j are randomly selected from a uniform distribution from $[-0.5, 0.5]$.

Various update laws for \hat{V}_{0-5} can be selected and designed for learning the inner-layer DNN weights while guaranteeing tracking performance. The constraints in (13) provide general guidelines and enable the user to select or design update laws accordingly, such as gradient tuning laws based on back-propagated errors [22] or a Hebbian tuning law [23].

In the first simulation study (Study 1), the inner-layer weight update laws, which are heuristically selected and inspired by the methods in [24] and [25], are selected as

$$v_j = \Gamma_{V_j} e \cdot \text{re} \left(\left(\tanh \circ \hat{\phi}_j^{-1} \circ \hat{V}_j^T \hat{\phi}_{j+1}^{-1} \circ \dots \right. \right. \\ \left. \left. \dots \circ \hat{V}_5^T \hat{\sigma}^{-1} \right) \left(\hat{W}^T \hat{x} \right) \right) \hat{V}_j \quad (21)$$

for all $j \in \{0, 1, \dots, 5\}$, where the inner-layer weight learning parameters are set to $\Gamma_{V_j} = 1000 \cdot \mathbf{1}_{L_j \times 2}$, \hat{V}_j^+ is the right-pseudo inverse of \hat{V}_j , the $\text{re}(\cdot)$ operator outputs element-wise the real component of each entry in \hat{V}_j , and $\hat{x} \in \mathbb{R}^n$ denotes the numerically generated state derivative. Due to the projection bounds \hat{V}_j and \bar{V}_j , \hat{V}_j^+ exists and is bounded. The selected update law satisfies the modular adaptive control constraint in (13).

⁴There are 955 individual weights in the simulation. As guaranteed by the analysis, the developed method can update each layer separately, i.e., a subset of the total number of weights are updated at a given time. It may be computationally burdensome for some systems to update a large number of weights online. One purpose of this simulation is to highlight the developed method's ability update different DNN layers separately.

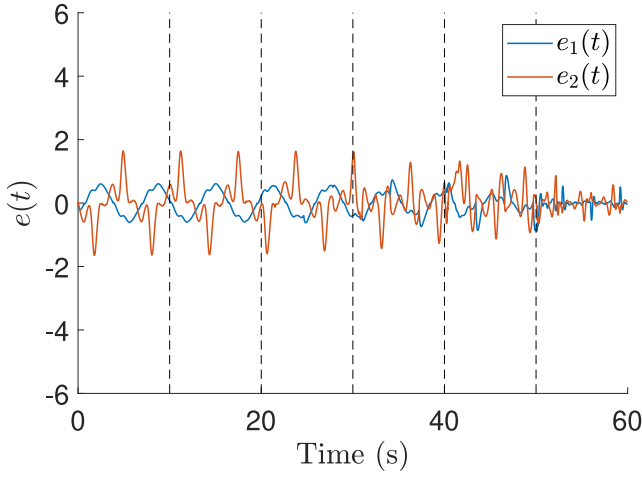


Fig. 1. Tracking error e in the system with the inner-layer DNN weight update law in (21). The vertical lines denote when a new inner-layer update law is activated.

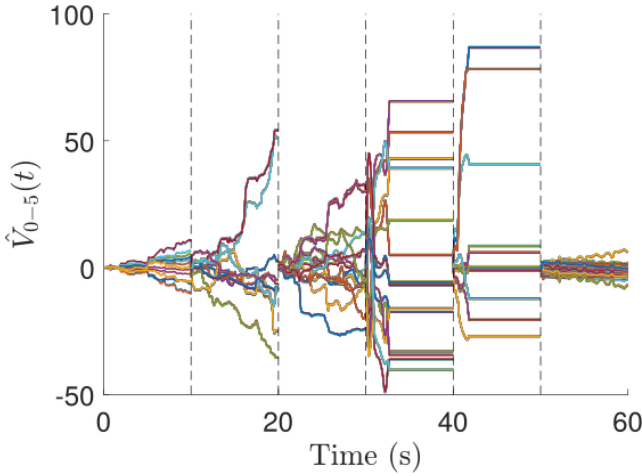


Fig. 2. Value of the active DNN inner-layer weight estimates \hat{V}_{0-5} when each inner-layer update law is active with the inner-layer DNN weight update law in (21). The vertical lines denote when a new inner-layer update law is activated. Based on the switching signals in (20), \hat{V}_0 is active from 0 to 10 seconds, \hat{V}_1 is active from 10 to 20 seconds, etc.

Figure 1 shows the tracking error $e \triangleq [e_1, e_2]^T$ with the inner-layer DNN weight update law in (21). From initialization to approximately 30 seconds, the system exhibits poor tracking performance with oscillatory behavior. Poor controller performance is likely due to randomly initialized weights which signifies no *a priori* model knowledge in the drift dynamic approximation. However, as each subsequent inner-layer weight matrix \hat{V}_{0-5} is updated, the tracking performance improves and the amplitude of the error signals decreases, which indicates the DNNs improved approximation of the drift dynamics.

Figure 2 shows the online adjustment of the inner-layer weights with the inner-layer DNN weight update law in (21). Each set of weights is divided by a set of black dashed lines. Note that some of the weights are unaffected by the indicator function since the upper bounds \hat{V}_j are sufficiently large. However, the weights of layers 3 and 4 are affected

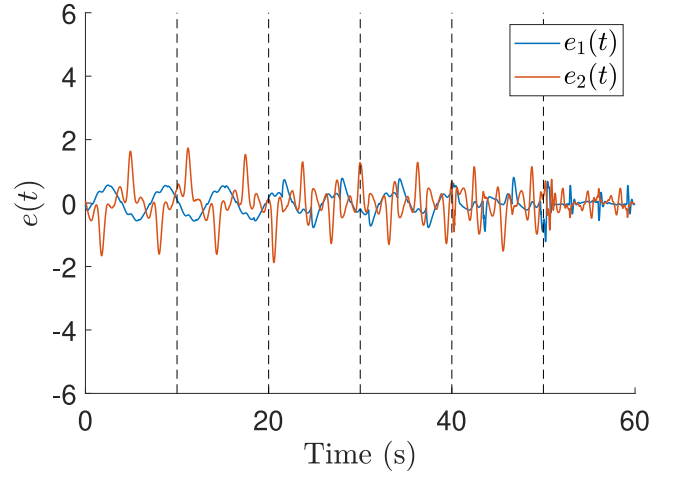


Fig. 3. Tracking error e in the system with inner-layer DNN weight update law in (22). The vertical lines denote when a new inner-layer update law is activated.

by the indicator function. When layers 3 and 4 are activated, their weight estimates quickly reach the bounds defined by the indicator function.

To provide a comparison in the selection of inner-layer weight update laws, a second simulation study (Study 2) is performed where the DNN is structured similarly as the first simulation study, and each inner-layer update is activated according to the switching signal in (20). To illustrate the modularity in the selection of inner-layer weight update laws, various inner-layer weight update laws are arbitrarily selected to satisfy (13) and are selected as

$$\begin{aligned} v_0 &= \Gamma_{V_0} e^{-\frac{e_1^2}{2}} e^{-\frac{e_2^2}{2}} \|e\|, \\ v_1 &= \Gamma_{V_1} e^{-\frac{e_1^2}{2}} \tanh(e_2) \|e\|, \\ v_2 &= \Gamma_{V_2} \tanh(e_1) \tanh(e_2) \|e\|, \\ v_3 &= \Gamma_{V_3} \frac{1}{1 + e^{-e_1}} e^{-\frac{e_2^2}{2}} \|e\|, \\ v_4 &= \Gamma_{V_4} \tanh(e_1) \frac{1}{1 + e^{-e_2}} \|e\|, \\ v_5 &= \Gamma_{V_5} \frac{1}{1 + e^{-e_1}} \tanh(e_2) \|e\|, \end{aligned} \quad (22)$$

where the inner-layer weight learning parameters are set to $\Gamma_{V_j} = 1000 \cdot \mathbf{1}_{L_j \times 2}$ for all $j \in \{0, 1, \dots, 5\}$.

Figure 3 shows the tracking error e with the inner-layer DNN weight update law in (22). Similar to Study 1, the tracking error results in Figure 3 show poor tracking performance at the start of the simulation. However, as the simulation progresses and each subsequent inner-layer is activated, the tracking performance improves.

Table I shows the root mean squared (RMS) error and standard deviation (SD) of the tracking error. The leftmost column indicates the active inner-layer weight law, e.g., the first row is data collected while the \hat{V}_0 update law is active. Study 1 uses the heuristically selected update law in (21), and the RMS error and SD corresponding to \hat{V}_0 are 0.735 and 0.145, respectively. Study 2 uses the update law in (22),

TABLE I
RMS AND SD ERROR

Active \hat{V}_j	Study 1		Study 2	
	RMS Error	SD Error	RMS Error	SD Error
\hat{V}_0	0.735	0.145	0.705	0.145
\hat{V}_1	0.727	0.142	0.700	0.149
\hat{V}_2	0.710	0.136	0.651	0.137
\hat{V}_3	0.640	0.116	0.612	0.098
\hat{V}_4	0.658	0.087	0.591	0.094
\hat{V}_5	0.289	0.039	0.327	0.050

and the RMS error and SD are 0.705 and 0.145, respectively. In Study 1, as each subsequent inner-layer update law is activated, the tracking performance improves and results in an RMS error and SD of 0.289 and 0.039, respectively. Although a similar trend is seen in Study 2, the inner-layer update law in Study 1 outperforms Study 2 which indicates the inner-layer update law yielded better function approximation performance, as expected. However, the tracking objective is achieved in both studies, despite different inner-layer update laws, as guaranteed by the analysis.

VI. CONCLUSION

This letter develops DNN-based modular adaptive control update laws and constraints, which were inspired by existing modular adaptive control constraints, to achieve trajectory tracking objectives. The modular adaptive control framework provides general constraints and enables users to design update laws accordingly. The developed method also allows for different sets of weights to be arbitrarily switched. A simulation study was performed to compare the performance of different inner-layer weight update laws selected. Inner-layer weight update laws were designed and selected that satisfy the developed design constraints. Despite different inner-layer update laws used and arbitrary switching, the implemented controllers achieved the tracking objective.

ACKNOWLEDGMENT

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.

REFERENCES

- [1] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Philadelphia, PA, USA: CRC Press, 1998.
- [2] N. E. Cotter, "The Stone-Weierstrass theorem and its application to neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 4, pp. 290–295, Dec. 1990.
- [3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [4] G. Joshi and G. Chowdhary, "Deep model reference adaptive control," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 4601–4608.
- [5] G. Joshi, J. Virdi, and G. Chowdhary, "Design and flight evaluation of deep model reference adaptive controller," in *Proc. AIAA Scitech Forum*, 2020, p. 1336.
- [6] R. Sun, M. Greene, D. Le, Z. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Syst. Lett.*, early access, Jan. 28, 2021, doi: [10.1109/LCSYS.2021.3055454](https://doi.org/10.1109/LCSYS.2021.3055454).
- [7] G. Joshi, J. Virdi, and G. Chowdhary, "Asynchronous deep model reference adaptive control," 2020. [Online]. Available: [arXiv:2011.02920](https://arxiv.org/abs/2011.02920).
- [8] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.
- [9] P. Patre, W. Mackunis, K. Dupree, and W. E. Dixon, "Modular adaptive control of uncertain Euler-Lagrange systems with additive disturbances," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 155–160, Jan. 2011.
- [10] M. S. De Queiroz, D. M. Dawson, and M. Agarwal, "Adaptive control of robot manipulators with controller/update law modularity," *Automatica*, vol. 35, no. 8, pp. 1379–1390, 1999.
- [11] W. E. Dixon, M. S. de Queiroz, D. M. Dawson, and T. J. Flynn, "Adaptive tracking and regulation control of a wheeled mobile robot with controller/update law modularity," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 1, pp. 138–147, Jan. 2004.
- [12] M. Krstic and P. V. Kokotovic, "Adaptive nonlinear design with controller-identifier separation and swapping," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 426–440, Mar. 1995.
- [13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [14] H. D. Patino, R. Carelli, and B. R. Kuchen, "Neural networks for advanced control of robot manipulators," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 343–354, Mar. 2002.
- [15] M. Chen, S. S. Ge, and B. V. E. How, "Robust adaptive neural network control for a class of uncertain MIMO nonlinear systems with input nonlinearities," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 796–812, May 2010.
- [16] R. Kamalapurkar, J. A. Rosenfeld, A. Parikh, A. R. Teel, and W. E. Dixon, "Invariance-like results for nonautonomous switched systems," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 614–627, Feb. 2019.
- [17] P. M. Patre, K. Dupree, W. MacKunis, and W. E. Dixon, "A new class of modular adaptive controllers, part II: Neural network extension for non-LP systems," in *Proc. Amer. Control Conf.*, Seattle, WA, USA, Jun. 2008, pp. 1214–1219.
- [18] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 73–82, Jan. 1987.
- [19] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, Sep. 1994.
- [20] N. Fischer, R. Kamalapurkar, and W. E. Dixon, "LaSalle-Yoshizawa corollaries for nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2333–2338, Sep. 2013.
- [21] R. Kamalapurkar, J. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, Dec. 2016.
- [22] P. J. Werbos, "Back propagation: Past and future," in *Proc. Int. Conf. Neural Netw.*, vol. 1, 1989, pp. 1343–1353.
- [23] D. O. Hebb, *The Organization of Behavior*. New York, NY, USA: Wiley, 1949.
- [24] K.-A. Toh, "Analytic network learning," 2018. [Online]. Available: [arXiv:1811.08227](https://arxiv.org/abs/1811.08227).
- [25] H.-T. Nguyen, C. C. Cheah, and K.-A. Toh, "An analytic layer-wise deep learning framework with applications to robotics," 2021. [Online]. Available: [arXiv:2102.03705](https://arxiv.org/abs/2102.03705).

Exponential Stability With RISE Controllers

Omkar Sudhir Patil^{ID}, Axton Isaly, Bin Xian^{ID}, *Senior Member, IEEE*,
and Warren E. Dixon^{ID}, *Fellow, IEEE*

Abstract—A class of continuous robust controllers termed Robust Integral of the Sign of the Error (RISE) have been published over the past two decades as a means to yield asymptotic tracking error convergence and asymptotic identification of time-varying uncertainties, for classes of nonlinear systems that are subject to sufficiently smooth bounded exogenous disturbances and/or modeling uncertainties. Despite the wide application of RISE-based techniques, an open question that has eluded researchers during this time-span is whether the asymptotic tracking error convergence is also uniform or exponential. This question has remained open due to certain limitations in the traditional construction of a Lyapunov function for RISE-based error systems. In this letter, new insights on the construction of a Lyapunov function are used that result in an exponential stability result for RISE-based controllers. As an outcome of this breakthrough, the inherent learning capability of RISE-based controllers is shown to yield exponential identification of state-dependent disturbances/uncertainty.

Index Terms—Robust control, Lyapunov methods, nonlinear control systems.

I. INTRODUCTION

A CLASS of continuous robust controllers termed Robust Integral of the Sign of the Error (RISE) have been published over the past two decades [1]–[14] as a means to yield asymptotic tracking error convergence and asymptotic identification of time-varying uncertainties, for classes of nonlinear systems that are subject to sufficiently smooth bounded exogenous disturbances and/or modeling uncertainties. RISE-based methods have been used for a wide variety of applications involving control [4]–[20], estimation [2], [21], [22], and optimization [23]. Despite the wide application of RISE-based

techniques, an open question that has eluded researchers during this time-span is whether the asymptotic tracking error convergence is also uniform or exponential. This question has remained open due to certain limitations in the traditional construction of a Lyapunov function for RISE-based error systems.

The traditional analysis methodology for a RISE-based error system involves a Lyapunov-based approach, where the candidate Lyapunov function (denoted by V_L) includes a P-function (denoted by P) in addition to a typical sum of norm squared error terms. The P-function is designed by selecting \dot{P} to cancel disturbance-based terms in \dot{V}_L and is the essential analysis and design tool to enable asymptotic convergence (instead of uniformly ultimately bounded tracking) despite the presence of a disturbance term that is only upper bounded by a constant. Previous results, including the current result, determine P as a function of the initial conditions of the system that is proven to be non-negative under certain gain conditions. Evaluating \dot{V}_L along the closed-loop error trajectories yields a negative semi-definite \dot{V}_L . Then the extension of the LaSalle-Yoshizawa theorem for nonsmooth systems in [24] is invoked to prove asymptotic tracking error convergence. Since the LaSalle-Yoshizawa theorem is based on Barbalat's lemma, the traditional analysis methodology does not guarantee uniform tracking error convergence, and the non-strictness of V_L precluded exponential stability of the closed-loop error system's origin.

To prove exponential stability, it would be sufficient to select a positive-definite V_L such that $\dot{V}_L \leq -\lambda_L V_L$ for almost all time, with some positive constant λ_L . Then exponential stability can be established using the comparison principle. Such a Lyapunov function is developed in [14], which is the only known RISE-based exponential tracking result. The result in [14] was developed for a specific application under the assumption that the first and second derivatives of the uncertainty are bounded by known constants. However, RISE-based controllers have been applied to a broader set of applications where this assumed bound would not hold. For example, in results like [8], [10] and [11] that involve dynamic compensator-based auxiliary control terms, the first or second derivative of the uncertainty have bounds that are state-dependent. It is not clear how the analysis approach in [14] can be extended for such cases.

In this letter, a novel P-function design is developed that results in a strict Lyapunov function. The new analysis results

Manuscript received August 17, 2021; revised October 21, 2021; accepted November 8, 2021. Date of publication November 10, 2021; date of current version November 19, 2021. This work was supported in part by NSF under Award 1762829; in part by the Office of Naval Research under Grant N00014-13-1-0151; and in part by Air Force Office of Scientific Research (AFOSR) under Award FA9550-18-1-0109 and Award FA9550-19-1-0169. Recommended by Senior Editor C. Prieur. (Corresponding author: Omkar Sudhir Patil.)

Omkar Sudhir Patil, Axton Isaly, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: patilomkarsudhir@ufl.edu; axtonisaly1013@ufl.edu; wdixon@ufl.edu).

Bin Xian is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: xbin@tju.edu.cn).

Digital Object Identifier 10.1109/LCSYS.2021.3127134

in exponential stability of the closed-loop error system's origin using a comparison theorem-based argument. The novel P-function is shown to be non-negative under certain gain conditions by examining the analytically derived solution to the dynamics in \dot{P} . Unlike the analysis approach in [14], the developed P-function can be easily modified for various bounds on the first and second derivatives of uncertainty. To rule out the existence of extra solutions for P that could be potentially negative over some time interval, the derived solution for P is shown to be unique corresponding to a given closed-loop error trajectory. Additionally, solution-dependent arguments are employed to show the sign of the error term is integrable, and $\dot{V}_L \leq -\lambda_L V_L$ for almost all time, which involves showing that the set of time-instants where $\dot{V}_L \leq -\lambda_L V_L$ may not be true have Lebesgue measure zero. Furthermore, the disturbance/uncertainty is shown to be estimated with exponential convergence of the disturbance identification error, while prior results only indicated asymptotic convergence.

II. CONTROL DESIGN

A. Preliminaries

A function $y : \mathcal{I}_y \rightarrow \mathbb{R}^n$ is called a *Filippov solution* of $\dot{y} = h(y, t)$ on the interval $\mathcal{I}_y \subseteq \mathbb{R}_{\geq 0}$, given some Lebesgue measurable and locally essentially bounded function $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$, if y is absolutely continuous on \mathcal{I}_y , and $\dot{y} \in K[h](y, t)$ for almost all $t \in \mathcal{I}_y$, where $K[\cdot]$ denotes the Filippov set-valued map defined in [25, eq. (2b)]. A solution is called *complete* if \mathcal{I}_y is unbounded. A solution $y_2 : [t_0, t_2) \rightarrow \mathbb{R}^n$ to $\dot{y} = h(y, t)$ is called a *proper right extension* of a solution $y_1 : [t_0, t_1) \rightarrow \mathbb{R}^n$ to $\dot{y} = h(y, t)$ if $t_2 > t_1$ and $y_2(t) = y_1(t)$, $\forall t \in [t_0, t_1)$. A solution to $\dot{y} = h(y, t)$ is called *maximal* if it does not have a proper right extension which is also a solution to $\dot{y} = h(y, t)$. If a solution is maximal and if the closure of its range, $\overline{\{y(t) \in \mathbb{R}^n | t \in \mathcal{I}_y\}}$, is compact, then the solution is called *precompact*. The space of continuous functions with continuous first m derivatives is denoted by C^m . The space of essentially bounded Lebesgue measurable functions is denoted by \mathcal{L}_∞ , and $\|\cdot\|_p$ denotes the p -norm, where the subscript is suppressed when $p = 2$. The notation $[a; b]$ denotes the column vector $[a^T \ b^T]^T$.

B. Control Objective

Consider a control affine system with the nonlinear dynamics

$$\dot{x} = d(x, v, t) + u, \quad (1)$$

where $t \in \mathbb{R}_{\geq 0}$ denotes time, $x : \mathcal{I} \rightarrow \mathbb{R}^n$ denotes a Filippov solution¹ to (1), with the interval of existence $\mathcal{I} = [t_0, t_1)$ for some $t_0, t_1 \in \mathbb{R}_{\geq 0}$ s.t. $t_1 > t_0$, $v : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ denotes an auxiliary function representing some external dynamic

¹We consider Filippov solutions instead of classical solutions to facilitate a nonsmooth control design. Alternatively, Krasovskii solutions can also be considered. Generalized solutions such as Filippov or Krasovskii solutions are guaranteed to exist for nonsmooth systems with Lebesgue measurable and locally essentially bounded right-hand-sides [26, Proposition 3], whereas classical solutions might not exist.

compensator-based terms (e.g., adaptive feedforward terms, observer-based terms), $d : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ represents \mathcal{C}^2 modeling uncertainty in the system, and $u : \mathcal{I} \rightarrow \mathbb{R}^n$ represents the control input. Let $\dot{d}(x, \dot{x}, v, \dot{v}, t) \triangleq \nabla d^T(x, v, t)[\dot{x}; \dot{v}; 1]$ and $\ddot{d}(x, \dot{x}, \ddot{x}, v, \dot{v}, \ddot{v}, t) = [\dot{x}; \dot{v}; 1]^T \nabla^2 d(x, v, t)[\dot{x}; \dot{v}; 1] + \nabla d^T(x, v, t)[\ddot{x}; \ddot{v}; 0]$, respectively, where ∇ and ∇^2 denote the gradient and Hessian operators, respectively. It is assumed that for each $(a, b, p, v, w, s) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^m$, the mappings $t \mapsto d(a, v, t)$, $t \mapsto \dot{d}(a, b, v, w, t)$, and $t \mapsto \ddot{d}(a, b, p, v, w, s, t)$ are bounded. The objective is to design a controller such that the state tracks a smooth bounded reference trajectory. The objective is quantified by defining the tracking error

$$e \triangleq x - x_d, \quad (2)$$

where $x_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ is a \mathcal{C}^2 reference trajectory such that $x_d, \dot{x}_d \in \mathcal{L}_\infty$.

C. Control Law Development

To facilitate the subsequent analysis, a filtered tracking error $r : \mathcal{I} \rightarrow \mathbb{R}^n$ is defined as $r \triangleq d(x, v, t) + u - \dot{x}_d + \alpha e$, where $\alpha \in \mathbb{R}_{>0}$ is a constant control gain. To facilitate the subsequent analysis, the dynamics in terms of \dot{e} can be rewritten using (1) and (2) as

$$\dot{e} = r - \alpha e. \quad (3)$$

Let $z : \mathcal{I} \rightarrow \mathbb{R}^{2n}$ denote the augmented tracking error, $z \triangleq [e^T \ r^T]^T$. From the subsequent stability analysis, a continuous RISE control input is designed as [3]

$$u \triangleq \dot{x}_d - \alpha e - \hat{d}, \quad (4)$$

where $\hat{d} : \mathcal{I} \rightarrow \mathbb{R}^n$ is an auxiliary term designed as a Filippov solution² to

$$\dot{\hat{d}} = kr + e + \beta \text{sgn}(e), \quad (5)$$

given any user-selected $\hat{d}(t_0) \in \mathbb{R}^n$. In (5), $k, \beta \in \mathbb{R}_{>0}$ are constant control gains. Using (1)-(4) yields

$$r = d(x, v, t) - \hat{d}. \quad (6)$$

It follows from (5) and (6) that r is a Filippov solution to the closed-loop error system

$$\dot{r} = \tilde{N} + N_B - kr - e - \beta \text{sgn}(e), \quad (7)$$

where $\tilde{N} \triangleq \dot{d}(x, \dot{x}, v, \dot{v}, t) - \dot{d}(x_d, \dot{x}_d, v, \dot{v}, t)$ and $N_B \triangleq \dot{d}(x_d, \dot{x}_d, v, \dot{v}, t)$.

Assumption 1: The function v is a solution to some external dynamics such that there exist known constants, $\eta_1, \eta_2, \eta_3, \eta_4 \in \mathbb{R}_{\geq 0}$, and a known strictly increasing function, $\rho_{21} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$, such that $\|v\| \leq \eta_1$, $\|\dot{v}\| \leq \eta_2$, and $\|\ddot{v}\| \leq \eta_3 + \eta_4 \|z\| + \rho_{21}(\|z\|)\|z\|$.

Then, there exist known constants $\gamma_1, \gamma_3, \gamma_4 \in \mathbb{R}_{\geq 0}$ and a known strictly increasing function $\rho_{21} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ such that $\|N_B\| \leq \gamma_1$ and $\|\dot{N}_B\| \leq \gamma_3 + \gamma_4 \|z\| + \rho_{21}(\|z\|)\|z\|$, $\forall t \in \mathbb{R}_{\geq 0}$.

²Since r may not be commonly available, $\hat{d}(t)$ is evaluated using $\hat{d}(t) = \hat{d}(t_0) + ke(t) - ke(t_0) + \int_{t_0}^t ((k\alpha + 1)e(\tau) + \beta \text{sgn}(e(\tau)))d\tau$ for closed-loop implementation. Note that $\beta \text{sgn}(e(\cdot))$ is Riemann integrable on $[t_0, t]$, $\forall t \in \mathcal{I}$ according to Lemma 2.

Additionally, since v is bounded and $t \mapsto \dot{d}(a, b, v, w, t)$ is bounded for each $(a, b, v, w) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$, $\|\tilde{N}\| \leq \gamma_2 \|z\| + \rho_1 (\|z\|) \|z\|$, $\forall t \in \mathbb{R}_{\geq 0}$, according to the Mean Value Theorem-based inequality in [27, Lemma 5], where $\gamma_2 \in \mathbb{R}_{\geq 0}$ is a known constant, and $\rho_1 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a known strictly increasing function. Note that the type of state-dependent bounds considered in Assumption 1 are general and often required in various applications where the RISE method is used (e.g., [8], [10] and [11]), typically as a consequence of augmenting adaptive feedforward controllers with a RISE term. In the case where v represents adaptive feedforward terms, the developed approach offers modularity of design in the sense that \dot{d} and v can be designed independently, as long as v satisfies Assumption 1. The following example illustrates a type of system satisfying Assumption 1.

Example 1: Consider a dynamic neural network given by

$$\dot{v} = \text{proj}\{W^T \sigma(V^T x), v\}, \quad (8)$$

where $\text{proj}\{\cdot, \cdot\}$ denotes the smooth projection operator in [28] that guarantees $\|v\| \leq \eta_1$, $\sigma : \mathbb{R}^L \rightarrow \mathbb{R}^L$ denotes a globally bounded continuous activation function, and $W \in \mathbb{R}^{m \times L}$ and $V \in \mathbb{R}^{L \times n}$ are constant³ matrices of outer and inner-layer weights, respectively. Using [28, Property 3] and the fact that $\sigma(\cdot)$ is globally bounded, \dot{v} can be bounded by a constant, i.e., $\|\dot{v}\| \leq \eta_2$. Taking the time-derivative of \dot{v} yields $\ddot{v} = \frac{d}{dt}(\text{proj}\{W^T \sigma(V^T x), v\}) = \frac{\partial}{\partial y}(\text{proj}\{y, v\})|_{y=W^T \sigma(V^T x)} \frac{d}{dt}(W^T \sigma(V^T x)) + \frac{\partial}{\partial y}(\text{proj}\{W^T \sigma(V^T x), y\})|_{y=v} \dot{v}$. Based on the structure of the projection operator in [28, eq. (7)], the terms $\frac{\partial}{\partial y}(\text{proj}\{y, v\})|_{y=W^T \sigma(V^T x)}$ and $\frac{\partial}{\partial y}(\text{proj}\{W^T \sigma(V^T x), y\})|_{y=v}$ can be bounded by some known functions of x . Additionally, based on the right-hand-side of (3), the term $\frac{d}{dt}(W^T \sigma(V^T x)) = W^T \frac{\partial}{\partial y} \sigma(y)|_{y=V^T x} V^T \dot{x} = W^T \frac{\partial}{\partial y} \sigma(y)|_{y=V^T x} V^T (\dot{x} + r - \alpha e)$ can be bounded by some known continuous function of z . Therefore, \ddot{v} can be bounded as $\|\ddot{v}\| \leq \eta_3 + \eta_4 \|z\| + \rho_{21} (\|z\|) \|z\|$. Thus, the dynamic neural network in (8) satisfies Assumption 1.

The structure of the closed-loop error system in (7) may appear similar to a higher order sliding-mode design (see [29]); however, there are some remarkable differences to highlight. Specifically, the $\beta \text{sgn}(e)$ term in (7) would need to be $\beta \text{sgn}(r)$ to facilitate the analysis for a standard continuous higher-order sliding-mode design. Since sensor measurements for the highest order derivative (e.g., \dot{e} or r) may not be available for feedback, the controller in (4) is designed to depend only on state measurements. Additionally, the closed-loop error system in (3) and (7) is also different from a super-twisting system, since (3) would require an additional $-|e|^{1/2} \text{sgn}(e)$ term to facilitate a super-twisting design, which needs a different analysis approach [30].

We now present some supporting lemmas which facilitate the subsequent analysis. Proofs of all lemmas can be found in the Appendix.

Lemma 1: Given some Filippov solutions, e and r , to (3) and (7), respectively, the set of time-instants $T \triangleq \{t \in \mathcal{I} | \exists i \in$

$\{1, 2, \dots, n\} \text{ s.t. } e_i(t) = 0 \wedge r_i(t) \neq 0\}$ has Lebesgue measure zero, where e_i and r_i denote the i^{th} element of e and r , respectively.

Lemma 2: Given some Filippov solution, e , to (3), $\text{sgn}(e(\cdot))$ is Riemann integrable on $[t_0, t_1]$, $\forall t_1 \in \mathcal{I}$.

III. STABILITY ANALYSIS

Following the development in Section II, every Filippov solution to (1) and (5) with the controller in (4) corresponds to a Filippov solution of the transformed system in (3) and (7). Additionally, a P-function is introduced to facilitate the construction of a candidate Lyapunov function for analyzing the stability and convergence properties of z . The P-function is denoted by $P : \mathcal{I} \rightarrow \mathbb{R}$ and is defined as a Filippov solution to

$$\dot{P} = -\lambda_P P - L, \quad (9)$$

where $\lambda_P \in \mathbb{R}_{>0}$ is an auxiliary constant, and

$$L \triangleq r^T N_B - r^T \beta \text{sgn}(e) - (\gamma_4 + \rho_2(\|z\|)) \|z\| \|e\|_1, \quad (10)$$

where $\|\cdot\|_1$ denotes the 1-norm, and

$$P(t_0) = 0. \quad (11)$$

The analytical solution to (9) is derived in Lemma 3. To facilitate the inclusion of the P-function in the candidate Lyapunov function, P is designed to be non-negative under certain gain conditions as described in Lemma 4.

Lemma 3: Given some Filippov solutions, e and r , to (3) and (7), respectively,

$$P = \beta \|e\|_1 - e^T N_B + e^{-\lambda_P t} * ((\alpha - \lambda_P)(\beta \|e\|_1 - e^T N_B) + e^T \dot{N}_B) + e^{-\lambda_P t} * ((\gamma_4 + \rho_2(\|z\|)) \|z\| \|e\|_1), \quad (12)$$

is the unique Filippov solution to the differential equation in (9) initialized according to (11), where $*$ denotes the convolution operator, i.e., $p(t) * q(t) = \int_{t_0}^t p(t-\tau) q(\tau) d\tau$, for any given $p, q : [t_0, \infty) \rightarrow \mathbb{R}$.

Lemma 4: Given any pair of Filippov solutions, e and r , to (3) and (7), respectively, provided that P is initialized according to (11), and the gain conditions

$$\alpha > \lambda_P, \quad (13)$$

$$\beta > \gamma_1 + \frac{\gamma_3}{\alpha - \lambda_P}, \quad (14)$$

are satisfied, $P(t) \geq 0$, $\forall t \in \mathcal{I}$, where the gains α and β are introduced in (3) and (5), respectively, and γ_1, γ_4 are introduced in Assumption 1.

Let $\psi \triangleq [e^T \ r^T \ P]^T$, and $\dot{\psi} = g(\psi, t)$ denote the differential equations in (3), (7) and (9), where $g : \mathbb{R}^{2n+1} \times [t_0, \infty) \rightarrow \mathbb{R}^{2n+1}$ is Lebesgue measurable and locally essentially bounded (i.e., bounded on a neighborhood of every point, excluding sets of measure zero), since it is continuous except in the set $\{(\psi, t) \in \mathbb{R}^{2n+1} \times [t_0, \infty) | e = 0\}$. To facilitate the stability analysis, let $V_L : \mathbb{R}^{2n+1} \rightarrow \mathbb{R}_{\geq 0}$ be defined as

$$V_L(\psi) \triangleq \frac{1}{2} e^T e + \frac{1}{2} r^T r + P. \quad (15)$$

Let $c \triangleq \min\{k - \gamma_2 - n\gamma_4, \alpha - \gamma_2 - n\gamma_4, \frac{\lambda_P}{2}\}$, $\rho(\cdot) \triangleq \rho_1(\cdot) + n\rho_2(\cdot)$, and consider the regions, $\mathcal{D} \triangleq \{\sigma \in \mathbb{R}^{2n+1} | V_L(\sigma) <$

³See [8] for continuous adaptive weight updates.

$\frac{(\rho^{-1}(c-\lambda_V))^2}{2}\}$ and $\mathcal{S} \triangleq \{\sigma \in \mathbb{R}^{2n} \mid \|\sigma\| < c - \lambda_V\}$, where $\lambda_V \in \mathbb{R}_{>0}$ is a user-defined constant.

Theorem 1: Given any initial condition $z(t_0) \in \mathbb{R}^{2n}$, every maximal solution to (3), (7), and (9) with $P(t_0)$ initialized according to (11) is complete, and the zero solution to (3) and (7), $(e(t), r(t)) \equiv (0, 0)$, is semi-globally exponentially stable in the sense that $\|z(t)\| \leq \|z(t_0)\| \exp(-\lambda_V(t - t_0))$, $\forall (z(t_0), t) \in \mathbb{R}^{2n} \times [t_0, \infty)$, provided that the gains α, β, k and λ_P are selected according to the gain conditions in (13), (14), and

$$c > \lambda_V + \|z(t_0)\|. \quad (16)$$

Proof: The existence of a Filippov solution, $\psi : \mathcal{I} \rightarrow \mathbb{R}^{2n+1}$, to $\dot{\psi} = g(\psi, t)$ is guaranteed⁴ by [26, Proposition 3]. The time-derivative of V_L along ψ , starting from the specified initial conditions, exists a.e., and $\dot{V}_L(\psi, t) \stackrel{a.e.}{=} \tilde{V}_L(\psi, t)$ [31, eqs. (12) and (13)], where the notation (\cdot) implies that the relation holds for almost all time $t \in \mathcal{I}$, and

$$\begin{aligned} \tilde{V}_L(\psi, t) &= \bigcap_{\xi \in \partial V_L(\psi)} \xi^T K[g](\psi, t) \\ &= \nabla V_L^T K[g](\psi, t) \\ &= [z^T \ 1]^T K[g](\psi, t). \end{aligned} \quad (17)$$

In (17), $\partial V_L(\psi)$ denotes Clarke's generalized gradient [31, eq. (7)], and $K[\cdot]$ is defined in [25, eq. (2b)]. Since $\psi \mapsto V_L(\psi)$ is continuously differentiable, $\partial V_L = \{\nabla V_L\}$ using [26, Proposition 6]. Substituting (3), (7), and (9) into (17), utilizing $\|\cdot\|_1 \leq \sqrt{n}\|\cdot\|$, and applying Young's inequality on $\|z\|\|e\|_1$ yields

$$\begin{aligned} \tilde{V}_L &= r^T(\tilde{N} + N_B - kr - e - \beta K[\text{sgn}](e)) \\ &\quad + e^T(r - \alpha e) - \lambda_P P - r^T N_B \\ &\quad + r^T \beta K[\text{sgn}](e) + (\gamma_4 + \rho_2(\|z\|))\|z\|\|e\|_1 \\ &\leq -k\|r\|^2 - \alpha\|e\|^2 + (\gamma_2 + \rho_1(\|z\|))\|r\|\|z\| \\ &\quad - \lambda_P P + (\gamma_4 + \rho_2(\|z\|))\|z\|\|e\|_1 \\ &\leq -2(c - \rho(\|z\|))V_L \\ &\leq -2\left(c - \rho\left(\sqrt{2V_L}\right)\right)V_L, \end{aligned} \quad (18)$$

for almost all $t \in \mathcal{I}$, where c and ρ are introduced before the theorem statement, the term $t \mapsto r^T(t)\beta K[\text{sgn}](e(t))$ is set-valued only for the set of time instants $T = \{t \in [t_0, \infty) \mid \exists i \in \{1, 2, \dots, n\} \text{ s.t. } e_i(t) = 0 \wedge r_i(t) \neq 0\}$. According to Lemma 1, the set T has Lebesgue measure zero. It follows from (18) that V_L is non-increasing along all trajectories initialized such that $\psi(t_0) \in \mathcal{D}$. Selecting c according to (16) and using (11) yields $c > \lambda_V + \|z(t_0)\| = \lambda_V + \rho(\sqrt{2V_L(\psi(t_0))})$. Then, $V_L(\psi(t))$ is non-increasing, implying $\psi(t) \in \mathcal{D}$, $\forall t \in \mathcal{I}$. It follows from (16) and (18) that \dot{V}_L can be upper-bounded as

$$\dot{V}_L \leq -2\lambda_V V_L, \quad (19)$$

⁴The solution ψ may not be unique; however, P is unique according to Lemma 3 for a given pair (e, r) . Moreover, the results in this letter are applicable to all the trajectories even when ψ is non-unique, since we consider a generalized Filippov solution in the analysis.

for almost all $t \in \mathcal{I}$. Using the comparison principle [32, Lemma 4.4] in (19) yields

$$V_L(\psi(t)) \leq V_L(\psi(t_0)) \exp(-2\lambda_V(t - t_0)), \quad (20)$$

$\forall (\psi(t_0), t) \in \mathcal{D} \times \mathcal{I}$. Since $(\psi, t) \mapsto K[g](\psi, t)$ is locally bounded over $\mathbb{R}^{2n+1} \times [t_0, \infty)$ and (20) implies that ψ is precompact, then [33, Lemma 3.3 and Remark 3.4] can be invoked to show that every maximal solution ψ with $P(t_0)$ initialized according to (11) is complete, i.e., $\mathcal{I} = [t_0, \infty)$. Using the definition of z , (20) and Lemma 4, $V_L(\psi(t)) = \frac{1}{2}\|z(t)\|^2 + P(t) \geq \frac{1}{2}\|z(t)\|^2$, $\forall t \in [t_0, \infty)$. Therefore,

$$\|z(t)\| \leq \sqrt{2V_L(\psi(t))}, \quad \forall t \in [t_0, \infty). \quad (21)$$

Using (20) and (21), $\|z(t)\|$ can further be upper-bounded as

$$\|z(t)\| \leq \sqrt{2V_L(\psi(t_0))} \exp(-\lambda_V(t - t_0)), \quad (22)$$

$\forall (\psi(t_0), t) \in \mathcal{D} \times [t_0, \infty)$. Moreover, substituting (11) in the expression for $V_L(\psi(t_0))$ yields $V_L(\psi(t_0)) = \frac{1}{2}\|z(t_0)\|^2$. Consequently, $\psi(t_0) \in \mathcal{D}$ implies $z(t_0) \in \mathcal{S}$. Using (22) yields

$$\|z(t)\| \leq \|z(t_0)\| \exp(-\lambda_V(t - t_0)), \quad (23)$$

$\forall (z(t_0), t) \in \mathcal{S} \times [t_0, \infty)$, implying the zero solution to (3) and (7), $(e(t), r(t)) \equiv (0, 0)$, is semi-globally exponentially stable. Note that the exponential stability result is semi-global (cf., [34, Remark 2]) because the size of the set \mathcal{S} can be arbitrarily increased using (16) to include any $z(t_0) \in \mathbb{R}^{2n}$. Moreover, $x \in \mathcal{L}_\infty$ since $e, x_d \in \mathcal{L}_\infty$. Since $d \in \mathcal{L}_\infty$ by Assumption 1, it follows from (6) that $\mu \in \mathcal{L}_\infty$. Therefore, since all the terms on the right hand side of (4) are bounded and continuous, $u \in \mathcal{L}_\infty$ and is continuous. Moreover, since $e(t) = e(t_0) + \int_{t_0}^t (r(\tau) - \alpha e(\tau)) d\tau$ for all $t \in [t_0, \infty)$ using (3), the continuity of $r - \alpha e$ implies that the Filippov solution e is also continuously differentiable. ■

Remark 1: The relation in (6) indicates that r is the estimation error between the RISE term $\hat{d}(t)$ and the uncertainty $d(x, v, t)$. Therefore, (23) implies that the RISE term is an exponentially convergent estimator of the uncertainty, i.e., $\hat{d}(t) \rightarrow d(x, v, t)$ with a uniform and exponential convergence as $t \rightarrow \infty$.

Remark 2: For the special case when $\dot{d}(t)$ and $\ddot{d}(t)$ are bounded by known constants, the analysis approach in [14] can also be considered.

Remark 3: The exponential stability result is global when the bounds on \tilde{N} and \dot{N}_B are linear in $\|z\|$, i.e., $\rho_1 = \rho_2 = 0$.

IV. CONCLUSION

In this letter, new insights on the construction of a P-function are used to yield exponential stability with RISE-based controllers. As an outcome of this breakthrough, the inherent learning capability of RISE-based controllers is shown to yield exponential identification of disturbances/uncertainty, as compared to all previous asymptotic results. Future work could involve extension of the proposed stability analysis methodology for RISE-based error systems with sensor noise, and delays in input and state measurements.

APPENDIX

Proof of Lemma 1: The set T can also be represented as $T = \{t \in \mathcal{I} | \exists i \in \{1, 2, \dots, n\} \text{ s.t. } e_i(t) = 0 \wedge r_i(t) - \alpha e_i(t) \neq 0\} = \{t \in \mathcal{I} | \exists i \in \{1, 2, \dots, n\} \text{ s.t. } e_i(t) = 0 \wedge \dot{e}_i(t) \neq 0\}$ to facilitate the subsequent analysis using the dynamics in (7). Select $a \in T$, which yields $e_i(a) = 0$ using the definition of T . Given $\dot{e}_i(a) \neq 0$, we can assume without loss of generality that $\dot{e}_i(a) > 0$; the proof easily extends for $\dot{e}_i(a) < 0$. Since e and r are absolutely continuous, it follows from (3) that \dot{e} is continuous. Given $\dot{e}_i(a) > 0$ and continuity of \dot{e} , there exists $\delta > 0$ such that $\dot{e}_i(t) > 0$ for all $t \in (a - \delta, a + \delta)$. Based on $\dot{e}_i(t) > 0$, it follows that $\int_a^t \dot{e}_i(\tau) d\tau > 0$ for all $t \in (a, a + \delta)$. Then, using $e_i(a) = 0$ yields $\int_a^t \dot{e}_i(\tau) d\tau = e_i(t) - e_i(a) = e_i(t) > 0$. Similarly, $-\int_t^a \dot{e}_i(\tau) d\tau = e_i(t) - e_i(a) = e_i(t) < 0$ for all $t \in (a - \delta, a)$, implying $e_i(t) \neq 0$ for all $t \in (a - \delta, a + \delta) \setminus \{a\}$. If more than one component has $e_i(a) = 0$, we select the intersection of each neighborhood found above, represented by $U(a)$. Therefore, there exists a neighborhood, $U(a)$, for any $a \in T$, s.t. $e_i(t) \neq 0$ for all time-instants $t \in U(a) \setminus \{a\}$ and $i \in \{1, 2, \dots, n\}$. When $t \in T$, $e_i(t) = 0$ for some $i \in \{1, 2, \dots, n\}$, which implies $U(a) \cap T = \{a\}$ for all $a \in T$; therefore, T is discrete and consequently has measure zero. ■

Proof of Lemma 2: The function $\text{sgn}(e(\cdot))$ is discontinuous only at time-instants where it changes sign, i.e., the set $\{t \in \mathcal{I} | \exists i \in \{1, 2, \dots, n\} \text{ s.t. } e_i(t) = 0 \wedge \dot{e}_i(t) \neq 0\} = \{t \in \mathcal{I} | \exists i \in \{1, 2, \dots, n\} \text{ s.t. } e_i(t) = 0 \wedge r_i(t) - \alpha e_i(t) \neq 0\} = T$. Since T has Lebesgue measure zero according to Lemma 1, $\text{sgn}(e(\cdot))$ is continuous a.e., implying it is Riemann integrable [35, Th. 11.33] on $[t_0, t_1]$, $\forall t_1 \in \mathcal{I}$. ■

Proof of Lemma 3: The right hand side (RHS) of (12) is almost everywhere (a.e.) differentiable with respect to time, because every term on the RHS is absolutely continuous, including $\|e\|_1$, since $\|\cdot\|_1$ is globally Lipschitz and e is absolutely continuous. The time-derivative of $\|e\|_1$, whenever it exists, is $\dot{e}^T \text{sgn}(e)$, using the chain rule. Therefore, taking the time-derivative of both the sides of (12) at points where P is differentiable yields

$$\begin{aligned} \dot{P} \stackrel{a.e.}{=} & \beta \dot{e}^T \text{sgn}(e) - \dot{e}^T N_B - e^T \dot{N}_B \\ & + \frac{d}{dt} (e^{-\lambda_P t} * ((\alpha - \lambda_P)(\beta \|e\|_1 - e^T N_B) + e^T \dot{N}_B)) \\ & + \frac{d}{dt} (e^{-\lambda_P t} * ((\gamma_4 + \rho_2(\|z\|))\|z\| \|e\|_1)). \end{aligned} \quad (24)$$

Based on the Leibniz rule, for any given $q: [t_0, \infty) \rightarrow \mathbb{R}$, the function $e^{-\lambda_P t}$ satisfies $\frac{d}{dt} (e^{-\lambda_P t} * q) = \frac{d}{dt} (\int_{t_0}^t e^{-\lambda_P(t-\tau)} q(\tau) d\tau) = q(t) - \lambda_P \int_{t_0}^t e^{-\lambda_P(t-\tau)} q(\tau) d\tau = -\lambda_P e^{-\lambda_P t} * q + q$. Additionally, $L = \dot{e}^T N_B + \alpha e^T N_B - \beta \dot{e}^T \text{sgn}(e) - \alpha \beta \|e\|_1 - (\gamma_4 + \rho_2(\|z\|))\|z\| \|e\|_1$ is obtained after substituting (3) into (10). Therefore, the expression for \dot{P} in (24) can be rewritten as

$$\begin{aligned} \dot{P} \stackrel{a.e.}{=} & -\lambda_P e^{-\lambda_P t} * ((\gamma_4 + \rho_2(\|z\|))\|z\| \|e\|_1) \\ & - \lambda_P e^{-\lambda_P t} * ((\alpha - \lambda_P)(\beta \|e\|_1 - e^T N_B) + e^T \dot{N}_B) \\ & - \lambda_P \beta \|e\|_1 + \lambda_P e^T N_B + \beta \dot{e}^T \text{sgn}(e) \\ & - \dot{e}^T N_B + \alpha \beta \|e\|_1 - \alpha e^T N_B + (\gamma_4 + \rho_2(\|z\|))\|z\| \|e\|_1 \\ = & -\lambda_P P - L. \end{aligned} \quad (25)$$

Filippov's differential inclusion for (9) is given by

$$\dot{P} \in -\lambda_P P - K[L]. \quad (26)$$

To prove the uniqueness of the solution to (26), consider any two solutions with the same initial conditions, i.e., P_1 and P_2 , with $P_1(t_0) = P_2(t_0) = 0$, implying

$$\dot{P}_1 \stackrel{a.e.}{\in} -\lambda_P P_1 - K[L], \quad (27)$$

$$\dot{P}_2 \stackrel{a.e.}{\in} -\lambda_P P_2 - K[L]. \quad (28)$$

Based on (10), $t \mapsto K[L](\psi(t))$ is set-valued only when there exists some $i \in \{1, 2, \dots, n\}$ such that $t \rightarrow K[\text{sgn}](e_i(t))$ is set-valued and $r_i(t) \neq 0$. Using Lemma 1, $t \mapsto K[L](\psi(t))$ is set-valued only for a set of time-instants of measure zero. Therefore, defining $\Delta(t) = P_2(t) - P_1(t)$, and using (27) and (28) yields

$$\dot{\Delta} \stackrel{a.e.}{=} -\lambda_P \Delta, \quad (29)$$

with $\Delta(t_0) = 0$. Since $\Delta \equiv 0$ is an equilibrium point of (29), $\Delta(t_0) = 0$ implies $\|\Delta(t)\| = 0$, $\forall t \in \mathcal{I}$, therefore $P_1(t) = P_2(t)$, $\forall t \in [t_0, \infty)$, i.e., any two solutions are equal, implying the solution is unique. ■

Proof of Lemma 4: Using Holder's inequality and Assumption 1 yields lower bounds on $-e^T N_B$ and $e^T \dot{N}_B$,

$$-e^T N_B \geq -\|e\|_1 \|N_B\|_1 \geq -\gamma_1 \|e\|_1, \quad (30)$$

and

$$\begin{aligned} e^T \dot{N}_B & \geq -\|e\|_1 \|\dot{N}_B\|_1 \\ & \geq -(\gamma_3 + \gamma_4 \|z\| + \rho_2(\|z\|)\|z\|) \|e\|_1, \end{aligned} \quad (31)$$

$\forall t \in \mathcal{I}$. Substituting the bounds in (30) and (31) into the expression for P in (12) yields

$$\begin{aligned} P & \geq \beta \|e\|_1 - \gamma_1 \|e\|_1 \\ & + e^{-\lambda_P t} * (((\alpha - \lambda_P)(\beta - \gamma_1) - \gamma_3) \|e\|_1). \end{aligned} \quad (32)$$

Selecting $P(t_0)$ according to (11), and α and β according to the gain conditions (13) and (14) yields $P(t) \geq 0$, $\forall t \in \mathcal{I}$ using (32). ■

REFERENCES

- [1] Z. Qu and J. X. Xu, "Model-based learning controls and their comparisons using Lyapunov direct method," *Asian J. Control*, vol. 4, no. 1, pp. 99–110, Mar. 2002.
- [2] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, "Range identification for perspective vision systems," *IEEE Trans. Autom. Control*, vol. 48, no. 12, pp. 2232–2238, Dec. 2003.
- [3] B. Xian, D. M. Dawson, M. S. de Queiroz, and J. Chen, "A continuous asymptotic tracking control strategy for uncertain nonlinear systems," *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1206–1211, Jul. 2004.
- [4] Z. Cai, M. S. de Queiroz, and D. M. Dawson, "Robust adaptive asymptotic tracking of nonlinear systems with additive disturbance," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 524–529, Mar. 2006.
- [5] C. Makkar, G. Hu, W. G. Sawyer, and W. E. Dixon, "Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction," *IEEE Trans. Autom. Control*, vol. 52, no. 10, pp. 1988–1994, Oct. 2007.
- [6] B. Xian, M. S. de Queiroz, D. M. Dawson, and M. McIntyre, "A discontinuous output feedback controller and velocity observer for nonlinear mechanical systems," *Automatica*, vol. 40, no. 4, pp. 695–700, 2004.
- [7] P. M. Patre, W. Mackunis, C. Makkar, and W. E. Dixon, "Asymptotic tracking for systems with structured and unstructured uncertainties," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 2, pp. 373–379, Mar. 2008.

- [8] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, Oct. 2008.
- [9] P. Patre, W. Mackunis, M. Johnson, and W. E. Dixon, "Composite adaptive control for Euler-Lagrange systems with additive disturbances," *Automatica*, vol. 46, no. 1, pp. 140–147, 2010.
- [10] P. Patre, S. Bhasin, Z. D. Wilcox, and W. E. Dixon, "Composite adaptation for neural network-based controllers," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 944–950, Apr. 2010.
- [11] P. Patre, W. Mackunis, K. Dupree, and W. E. Dixon, "Modular adaptive control of uncertain Euler-Lagrange systems with additive disturbances," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 155–160, Jan. 2011.
- [12] N. Fischer, Z. Kan, R. Kamalapurkar, and W. E. Dixon, "Saturated RISE feedback control for a class of second-order nonlinear systems," *IEEE Trans. Autom. Control*, vol. 59, pp. 1094–1099, Apr. 2014.
- [13] N. Sharma, S. Bhasin, Q. Wang, and W. E. Dixon, "RISE-based adaptive control of a control affine uncertain nonlinear system with unknown state delays," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 255–259, Jan. 2012.
- [14] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2891–2902, May 2015.
- [15] N. Fischer, D. Hughes, P. Walters, E. Schwartz, and W. E. Dixon, "Nonlinear RISE-based control of an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 845–852, Aug. 2014.
- [16] J. Yao, W. Deng, and Z. Jiao, "RISE-based adaptive control of hydraulic systems with asymptotic tracking," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1524–1531, Jul. 2017.
- [17] Z. Yao, J. Yao, and W. Sun, "Adaptive RISE control of hydraulic systems with multilayer neural-networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8638–8647, Nov. 2019.
- [18] T. Dierks and S. Jagannathan, "Neural network control of mobile robot formations using RISE feedback," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 332–347, Apr. 2009.
- [19] B. Bidikli, E. Tatlicioglu, A. Bayrak, and E. Zergeroglu, "A new robust 'integral of sign of error' feedback controller with adaptive compensation gain," in *Proc. IEEE Conf. Decis. Control*, 2013, pp. 3782–3787.
- [20] B. Bidikli, E. Tatlicioglu, and E. Zergeroglu, "A self tuning RISE controller formulation," in *Proc. Amer. Control Conf.*, 2014, pp. 5608–5613.
- [21] S. Bhasin, R. Kamalapurkar, H. T. Dinh, and W. Dixon, "Robust identification-based state derivative estimation for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 187–192, Jan. 2013.
- [22] H. T. Dinh, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Dynamic neural network-based robust observers for uncertain nonlinear systems," *Neural Netw.*, vol. 60, pp. 44–52, Dec. 2014.
- [23] C. Sun, M. Ye, and G. Hu, "Distributed time-varying quadratic optimization for multiple agents under undirected graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3687–3694, Jul. 2017.
- [24] N. Fischer, R. Kamalapurkar, and W. E. Dixon, "LaSalle-Yoshizawa corollaries for nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2333–2338, Sep. 2013.
- [25] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 1, pp. 73–82, Jan. 1987.
- [26] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Syst.*, vol. 28, no. 3, pp. 36–73, Jun. 2008.
- [27] R. Kamalapurkar, J. A. Rosenfeld, J. Klotz, R. J. Downey, and W. E. Dixon, "Supporting lemmas for RISE-based control methods," 2014, [arXiv:1306.3432](https://arxiv.org/abs/1306.3432).
- [28] Z. Cai, M. S. de Queiroz, and D. M. Dawson, "A sufficiently smooth projection operator," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 135–139, Jan. 2006.
- [29] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, 1993.
- [30] T. Gonzalez, J. A. Moreno, and L. Fridman, "Variable gain super-twisting sliding mode control," *IEEE Trans. Autom. Control*, vol. 57, no. 8, pp. 2100–2105, Aug. 2012.
- [31] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, Sep. 1994.
- [32] Y. Lin, E. D. Sontag, and Y. Wang, "A smooth converse Lyapunov theorem for robust stability," *SIAM J. Control Optim.*, vol. 34, no. 1, pp. 124–160, 1996.
- [33] R. Kamalapurkar, W. E. Dixon, and A. Teel, "On reduction of differential inclusions and Lyapunov stability," *ESAIM Control Optim. Calc. Var.*, vol. 26, no. 24, pp. 1–16, 2020.
- [34] K. Y. Pettersen, "Lyapunov sufficient conditions for uniform semiglobal exponential stability," *Automatica*, vol. 78, pp. 97–102, Apr. 2017.
- [35] W. Rudin, *Principles of Mathematical Analysis*. New York, NY, USA: McGraw-Hill, 1976.

Lyapunov-Derived Control and Adaptive Update Laws for Inner and Outer Layer Weights of a Deep Neural Network

Omkar Sudhir Patil^{ID}, Graduate Student Member, IEEE, Duc M. Le^{ID}, Max L. Greene^{ID},
and Warren E. Dixon^{ID}, Fellow, IEEE

Abstract—Lyapunov-based real-time update laws are well-known for neural network (NN)-based adaptive controllers that control nonlinear dynamical systems using single-hidden-layer NNs. However, developing real-time weight update laws for deep NNs (DNNs) remains an open question. This letter presents the first result with Lyapunov-based real-time weight adaptation laws for each layer of a feedforward DNN-based control architecture, with stability guarantees. Additionally, the developed method allows nonsmooth activation functions to be used in the DNN to facilitate improved transient performance. A nonsmooth Lyapunov-based stability analysis proves global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs with leaky rectified linear unit (LReLU) and hyperbolic tangent activation functions to demonstrate the efficacy of the developed method.

Index Terms—Deep neural networks, deep learning, adaptive control, Lyapunov methods, nonlinear control systems.

I. INTRODUCTION

NEURAL networks (NNs) are universal function approximators that are capable of modeling continuous functions over a compact domain [1]. Although NNs with a single hidden-layer are capable of approximating general nonlinear functions, deep NNs (DNNs) provide improved performance [2]. Moreover, DNNs are exponentially more expressive than shallow NNs in terms of the total number of neurons required to achieve the same accuracy in function approximation [3].

Manuscript received September 14, 2021; revised November 24, 2021; accepted December 9, 2021. Date of publication December 14, 2021; date of current version December 22, 2021. This work was supported in part by NSF under Award 1762829; in part by the Office of Naval Research under Grant N00014-13-1-0151; and in part by the Air Force Office of Scientific Research (AFOSR) under Award FA9550-18-1-0109 and Award FA9550-19-1-0169. Recommended by Senior Editor C. Seatzu. (Corresponding author: Omkar Sudhir Patil.)

The authors are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: patilomkarsudhir@ufl.edu; ledan50@ufl.edu; maxgreene12@ufl.edu; wdixon@ufl.edu).

Digital Object Identifier 10.1109/LCSYS.2021.3134914

Motivated by recent advances in DNNs, researchers have explored the use of DNN-based control architectures. DNN-based techniques often employ optimization methods to train the DNN weights by minimizing a loss function over a training dataset [4]. Results in [5]–[7] utilize such offline DNN training techniques to approximate explicit model predictive control laws. However, such offline methods pose limitations since training typically requires large amounts of data, and the resulting feedforward terms are implemented as an open-loop approximator based on the offline training.

In contrast to offline training, NN weight update laws derived from Lyapunov-based stability analysis methods have been developed to adjust the NN weights in real-time as an adaptive closed-loop feedforward term [8]. Although NN-based adaptive architectures are well-established, these methods only apply to NNs with a single hidden-layer. The complex nature of DNNs being nested nonlinear parameterizations of inner-layer activation functions, weights, and bias terms presents challenges that preclude development of real-time adaptation laws with Lyapunov-based methods.

Motivated by function approximation abilities of DNNs, emerging results in [9]–[12] develop real-time DNN-based adaptive architectures. In [9] and [10], real-time DNN-based adaptive architectures are developed for model reference adaptive control. Similarly, results in [11] generalize the DNN-based adaptive architecture to general nonlinear systems. However, such results only update the output-layer weights in real-time. While the output-layer weights are updated in real-time, data is collected and used to train the inner-layer weights iteratively over discrete training periods via traditional offline techniques. In [12], insights are provided into the development of real-time adaptive weight update laws for individual layers of a feedforward DNN based on a modular design. Modular adaptive designs develop mild constraints on the adaptation laws and provide stability guarantees based on the worst-case scenario of the developed constraints. Although the modular adaptive approach provides constraints on the weight update laws, these constraints are only sufficient and lack insights on how to best design the inner-layer weight adaptation laws.

This letter presents the first result with Lyapunov-based real-time weight adaptation laws for each layer of a DNN. A

general uncertain nonlinear system is considered. To address the challenges posed by nested nonlinear parameterizations of the inner-layer DNN weights, we develop a recursive representation of the inner-layer DNN structure to facilitate the analysis. Then, a Taylor's first order approximation of the uncertainty is recursively derived. Subsequently, the update laws are derived from a Lyapunov-based stability analysis, in which the first-order terms are canceled by the weight update law-based terms. The remaining terms in the Lyapunov-based analysis are eliminated using a robust control approach.

The adaptation laws developed in this letter depend on gradients of activation functions. The adaptation laws contain discontinuities if an activation function with a discontinuous gradient is used in the DNN architecture. Nonsmooth activation functions such as rectified linear units (ReLU), leaky ReLUs (LReLU) [13], maxout [14], etc. are often preferred over sigmoidal activation functions, since they empirically exhibit improved function approximation performance while also overcoming the vanishing gradient problem [4, Ch. 6]. As previously noted, the discontinuities in gradients of these activation functions pose difficulties in facilitating the standard Lyapunov-based analysis. In this letter, a nonsmooth analysis is performed to address the challenges of including nonsmooth activation functions. The nonsmooth Lyapunov-based analysis guarantees global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs with leaky ReLU and hyperbolic tangent activation functions to demonstrate the efficacy of the developed method. A comparison of a DNN with leaky ReLU activation functions to a DNN with hyperbolic tangent activation functions shows improved tracking and function approximation performance while using the DNN with leaky ReLU activation functions.

Notation and Preliminaries: The space of essentially bounded Lebesgue measurable functions is denoted by \mathcal{L}_∞ . The right-to-left matrix product operator is represented by \prod , i.e., $\prod_{p=1}^m A_p = A_m \dots A_2 A_1$ and $\prod_{p=a}^m A_p = 1$ if $a > m$. The vectorization operator is denoted by $\text{vec}(\cdot)$, i.e., given $A \triangleq [a_{i,j}] \in \mathbb{R}^{n \times m}$, $\text{vec}(A) \triangleq [a_{1,1}, \dots, a_{1,m}, \dots, a_{n,1}, \dots, a_{n,m}]^T$. The p -norm is denoted by $\|\cdot\|_p$, where the subscript is suppressed when $p = 2$. The Frobenius norm is denoted by $\|\cdot\|_F \triangleq \|\text{vec}(\cdot)\|$. The Kronecker product is denoted by \otimes . Function compositions are denoted using the symbol \circ , e.g., $(g \circ h)(x) = g(h(x))$, given suitable functions f and g . The Filippov set-valued map defined in [15, Equation 2b] is denoted by $K[\cdot]$. Consider a Lebesgue measurable and locally essentially bounded function $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$. Then, the function $y : \mathcal{I} \rightarrow \mathbb{R}^n$ is called a *Filippov solution* of $\dot{y} = h(y, t)$ on the interval $\mathcal{I} \subseteq \mathbb{R}_{\geq 0}$ if y is absolutely continuous on \mathcal{I} and $\dot{y} \in K[h](y, t)$ for almost all $t \in \mathcal{I}$. The notation $F : A \rightrightarrows B$ denotes a set-valued map from set A to set B . Given some functions f and g , the notation $f(y) = \mathcal{O}^m(g(y))$ means that there exists some constants $M \in \mathbb{R}_{>0}$ and $y_0 \in \mathbb{R}$ such that $\|f(y)\| \leq M\|g(y)\|^m$ for all $y \geq y_0$.

Fact 1 [16, Proposition 7.1.9]: Given any $A \in \mathbb{R}^{p \times a}$, $B \in \mathbb{R}^{a \times r}$, and $C \in \mathbb{R}^{r \times s}$, $\text{vec}(ABC) = (C^T \otimes A)\text{vec}(B)$.

II. UNKNOWN SYSTEM DYNAMICS AND CONTROL DESIGN

Consider a control-affine nonlinear dynamic system modeled as

$$\dot{x} = f(x) + u, \quad (1)$$

where $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ denotes a Filippov solution¹ to (1), $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes an unknown differentiable function, and $u : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ denotes a control input. The control objective is to track a user-defined reference trajectory $x_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$. The reference trajectory is designed to be continuously differentiable, such that $x_d(t) \in \Omega$, $\forall t \in \mathbb{R}_{\geq 0}$, and $\dot{x}_d \in \mathcal{L}_\infty$, where $\Omega \subset \mathbb{R}^n$ denotes a known compact set. To quantify the tracking objective, the tracking error $e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ is defined as

$$e \triangleq x - x_d. \quad (2)$$

A. Deep Neural Network Architecture

A variety of DNN architectures are known to approximate any given continuous function on a compact set, based on universal approximation theorems that can be invoked case-by-case for DNN architectures [18]. Let $\Phi : \mathbb{R}^n \times \mathbb{R}^{L_0 \times L_1} \times \dots \times \mathbb{R}^{L_k \times L_{k+1}} \rightarrow \mathbb{R}^n$ denote the feedforward DNN architecture defined as

$$\Phi(x_d, V_0, V_1, \dots, V_k) \triangleq (V_k^T \phi_k \circ \dots \circ V_1^T \phi_1)(V_0^T x_{da}), \quad (3)$$

where $x_{da} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n+1}$ denotes the augmented desired state $x_{da} \triangleq [x_d^T \ 1]^T$, and $k \in \mathbb{N}$ denotes the total number of hidden-layers. The matrix of weights and biases at the j^{th} layer is denoted by $V_j \in \mathbb{R}^{L_j \times L_{j+1}}$, where $L_j \in \mathbb{N}$ denotes the number of nodes in the j^{th} inner-layer for all $j \in \{0, \dots, k\}$, with $L_0 \triangleq n + 1$ and $L_{k+1} \triangleq n$. The vector of smooth² activation functions at the j^{th} layer is denoted by $\phi_j : \mathbb{R}^{L_j} \rightarrow \mathbb{R}^{L_j}$. If the DNN involves multiple types of activation functions at each layer, then ϕ_j may be represented as $\phi_j \triangleq [\varsigma_{j,1} \dots \varsigma_{j,L_{j-1}} \ 1]^T$, where $\varsigma_{j,i} : \mathbb{R} \rightarrow \mathbb{R}$ denotes the activation function at the i^{th} node of j^{th} layer. Note that x_{da} and ϕ_j are augmented with 1 to facilitate the inclusion of a bias term. The DNN architecture in (3) can also be represented recursively as

$$\Phi_j \triangleq \begin{cases} V_j^T \phi_j(\Phi_{j-1}), & j \in \{1, \dots, k\}, \\ V_0^T x_{da}, & j = 0, \end{cases} \quad (4)$$

and $\Phi(x_d, V_0, \dots, V_k) = \Phi_k$, where $\Phi_j : \mathbb{R}^n \times \mathbb{R}^{L_0 \times L_1} \times \dots \times \mathbb{R}^{L_j \times L_{j+1}} \rightarrow \mathbb{R}^{L_{j+1}}$ denotes $(x_d, V_0, \dots, V_j) \mapsto \Phi_j(x_d, V_0, \dots, V_j)$. The universal function approximation property states that the function space of DNNs given by (3) is dense in $\mathcal{C}(\Omega)$ [18, Th. 3.2], where $\mathcal{C}(\Omega)$ denotes the space of functions continuous over Ω . For any given

¹We consider generalized solutions such as Filippov or Krasovskii solutions instead of classical solutions to facilitate a nonsmooth control design. These solutions are guaranteed to exist for nonsmooth systems with Lebesgue measurable and locally essentially bounded right-hand-sides [17, Proposition 3], whereas classical solutions might not exist.

²Although ϕ_j is defined as a smooth function, the subsequent analysis allows the inclusion of nonsmooth activation functions by modeling them via a switching mechanism involving smooth functions.

$f \in \mathcal{C}(\Omega)$ and prescribed $\bar{\varepsilon} \in \mathbb{R}_{>0}$, there exist some $k, L_j \in \mathbb{N}$, and corresponding ideal weights and biases, $V_j^* \in \mathbb{R}^{L_j \times L_{j+1}}$, $\forall j \in \{0, \dots, k\}$, such that $\sup_{x_d \in \Omega} \|f(x_d) - \Phi(x_d, V_0^*, V_1^*, \dots, V_k^*)\| \leq \bar{\varepsilon}$. Then the unknown function in (1) can be modeled as

$$f(x_d) = \Phi(x_d, V_0^*, V_1^*, \dots, V_k^*) + \varepsilon(x_d), \quad (5)$$

where $\varepsilon : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes the unknown function approximation error such that $\sup_{x_d \in \Omega} \|\varepsilon(x_d)\| \leq \bar{\varepsilon}$. It is assumed there exists a known constant $\bar{V} \in \mathbb{R}_{>0}$ such that $\sup_{x_d \in \Omega, \forall j} \|V_j^*\|_F \leq \bar{V}$ (cf., [19, Assumption 1]).

B. Control Law Development

The universal approximation property makes DNN-based adaptive control architectures well-suited for unknown dynamics, as in (1) where $f(\cdot)$ is unknown [18, Th. 3.2]. The adaptive feedforward DNN term is designed as $\hat{\Phi} \triangleq \Phi(x_d, \hat{V}_0, \dots, \hat{V}_k)$, where $\hat{V}_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_j \times L_{j+1}}$ for all $j \in \{0, \dots, k\}$ denotes the estimated weight matrix for the j^{th} layer. The weight estimation error of the ideal inner-layer weights $\tilde{V}_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_j \times L_{j+1}}$ for all $j \in \{0, \dots, k\}$ is defined as $\tilde{V}_j \triangleq V_j^* - \hat{V}_j$. The gradient of the activation function vector at the j^{th} layer is denoted as $\phi_j' : \mathbb{R}^{L_j} \rightarrow \mathbb{R}^{L_j \times L_j}$, and $\phi_j'(y) \triangleq \frac{\partial}{\partial z} \phi_j(z)|_{z=y}$, $\forall y \in \mathbb{R}^{L_j}$. To facilitate the subsequent stability analysis, let the function $f_e : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}^n$ be defined as $f_e \triangleq f(x) - f(x_d)$. By [20, Lemma 5], the function $(x, x_d) \mapsto f_e$ is bounded as $\|f_e\| \leq \rho(\|e\|)\|e\|$ for all $x \in \mathbb{R}^n$ and $x_d \in \Omega$, where $\rho : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ denotes a known strictly increasing function. Based on the subsequent stability analysis, the control input is designed as

$$u \triangleq \dot{x}_d - \rho(\|e\|)e - k_1 e - k_s \text{sgn}(e) - \hat{\Phi}, \quad (6)$$

where $k_1, k_s \in \mathbb{R}_{>0}$ are user-defined control gains, and $\text{sgn}(\cdot)$ denotes the vector signum function. The following short-hand notations are introduced for brevity in the subsequent analysis: $\Phi_j^* \triangleq \Phi_j(x_d, V_0^*, \dots, V_j^*)$, $\hat{\Phi}_j \triangleq \Phi_j(x_d, \hat{V}_0, \dots, \hat{V}_j)$, $\tilde{\Phi}_j \triangleq \Phi_j^* - \hat{\Phi}_j$, $\Phi^* \triangleq \Phi_k^*$, $\tilde{\Phi} \triangleq \tilde{\Phi}_k = \Phi^* - \hat{\Phi}$, $\phi_j^* \triangleq \phi_j(\Phi_{j-1}^*)$, $\hat{\phi}_j \triangleq \phi_j(\hat{\Phi}_{j-1})$, and $\hat{\phi}_j' \triangleq \phi_j'(\hat{\Phi}_{j-1})$.

Based on the subsequent analysis, the input layer weight adaptation law is designed as

$$\text{vec}(\dot{\hat{V}}_0) \triangleq \text{proj}(\Gamma_0((\prod_{l=1}^k \hat{V}_l^T \hat{\phi}_l')(I_{L_1} \otimes x_{da}^T))^T e), \quad (7)$$

and the j^{th} layer weight adaptation law is designed as

$$\text{vec}(\dot{\hat{V}}_j) \triangleq \text{proj}(\Gamma_j((\prod_{l=j+1}^k \hat{V}_l^T \hat{\phi}_l')(I_{L_{j+1}} \otimes \hat{\phi}_j^T))^T e), \quad (8)$$

$\forall j \in \{1, \dots, k\}$, where $\Gamma_j \in \mathbb{R}^{L_{j+1} \times L_{j+1}}$ is a positive-definite adaptation gain matrix for all $j \in \{0, \dots, k\}$. The operator $\text{proj}(\cdot)$ denotes the projection operator defined in [21, Appendix E, eq. (E.4)], which is used to ensure $\hat{V}_j(t) \in \mathcal{B}_j \triangleq \{\theta \in \mathbb{R}^{L_j \times L_{j+1}} : \|\theta\|_F \leq \bar{V}\}$, $\forall (t, j) \in \mathbb{R}_{\geq 0} \times \{0, 1, \dots, k\}$.

III. STABILITY ANALYSIS

A. Closed-Loop Error System Development

Subtracting $\hat{\Phi}_j$ from Φ_j^* , using (4), adding and subtracting $V_j^{*T} \hat{\phi}_j$, and rearranging terms yields

$$\tilde{\Phi}_j = \tilde{V}_j^T \hat{\phi}_j + V_j^{*T} (\phi_j^* - \hat{\phi}_j), \quad (9)$$

$\forall j \in \{1, \dots, k\}$, and $\tilde{\Phi}_0 = \tilde{V}_0^T x_{da}$. Using (1), (2), and (6) yields the closed-loop error system

$$\dot{e} = f_e + \tilde{\Phi} + \varepsilon(x_d) - \rho(\|e\|)e - k_1 e - k_s \text{sgn}(e). \quad (10)$$

The term $\tilde{\Phi}$ in (10) has a nested nonlinear parameterization in V_j^* and \hat{V}_j , which precludes the application of traditional analysis techniques that are used for linearly parameterized adaptive systems. A first-order Taylor series approximation is developed in [19] to overcome the challenges presented by the nonlinear parameterization for three-layer neural networks. To overcome the nested structure of nonlinear parameterization in DNNs, we use a recursive approach to develop a first-order Taylor series approximation for ϕ_j^* , $\tilde{\Phi}_j$, and $\tilde{\Phi}$. Using the first-order Taylor series approximation in [19, eq. (22)] yields

$$\phi_j^* = \hat{\phi}_j + \hat{\phi}_j' \tilde{\Phi}_{j-1} + \mathcal{O}^2(\tilde{\Phi}_{j-1}), \quad (11)$$

$\forall j \in \{1, \dots, k\}$. Substituting (11) into (9), adding and subtracting $\hat{V}_j^T \hat{\phi}_j' \tilde{\Phi}_{j-1}$, and rearranging terms yields

$$\tilde{\Phi}_j = \tilde{V}_j^T \hat{\phi}_j + \hat{V}_j^T \hat{\phi}_j' \tilde{\Phi}_{j-1} + \Delta_j, \quad (12)$$

where $\Delta_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_{j+1}}$ is defined as

$$\Delta_j \triangleq \tilde{V}_j^T \hat{\phi}_j' \tilde{\Phi}_{j-1} + V_j^{*T} \mathcal{O}^2(\tilde{\Phi}_{j-1}), \quad (13)$$

$\forall j \in \{1, \dots, k\}$. Since the term $\tilde{V}_j^T \hat{\phi}_j$ is a vector, $\tilde{V}_j^T \hat{\phi}_j = \text{vec}(\tilde{V}_j^T \hat{\phi}_j) = \text{vec}(\hat{\phi}_j^T \tilde{V}_j) = \text{vec}(\hat{\phi}_j^T \tilde{V}_j I_{L_{j+1}})$. Applying Fact 1 on $\text{vec}(\hat{\phi}_j^T \tilde{V}_j I_{L_{j+1}})$ yields

$$\tilde{V}_j^T \hat{\phi}_j = (I_{L_{j+1}} \otimes \hat{\phi}_j^T) \text{vec}(\tilde{V}_j). \quad (14)$$

Substituting (14) into (12) yields the recursive representation

$$\tilde{\Phi}_j = (I_{L_{j+1}} \otimes \hat{\phi}_j^T) \text{vec}(\tilde{V}_j) + \hat{V}_j^T \hat{\phi}_j' \tilde{\Phi}_{j-1} + \Delta_j, \quad (15)$$

$\forall j \in \{1, \dots, k\}$. To facilitate the subsequent analysis, the following lemma yields a generalized expression for $\tilde{\Phi}_j$.

Lemma 1: For all $j \in \{0, \dots, k\}$, the term $\tilde{\Phi}_j$ can be expressed as

$$\begin{aligned} \tilde{\Phi}_j = & \sum_{i=1}^j \left(\prod_{l=i+1}^j \hat{V}_l^T \hat{\phi}_l' \right) (I_{L_{i+1}} \otimes \hat{\phi}_i^T) \text{vec}(\tilde{V}_i) \\ & + \left(\prod_{l=1}^j \hat{V}_l^T \hat{\phi}_l' \right) (I_{L_1} \otimes x_{da}^T) \text{vec}(\tilde{V}_0) \\ & + \sum_{i=1}^j \left(\prod_{l=i+1}^j \hat{V}_l^T \hat{\phi}_l' \right) \Delta_i. \end{aligned} \quad (16)$$

Proof: See the Appendix. ■

B. Nonsmooth Analysis

Let $\Xi_0 \triangleq \prod_{l=1}^k \hat{V}_l^T \hat{\phi}'_l$, $\Xi_0 \triangleq \prod_{l=j+1}^k \hat{V}_l^T \hat{\phi}'_l$, $\Lambda_0 \triangleq \Xi_0(I_{L_1} \otimes x_{da}^T)$, and $\Lambda_j \triangleq \Xi_j(I_{L_{j+1}} \otimes \hat{\phi}_j^T)$, $\forall j \in \{1, \dots, k\}$, for notational brevity, where $\Xi_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times L_{j+1}}$ and $\Lambda_j : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{L_{j+1} \times L_{j+1}}$, respectively, $\forall j \in \{0, \dots, k\}$. The subsequent analysis is structured to account for nonsmooth systems. Thus, state-dependent switching between smooth activation functions can also be considered in the analysis. Specifically, a nonsmooth activation function with a finite number of discontinuities in its gradient can be modeled by a switched function involving a collection of smooth activation functions. Let $\sigma \in \mathcal{N}$ denote the switching index considering the total number of switching between activation functions in the entire DNN, where $\mathcal{N} \subset \mathbb{N}$ denotes the set of all possible switching indices. Then, the function approximation in (5) can be represented as $f(x_d) = \Phi_{k,\sigma}(x_d, V_0^*, V_1^*, \dots, V_k^*) + \varepsilon_\sigma(x_d)$, such that $(x_d, V_0, \dots, V_j) \mapsto \Phi_{k,\sigma}(x_d, V_0, \dots, V_j)$ is smooth for each σ with the corresponding approximation error $\varepsilon_\sigma(x_d)$. Thus, ϕ_j^* , $\hat{\phi}_j$, $\hat{\phi}'_j$, $\tilde{\Phi}_j$, Ξ_j , Λ_j , and Δ_j can also be represented as the switched functions $\phi_{j,\sigma}^*$, $\hat{\phi}_{j,\sigma}$, $\hat{\phi}'_{j,\sigma}$, $\tilde{\Phi}_{j,\sigma}$, $\Xi_{j,\sigma}$, $\Lambda_{j,\sigma}$, and $\Delta_{j,\sigma}$, respectively, such that they are continuous for each σ . It is assumed that the bound $\sup_{x_d \in \Omega} \|\varepsilon_\sigma(x_d)\| \leq \bar{\varepsilon}$ holds for all $\sigma \in \mathcal{N}$. Using Lemma 1 yields $\tilde{\Phi} = \tilde{\Phi}_{k,\sigma} = \sum_{j=0}^k \Lambda_{j,\sigma} \text{vec}(\tilde{V}_j) + \sum_{j=1}^k \Xi_{j,\sigma} \Delta_{j,\sigma}$. Substituting $\tilde{\Phi}$ into (10) yields

$$\begin{aligned} \dot{e} = & f_e + \sum_{j=0}^k \Lambda_{j,\sigma} \text{vec}(\tilde{V}_j) + \sum_{j=1}^k \Xi_{j,\sigma} \Delta_{j,\sigma} \\ & + \varepsilon_\sigma(x_d) - \rho(\|e\|)e - k_1 e - k_s \text{sgn}(e). \end{aligned} \quad (17)$$

Additionally, the adaptation laws in (7) and (8) can be represented using $\text{vec}(\hat{V}_j) \triangleq \text{proj}(\Gamma_j \Lambda_{j,\sigma}^T e)$, $\forall j \in \{0, \dots, k\}$. Consequently, $\text{vec}(\tilde{V}_j) = -\text{proj}(\Gamma_j \Lambda_{j,\sigma}^T e)$, $\forall j \in \{0, \dots, k\}$. Since $\|V_j^*\|_F \leq \bar{V}$ and $\|\hat{V}_j\|_F \leq \bar{V}$, $\forall j \in \{0, \dots, k\}$, it follows that $\|\tilde{V}_j\|_F = \|\hat{V}_j - V_j^*\|_F \leq 2\bar{V}$. Moreover, since x_{da} is bounded, and $\phi_{j,\sigma}$ and $\phi'_{j,\sigma}$ are continuous for each $\sigma \in \mathcal{N}$, it follows from (9) that $\phi_{j,\sigma}^*$, $\hat{\phi}_{j,\sigma}$, $\hat{\phi}'_{j,\sigma}$, $\tilde{\Phi}_{j,\sigma}$, and $\Xi_{j,\sigma}$ can be bounded by known constants for all $(j, \sigma) \in \{0, \dots, k\} \times \mathcal{N}$. Therefore, based on (13), Δ_j can be bounded by known constants for all $j \in \{1, \dots, k\}$, and it follows that there exists a known constant $c \in \mathbb{R}_{>0}$ such that

$$\left\| \sum_{i=1}^k \Xi_{i,\sigma} \Delta_{i,\sigma} \right\| \leq c. \quad (18)$$

Let $z : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^\Psi$ denote the concatenated function, $z \triangleq [e^T, \text{vec}(\tilde{V}_0)^T, \dots, \text{vec}(\tilde{V}_k)^T]^T$, where $\Psi \triangleq n + \sum_{j=0}^k L_j L_{j+1}$ is defined for notational brevity. Let $w_\sigma : \mathbb{R}^\Psi \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^\Psi$ denote the concatenated right hand sides of (17) and $\text{vec}(\tilde{V}_j) = -\text{proj}(\Gamma_j \Lambda_{j,\sigma}^T e)$. Then (17) and $\text{vec}(\tilde{V}_j)$ can be represented by the collection of subsystems $\dot{z} = w_\sigma(z, t)$, and the corresponding switched system is represented by

$$\dot{z} = w_{\varrho(z,t)}(z, t), \quad (19)$$

where $\varrho : \mathbb{R}^\Psi \times \mathbb{R}_{\geq 0} \rightarrow \mathcal{N}$ denotes a state-dependent switching signal that satisfies [22, Assumption 1].³ Based on the result in [22], we establish the invariance properties of (19) by establishing the invariance properties of $\dot{z} = w_\sigma(z, t)$ for each $\sigma \in \mathcal{N}$. Let $F_\sigma : \mathbb{R}^\Psi \times \mathbb{R}_{\geq 0} \rightrightarrows \mathbb{R}^\Psi$ denote $K[w_\sigma](z, t)$. Then $F_\sigma(z, t) \subseteq F'_\sigma(z, t)$, where $F'_\sigma : \mathbb{R}^\Psi \times \mathbb{R}_{\geq 0} \rightrightarrows \mathbb{R}^\Psi$ is defined as $F'_\sigma(z, t) \triangleq \{[\sum_{j=0}^k \Lambda_{j,\sigma} \text{vec}(\tilde{V}_j) + \sum_{j=1}^k \Xi_{j,\sigma} \Delta_{j,\sigma} + f_e + \varepsilon_\sigma(x_d) - \rho(\|e\|)e - k_1 e] - k_s K[\text{sgn}(e)]; -K[\text{proj}](\Gamma_0 \Lambda_{0,\sigma}^T e); \dots; -K[\text{proj}](\Gamma_k \Lambda_{k,\sigma}^T e)\}$.

Theorem 1: For the dynamical system in (1), the controller in (6) and the adaptation laws in (7) and (8) ensure global asymptotic tracking error convergence in the sense that $\lim_{t \rightarrow \infty} \|e(t)\| = 0$, $\forall (e(0), \tilde{V}_0, \dots, \tilde{V}_k) \in \mathbb{R}^n \times \mathcal{B}_0 \times \dots \times \mathcal{B}_k$, provided the gain condition $k_s > \bar{\varepsilon} + c$ is satisfied.

Proof: Consider the candidate common Lyapunov function $\mathcal{V}_L : \mathbb{R}^\Psi \rightarrow \mathbb{R}_{\geq 0}$ defined as

$$\mathcal{V}_L(z) \triangleq \frac{1}{2} e^T e + \frac{1}{2} \sum_{j=0}^k \text{vec}(\tilde{V}_j)^T \Gamma_j^{-1} \text{vec}(\tilde{V}_j), \quad (20)$$

which satisfies the inequality $\underline{\alpha} \|z\|^2 \leq \mathcal{V}_L(z) \leq \bar{\alpha} \|z\|^2$, where $\underline{\alpha}, \bar{\alpha} \in \mathbb{R}_{\geq 0}$ are known constants. Using [22, Def. 3], the generalized time-derivative of \mathcal{V}_L can be computed as $\dot{\mathcal{V}}_\sigma(z, t) \triangleq \max_{p \in \partial \mathcal{V}_L(z)} \max_{q \in F_\sigma(z, t)} p^T q$, where $\partial \mathcal{V}_L$ denotes the Clarke gradient of \mathcal{V}_L defined in [23, pp. 39]. Since $z \mapsto \mathcal{V}_L(z)$ is continuously differentiable, $\partial \mathcal{V}_L(z) = \{\nabla \mathcal{V}_L(z)\}$, where ∇ denotes the standard gradient operator. Thus,

$$\begin{aligned} \dot{\mathcal{V}}_\sigma(z, t) &= \max_{q \in F_\sigma(z, t)} (\nabla \mathcal{V}_L(z))^T q \\ &\stackrel{a.e.}{\leq} \max_{q \in F'_\sigma(z, t)} (\nabla \mathcal{V}_L(z))^T q, \end{aligned}$$

where the notation $\stackrel{a.e.}{\cdot}$ denotes that the relation (\cdot) holds for almost all $t \in \mathbb{R}_{\geq 0}$. Additionally, using [21, Lemma E.1.IV],⁴ the update law-based terms that appear after evaluating $\max_{q \in F'_\sigma(z, t)} (\nabla \mathcal{V}_L(z))^T q$ can be upper-bounded as

$$-\text{vec}(\tilde{V}_j)^T \Gamma_j^{-1} K[\text{proj}](\Gamma_j \Lambda_{j,\sigma}^T e) \leq -\text{vec}(\tilde{V}_j)^T \Lambda_{j,\sigma}^T e, \quad (21)$$

$\forall j \in \{0, \dots, k\}$. Thus, evaluating $\max_{q \in F'_\sigma(z, t)} (\nabla \mathcal{V}_L(z))^T q$, using (21) and the fact that $e^T K[\text{sgn}(e)] = \{\|e\|_1\}$ yields

$$\begin{aligned} \dot{\mathcal{V}}_\sigma(z, t) &\stackrel{a.e.}{\leq} e^T (f_e + \sum_{j=1}^k \Xi_{j,\sigma} \Delta_{j,\sigma} + \varepsilon_\sigma(x_d) - \rho(\|e\|)e - k_1 e) \\ &\quad + \max_{j=0}^k \{e^T \Lambda_{j,\sigma} \text{vec}(\tilde{V}_j) - \text{vec}(\tilde{V}_j)^T \Lambda_{j,\sigma}^T e\} - k_s \|e\|_1. \end{aligned} \quad (22)$$

Noting that $e^T \Lambda_{j,\sigma} \text{vec}(\tilde{V}_j) = (e^T \Lambda_{j,\sigma} \text{vec}(\tilde{V}_j))^T = \text{vec}(\tilde{V}_j)^T \Lambda_{j,\sigma}^T e$ since they are scalar, the term $\max_{j=0}^k \{e^T \Lambda_{j,\sigma} \text{vec}(\tilde{V}_j) - \text{vec}(\tilde{V}_j)^T \Lambda_{j,\sigma}^T e\} = 0$, $\forall \sigma \in \mathcal{N}$.

³The assumption [22, Assumption 1] is equivalent to the assumption that ϱ is locally bounded. Since the switched system in (19) involves a finite number of subsystems, the assumption is always satisfied in this letter.

⁴The lemma says $-\partial^T \Gamma^{-1} \text{proj}(\mu) \leq -\partial^T \Gamma^{-1} \mu$. This property also holds after replacing $\text{proj}(\mu)$ with $K[\text{proj}](\mu)$, since $K[\text{proj}](\mu)$ evaluates as the set of convex combinations of $\text{proj}(\mu)$ and μ , whenever $\text{proj}(\mu)$ is discontinuous.

Thus, substituting $\|\sum_{j=1}^k \Xi_j \Delta_j\| \leq c$, $\|\varepsilon_\sigma(x_d)\| \leq \bar{\varepsilon}$, and $e^T f_e \leq \rho(\|e\|)\|e\|^2$, (22) can be upper bounded as

$$\dot{\tilde{V}}_\sigma(z, t) \stackrel{a.e.}{\leq} e^T(c + \bar{\varepsilon} - k_1 e) - k_s \|e\|_1.$$

Using $-k_s \|e\|_1 \leq -k_s \|e\|$, and selecting k_s according to the theorem statement yields

$$\dot{\tilde{V}}_\sigma(z, t) \stackrel{a.e.}{\leq} -k_1 \|e\|^2. \quad (23)$$

By invoking [22, Th. 2], $z \in \mathcal{L}_\infty$ and $\|e(t)\| \rightarrow 0$ as $t \rightarrow \infty$. Additionally, $z \in \mathcal{L}_\infty$ implies $\tilde{V}_j, \hat{V}_j \in \mathcal{L}_\infty, \forall j \in \{0, \dots, k\}$. Since Φ is a locally essentially bounded function, it follows that $\hat{\Phi}$ is bounded. Therefore, since all terms on the right hand side of (6) are bounded, it follows that $u \in \mathcal{L}_\infty$. Moreover, since ϕ_j and ϕ'_j are locally essentially bounded functions for all $j \in \{0, \dots, k\}$, it follows from (7) and (8) that $\hat{V}_j \in \mathcal{L}_\infty, \forall j \in \{0, \dots, k\}$. ■

IV. SIMULATIONS

Four simulation examples are provided to demonstrate the efficacy of the developed method, and the results are quantitatively compared with known baseline methods such as offline pre-training and output-layer adaptation [11]. The nonlinear system in (1) is considered with $f(x) = [x_1 x_2 \tanh(x_2) + \text{sech}^2(x_1), \text{sech}^2(x_1 + x_2) - \text{sech}^2(x_2)]^T$, where $x = [x_1, x_2]^T$. The desired trajectory is $x_d(t) = [\sin(2t), -\cos(t)]^T$, the initial condition is $x(0) = [1, 2]^T$, the control gains are selected as $k_1 = 20$ and $k_s = 1$, and the bound for projection operator is selected as $\bar{V} = 5000$. The DNNs in the first and second examples, i.e., DNN1 and DNN2, consist of 6 layers, with 7 neurons in each layer; hence, there is a total of 231 individual weights in the first two examples. The DNNs in the third and fourth examples, i.e., DNN3 and DNN4, consist of 10 layers, with 30 neurons in each layer; hence, there is a total of 90150 individual weights in the third and fourth examples. Each simulation is performed for 10 seconds. To prevent the DNN term from having a large initial value, the inner and output layer weights are initialized as random values from the uniform distributions $U(0, 0.5)$ and $U(0, 0.01)$, respectively.

DNN1 and DNN3 contain LReLU activation functions given by $\varsigma(y) = y$ for $y \geq 0$, and $\varsigma(y) = 0.01y$, otherwise. The adaptation gain for DNN1 and DNN3 is selected using the switched rule: $\Gamma_j = 10I_{L_j L_{j+1}}$, if $\|[\text{vec}^T(\tilde{V}_0) \dots \text{vec}^T(\tilde{V}_k)]^T\| \leq 5$, and $\Gamma_j = I_{L_j L_{j+1}}$, otherwise, for all $j \in \{0, \dots, k\}$. DNN2 and DNN4 contain hyperbolic tangent activation functions given by $\varsigma(y) = \tanh(y)$. Unlike LReLU, saturating activation functions like hyperbolic tangents suffer from the vanishing gradient problem [4], i.e., the gradient terms in the update law vanish as the activation function saturates, which slows down the weight updates. To compensate for vanishing gradients and for a fair comparison with the LReLU-based DNNs, the adaptation gain for the hyperbolic tangent activation function-based DNNs is selected with a relatively larger value of $\Gamma_j = 500I_{L_j L_{j+1}}$.

Figure 1 shows the plots of DNN weight estimates, tracking error, and function estimation error for DNN3 and DNN4, where $\tilde{f} \triangleq f(x) - \hat{\Phi}$ denotes the function estimation error. The

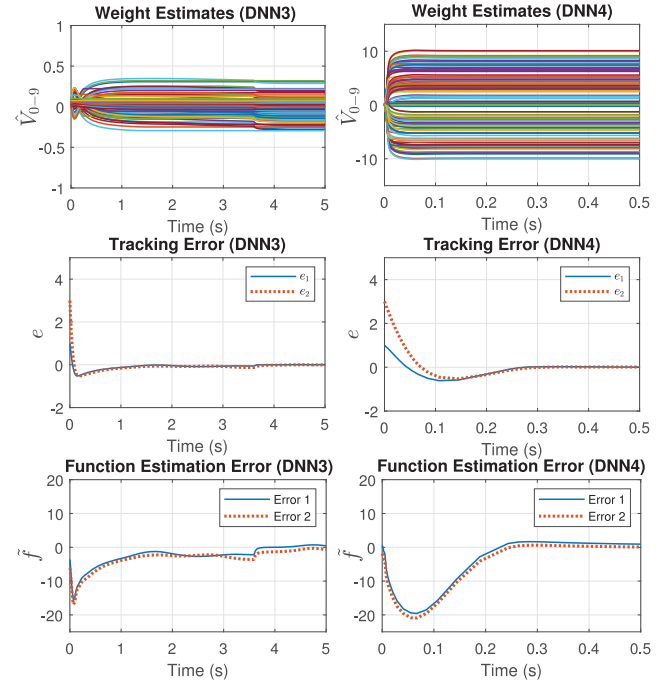


Fig. 1. Plots of DNN weight estimates, tracking error, and function approximation error for DNN3 and DNN4. The simulation is performed for 10 seconds. For a better visualization of the transient performance, the plots for LReLU and tanh are shown for 5 and 0.5 seconds, respectively. Additionally, we show 150 arbitrarily selected weight estimates out of the total 90150 weights for a tractable visualization.

TABLE I
DNN PERFORMANCE COMPARISON

Method	$\ e_{\text{RMS}}\ $	$\ e_{\text{RMS,SS}}\ $	$\ \tilde{f}_{\text{RMS,SS}}\ $
Developed (DNN1)	0.7014	0.0059	0.1204
Developed (DNN2)	1.2687	0.0108	0.2178
Developed (DNN3)	0.4326	0.0056	0.6824
Developed (DNN4)	0.3844	0.0063	0.7823
Offline (DNN1)	0.2952	0.0216	1.2165
[11] (DNN1)	0.6831	0.0105	0.2143

plots demonstrate that asymptotic convergence of the tracking error e is achieved in 0.5 s for both the examples. Table I provides a quantitative comparison of the developed method with offline training and output-layer adaptation [11], where e_{RMS} denotes the root mean square (RMS) of e over the time interval $[0, 10]$, and $e_{\text{RMS,SS}}$ and $\tilde{f}_{\text{RMS,SS}}$ denote the RMS of e and \tilde{f} , respectively, over the time interval $[5, 10]$ (i.e., in steady state). For the simulations in Tab. I, the robustifying term $k_s \text{sgn}(e)$ is removed to better quantitatively compare the effects of the DNN term. The offline pre-trained DNN is trained using data collected from 600 seconds of an *a priori* simulation. Using LReLU yields improvement in the steady state tracking and function estimation performance as compared to hyperbolic tangent units as evident from the $\|e_{\text{RMS,SS}}\|$ and $\|\tilde{f}_{\text{RMS,SS}}\|$ values for DNN1 vs. DNN2 and DNN3 vs. DNN4. The developed method provides a decreased $\|e_{\text{RMS,SS}}\|$ but an increased $\|e_{\text{RMS}}\|$ as compared to offline pre-training or using adaptation for only the output-layer. This discrepancy is due

to the initial overshoot in tracking error due to weight adaptation. The developed method provides a tenfold and twofold improvement in steady-state function estimation as compared to offline pre-training and output-layer adaptation, respectively.

V. CONCLUSION

This letter presents Lyapunov-based real-time weight update laws for each layer of a feedforward DNN. Additionally, the developed method also allows nonsmooth activation functions to be used in the DNN architecture. A nonsmooth Lyapunov-based stability analysis is provided to guarantee global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs involving leaky ReLU and hyperbolic tangent activation functions to demonstrate the efficacy of the developed method. Using LReLU yields improvement in the steady-state tracking and function estimation performance when compared to hyperbolic tangent activation functions. Although adapting for more layers might cause initial overshoot in the tracking error, the developed method provides tenfold and twofold improvement in steady-state function estimation as compared to offline pre-training and output-layer adaptation, respectively.

APPENDIX

Proof of Lemma 1: We prove this lemma by mathematical induction. Using (4) and Fact 1 yields $\tilde{\Phi}_0 = \tilde{V}_0^T x_{da} = (I_{L_1} \otimes x_{da}^T) \text{vec}(\tilde{V}_0)$. Since $\prod_{l=1}^0 \hat{V}_l^T \hat{\phi}_l' = 1$, it can be verified that (16) also yields $\tilde{\Phi}_0 = (I_{L_1} \otimes x_{da}^T) \text{vec}(\tilde{V}_0)$. Thus, Lemma 1 holds for $j = 0$. To use induction, assume (16) applies for $j = h - 1$, given any arbitrary $h \in \{1, \dots, k\}$, and evaluate $\tilde{\Phi}_{h-1}$. Then, using (15) with $j = h$ yields

$$\tilde{\Phi}_h = (I_{L_{h+1}} \otimes \hat{\phi}_h^T) \text{vec}(\tilde{V}_h) + \hat{V}_h^T \hat{\phi}_h' \tilde{\Phi}_{h-1} + \Delta_h \quad (24)$$

Substituting $\tilde{\Phi}_{h-1}$ into (24) and rearranging terms, it can be verified that the obtained expression is the same as that obtained using (16). Thus, Lemma 1 also applies for $j = h$. ■

REFERENCES

- [1] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [3] D. Rolnick and M. Tegmark, "The power of deeper networks for expressing natural functions," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.
- [4] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.
- [5] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3866–3878, Sep. 2020.
- [6] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Contr. Syst. Lett.*, vol. 2, no. 3, pp. 543–548, Jul. 2018.
- [7] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3050–3057, Apr. 2020.
- [8] F. L. Lewis, "Nonlinear network structures for feedback control," *Asian J. Control*, vol. 1, no. 4, pp. 205–228, 1999.
- [9] G. Joshi and G. Chowdhary, "Deep model reference adaptive control," in *Proc. IEEE Conf. Decis. Control*, Nice, France, 2019, pp. 4601–4608.
- [10] G. Joshi, J. Virdi, and G. Chowdhary, "Asynchronous deep model reference adaptive control," in *Proc. Conf. Robot Learn.*, 2020, pp. 984–1000.
- [11] R. Sun, M. L. Greene, D. M. Le, Z. I. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 193–198, Jan. 2021. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9337905>
- [12] D. M. Le, M. L. Greene, W. A. Makumi, and W. E. Dixon, "Real-time modular deep neural network-based adaptive control of nonlinear systems," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 476–481, May 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9432951>
- [13] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, vol. 30, 2013, p. 3.
- [14] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1319–1327.
- [15] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 73–82, Jan. 1987.
- [16] D. S. Bernstein, *Matrix Mathematics*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
- [17] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Syst. Mag.*, vol. 28, no. 3, pp. 36–73, Jun. 2008.
- [18] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Proc. Conf. Learn. Theory*, 2020, pp. 2306–2327.
- [19] F. L. Lewis, A. Yegildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, Mar. 1996.
- [20] R. Kamalapurkar, J. A. Rosenfeld, J. Klotz, R. J. Downey, and W. E. Dixon, "Supporting lemmas for RISE-based control methods," 2014, *arXiv:1306.3432*.
- [21] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York, NY, USA: Wiley, 1995.
- [22] R. Kamalapurkar, J. A. Rosenfeld, A. Parikh, A. R. Teel, and W. E. Dixon, "Invariance-like results for nonautonomous switched systems," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 614–627, Feb. 2019.
- [23] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Philadelphia, PA, USA: SIAM, 1990.

Event/Self-Triggered Multi-Agent System Rendezvous with Graph Maintenance

Federico M. Zegers, Dan P. Guralnik, and Warren E. Dixon

Abstract—This paper explores the rendezvous problem for a multi-agent system (MAS) with distance-limited, intermittent communication and sensing. Unlike previous works that provide specific event-triggered controllers, we provide a framework that characterizes a family of distributed event-triggered controllers leveraging non-singular edge-potentials to achieve approximate rendezvous while maintaining the initial distance-limited graph. The proposed framework excludes the possibility of Zeno behavior and accommodates the development of self-triggered controllers. The combination of continuous and impulsive dynamics results in a hybrid system, where the closed-loop dynamics of the MAS are presented and analyzed using hybrid differential inclusions. The approximate rendezvous problem is recast into a set stabilization problem and sufficient conditions of the rendezvous set are obtained through a Lyapunov-based analysis. Simulation results are provided to validate the development.

I. INTRODUCTION

Network communication is often expressed in terms of distance-based constraints: two agents may communicate if they are sufficiently close. Even though network connectivity is a necessary component for achieving MAS objectives, it is often the case that assumptions are introduced about the network remaining connected or that continuous communication along edges is always possible [1]–[5]. Ensuring connectivity in real-time poses non-trivial challenges, even in the presence of continuous communication [6].

Centralized and distributed variants of the connectivity maintenance problem under continuous inter-agent communication have been extensively studied [6]–[11], as well as in discrete-time [12], where connectivity maintenance can be addressed inductively. Conversely, few results on distributed event-triggered control with connectivity maintenance are available [13]–[15], all based on some version of consensus dynamics (i.e., $\dot{x} = -Lx$, where L is a weighted graph Laplacian). In general, there is no straightforward reduction of the event-triggered version of this problem to either the smooth or discrete-time settings.

The result in [13] presents an event-triggered control approach to approximate rendezvous for a MAS. Piecewise con-

tinuous state-dependent gains on the standard (unweighted) consensus dynamics are used to preserve the edges of a distance-based communication graph. However, the proposed event trigger requires continuous access to inter-agent displacements, which implies continuous demand for sensing and/or communication, where only the controller updates are intermittent. The result in [14] investigates formation control with event-triggered connectivity preservation. The objective is achieved with intermittent sensing and broadcasting. Maintenance of the initial graph is achieved using unbounded edge-tension functions, originally presented in [16], where the Laplacian is weighted by the tension functions. In practice, unbounded tension functions impose unnecessary restrictions on the space of initial configurations, due to actuation limitations. We propose an alternative treatment, inspired by [11], using new edge tension functions designed to support the connectivity maintenance argument while avoiding restrictions on the admissible set of initial configurations. Recently, the authors in [15] developed an event-triggered controller and observer that achieve leader-follower consensus while maintaining the edges of an initially connected graph. Edge preservation is achieved through the use of bounded edge-potentials and intermittent communication determined by event triggers.

The results in [14] and [15] provide inroads to MAS coordination with graph maintenance. These results share an approach of matching specific pre-selected time-based triggers to the MAS dynamics as a means to guarantee the desired properties (graph-maintenance, continuous-time completeness, and stability). In contrast, we derive characterizations of trigger properties required to guarantee each of these desiderata directly from the event-triggered consensus dynamics. In particular, the triggers proposed in this paper need not depend on time, unlike methods such as in [13]–[15].

This paper studies the event-triggered rendezvous problem while also seeking to preserve the edges of an initially connected graph for a MAS consisting of single-integrator agents. Our result differs from [13]–[15] in that we characterize a family of distributed event/self-triggered controllers that (a) utilize edge-potentials of class C^1 for graph maintenance; (b) ensure completeness of maximal solutions while excluding Zeno behavior; and (c) given a user-defined parameter $\nu > 0$, exponentially drive the MAS to ν -approximate rendezvous. By ν -approximate rendezvous, we mean a state of the MAS where all agent states are within a distance of at most ν from each other.

Federico M. Zegers, Dan P. Guralnik, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida 32603, e-mail: {fredzeg, danguralnik, wdixon}@ufl.edu.

This research is supported in part by A Task Order contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, Office of Naval Research Grant N00014-21-1-2481, NEEC award number N00174-18-1-0003, AFOSR award numbers FA9550-18-1-0109 and 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.

The novelty of this result is that it provides a general formal treatment of event/self-triggered control for MAS rendezvous with graph maintenance within the hybrid differential inclusions framework of [17]. Using hybrid time enables a particularly concise analysis of the interactions between the clocks of the individual agents, event triggers, and dynamical properties of the overall system in our setting. The theoretical guarantees are independent of the particular form of the trigger, allowing for a variety of trigger designs.

Finally, this paper expands the result of [11] in two significant ways. First, the new proposed edge-tension function removes the restriction from [11] on the length of the initial edges (which are required to be significantly shorter than the communication radius), and illuminate the trade-off between how close an edge is to breaking and the control effort required to preserve the edge. Second, we relax the need for continuous communication/sensing by employing an event/self-triggered control strategy. Simulation results for a particular instance of an event trigger using our theorems are presented to validate the development.

II. PRELIMINARIES

A. Notation

For any sets A, B , a function f of A with values in B is denoted by $f: A \rightarrow B$, whereas $f: A \rightrightarrows B$ refers to a *set-valued* function $f: A \rightarrow 2^B$. The set-complement of A is denoted by A^c . For $p, q, n \in \mathbb{Z}_{>0}$, the $p \times q$ zero matrix and the $p \times 1$ zero column vector are denoted by $0_{p \times q}$ and 0_p , respectively. The $p \times p$ identity matrix and the $p \times 1$ column vector of 1s are denoted by I_p and 1_p , respectively.

B. Graphs

Let \mathcal{V} be a finite non-empty set of cardinality N , and let n be a fixed positive integer. The *configuration space* over \mathcal{V} is defined as $\text{Conf}(\mathcal{V}) \triangleq (\mathbb{R}^n)^\mathcal{V}$. We will refer to $\mathbf{x} \triangleq (x_p)_{p \in \mathcal{V}} \in \text{Conf}(\mathcal{V})$ as a *configuration of particles in \mathbb{R}^n* . For $R > 0$, the R -threshold graph $\mathcal{G}_R(\mathbf{x})$ on a configuration \mathbf{x} is the undirected graph with vertex set \mathcal{V} and edge set $\mathcal{E}_R(\mathbf{x})$ defined by setting $pq \in \mathcal{E}_R(\mathbf{x})$ if and only if $\|x_p - x_q\| \leq R$, where we denote $pq \triangleq \{p, q\}$ for all $p, q \in \mathcal{V}$, $p \neq q$. Recall, for any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a path in \mathcal{G} connecting a vertex p to a vertex q is a sequence of vertices $(v_0 = p, \dots, v_k = q)$ where $k \in \mathbb{Z}_{\geq 0}$ and $v_{s-1}v_s \in \mathcal{E}$ for all $s = 1, 2, \dots, k$. The graph \mathcal{G} is connected, if every two vertices $p, q \in \mathcal{V}$ may be connected by a path in \mathcal{G} . The *neighborhood* \mathcal{N}_p of a vertex $p \in \mathcal{V}$ is the set of all $q \in \mathcal{V}$ with $pq \in \mathcal{E}$. The *degree* of p is $d_p \triangleq |\mathcal{N}_p|$ and $\Delta(\mathcal{G})$ denotes the maximum degree in \mathcal{G} . Let $\mathcal{A} \triangleq [a_{pq}] \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$ denote the adjacency matrix of \mathcal{G} , where $a_{pq} = 1$ if and only if $pq \in \mathcal{E}$ and $a_{pq} = 0$ otherwise. Within this work, no self-loops are considered. Therefore, $a_{pp} \triangleq 0$ for all $p \in \mathcal{V}$. The degree matrix Δ of \mathcal{G} is the diagonal matrix whose p -th diagonal entry is $\sum_{q \in \mathcal{V}} a_{pq}$. The Laplacian matrix of \mathcal{G} is defined as $\mathbf{L} \triangleq \Delta - \mathcal{A} \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$. More generally, if $c = [c_{pq}] \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$ is a non-negative symmetric matrix, the weighted Laplacian is defined as $\mathbf{L}_c \triangleq \Delta_c - \mathcal{A}_c$, where¹ $\mathcal{A}_c \triangleq \mathcal{A} \odot c$, $\Delta_c \triangleq$

$\text{diag}(\mathcal{A}_c \cdot 1_N)$, and known to be positive semidefinite. Let $\lambda_i(\mathbf{L}_c)$ denote the eigenvalues of \mathbf{L}_c in non-decreasing order, and let $\lambda_i(\mathcal{G}) \triangleq \lambda_i(\mathbf{L})$. If \mathcal{G} is connected then $\lambda_1(\mathcal{G}) = 0$ is a simple eigenvalue, and $\lambda_2(\mathcal{G})$, known as the *Fiedler value* of \mathcal{G} , is positive. Also, $\lambda_N(\mathbf{L}_c)$ coincides with the operator norm $\|\mathbf{L}_c\|$, since \mathbf{L}_c is self-adjoint.

C. Hybrid Systems

A hybrid differential inclusion \mathcal{H} takes the form [17]:

$$\mathcal{H}: \begin{cases} \dot{z} \in F(z), & z \in C, \text{ (flow constraint)} \\ z^+ \in G(z), & z \in D, \text{ (jump constraint)} \end{cases}$$

where $F: C \rightrightarrows \mathbb{R}^n$, $C \subset \mathbb{R}^n$ are the *flow map and set*; and $G: D \rightrightarrows \mathbb{R}^n$, $D \subset \mathbb{R}^n$ are the *jump map and set*, respectively, and z^+ indicates the value of the state after a jump.

Solutions of \mathcal{H} evolve continuously over the flow set according to the dynamics given by the flow map, and are allowed to execute discrete jumps over the jump set, constrained to the sets specified by the jump map. Formally, a set $A \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is a *hybrid time domain*, if there is a non-decreasing sequence of non-negative reals $(t_j)_{j=0}^m$, $m \in \mathbb{Z}_{\geq 0} \cup \{\infty\}$, $t_0 = 0$, $t_m \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, such that $A = \bigcup_{j=1}^m (I_j \times \{j-1\})$, where all the I_j , $j < m$ are of the form $[t_{j-1}, t_j]$, and I_m is of the form² $[t_{m-1}, t_m]$ or $[t_{m-1}, t_m)$ when $m < \infty$. We refer to $(t_j)_{j=0}^m$ as the *jump sequence* of the time domain A . A *hybrid arc* ϕ is a function $\phi: \text{dom } \phi \rightarrow \mathbb{R}^n$, where (1) $\text{dom } \phi \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is a hybrid time domain with jump sequence $(t_j)_{j=0}^m$; and (2) ϕ is a locally absolutely continuous function of I_j , for every j . A *solution* of \mathcal{H} is a hybrid arc ϕ such that, for all $j > 0$, (1) $\phi(t, j-1) \in C$ for almost all $t \in [t_{j-1}, t_j]$, and $\dot{\phi}(t, j-1) \in F(\phi(t, j-1))$ for almost all $t \in I_j$ (the *flow condition*); and (2) $\phi(t_{j-1}, j-1) \in D$ and $\phi(t_{j-1}, j) \in G(\phi(t_{j-1}, j-1))$ (the *jump condition*). A solution ϕ to \mathcal{H} is called *maximal* if ϕ cannot be extended, that is: if ψ is a solution with $\text{dom } \phi \subseteq \text{dom } \psi$, which coincides with ϕ on $\text{dom } \phi$, then $\psi = \phi$. A solution ϕ is called *complete* if $\text{dom } \phi$ is unbounded.

Definition 1. A solution ϕ of \mathcal{H} is said to be *t-complete* if the sequence (t_j) is unbounded.³

III. PROBLEM FORMULATION AND CONTROLLER DESIGN

Consider a cooperative MAS composed of $N \in \mathbb{Z}_{>0}$ agents indexed by a set \mathcal{V} , with states $x_p \in \mathbb{R}^n$, $p \in \mathcal{V}$. Any two agents $p, q \in \mathcal{V}$ are capable of exchanging information with each other whenever the distance between them does not exceed $R > 0$. Then, the possible connections among the agents are encoded by the R -threshold graph $\mathcal{G}_R(\mathbf{x})$. For each $p \in \mathcal{V}$, the model of agent p is a fully actuated single integrator, $\dot{x}_p = u_p$, where $u_p \in \mathbb{R}^n$ denotes a control input.

Assumption 1. The initial R -threshold graph, $\mathcal{G} \triangleq \mathcal{G}_R(\mathbf{x}(0))$, is connected, and every edge pq in $\mathcal{E} \triangleq \mathcal{E}_R(\mathbf{x}(0))$ satisfies $\|x_p - x_q\| < R$.

²Note $t_m = \infty$ is allowed when $m < \infty$; for $m = \infty$ there is no t_m .

³Clearly, a *t-complete* solution is complete.

¹ \odot, \otimes denote the Hadamard and Kronecker matrix products, respectively.

Assumption 2. For each $p \in \mathcal{V}$ and $q \in \mathcal{N}_p$, agent p is capable of measuring $x_p - x_q$ for all $t \geq 0$.

Definition 2. Let $\nu > 0$. The MAS is in a state of ν -approximate rendezvous if $\|x_p - x_q\| \leq \nu$ for all $p, q \in \mathcal{V}$.

A distributed controller is developed for driving the MAS to ν -approximate rendezvous while maintaining the initial graph structure throughout the process, in the sense that $\mathcal{E} \subseteq \mathcal{E}_R(\mathbf{x}(t))$ holds for all $t \geq 0$.

The developed controller limits communications and/or sensing to edges of the graph \mathcal{G} to remove the need for continual monitoring of their R -neighborhood. Instead, each agent can rely on peer-to-peer communication with a fixed set of neighbors that was established at time $t = 0$, as long as this communication can be guaranteed. In addition, any properties of the communication graph, such as the spectrum of $\mathbf{L}_\mathcal{G}$, may be computed in advance at time $t = 0$.

A. Potential Functions

Inspired by [11], edge-potentials are employed to preserve the edges of configurations \mathbf{x} , which support a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the sense that $\mathcal{E} \subseteq \mathcal{E}_R(\mathbf{x})$. Let $r : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ be a non-decreasing continuous function with $r(0) > 0$. Furthermore, let $P : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ be given by

$$P(\rho) \triangleq \int_0^\rho r(s) ds, \quad \rho \in \mathbb{R}_{\geq 0}. \quad (1)$$

The potential $V_{pq} : \text{Conf}(\mathcal{V}) \rightarrow \mathbb{R}_{\geq 0}$ for each $pq \in \mathcal{E}$ is

$$V_{pq}(\mathbf{x}) \triangleq P(\|x_p - x_q\|), \quad w_{pq} \triangleq r(\|x_p - x_q\|), \quad (2)$$

noting $V_{pq} = V_{qp}$, and that $w \triangleq [w_{pq}] \in \mathbb{R}_{\geq 0}^{\mathcal{V} \times \mathcal{V}}$ is a state-dependent symmetric matrix. The function r is specified as follows. First, $\varepsilon \in \mathbb{R}$ is selected so that $\tilde{R} \triangleq R(1 - \varepsilon)$ satisfies

$$R > \tilde{R} > \frac{2}{3}R \Leftrightarrow \frac{1}{3} > \varepsilon > 0. \quad (3)$$

Next, let $\mu, \omega > 0$, and let $r(s)$ be selected as

$$r(s) \triangleq \mu \cdot \begin{cases} 1, & s \in [0, \tilde{R}] \\ 1 + \omega(s^2 - \tilde{R}^2), & s \in [\tilde{R}, R] \\ 1 + \omega(R^2 - \tilde{R}^2), & s \geq R. \end{cases} \quad (4)$$

After some algebra, we obtain

$$\omega \geq \frac{2|\mathcal{E}|(1 - \varepsilon)^2}{R^2\varepsilon^2(2 - \varepsilon)^2} \Rightarrow |\mathcal{E}|P(\tilde{R}) \leq P(R). \quad (5)$$

Selecting ω according to (5) is required for graph maintenance (see Theorem 1). As in [11, Proposition 3.2], \tilde{R} plays the role of a safe communication distance below which the edge potential is the usual square of the distance. The goal is to prove that the initial graph will be preserved if all its edges are initially safe. However, our design differs from those considered in [11] in that r ramps up quadratically in the unsafe zone $[\tilde{R}, R]$, whereas the latter are either held constant or grow linearly there, which results in lower bounds on the buffer: $\varepsilon \geq 1 - |\mathcal{E}|^{-1/2}$ for constant r and $\varepsilon \geq 1 - (\frac{2}{3|\mathcal{E}| - 1})^{1/3}$ for linearly growing r . In contrast, our design enables the selection of a small enough ε at initialization.

B. Hybrid Controller and Closed-Loop Dynamics

Let $\mathcal{X} \triangleq \text{Conf}(\mathcal{V}) \times \text{Conf}(\mathcal{V}) \times [0, \infty)^\mathcal{V}$ denote the extended configuration space of the MAS. The controller of agent $p \in \mathcal{V}$ is designed as $u_p \triangleq \eta_p$, where $\eta_p \in \mathbb{R}^n$, $\tau_p \in \mathbb{R}$ are auxiliary variables subject to the following closed-loop dynamics:

$$\mathcal{H}: \begin{cases} \dot{x}_p = \eta_p, \\ \dot{\tau}_p = 1, \quad \dot{\eta}_p = 0_n, & \mathbf{T}_p(\boldsymbol{\xi}) > 0 \\ \tau_p^+ = 0, \quad \eta_p^+ = \sum_{q \in \mathcal{N}_p} w_{pq}(x_q - x_p), & \mathbf{T}_p(\boldsymbol{\xi}) = 0, \end{cases} \quad (6)$$

where:

- 1) $\boldsymbol{\xi} \triangleq [\mathbf{x}^\top, \boldsymbol{\eta}^\top, \boldsymbol{\tau}^\top]^\top \in \mathcal{X}$, $\boldsymbol{\eta} \triangleq (\eta_p)_{p \in \mathcal{V}} \in \text{Conf}(\mathcal{V})$ denotes the stacked vector of auxiliary variables η_p , and $\boldsymbol{\tau} \triangleq (\tau_p)_{p \in \mathcal{V}} \in [0, \infty)^\mathcal{V}$ is a vector of *personal clocks*, each of which resets whenever the agent triggers, i.e., $\mathbf{T}_p = 0$.
- 2) The *triggers*, $\mathbf{T}_p : \mathcal{X} \rightarrow \mathbb{R}$, $p \in \mathcal{V}$ are continuous functions satisfying $\mathbf{T}_p(\boldsymbol{\xi}^+) > 0$ for all $p \in \mathcal{V}$ whenever $\mathbf{T}_q(\boldsymbol{\xi}) = 0$ for any $q \in \mathcal{V}$,

and where it is understood that $x_p^+ = x_p$, $\eta_p^+ = \eta_p$ and $\tau_p^+ = \tau_p$ hold for any jump not triggered by the jump condition of agent p . All functions of \mathbf{x} will be regarded as functions of $\boldsymbol{\xi}$. Also, for any particular value of $\boldsymbol{\xi}$, we denote the projection of $\boldsymbol{\xi}$ to the first component of \mathcal{X} by $\mathbf{x}(\boldsymbol{\xi})$. Since the \mathbf{T}_p are continuous, the sets

$$C \triangleq \bigcap_{p \in \mathcal{V}} [\mathbf{T}_p > 0], \quad D \triangleq \overline{C} \cap \bigcup_{p \in \mathcal{V}} [\mathbf{T}_p = 0],$$

—the flow and jump sets of \mathcal{H} —are open and closed⁴, respectively, with $\overline{C} \subseteq C \cup D$. Both the flow and jump maps are single-valued and continuous, and solutions to the flow equations are global and unique.

Lemma 1. Every initial condition $\phi(0, 0) \in \overline{C}$ determines one and only one maximal solution ϕ of the hybrid system \mathcal{H} given in (6). Moreover, every maximal solution of \mathcal{H} is either t -complete or Zeno.

Proof. By [17, Proposition 2.10], and due to the preceding observations, maximal solutions of \mathcal{H} are complete. The condition on the triggers gives $G(D) \subset D^\complement$, implying no solution of \mathcal{H} has a pair of consecutive jumps. ■

C. MAS Control Objective

A configuration \mathbf{x} is a *rendezvous state*, if $x_p = x_q$ for all $p, q \in \mathcal{V}$. The set $\Delta_\mathcal{V}$ of all rendezvous states is a linear subspace of $\text{Conf}(\mathcal{V})$. Henceforth, let $\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{x}^\perp$ denote the orthogonal decomposition of \mathbf{x} with $\tilde{\mathbf{x}} \in \Delta_\mathcal{V}$ and $\mathbf{x}^\perp \in \Delta_\mathcal{V}^\perp$, and let $\Delta\mathbf{x} \triangleq (x_p - x_q)_{pq \in \mathcal{E}}$. It can be shown that:

$$\tilde{\mathbf{x}} = \mathbf{1}_N \otimes \frac{1}{N} \sum_{p \in \mathcal{V}} x_p, \quad \frac{1}{\sqrt{N}} \|\mathbf{x}^\perp\| \leq \|\Delta\mathbf{x}\|_\infty \leq \sqrt{2} \|\mathbf{x}^\perp\|. \quad (7)$$

Achieving rendezvous is equivalent to globally⁵ stabilizing the set $\mathcal{R} \triangleq \{\boldsymbol{\xi} \in \mathcal{X} : \|\Delta\mathbf{x}\|_\infty = 0\}$. Moreover, ν -approximate rendezvous is achieved whenever $\|\Delta\mathbf{x}\|_\infty \leq \nu$.

⁴Continuous pre-images of open/closed sets are open/closed, respectively.

⁵Over the space of configurations with connected communication graphs.

Therefore, it suffices to show that every trajectory of \mathcal{H} satisfying Assumption 1 is eventually contained in the set

$$\mathcal{R}_\nu \triangleq \{\xi \in \mathcal{X} : \|\Delta \mathbf{x}\|_\infty \leq \nu\}. \quad (8)$$

In addition, we require maintenance of the initial communication graph. To this end, if $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a connected graph, for any $\rho > 0$, let

$$\mathcal{C}_\rho(\mathcal{G}) \triangleq \{\xi \in \mathcal{X} : \mathcal{E} \subseteq \mathcal{E}_\rho(\mathbf{x})\}, \quad (9)$$

and note Assumption 1 means $\xi(0) \in \text{int}(\mathcal{C}_R(\mathcal{G}))$. Given an initial configuration $\mathbf{x}(0)$ satisfying Assumption 1, the initial condition $\xi(0) = \phi(0, 0)$ is set to satisfy

$$\phi(0, 0) = (\mathbf{x}(0), (\mathbf{L}_w \otimes I_n)\mathbf{x}(0), 0_N), \quad (10)$$

and a controller is provided—that is, the values of ε, μ, ω are determined—to guarantee $\xi(t) \in \mathcal{C}_R(\mathcal{G})$ for all $t \geq 0$, for the corresponding maximal solution ϕ of \mathcal{H} .

IV. GRAPH MAINTENANCE AND t -COMPLETENESS

The edge weights w_{pq} in (2) give rise to the weighted graph Laplacian matrix of the MAS, \mathbf{L}_w , defined in Section II-B, using the weights w_{pq} introduced in (2). Set

$$\zeta \triangleq -(\mathbf{L}_w \otimes I_n)\mathbf{x} - \eta. \quad (11)$$

Writing $\zeta = (\zeta_p)_{p \in \mathcal{V}} \in \text{Conf}(\mathcal{V})$, observe that

$$\zeta_p = \sum_{q \in \mathcal{N}_p} w_{pq} (x_q - x_p) - \eta_p \quad (12)$$

is the error between the instantaneous consensus term over \mathcal{N}_p and the sampled consensus term for agent p .

Definition 3. Let $\mathbf{T}_p : \mathcal{X} \rightarrow \mathbb{R}$, $p \in \mathcal{V}$ be a continuously differentiable function, let $\sigma \in (0, 1]$, and let

$$f_p(\xi) \triangleq \|\eta_p\|^2 - \|\zeta_p\|^2 + \sigma K \tilde{R}^2, \quad K \triangleq \frac{\lambda_2^2(\mathcal{G})\mu^2}{2N}. \quad (13)$$

\mathbf{T}_p is an *admissible trigger*, if $\mathbf{T}_p \leq f_p$ throughout \mathcal{X} and there exist $\mathfrak{h}, \mathfrak{m} > 0$ such that (a) $\mathbf{T}_p(\xi^+) \geq \mathfrak{h}$ at each jump of \mathcal{H} ; and (b) $\frac{d}{dt}\mathbf{T}_p \geq -\mathfrak{m}$ holds along any solution of \mathcal{H} .

Note that admissibility implies agent p must update η_p at some time earlier than dictated by the condition $f_p \leq 0$, which is only state-dependent (and not time-dependent). Also, the parameter σ , to be determined later, allows the user to control the degree of rendezvous approximation (see Theorem 2).

Theorem 1. Given $\tilde{R} = R(1 - \varepsilon)$ satisfying (3) and a connected graph \mathcal{G} , if $\{\mathbf{T}_p\}_{p \in \mathcal{V}}$ is a collection of admissible triggers, then every solution of \mathcal{H} satisfying (10) and initiating from $\mathcal{C}_{\tilde{R}}(\mathcal{G})$ remains in $\mathcal{C}_R(\mathcal{G})$, is t -complete, and the controllers u_p are bounded for all time.

Proof. Using (2), define the *total potential* $V_G : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ as

$$V_G(\xi) \triangleq \sum_{p \in \mathcal{V}} \sum_{q \in \mathcal{N}_p} V_{pq}(\mathbf{x}) = 2 \sum_{pq \in \mathcal{E}} V_{pq}(\mathbf{x}). \quad (14)$$

Note that $V_G(\xi) = 0$ for $\xi \in \mathcal{R}$ and $V_G(\xi) > 0$ otherwise. Let $\phi : \text{dom } \phi \rightarrow \mathcal{X}$ be a solution of \mathcal{H} with initial condition

$\phi(0, 0) \in \mathcal{C}_{\tilde{R}}(\mathcal{G})$. We will abuse notation by writing $\xi = \phi(t, j)$ for $(t, j) \in \text{dom } \phi$.

First we show that $\dot{V}_G(\xi) \leq 0$ whenever $\|x_p - x_q\| \geq \tilde{R}$ for some $pq \in \mathcal{E}$. During flows, the change in V_G is given by $\dot{V}_G(\xi) = \langle \nabla V_G(\xi), F(\xi) \rangle$, where F is the flow map of \mathcal{H} . It is known [8, Section 7.2] that $\nabla V_G(\xi) = 2(\mathbf{L}_w \otimes I_n)\mathbf{x}$ follows from (2). Therefore, upon substituting (11) we obtain

$$\dot{V}_G(\xi) = -2\mathbf{x}^\top (\mathbf{L}_w^2 \otimes I_n) \mathbf{x} - 2\mathbf{x}^\top (\mathbf{L}_w \otimes I_n) \zeta.$$

Splitting the leading term and applying (11) twice yields

$$\dot{V}_G(\xi) = -\mathbf{x}^\top (\mathbf{L}_w^2 \otimes I_n) \mathbf{x} - \|\eta\|^2 + \|\zeta\|^2.$$

By (2) and (4), $w_{pq} \geq \mu$ for all $pq \in \mathcal{E}$, which implies $\lambda_2(\mathbf{L}_w) \geq \mu\lambda_2(\mathcal{G})$. Together with (7), this yields

$$\begin{aligned} \dot{V}_G(\xi) &\leq -\lambda_2^2(\mathcal{G})\mu^2\|\mathbf{x}^\perp\|^2 - \|\eta\|^2 + \|\zeta\|^2 \\ &\leq -(NK\|\Delta \mathbf{x}\|_\infty^2 + \|\eta\|^2 - \|\zeta\|^2). \end{aligned} \quad (15)$$

Using $\|\Delta \mathbf{x}\|_\infty \geq \tilde{R}$, $\sigma K \tilde{R}^2 \leq K \tilde{R}^2$, and (13), we obtain

$$\dot{V}_G(\xi) \leq -\sum_{p \in \mathcal{V}} f_p(\xi) \leq -\sum_{p \in \mathcal{V}} \mathbf{T}_p(\xi) \leq 0 \quad (16)$$

by the admissibility of the \mathbf{T}_p and since all the \mathbf{T}_p are non-negative during flows.

We now show that the edges of \mathcal{G} are maintained. Proceeding by contradiction, let $(s_1, j_1) \in \text{dom } \phi$ be a point with $\phi(s_1, j_1) \notin \mathcal{C}_R(\mathcal{G})$, and let

$$(s_0, j_0) \triangleq \sup \left\{ (t, j) \in \text{dom } \phi \mid \begin{array}{l} (t, j) \leq (s_1, j_1) \\ \phi(t, j) \in \mathcal{C}_{\tilde{R}}(\mathcal{G}) \end{array} \right\},$$

using the order on \mathbb{R}^2 given by $\mathbf{a} \leq \mathbf{b} \Leftrightarrow \mathbf{b} - \mathbf{a} \in \mathbb{R}_{\geq 0}^2$. Since $\text{dom } \phi$ is closed in \mathbb{R}^2 under this order, $(s_0, j_0) \in \text{dom } \phi$. Let $V_1 \triangleq V_G(\phi(s_1, j_1))$ and $V_0 \triangleq V_G(\phi(s_0, j_0))$. Also, note $j_0 \leq j_1$ and $s_0 \in [t_{j_0}, t_{j_0+1})$. Since $\mathbf{x}(\phi(t_{j+1}, j)) = \mathbf{x}(\phi(t_{j+1}, j+1))$ for all $j \geq 0$, the expression $\mathbf{z}(t) \triangleq \mathbf{x}(\phi(t, j))$ is a well-defined function of $t \in J \triangleq [s_0, s_1]$. Moreover, it is C^1 -smooth at every $t \in J$ except for the points $t = t_j$, $j_0 \leq j \leq j_1$. Since V_G is a C^1 -function of \mathbf{x} , the function $v(t) \triangleq V_G(\mathbf{z}(t))$ is C^1 -smooth at every $t \in J$ except for the points $t = t_j$, $j_0 \leq j \leq j_1$. Therefore, $V_1 - V_0 = v(s_1) - v(s_0) = \int_{s_0}^{s_1} \dot{v}(t) dt$. Since $\phi(t, j) \notin \mathcal{C}_{\tilde{R}}(\mathcal{G})$ for $t \in (s_0, s_1]$, the integrand is non-positive by (16), which results in $V_1 \leq V_0$. Since \mathbf{x} evolves continuously with t and $\mathcal{C}_{\tilde{R}}(\mathcal{G})$ is closed, we have $\phi(s_0, j_0) \in \mathcal{C}_{\tilde{R}}(\mathcal{G})$. Therefore, by (5) and (14), $V_0 = 2 \sum_{pq \in \mathcal{E}} V_{pq}(\phi(s_0, j_0)) \leq 2|\mathcal{E}|P(\tilde{R}) \leq 2P(R)$. Since $\mathbf{x}(s_1)$ has at least one edge of length greater than R , $V_0 \leq 2P(R) < V_1$ —contradiction.

To prove the second assertion of the theorem, we suppose the solution ϕ is maximal and verify that ϕ is t -complete. By Lemma 1, a maximal solution $\phi : \text{dom } \phi \rightarrow \mathcal{X}$ of \mathcal{H} is either eventually continuous—in which case it is t -complete (and we are done)—or it has an infinite sequence of jump times t_j , $j \in \mathbb{Z}_{\geq 0}$. For each $p \in \mathcal{V}$, let J_p denote the set of indices $j > 0$ satisfying $\mathbf{T}_p(\phi(t_j, j-1)) = 0$, with the addition of the index $j = 0$. Fix a $p \in \mathcal{V}$ such that J_p is infinite, and denote the *hybrid jump times* $(t_j, j-1)$, $j \in J_p$ as $((t_i^p, j_i^p - 1))_{i=1}^\infty$, in increasing order. Also, let $t_0^p \triangleq 0$. Then, by Lemma 1, it

suffices to find $\delta > 0$ such that $t_{i+1}^p - t_i^p \geq \delta$ for all $i \geq 0$. For each $i \geq 1$, $\mathbf{T}_p(\phi(t_i^p, j_i^p) - 1) = 0$ and $\mathbf{T}_p(\phi(t_i^p, j_i^p)) \geq \mathfrak{h}$ by Definition 3(a). Moreover, the continuity of $\mathbf{T}_p(\phi(t, j))$ during flows implies $t_{i+1}^p - t_i^p > 0$.

We claim that $\delta = \mathfrak{h}/m$ satisfies our needs. Proceeding by contradiction, assume $t_{i+1}^p - t_i^p < \delta$. Substituting $\xi = \phi(t_{i+1}^p, j_i^p)$ into \mathbf{T}_p , together with the continuity of $\mathbf{T}_p(\phi(s, j_i^p))$ over all but finitely many $s \in [t_i^p, t_{i+1}^p]$, produces

$$\mathbf{T}_p(\xi) = \int_{t_i^p}^{t_{i+1}^p} \dot{\mathbf{T}}_p(\phi(s, j_i^p)) ds \geq \mathfrak{h} - m(t_{i+1}^p - t_i^p) > 0,$$

contradicting the fact that ξ is a jump point for agent p .

Finally, since $\|\Delta \mathbf{x}\|_\infty \leq R$ for all time, we see $w_{pq} \leq r(R)$ by (2), and $\|u_p\| = \|\eta_p\| \leq \Delta(\mathcal{G})Rr(R)$ by (4), for each $pq \in \mathcal{E}$ and $p \in \mathcal{V}$ —hence bounded. ■

V. STABILITY ANALYSIS

Theorem 2. Let $\nu > 0$, $\tilde{R} = R(1 - \varepsilon)$ satisfy (3), and \mathcal{G} be a connected graph. Suppose $0 < \beta \leq \sigma K \tilde{R}^2$ and $\{\mathbf{T}_p\}_{p \in \mathcal{V}}$ is a collection of admissible triggers such that, over solutions of \mathcal{H} satisfying (10) and initiating from $\mathcal{C}_{\tilde{R}}(\mathcal{G})$,

$$\mathbf{T}_p + \sigma K \tilde{R}^2 \leq f_p + \beta \quad (17)$$

holds for all $p \in \mathcal{V}$. Then, any such solution satisfies

$$\|\Delta \mathbf{x}(t)\|_\infty^2 \leq \frac{r(R)|\mathcal{E}|}{r(0)} \left(\|\Delta \mathbf{x}(0)\|_\infty^2 e^{-\frac{N K t}{r(R)|\mathcal{E}|}} + \sigma \tilde{R}^2 \right). \quad (18)$$

In this sense, \mathcal{R}_ν is exponentially stable⁶ for $\sigma = \frac{\nu^2 r(0)}{|\mathcal{E}| \tilde{R}^2 r(R)}$. In particular, for any $\nu' > \nu$, ν' -approximate rendezvous is achieved for every such maximal solution.

Proof. Let $\phi : \text{dom } \phi \rightarrow \mathcal{X}$ be a maximal solution of \mathcal{H} with initial condition $\phi(0, 0) \in \mathcal{C}_{\tilde{R}}(\mathcal{G})$. By Theorem 1, ϕ is t -complete. Consider the same total potential $V_{\mathcal{G}}(\xi)$ defined in (14). Using (1), (2), (4), and (14), we can bound $V_{\mathcal{G}}$ as

$$r(0)\|\Delta \mathbf{x}\|^2 \leq V_{\mathcal{G}}(\xi) \leq r(R)\|\Delta \mathbf{x}\|^2. \quad (19)$$

By (17), $\|\eta_p\|^2 - \|\zeta_p\|^2 \geq -\beta + \mathbf{T}_p$. Then, (15) yields

$$\begin{aligned} \dot{V}_{\mathcal{G}}(\xi) &\leq -N K \|\Delta \mathbf{x}\|_\infty^2 - \sum_{p \in \mathcal{V}} (-\beta + \mathbf{T}_p) \\ &\leq -N K \|\Delta \mathbf{x}\|_\infty^2 + N \beta. \end{aligned} \quad (20)$$

Combining (19) with $\|\Delta \mathbf{x}\| \leq \sqrt{|\mathcal{E}|} \|\Delta \mathbf{x}\|_\infty$ implies $V_{\mathcal{G}}(\xi) \leq r(R) |\mathcal{E}| \|\Delta \mathbf{x}\|_\infty^2$. Then, (20) can be upper bounded as

$$\dot{V}_{\mathcal{G}}(\xi) \leq -\frac{N K}{r(R)|\mathcal{E}|} V_{\mathcal{G}}(\xi) + N \beta. \quad (21)$$

Since $V_{\mathcal{G}}$ only depends on \mathbf{x} (14), and \mathbf{x} evolves continuously under \mathcal{H} , $V_{\mathcal{G}}$ evolves continuously as well. By Theorem 1, $\dot{V}_{\mathcal{G}}(\xi)$, $\xi = \phi(t, j)$ has only finitely many discontinuities in any bounded sub-interval of $\text{dom } \phi$. Therefore, integrating (21) and recalling that $\beta \leq \sigma K \tilde{R}^2$ yields

$$V_{\mathcal{G}}(\phi(t, j)) \leq V_{\mathcal{G}}(\phi(0, 0)) e^{-\frac{N K t}{r(R)|\mathcal{E}|}} + \sigma r(R) |\mathcal{E}| \tilde{R}^2,$$

⁶Recalling (7), Equation (18) implies the exponential stability of the ν -neighborhood of $\Delta_{\mathcal{V}}$ with respect to $\|\cdot\|$, in the same restricted sense.

where using the bounds of $V_{\mathcal{G}}$ in (19) and $\|\Delta \mathbf{x}\|_\infty \leq \|\Delta \mathbf{x}\| \leq \sqrt{|\mathcal{E}|} \|\Delta \mathbf{x}\|_\infty$ produces (18) for any $(t, j) \in \text{dom } \phi$. ■

VI. TRIGGER DESIGN

Lemma 2. Let \tilde{R}, r, ω satisfy the requirements in Section III-A, and let \mathcal{G} be a connected graph. If $\{\mathbf{T}_p\}_{p \in \mathcal{V}}$ is a collection of admissible triggers and ϕ is a solution of \mathcal{H} with $\phi(0, 0) \in \mathcal{C}_{\tilde{R}}(\mathcal{G})$ satisfying (10), then the following holds over every flow interval of agent p : $\|\zeta_p(\phi(t, j))\| \leq Z \tau_p(\phi(t, j))$, where $Z \triangleq 2\Delta(\mathcal{G}) (\mu \omega R^2 + \Delta(\mathcal{G}) R r(R)^2)$.

Proof. Using the notation from the proof of Theorem 1, consider any $t \in [t_i^p, t_{i+1}^p]$. By (6), η_p is constant over the interval $[t_i^p, t_{i+1}^p] \times \{j_i^p\} \subset \text{dom } \phi$. Furthermore, Theorem 1 and Equations (2)–(4) imply $\|x_p - x_q\| \leq R$ and $|w_{pq}| \leq r(R)$ for all $q \in \mathcal{N}_p$, respectively. It then follows that

$$\|\eta_p\| \leq d_p R r(R) \leq \Delta(\mathcal{G}) R r(R). \quad (22)$$

Taking the time derivative of ζ_p over $[t_i^p, t_{i+1}^p] \times \{j_i^p\}$ yields

$$\dot{\zeta}_p = \sum_{q \in \mathcal{N}_p} \dot{w}_{pq}(x_q - x_p) + \sum_{q \in \mathcal{N}_p} w_{pq}(\eta_q - \eta_p). \quad (23)$$

From (4), it follows that $\dot{w}_{pq}(s) = 0$ for $s \in [0, \tilde{R}]$, and $\dot{w}_{pq}(s) = 2\mu \omega s \dot{s}$ for $s \in [\tilde{R}, R]$. Therefore, $\|\dot{w}_{pq}\| \leq 2\mu \omega R$ because \dot{s} is a unit vector. Combining these bounds with (23),

$$\|\dot{\zeta}_p\| \leq 2\mu \omega d_p R^2 + d_p r(R) \cdot 2\Delta(\mathcal{G}) R r(R) \leq Z,$$

where we recall that d_p is the degree of p in \mathcal{G} and $\Delta(\mathcal{G})$ is the maximal degree in \mathcal{G} . Over the interval $[t_i^p, t_{i+1}^p] \times \{j_i^p\}$, ζ_p is continuous as a function of t , and differentiable everywhere except $t = t_j$, $j_i^p \leq j \leq j_{i+1}^p$. From $\|\zeta_p(\phi(t_i^p, j_i^p))\| = 0$ obtain $\|\zeta_p(t)\| \leq \int_{t_i^p}^t \|\dot{\zeta}_p(s)\| ds \leq Z(t - t_i^p)$. Since $\tau_p = t - t_i^p$ over $[t_i^p, t_{i+1}^p] \times \{j_i^p\}$, $\|\zeta_p(t, j_i^p)\| \leq Z \tau_p$. ■

Corollary 1. The trigger $\mathbf{T}_p = f_p$ is admissible.⁷

Let $0 < \beta \leq \sigma K \tilde{R}^2$ and $\gamma > 0$. Consider the triggers: $\mathbf{T}_{p,1}(\xi) = \|\eta_p\|^2 - \|\zeta_p\|^2 + \alpha(\|\eta_p\|)$ and $\mathbf{T}_{p,2}(\xi) = \|\eta_p\|^2 - Z^2 \tau_p^2 + \alpha(\|\eta_p\|)$, where $\alpha(s) \triangleq \beta - \frac{\beta}{\gamma} s$ for $s \in [0, \gamma]$ and $\alpha(s) = 0$ otherwise. Observe that $\mathbf{T}_{p,1}$ is state-based, while $\mathbf{T}_{p,2}$ is a self-trigger.

Corollary 2. $\mathbf{T}_{p,1}$ and $\mathbf{T}_{p,2}$ are admissible and satisfy (17).⁷

On first glance, the event trigger $\mathbf{T}_{p,1}$ requires agent p to continuously measure $x_q - x_p$ for each $q \in \mathcal{N}_p$. This could be useful in settings where intermittent actuation is desired and continuous measurements are inexpensive, such as with satellite constellations where power is limited and visual measurements can be made over large distances.

For other scenarios, an alternative communication protocol can remove the need for continuously monitoring the neighbors' states. Recalling that $\dot{x}_p = \eta_p$ is constant over $[t_i^p, t_{i+1}^p]$, observe that $x_p(t) = \eta_p \tau_p + x_p(t_i^p)$ for all $t \in [t_i^p, t_{i+1}^p]$. Under the assumption of instantaneous communication, it suffices for each agent $q \in \mathcal{N}_p$ to broadcast η_q and x_q at each jump time of agent q . Consequently, p can compute

⁷Proof omitted due to space constraints. Available upon request.

ζ_p by using the solution for $x_q(t)$ over the appropriate time interval for each $q \in \mathcal{N}_p$.

VII. IMPLEMENTATION AND SIMULATION

Putting together the theoretical results, suppose we are given N agents with $\mathbf{x}(0)$ satisfying Assumption 1, and $\nu > 0$. At the initial time, $\mathcal{G} = \mathcal{G}_R(\mathbf{x}(0))$, $\lambda_2(\mathcal{G})$, and $|\mathcal{E}|$ are computed. The parameter ε is selected so that $\tilde{R} > \|\Delta\mathbf{x}(0)\|_\infty$, and ω is selected to satisfy (5). For any choice of $\mu, \gamma > 0$, σ is selected according to Theorem 2, guaranteeing that any collection $\{\mathbf{T}_p\}_{p \in \mathcal{V}}$ of admissible triggers generates a controller driving the MAS to ν' -approximate rendezvous at an exponential rate, uniformly over initial conditions $\phi(0,0) \in \mathcal{C}_{\tilde{R}}(\mathcal{G})$ satisfying (10). Theorem 1 guarantees that no edges of \mathcal{G} are broken at any time for any of the above initial conditions. Figure 1 shows the trajectories of a MAS with $N = 9$ agents in the Euclidean plane \mathbb{R}^2 , with randomized initial positions and with $\mathbf{T}_{p,1}$ (with the default choice of $\beta = \sigma K \tilde{R}^2$) serving as the event trigger function for each agent.

VIII. CONCLUSION

Distributed controllers with intermittent distance-limited communication were developed for the rendezvous problem. The hybrid differential inclusions framework of [17] enables a systematic approach to the analysis of the closed-loop system, tying together personal clocks, triggering conditions, stability, and topological properties of solutions. This results in a family of controllers which yield complete, Zeno-free solutions while maintaining the initial graph structure and keeping the prescribed approximate rendezvous set exponentially stable with respect to continuous time. In contrast with [11], where the growth of the edge potentials is insufficient for providing a graph maintenance guarantee for all initial configurations with a connected communication graph, and where this guarantee shrinks as the number of agents grows, our design fits any such configuration with a controller capable of maintaining the initial graph. Future research may address less conservative triggers, more complex agent dynamics, and heterogeneous networks. Accounting for perturbations is especially important in the context of the current construction, since it does not satisfy the hybrid basic conditions.

REFERENCES

- [1] F. Zegers, M. Hale, J. M. Shea, and W. E. Dixon, "Reputation-based event-triggered formation control and leader tracking with resilience to byzantine adversaries," in *Proc. Am. Control Conf.*, 2020, pp. 761–766.
- [2] F. M. Zegers, M. T. Hale, J. M. Shea, and W. E. Dixon, "Event-triggered formation control and leader tracking with resilience to byzantine adversaries: A reputation-based approach," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 3, pp. 1417–1429, 2021.
- [3] C. Nowzari and J. Cortés, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.
- [4] W. Ren, "Consensus based formation control strategies for multi-vehicle systems," in *Proc. Am. Control Conf.*, 2006, pp. 4237–4242.
- [5] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.

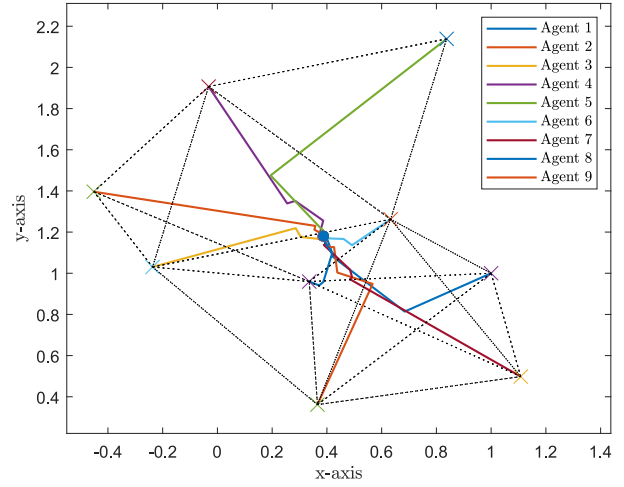


Fig. 1. Trajectories of a MAS with $N = 9$ agents, communication radius $R = 1$, and a random initial configuration (\times 's) with a connected initial graph \mathcal{G} (black-dashed lines). \mathcal{G} has $|\mathcal{E}| = 20$ and $\lambda_2(\mathcal{G}) = 1.578$. The final configuration (\bullet 's) were required to be in ν -approximate rendezvous with $\nu = 0.1$. The parameter \tilde{R} is selected as the length of the longest edge of \mathcal{G} plus the machine epsilon of our computer (10^{-16}), which yields $\tilde{R} = 0.9392$, corresponding to $\varepsilon = 0.0608$. Then, $\omega = 2541.5$, and $\sigma = 1.8863 \times 10^{-6}$. With the gain $\mu = 1$, approximate rendezvous was achieved within 4 time units. The triggers $\mathbf{T}_{p,1}$ were used with $\beta = 3.8298 \times 10^{-13}$ and γ set to $\gamma = 0.1$. Note how each cusp along an agent's trajectory coincides with a jump time for that agent. Simulation executed in MATLAB, using [18].

- [6] M. Zavlanos and G. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 812–816, Aug. 2007.
- [7] D. Spanos and R. Murray, "Robust connectivity of networked vehicles," in *Proc. IEEE Conf. Decis. Control*, vol. 3, 2004, pp. 2893–2898.
- [8] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010, vol. 33.
- [9] Z. Kan, A. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1827–1832, 2012.
- [10] Z. Kan, E. Doucette, and W. E. Dixon, "Distributed connectivity preserving target tracking with random sensing," *IEEE Trans. Autom. Control*, vol. 64, no. 5, pp. 2166–2173, 2019.
- [11] D. Boskos and D. Dimarogonas, "Robustness and invariance of connectivity maintenance control for multiagent systems," *SIAM J. Control Optim.*, vol. 55, no. 3, pp. 1887–1914, 2017.
- [12] M. Fiacchini and I.-C. Morarescu, "Convex conditions on decentralized control for graph topology preservation," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1640–1645, 2014.
- [13] Y. Fan and G. Hu, "Connectivity-preserving rendezvous of multi-agent systems with event-triggered controllers," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 234–239.
- [14] X. Yi, J. Wei, D. V. Dimarogonas, and K. H. Johansson, "Formation control for multi-agent systems with connectivity preservation and event-triggered controllers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9367–9373, 2017.
- [15] Y. Dong and S. Xu, "Rendezvous with connectivity preservation problem of linear multiagent systems via parallel event-triggered control strategies," *IEEE Trans. Cybern.*, to appear.
- [16] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, Aug. 2007.
- [17] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling Stability, and Robustness*. Princeton University Press, Princeton, NJ, 2012.
- [18] R. Sanfelice, D. Copp, and P. Nanez, "A toolbox for simulation of hybrid systems in Matlab/Simulink: Hybrid equations (HyEQ) toolbox," in *Proc. Intern. Conf. Hybrid Syst.: Comput. Control*, 2013, pp. 101–106.

Consensus over Clustered Networks with Asynchronous Inter-Cluster Communication

Federico M. Zegers, Sean Phillips, and Warren E. Dixon

Abstract—Within this work, the consensus problem for a clustered multi-agent system (C-MAS) is investigated. Given a multi-agent system (MAS), the agents are organized into disjoint clusters, where each cluster forms a connected network. The agents that are within the same cluster can communicate continuously with their neighbors. Between some cluster pairs, there exists an inter-cluster that enables the relay of information between the two clusters. Agents that have neighbors in clusters different from their own can communicate intermittently and asynchronously with their different-cluster neighbors. The goal of each agent in the C-MAS is to converge to an agreement value by sharing local information of a continuous-time homogeneous process. Note that the intermittent communication events of the continuous-time process are inherently hybrid. Therefore, a unique coupling between a static consensus protocol and a hybrid consensus protocol is designed. The closed-loop network model is presented using a hybrid systems framework. The consensus problem is then recast into a set stability problem and sufficient conditions of the consensus set are presented through leveraging a Lyapunov-based stability analysis. A C-MAS composed of 16 agents that is partitioned into 3 clusters and 2 inter-clusters is used to illustrate the results through numerical simulations and shown to achieve consensus using the proposed control strategy.

I. INTRODUCTION

The consensus problem is a fundamental and widely explored area of research within the controls and network science communities [1]–[5]. Consensus can be found in numerous applications, such as in the rendezvous of mobile robots [6], [7], attitude synchronization between multiple satellites [8], [9], and verification of distributed ledgers to promote blockchain security [10]. While early results on consensus were built on undirected connected communication graphs [11], works like [12] enable consensus through the use of weight-balanced directed communication graphs that are infinitely often jointly strongly connected. Because continuous communication between neighbors in a network places a constraint on the physical distance between agent pairs, control algorithms constructed on infinitely often jointly connected graphs, which allow for the intermittent relaxation of physical distance constraints, have grown in interest [13]–[15].

Federico M. Zegers and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida 32603, e-mail: fredzeg@ufl.edu, wdixon@ufl.edu.

Sean Phillips is with the Air Force Research Laboratory, Space Vehicles Directorate, Kirtland AFB, NM 87117

This research is supported, in part, by the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, Office of Naval Research Grant N00014-13-1-0151, NEEC award number N00174-18-1-0003, AFOSR award numbers FA9550-18-1-0109 and FA9550-19-1-0169.

This document is approved for public release AFRL-2020-0195.

978-1-6654-4197-1/\$31.00 ©2021 AACC

Akin to the idea of intermittently connected graphs, clustered multi-agent systems (MASs) have also been an active area of research since they impart high-level task division among sub-groups of the whole MAS. For example, the work in [16] presents a distributed cluster consensus control algorithm, where if the communication between clusters is directed and acyclic, then each cluster can achieve local consensus for both fixed and switching graphs. The authors in [17] study the clustered formation control problem for MASs with intermittent and delayed communication. Given a MAS that is divided into M clusters, the result presents a distributed controller that enables each cluster to assemble a unique formation.

While cluster control strategies grant each component of a MAS autonomy, these same techniques can be leveraged to obtain system-wide consensus while reducing the communication between clusters. Recently, [18] proposed a distributed controller that enables system-wide consensus for a clustered MAS (C-MAS), i.e., a MAS that is partitioned into clusters, consisting of agents with Linear Time-Invariant (LTI) dynamics. The result designates each cluster a single leader, where only the leaders of each cluster are allowed to intermittently communicate through a fixed directed graph. Furthermore, the communication topology within each cluster is modeled as a fixed and directed graph, where agents in the same cluster can continuously communicate with their neighbors. While [18] relaxes the communication load when compared to systems that require continuous communication in the entire MAS, the result is limited in the sense that each cluster can only have a single leader, where all cluster leaders communicate intermittently at the same time. This implies that all cluster leaders must be physically close to enable communication.

Motivated by [18] and [19], we develop a distributed controller that enables a C-MAS composed of agents with LTI dynamics to obtain system-wide consensus, where we do not restrict the number of agents that can communicate between clusters and allow for the communication between clusters to be both intermittent and asynchronous. Therefore, cluster pairs need only be close to intermittently satisfy communication requirements. For simplicity, all agents in the same cluster are allowed to continuously communicate with their neighbors, where the communication topology is modeled with an undirected graph. Similarly, the communication topology between cluster pairs is also modeled with an undirected graph. Due to the combination of continuous and impulsive dynamics, we use hybrid systems theory to model the C-MAS, the controller, and the network topologies, where we apply a Lyapunov theorem for asymptotic stability of sets for hybrid systems.

4249

A simulation study for a C-MAS composed of 16 agents that is partitioned into 3 clusters and 2 inter-clusters is presented to confirm the result. Due to space constraints, all proofs are omitted but are available upon request.

II. PRELIMINARIES

A. Notation

Let \mathbb{R} and \mathbb{Z} denote the set of real numbers and integers, respectively. We also write $\mathbb{R}_{\geq x} \triangleq [x, \infty)$, $\mathbb{R}_{> x} \triangleq (x, \infty)$, $\mathbb{R}_{< x} \triangleq (-\infty, x)$, $\mathbb{Z}_{\geq x} \triangleq \mathbb{R}_{\geq x} \cap \mathbb{Z}$, and $\mathbb{Z}_{> x} \triangleq \mathbb{R}_{> x} \cap \mathbb{Z}$ for $x \in \mathbb{R}$. For $p, q, n \in \mathbb{Z}_{>0}$, the $p \times q$ zero matrix and the $p \times 1$ zero column vector are denoted by $0_{p \times q}$ and 0_p , respectively. The $p \times p$ identity matrix and the $p \times 1$ column vector of ones are denoted by I_p and 1_p , respectively. The Euclidean norm of a vector $r \in \mathbb{R}^p$ is denoted by $\|r\| \triangleq \sqrt{r^\top r}$. For $x, y \in \mathbb{R}^n$, the inner product between x and y is denoted by $\langle x, y \rangle$, where $\langle x, y \rangle = x^\top y$. Given a positive integer M , define $[M]$ as the set $\{1, 2, \dots, M\}$. The Kronecker product of $A \in \mathbb{R}^{p \times q}$ and $B \in \mathbb{R}^{u \times v}$ is denoted by $(A \otimes B) \in \mathbb{R}^{pu \times qv}$. The block diagonal matrix whose diagonal blocks consist of G_1, G_2, \dots, G_n is denoted by $\text{diag}(G_1, G_2, \dots, G_n)$. The i^{th} eigenvalue of a symmetric matrix $G \in \mathbb{R}^{p \times p}$ is denoted by $\lambda_i(G) \in \mathbb{R}$. The maximum and minimum eigenvalues of G are denoted by $\lambda_{\max}(G) \in \mathbb{R}$ and $\lambda_{\min}(G) \in \mathbb{R}$, respectively.

B. Graph Theory

Let $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E}, \mathcal{A})$ denote a static, weighted, and undirected graph with node set $\mathcal{V} \triangleq [N]$, for some $N \in \mathbb{Z}_{>0}$, edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and symmetric weighted adjacency matrix $\mathcal{A} \triangleq [a_{ik}] \in \mathbb{R}^{N \times N}$. The edge $(i, k) \in \mathcal{E}$ if and only if node i can send information to node k . Since \mathcal{G} is undirected, $(i, k) \in \mathcal{E}$ if and only if $(k, i) \in \mathcal{E}$. An undirected graph is connected if and only if there exists a sequence of edges in \mathcal{E} between any two distinct nodes. The neighbor set of node i is denoted by $\mathcal{N}_i \triangleq \{k \in \mathcal{V} : (k, i) \in \mathcal{E}\}$. Within this work, no self-loops are considered, and therefore, $a_{ii} \triangleq 0$ for all $i \in \mathcal{V}$. Moreover, $a_{ik} > 0$ if $(k, i) \in \mathcal{E}$, and $a_{ik} = 0$ if $(k, i) \notin \mathcal{E}$. The degree matrix of \mathcal{G} is defined as a diagonal matrix such that $\Delta \triangleq [\Delta_{ik}] \in \mathbb{R}^{N \times N}$, where $\Delta_{ik} \triangleq 0$ for all $i \neq k$, and $\Delta_{ii} \triangleq \sum_{k \in \mathcal{V}} a_{ik}$. The Laplacian matrix of \mathcal{G} is denoted by $L \in \mathbb{R}^{N \times N}$ and defined as $L \triangleq \Delta - \mathcal{A}$. Let $L_C \in \mathbb{R}^{N \times N}$ denote the Laplacian of the complete graph on N nodes.

C. Cluster and Inter-Cluster Subgraphs

In this section, we introduce notation of the particular graph used in the remainder of this paper. The nodes in \mathcal{V} can be grouped into $M \in \mathbb{Z}_{>0}$, $M < N$, disjoint clusters indexed by $[M]$ and defined by the cluster set $\mathcal{C} \triangleq \{\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_M\}$, where \mathcal{C} is a partition of \mathcal{V} . Specifically, \mathcal{C} being a partition of \mathcal{V} means $\mathcal{V}_p \subset \mathcal{V}$ for all $p \in [M]$, $\mathcal{V}_p \cap \mathcal{V}_q = \emptyset$ for each distinct $p, q \in [M]$, and $\cup_{p \in [M]} \mathcal{V}_p = \mathcal{V}$. Note that a cluster can contain a single node, i.e., $|\mathcal{V}_p| = 1$.

Each cluster induces a subgraph of \mathcal{G} . The induced subgraph of cluster p is defined by $\mathcal{G}[\mathcal{V}_p] \triangleq (\mathcal{V}_p, \mathcal{E}[\mathcal{V}_p], \mathcal{A}[\mathcal{V}_p])$, where $\mathcal{E}[\mathcal{V}_p] \triangleq \{(i, k) \in \mathcal{E} : i, k \in \mathcal{V}_p\}$ and $\mathcal{A}[\mathcal{V}_p] \in \mathbb{R}^{|\mathcal{V}_p| \times |\mathcal{V}_p|}$ is determined from $\mathcal{E}[\mathcal{V}_p]$ and the edge weights in \mathcal{A} . Given a set of nodes $S \subset \mathcal{V}$, the induced subgraph of \mathcal{G} is undirected since \mathcal{G} is undirected. Moreover, we only consider $S \subset \mathcal{V}$ such

that $\mathcal{G}[S]$ is connected. The induced inter-cluster subgraph of \mathcal{G} between distinct clusters p and q is defined by $\mathcal{G}[\mathcal{V}_{pq}] \triangleq (\mathcal{V}_{pq}, \mathcal{E}[\mathcal{V}_{pq}], \mathcal{A}[\mathcal{V}_{pq}])$, where $\mathcal{V}_{pq} \triangleq \{i \in \mathcal{V}_p : (i, k) \in \mathcal{E}, k \in \mathcal{V}_q\} \cup \{i \in \mathcal{V}_q : (i, k) \in \mathcal{E}, k \in \mathcal{V}_p\}$, $\mathcal{E}[\mathcal{V}_{pq}] \triangleq \{(i, k) \in \mathcal{E} : i, k \in \mathcal{V}_{pq} \text{ and } (i, k) \notin \mathcal{E}[\mathcal{V}_p] \cup \mathcal{E}[\mathcal{V}_q]\}$, and $\mathcal{A}[\mathcal{V}_{pq}] \in \mathbb{R}^{|\mathcal{V}_{pq}| \times |\mathcal{V}_{pq}|}$ is determined from $\mathcal{E}[\mathcal{V}_{pq}]$ and the edge weights in \mathcal{A} .

To simplify the development, we only focus on clustered graph structures with at most one inter-cluster subgraph between any two distinct cluster subgraphs. Hence, given M cluster subgraphs, there are $M^* \in \{0, \dots, M(M-1)/2\}$ inter-cluster subgraphs. The inter-cluster subgraphs can be indexed by $[M^*]$, where every $pq \in [M] \times [M]$ uniquely corresponds to some $r \in [M^*]$ through the use of an enumeration.¹ Therefore, for all $pq \in [M] \times [M]$ there exists an $r \in [M^*]$, where we now refer to inter-cluster pq by inter-cluster r , i.e., $\mathcal{V}^r \triangleq \mathcal{V}_{pq}$. Note that $\mathcal{G}[\mathcal{V}_{pq}] = \mathcal{G}[\mathcal{V}^r]$ if and only if $\mathcal{V}_{pq} = \mathcal{V}^r$. Hence, the cluster subgraphs of \mathcal{G} are denoted by $\mathcal{G}[\mathcal{V}_p]$ for $p \in [M]$, and the inter-cluster subgraphs of \mathcal{G} are denoted by $\mathcal{G}[\mathcal{V}^r]$ for $r \in [M^*]$. By construction, the union graph composed from all cluster and inter-cluster subgraphs is equal to \mathcal{G} . A MAS composed of N agents that is divided into clusters according to some cluster set \mathcal{C} is henceforth referred to as a *clustered MAS*.

Each subgraph may have an adjacency matrix with size that is different from $N \times N$. To facilitate the subsequent analysis, we now construct an $N \times N$ augmented adjacency matrix for each subgraph of \mathcal{G} . Let $S \subset \mathcal{V}$. The augmented adjacency matrix of $\mathcal{G}[S]$ is defined by $A[S] \triangleq [a_{ik}] \in \mathbb{R}^{N \times N}$, where $a_{ik} \triangleq a_{ik}$ if $(i, k) \in \mathcal{E}[S]$ and $a_{ik} \triangleq 0$ otherwise. Hence, the augmented adjacency matrix of cluster p is denoted by $A[\mathcal{V}_p]$. Similarly, the augmented adjacency matrix of inter-cluster r is denoted by $A[\mathcal{V}^r]$. The augmented adjacency matrix corresponding to the union of all clusters is defined by $A_0 \triangleq \sum_{p \in [M]} A[\mathcal{V}_p]$. The Laplacian matrix of the union of all cluster subgraphs is defined by $L_0 \triangleq \text{diag}(A_0 \cdot 1_N) - A_0 \in \mathbb{R}^{N \times N}$. The Laplacian matrix of inter-cluster $r \in [M^*]$ is defined by $L_r \triangleq \text{diag}(A[\mathcal{V}^r] \cdot 1_N) - A[\mathcal{V}^r] \in \mathbb{R}^{N \times N}$. Hence, $L = L_0 + \sum_{r \in [M^*]} L_r$.

D. Hybrid Systems

A hybrid system \mathcal{H} with state $z \in \mathbb{R}^n$ and data (C, f, D, G) is defined as

$$\mathcal{H} \triangleq \begin{cases} \dot{z} = f(z), & z \in C, \\ z^+ \in G(z), & z \in D, \end{cases} \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes the flow map that models continuous behavior, $C \subset \mathbb{R}^n$ denotes the flow set over which the system continuously evolves, $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ denotes the set-valued jump map that models discrete behavior, and $D \subset \mathbb{R}^n$ denotes the jump set over which the system discretely evolves. A solution ϕ to the hybrid system \mathcal{H} is parameterized with respect to hybrid-time as denoted by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$. Observe that t denotes continuous time, and j denotes discrete time. The domain $\text{dom } \phi \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is called a hybrid time domain if for all $(T, J) \in \text{dom } \phi$, $\text{dom } \phi \cap ([0, T] \times$

¹An enumeration is a bijective mapping from $[M] \times [M]$ to $[M^*]$.

$\{0, 1, \dots, J\}$) can be expressed as $\cup_{j=0}^J (I_j \times \{j\})$, where $I_j \triangleq [t_j, t_{j+1}]$ for a time sequence $0 = t_0 \leq t_1 \leq \dots \leq t_{j+1}$.² Note that t_j indicates the j^{th} instance the state z jumps. A solution ϕ to \mathcal{H} is called maximal if ϕ cannot be extended, that is, ϕ is not a truncated version of another solution. A solution is called complete if its domain is unbounded. A solution is said to be precompact if it is complete and bounded. The set $\mathcal{S}_{\mathcal{H}}$ contains all maximal solutions to \mathcal{H} , where the set $\mathcal{S}_{\mathcal{H}}(\xi)$ contains all maximal solutions to \mathcal{H} from ξ . A hybrid system $\mathcal{H} = (C, f, D, G)$ with data in (1) is said to satisfy the *hybrid basic conditions* if it satisfies the conditions in [20, Assumption 6.5].

Definition 1. [21] (*global exponential stability*) Let a hybrid system \mathcal{H} be defined on \mathbb{R}^n . Let $\mathcal{A} \subset \mathbb{R}^n$ be closed. The set \mathcal{A} is said to be *globally exponentially stable* (GES) for \mathcal{H} if there exist $\kappa, \alpha > 0$ such that every maximal solution ϕ to \mathcal{H} is complete and satisfies

$$|\phi(t, j)|_{\mathcal{A}} \leq \kappa \exp(-\alpha(t + j)) |\phi(0, 0)|_{\mathcal{A}}$$

for each $(t, j) \in \text{dom } \phi$.

III. PROBLEM STATEMENT

Consider a C-MAS composed of N agents in the node set \mathcal{V} , where each agent is assigned to one cluster for all time. Each agent computes a homogeneous process³ such that the local version of the process for i^{th} agent evolves according to the following continuous-time differential equation

$$\dot{x}_i(t) = Ax_i(t) + Bu_i(t), \quad (2)$$

where $x_i : [0, \infty) \rightarrow \mathbb{R}^n$ defines the state, $A \in \mathbb{R}^{n \times n}$ denotes the known state matrix, $B \in \mathbb{R}^{n \times d}$ denotes the known control effectiveness matrix, and $u_i : [0, \infty) \rightarrow \mathbb{R}^d$ defines the control input.

Assumption 1. For all $i \in \mathcal{V}$ and $t \geq 0$, agent i can measure its state, $x_i(t)$.

As indicated in Section II-C, the underlying communication network is modeled through a partitioning of the node set \mathcal{V} to form disjoint clusters with inter-cluster links leading to the following assumption.

Assumption 2. Let M connected cluster subgraphs and $M^* \geq M - 1$ connected inter-cluster subgraphs be given.

In this work, the C-MAS under consideration is such that all agents in each cluster can communicate continuously with their neighbors within the same cluster, while all agents contained in each inter-cluster can only communicate intermittently with their neighbors within the same inter-cluster. More precisely, for each $p \in [M]$ and $i \in \mathcal{V}_p$, the process state of agent i , namely $x_i(t)$, is broadcast to all agents $k \in \mathcal{N}_i \cap \mathcal{V}_p$ for all $t \in \mathbb{R}_{\geq 0}$. However, for each $r \in [M^*]$ and $i \in \mathcal{V}^r$, the state of agent i is broadcast to each agent

$k \in \mathcal{N}_i \cap \mathcal{V}^r$ only at isolated times. Specifically, let $\{t_w^r\}_{w=1}^{\infty}$ be an increasing sequence of broadcast times, where t_w^r denotes the w^{th} instance that the state of agent $i \in \mathcal{V}^r$ is broadcast to all agents $k \in \mathcal{N}_i \cap \mathcal{V}^r$ for each $i \in \mathcal{V}^r$. For each inter-cluster $r \in [M^*]$, let $0 < T_1^r \leq T_2^r$ denote lower and upper bounds, respectively, for the difference between consecutive broadcast times in $\{t_w^r\}_{w=1}^{\infty}$, i.e.,

$$T_1^r \leq t_{w+1}^r - t_w^r \leq T_2^r \quad (3)$$

for all $w \in \mathbb{Z}_{\geq 1}$ and $t_1^r \leq T_2^r$. For conciseness, let $\mathcal{N}_i^0 \triangleq \mathcal{N}_i \cap \mathcal{V}_p$ such that \mathcal{V}_p denotes the cluster of agent i and $\mathcal{N}_i^q \triangleq \mathcal{N}_i \cap \mathcal{V}^q$ for $q \in [M^*]$. Therefore, \mathcal{N}_i^0 contains the neighbors of agent i that are in the same cluster as agent i , and \mathcal{N}_i^q contains the neighbors of agent i that are in inter-cluster q .

The objective is to design a distributed controller for each agent $i \in \mathcal{V}$ that ensures $\{x_i(t)\}_{i \in \mathcal{V}}$ achieves consensus through the use of the communication graph \mathcal{G} whose cluster subgraphs and inter-cluster subgraphs undergo continuous communication and intermittent communication, respectively. Let the disagreement between the state of agent i and the average state of the C-MAS be defined by $e_i : [0, \infty) \rightarrow \mathbb{R}^n$ such that

$$e_i(t) \triangleq x_i(t) - \frac{1}{N} \sum_{\ell \in \mathcal{V}} x_{\ell}(t). \quad (4)$$

The C-MAS is said to achieve consensus if it satisfies

$$\lim_{t \rightarrow \infty} \|e_i(t)\| = 0 \quad \forall i \in \mathcal{V}.$$

From the construction of (4), it follows that the difference in the state of agents $i, k \in \mathcal{V}$ can be alternatively expressed as

$$e_k - e_i = x_k - x_i. \quad (5)$$

IV. CONSENSUS CONTROL DESIGN AND HYBRID SYSTEM MODELING

For brevity, all time dependencies are henceforth omitted unless otherwise necessary. The broadcast times of inter-cluster r are determined by discrete time instances governed by the set of points $\{t_w^r\}_{w=1}^{\infty}$. Following our previous work in [19], we define a timer state $\tau_r : [0, \infty) \rightarrow [0, T_2^r]$ whose hybrid dynamics are given by⁴

$$\begin{aligned} \dot{\tau}_r &= -1, & \tau_r &\in [0, T_2^r] \\ \tau_r^+ &\in [T_1^r, T_2^r], & \tau_r &= 0, \end{aligned} \quad (6)$$

where $0 < T_1^r \leq T_2^r$ are the constraints on the broadcast times given in (3). At each timer reset, i.e., when $\tau_r = 0$, the w^{th} broadcast time corresponding to the agents in $\mathcal{G}[\mathcal{V}^r]$ is set equal to the current time.

Suppose agent i is assigned to cluster p , i.e., $i \in \mathcal{V}_p$. The information coming into each agent may occur at drastically different time instances, namely, both continuously and intermittently. The consensus controller of agent i is, therefore, designed as

$$u_i \triangleq K \left(\eta_i + \sum_{r \in [M^*]} \eta_{i,r} \right), \quad (7)$$

²Given a set-valued mapping $H : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$, the domain of H is the set $\text{dom } H \triangleq \{x \in \mathbb{R}^m : H(x) \neq \emptyset\}$ [20, Definition 2.1].

³Homogeneous process means that all agents in the C-MAS interact with the same system. In the LTI system in (2), the A and B matrices are the same for all agents.

⁴Timers like (6) can also be developed for each agent to enable intermittent state broadcasting.

where $K \in \mathbb{R}^{d \times n}$ is a to be designed matrix, $\eta_i : [0, \infty) \rightarrow \mathbb{R}^n$ defines agent i 's control component corresponding to state information coming from cluster p , and $\eta_{i,r} : [0, \infty) \rightarrow \mathbb{R}^n$ defines agent i 's control component corresponding to state information coming from inter-cluster r . With respect to agent i , the component of (7) that uses the continuously available state information from same-cluster neighbors is given by the following static controller

$$\eta_i \triangleq \sum_{k \in \mathcal{N}_i^0} a_{ik}(x_k - x_i). \quad (8)$$

Similarly, the component in (7) that uses the intermittently available state information from neighbors in inter-cluster r evolves according to the sample-and-hold dynamics given by the following hybrid system

$$\begin{aligned} \dot{\eta}_{i,r} &= 0_n, & \tau_r &\in [0, T_2^r] \\ \eta_{i,r}^+ &= \sum_{k \in \mathcal{N}_i^r} a_{ik}(x_k - x_i), & \tau_r &= 0. \end{aligned} \quad (9)$$

Recall that $\mathcal{N}_i^r = \mathcal{N}_i \cap \mathcal{V}^r$, which denotes the set of neighbors of agent i that are in inter-cluster r . We now derive the closed-loop error system for the local consensus error of agent i as defined in (4). Substituting (2) and (7) into the time derivative of (4) yields

$$\begin{aligned} \dot{e}_i &= BK \left(\eta_i - \sum_{\ell \in \mathcal{V}} \frac{\eta_\ell}{N} \right) + BK \sum_{r \in [M^*]} \left(\eta_{i,r} - \sum_{\ell \in \mathcal{V}} \frac{\eta_{\ell,r}}{N} \right) \\ &\quad + Ae_i. \end{aligned} \quad (10)$$

To facilitate the subsequent stability analysis for the closed-loop C-MAS, let $e \triangleq [e_1^\top, e_2^\top, \dots, e_N^\top]^\top \in \mathbb{R}^{nN}$ denote the stacked consensus error, $\eta \triangleq [\eta_1^\top, \eta_2^\top, \dots, \eta_N^\top]^\top \in \mathbb{R}^{nN}$ denote the cluster control components, $\theta_r \triangleq [\eta_{1,r}^\top, \eta_{2,r}^\top, \dots, \eta_{N,r}^\top]^\top \in \mathbb{R}^{nN}$ denote the control components for inter-cluster r , $\Theta \triangleq [\theta_1^\top, \theta_2^\top, \dots, \theta_{M^*}^\top]^\top \in \mathbb{R}^{nNM^*}$ denote the control components for all inter-clusters, and $\tau \triangleq [\tau_1, \tau_2, \dots, \tau_{M^*}]^\top \in \mathcal{T} \triangleq [0, T_2^1] \times \dots \times [0, T_2^{M^*}]$ denote the stacked timer states for all inter-clusters. Moreover, recall that $L_C \in \mathbb{R}^{N \times N}$ denotes the Laplacian of the complete graph on N nodes. It follows from (10) that the stacked consensus error dynamics are given by

$$\begin{aligned} \dot{e} &= \frac{1}{N} (L_C \otimes BK) \eta + \frac{1}{N} (L_C \otimes BK) \sum_{r \in [M^*]} \theta_r \\ &\quad + (I_N \otimes A) e. \end{aligned} \quad (11)$$

Substituting (5) and (8) into η for all $i \in \mathcal{V}$ yields

$$\eta = -(L_0 \otimes I_n) e. \quad (12)$$

Substituting (12) into (11) yields

$$\dot{e} = \bar{A}e + \bar{B} \sum_{r \in [M^*]} \theta_r, \quad (13)$$

where $\bar{A} \triangleq (I_N \otimes A) - (\frac{1}{N} L_C L_0 \otimes BK) \in \mathbb{R}^{nN \times nN}$ and $\bar{B} \triangleq \frac{1}{N} (L_C \otimes BK) \in \mathbb{R}^{nN \times nN}$. Note that the stacked consensus error denoted by e is naturally a continuous-time system. Therefore, at each jump, which occurs when $\tau_r = 0$

for some $r \in [M^*]$, the closed-loop dynamics of the stacked consensus error are updated as $e^+ = e$. Next, observe that substituting (9) into the time derivative of θ_r for all $i \in \mathcal{V}$ and $r \in [M^*]$ yields

$$\dot{\theta}_r = 0_{nN}. \quad (14)$$

At jumps, i.e., when $\tau_r = 0$ for some $r \in [M^*]$, the substitution of (5) and (9) into θ_r for all $i \in \mathcal{V}$ and $r \in [M^*]$ yields

$$\theta_r^+ = -(L_r \otimes I_n) e. \quad (15)$$

To further facilitate the development, we introduce the following variables. Let

$$\tilde{\theta}_r \triangleq -(L_r \otimes I_n) e - \theta_r \quad (16)$$

and $\tilde{\Theta} \triangleq [\tilde{\theta}_1^\top, \tilde{\theta}_2^\top, \dots, \tilde{\theta}_{M^*}^\top]^\top \in \mathbb{R}^{nNM^*}$. Substituting (13) and (14) into the time derivative of (16) yields

$$\dot{\tilde{\theta}}_r = -(L_r \otimes I_n) \bar{A}e - (L_r \otimes I_n) \bar{B} \sum_{\ell \in [M^*]} \theta_\ell. \quad (17)$$

Substituting $e^+ = e$ and (15) into the time difference of (16) for $r \in [M^*]$ yields

$$\tilde{\theta}_r^+ = 0_{nN}. \quad (18)$$

Substituting (16) into (13) and (17) yields

$$\dot{e} = (\bar{A} - \bar{B}\bar{L}) e - \bar{B} (1_{M^*} \otimes I_{nN}) \tilde{\Theta}, \quad (19)$$

and

$$\begin{aligned} \dot{\tilde{\theta}}_r &= -(L_r \otimes I_n) (\bar{A} - \bar{B}\bar{L}) e \\ &\quad + (L_r \otimes I_n) \bar{B} (1_{M^*} \otimes I_{nN}) \tilde{\Theta}, \end{aligned} \quad (20)$$

respectively, where

$$\bar{L} \triangleq \sum_{\ell \in [M^*]} (L_\ell \otimes I_n).$$

We now construct the closed-loop C-MAS hybrid dynamics denoted by \mathcal{H} . Let $\xi \triangleq [e^\top, \tilde{\Theta}^\top, \tau^\top]^\top \in \mathcal{X}$ denote the C-MAS state variable, where $\mathcal{X} \triangleq \mathbb{R}^{nN} \times \mathbb{R}^{nNM^*} \times \mathcal{T}$ denotes the state space of the hybrid system. Furthermore, let $z \triangleq [e^\top, \tilde{\Theta}^\top]^\top \in \mathbb{R}^{nN} \times \mathbb{R}^{nNM^*}$. The flow set of the hybrid system is defined as $C \triangleq \mathcal{X}$, and the flow map $f : \mathcal{X} \rightarrow \mathcal{X}$ is defined as

$$f(\xi) \triangleq \begin{bmatrix} \mathbf{A}z \\ -1_{M^*} \end{bmatrix}, \quad (21)$$

where

$$\begin{aligned} \mathbf{A} &\triangleq \begin{bmatrix} \bar{A} - \bar{B}\bar{L} & -\mathbf{A}^* \\ -\mathbf{L} (1_{M^*} \otimes (\bar{A} - \bar{B}\bar{L})) & -\mathbf{L} (1_{M^*} \otimes \mathbf{A}^*) \end{bmatrix} \\ \mathbf{A}^* &\triangleq \bar{B} (1_{M^*}^\top \otimes I_{nN}), \quad \mathbf{L} \triangleq \text{diag} (L_1 \otimes I_n, \dots, L_{M^*} \otimes I_n). \end{aligned}$$

Let $\kappa_1 \triangleq nN(1 + M^*)$, and observe that $\mathbf{A} \in \mathbb{R}^{\kappa_1 \times \kappa_1}$. The jump set is given by $D \triangleq \cup_{r \in [M^*]} D_r$, where $D_r \triangleq \{\xi \in \mathcal{X} : \tau_r = 0\}$. When $\tau_r = 0$, a jump for the r^{th} inter-cluster occurs, where e is mapped to itself by $e^+ = e$, $\tilde{\theta}_r$ is mapped to 0_{nN} by (18), and τ_r is mapped to some $\nu_r \in [T_1^r, T_2^r]$ by (6). For $k \in [M^*] \setminus \{r\}$, the variables $e, \tilde{\theta}_k,$

and τ_k evolve according to (19), (20), and (6), respectively. The jump map $G : \mathcal{X} \rightrightarrows \mathcal{X}$ is defined as

$$G(\xi) \triangleq \{G_r(\xi) : \xi \in D_r, r \in [M^*]\},$$

$$G_r(\xi) \triangleq \begin{bmatrix} e \\ [\tilde{\theta}_1^\top, \tilde{\theta}_2^\top, \dots, 0_{nN}^\top, \dots, \tilde{\theta}_{M^*}^\top]^\top \\ [\tau_1, \tau_2, \dots, [T_1^r, T_2^r], \dots, \tau_{M^*}]^\top \end{bmatrix}, \quad (22)$$

where 0_{nN} and $[T_1^r, T_2^r]$ are in the r^{th} position of $\tilde{\Theta}$ and τ , respectively.

Due to the definition of G , the solutions to the hybrid system \mathcal{H} are inherently non-unique. Nevertheless, using hybrid systems analysis, we will show that, under certain sufficient conditions, the subsequently defined consensus set \mathcal{A} is globally exponentially stable. With respect to the objective, it then follows that the set to stabilize is given by

$$\mathcal{A} \triangleq \left\{ \xi \in \mathcal{X} : \xi = [0_{nN}^\top, 0_{nNM^*}^\top, \nu^\top]^\top, \nu \in \mathcal{T} \right\}, \quad (23)$$

for the hybrid system \mathcal{H} with data (C, f, D, G) .⁵

V. STABILITY ANALYSIS

Before introducing the main results, we first begin by presenting two supporting lemmas.

Lemma 1. (cf. [22, Lemma 3.5]) Let $0 < T_1^r \leq T_2^r$ for each $r \in [M^*]$. Every $\phi \in \mathcal{S}_{\mathcal{H}}$ satisfies the following.

1. ϕ is complete, i.e., $\text{dom } \phi$ is unbounded.
2. For each $(t, j) \in \text{dom } \phi$, $(\frac{j}{N} - 1)T_1^{\min} \leq t \leq \frac{j}{N}T_2^{\max}$, where $T_1^{\min} \triangleq \min\{T_1^r : r \in [M^*]\}$ and $T_2^{\max} \triangleq \max\{T_2^r : r \in [M^*]\}$.
3. For all $j \in \mathbb{Z}_{\geq 0}$ such that $(t_{(j+1)N}, (j+1)N) \in \text{dom } \phi$, $(t_{jN}, jN) \in \text{dom } \phi$, $t_{(j+1)N} - t_{jN} \in [T_1^{\max}, T_2^{\max}]$.

Lemma 2. Assumption 2 is satisfied if and only if the graph \mathcal{G} is connected.

Next, we present useful objects that aid the stability analysis. Let $U \triangleq I_N - \frac{1}{N}1_N1_N^\top \in \mathbb{R}^{N \times N}$, where $U^2 = U$ [23]. Recall that $L \in \mathbb{R}^{N \times N}$ denotes the Laplacian of the C-MAS. Since \mathcal{G} is connected by Lemma 2 and undirected by construction, L is symmetric and positive semi-definite. Since L is symmetric, L is diagonalizable, where $L = SDS^{-1}$ such that $S \in \mathbb{R}^{N \times N}$ is an orthonormal eigenvector matrix and $D \in \mathbb{R}^{N \times N}$ is a diagonal eigenvalue matrix. Let $\lambda_1, \lambda_2, \dots, \lambda_N$ denote the eigenvalues of L , and recall that $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N$ for a connected and undirected graph \mathcal{G} . Let $s_i \in \mathbb{R}^N$ denote the i^{th} eigenvector of L that corresponds to λ_i , $\Psi \triangleq [s_2, s_3, \dots, s_N] \in \mathbb{R}^{N \times (N-1)}$, and $\tilde{\Psi} \triangleq \Psi \otimes I_n \in \mathbb{R}^{nN \times n(N-1)}$. Let

$$\Gamma \triangleq [I_{nN(1+M^*)} \quad 0_{nN(1+M^*) \times M^*}],$$

and define $|\xi|_{\mathcal{A}} \triangleq \inf \{\|\Gamma(\xi - y)\| : y \in \mathcal{A}\}$. Observe that $z = \Gamma\xi$ and $|\xi|_{\mathcal{A}} = \|z\|$. Let $\kappa_2 \triangleq n(N-1)(1+M^*)$.

Theorem 1. Suppose Assumptions 1 and 2 are satisfied. Given $0 < T_1^r \leq T_2^r$ for all $r \in [M^*]$, the set \mathcal{A} is globally exponentially stable for the hybrid system \mathcal{H} with

data in (21) and (22) if there exists a scalar $\sigma > 0$, gain matrix $K \in \mathbb{R}^{d \times n}$, and symmetric positive definite matrices $P \in \mathbb{R}^{n(N-1) \times n(N-1)}$ and $Q_r \in \mathbb{R}^{n(N-1) \times n(N-1)}$ for each $r \in [M^*]$ such that

$$M(\tau) \triangleq \tilde{\mathbf{A}}^\top R(\tau) + R(\tau) \tilde{\mathbf{A}} - \tilde{\mathbf{R}}(\tau) < 0_{\kappa_2 \times \kappa_2} \quad \forall \tau \in \mathcal{T}, \quad (24)$$

where $R(\tau) \triangleq \text{diag}(P, Q_1 e^{\sigma \tau_1}, \dots, Q_{M^*} e^{\sigma \tau_{M^*}}) \in \mathbb{R}^{\kappa_2 \times \kappa_2}$, $\tilde{\Psi} \triangleq I_{(1+M^*)} \otimes \tilde{\Psi} \in \mathbb{R}^{\kappa_1 \times \kappa_2}$, $\tilde{\mathbf{A}} \triangleq \tilde{\Psi}^\top \mathbf{A} \tilde{\Psi} \in \mathbb{R}^{\kappa_2 \times \kappa_2}$, and $\tilde{\mathbf{R}}(\tau) \triangleq \text{diag}(0_{nN \times nN}, \sigma Q_1 e^{\sigma \tau_1}, \dots, \sigma Q_{M^*} e^{\sigma \tau_{M^*}}) \in \mathbb{R}^{\kappa_2 \times \kappa_2}$.

The bilinear matrix inequality (BMI) in (24) must be satisfied for all $\tau \in \mathcal{T}$ to ensure the result in Theorem 1. Hence, given matrices P and Q_r for $r \in [M^*]$, one must verify that these matrices satisfy (24) for infinitely many points, i.e., for each $\tau \in \mathcal{T}$. However, we can leverage the BMI's structure to construct an equivalent BMI that is a convex combination of (24) evaluated at two subsequently defined endpoints. Hence, satisfying (24) at these endpoints ensures (24) is satisfied for all $\tau \in \mathcal{T}$. Before we present the equivalent BMI, we introduce the following objects. For each $r \in [M^*]$, let the function $\vartheta_r : [0, T_2^r] \rightarrow [0, 1]$ be defined as

$$\vartheta_r(w) \triangleq \frac{\exp(\sigma w) - \exp(\sigma T_2^r)}{1 - \exp(\sigma T_2^r)}. \quad (25)$$

Also, let $\epsilon \in [0, 1]$, $\nu \triangleq [\nu_1, \dots, \nu_{M^*}]^\top$, $\bar{\nu} \triangleq [\epsilon, \nu^\top]^\top$, $T_2 \triangleq [T_2^1, T_2^2, \dots, T_2^{M^*}]^\top$, $\tilde{Q}(\nu) \triangleq \text{diag}(Q_1 e^{\sigma \nu_1}, \dots, Q_{M^*} e^{\sigma \nu_{M^*}})$, $\bar{R}(\nu) \triangleq \text{diag}(\vartheta_1(\nu_1), \dots, \vartheta_{M^*}(\nu_{M^*})) \otimes I_{n(N-1)}$, and $\mathcal{R}(\bar{\nu}) \triangleq \text{diag}(\epsilon I_{n(N-1)}, \bar{R}(\nu))$.

Proposition 1. Let $0 < T_1^r \leq T_2^r$ for each $r \in [M^*]$. The inequality in (24) holds if there exists a scalar $\sigma > 0$ and symmetric positive definite matrices P and Q_r for each $r \in [M^*]$ satisfying $M(0_{M^*}) < 0_{\kappa_2 \times \kappa_2}$ and $M(T_2) < 0_{\kappa_2 \times \kappa_2}$, where

$$M(\nu) = \mathcal{R}(\bar{\nu}) M(0_{M^*}) + (I_{\kappa_2} - \mathcal{R}(\bar{\nu})) M(T_2). \quad (26)$$

Remark 1. Using Proposition 1, we can satisfy the BMI in (24) for all $\tau \in \mathcal{T}$ provided $M(0_{M^*})$ and $M(T_2)$ are negative definite.

VI. SIMULATION EXAMPLE

For the simulation, we consider a C-MAS composed of 16 agents that is partitioned into 3 clusters and 2 inter-clusters. Each agent computes a homogeneous process that is modeled by (2) and

$$A \triangleq \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad B \triangleq \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The C-MAS is defined using $\mathcal{V}_1 \triangleq \{1, 2, 3, 4, 5\}$, $\mathcal{V}_2 \triangleq \{6, 7, 8, 9, 10\}$, $\mathcal{V}_3 \triangleq \{11, 12, 13, 14, 15, 16\}$, $\mathcal{V}^1 \triangleq \{1, 3, 10\}$, and $\mathcal{V}^2 \triangleq \{9, 14\}$. The simulation was conducted with the Hybrid Systems Toolbox in MATLAB [24]. The network communication parameters are: $T_1^1 = 0.1$, $T_1^2 = 0.1$, $T_2^1 = 1$, $T_2^2 = 2$, and $\sigma = 1$. The controller gain matrix was set to $K = [26.5, 10.5]$, which was selected using pole placement and $-5, -5.5$ as desired eigenvalues. It can shown that K satisfies the BMI in (26), more specifically, it can be shown that there exists matrices P , Q_r for each $r \in [M^*]$ and scalar σ such that (26) holds. Each element in $e(0)$ and $\tilde{\Theta}(0)$

⁵Note that \mathcal{A} is closed since the countable product of closed sets is closed.

was randomly drawn from a uniform distribution over $[-5, 5]$. Moreover, each timer was initialized such that $\tau_r(0) = T_2^r$ for $r \in \{1, 2\}$. Figure 1 illustrates the simulation results and demonstrates that the consensus objective is achieved.

With respect to Figure 1, the first 10 time units of the simulation are presented since the norm of the errors in the top subplot converge below 0.2 for all $t \geq 10$. In the middle subplot, once the timer state τ_r reaches 0, the timer state is randomly reset by drawing a new initial condition from a uniform distribution over $[T_1^r, T_2^r]$. The asynchronous communication in the C-MAS is depicted by the distinct instances that the inter-cluster communication timers reach 0.

In the bottom subplot, the large initial values of V are caused by the large P , Q_1 , and Q_2 matrices used to satisfy the bilinear matrix inequality in (26). The P , Q_1 , and Q_2 matrices are omitted due to space constraints. The breakout window in the bottom plot depicts the end behavior of V . Observe that inter-cluster communication events result in the downward jump of V , which is otherwise exponentially decreasing. However, inter-cluster communication causes the norm of the corresponding consensus error to increase.

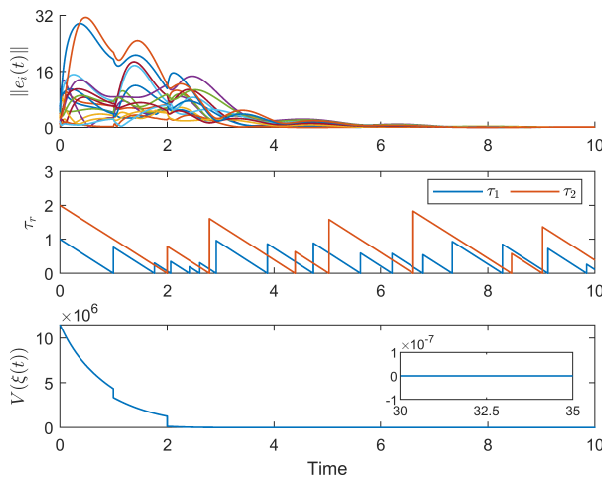


Figure 1. The top plot shows the norm of the consensus error for each agent in the C-MAS. The middle plot shows the evolution of the timer states for the two inter-clusters. The bottom plot shows the evolution of the Lyapunov functional V .

VII. CONCLUSION

This paper studies the consensus problem for clustered multi-agent systems, where the objective is to drive all agent states to a common value. In the C-MAS, each agent computes a homogeneous process with a distributed controller that only uses locally available information. The controller is continuously updated with state information from neighbors in the same cluster and intermittently updated with state information from neighbors in different clusters. Despite the clustered graph structure, which may result in disconnected distanced-based communication graphs as modeled through the timer dynamics, we can still achieve system-wide consensus. Future works may consider uncertain process models that are subject to disturbances. We may also further explore how large the maximum allowable transmission interval can be made before consensus is no longer possible.

REFERENCES

- [1] W. Yu, W. X. Zheng, G. Chen, W. Ren, and J. Cao, "Second-order consensus in multi-agent dynamical systems with sampled position data," *Automatica*, vol. 47, no. 7, pp. 1496–1503, 2011.
- [2] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. Am. Control Conf.*, 2005, pp. 1859–1864.
- [3] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.
- [4] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [5] L. Ding, Q.-L. Han, and G. Guo, "Network-based leader-following consensus for distributed multi-agent systems," *Automatica*, vol. 49, no. 7, pp. 2281–2286, 2013.
- [6] T. Gustavi, D. V. Dimarogonas, M. Egerstedt, and X. Hu, "Sufficient conditions for connectivity maintenance and rendezvous in leader-follower networks," *Automatica*, vol. 46, no. 1, pp. 133–139, 2010.
- [7] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, 2007.
- [8] X. Wang, J. Wu, and X. Wang, "Distributed attitude consensus of spacecraft formation flying," *J. Syst. Engin. Elec.*, vol. 24, no. 2, pp. 296–302, 2013.
- [9] J. Li and K. D. Kumar, "Decentralized fault-tolerant control for satellite attitude synchronization," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 3, pp. 572–586, 2011.
- [10] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *Proc. Int. Conf. Advan. Comp. Comm. Syst.*, 2017, pp. 1–5.
- [11] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. IFAC World Congress*, 2005, pp. 1–6.
- [12] S. S. Kia, J. Cortés, and S. Martinez, "Dynamic average consensus under limited control authority and privacy requirements," *Int. J. Robust Nonlin. Control*, vol. 25, no. 13, pp. 1941–1966, 2015.
- [13] H. Yu and X. Xia, "Adaptive consensus of multi-agents in networks with jointly connected topologies," *Automatica*, vol. 48, no. 8, pp. 1783–1790, 2012.
- [14] D. Mateos-Núñez and J. Cortés, "Distributed online convex optimization over jointly connected digraphs," *IEEE Trans. Netw. Sci. Engin.*, vol. 1, no. 1, pp. 23–37, 2014.
- [15] P. Lin and Y. Jia, "Consensus of a class of second-order multi-agent systems with time-delay and jointly-connected topologies," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 778–784, 2010.
- [16] J. Qin and C. Yu, "Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition," *Automatica*, vol. 49, no. 9, pp. 2898–2905, 2013.
- [17] X. Ge, Q.-L. Han, and X.-M. Zhang, "Achieving cluster formation of multi-agent systems under aperiodic sampling and communication delays," *IEEE Trans. Indus. Elec.*, vol. 65, no. 4, pp. 3417–3426, 2017.
- [18] V. T. Pham, N. Messai, and N. Manamanni, "Impulsive observer-based control in clustered networks of linear multi-agent systems," *IEEE Trans. Netw. Sci. Engin.*, 2019.
- [19] S. Phillips and R. G. Sanfelice, "Robust distributed synchronization of networked linear systems with intermittent information," *Automatica*, vol. 105, pp. 323–333, 2019.
- [20] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid dynamical systems: modeling stability, and robustness*. Princeton University Press, Princeton, NJ, 2012.
- [21] A. R. Teel, F. Forni, and L. Zaccarian, "Lyapunov-based sufficient conditions for exponential stability in hybrid systems," *IEEE Trans. Autom. Control*, vol. 58, no. 6, pp. 1591–1596, 2012.
- [22] Y. Li, S. Phillips, and R. G. Sanfelice, "Robust distributed estimation for linear systems under intermittent information," *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 973–988, 2017.
- [23] T. Liu, D. J. Hill, and B. Liu, "Synchronization of dynamical networks with distributed event-based communication," in *Proc. IEEE Conf. Decis. Control*, 2012, pp. 7199–7204.
- [24] R. Sanfelice, D. Copp, and P. Nanez, "A toolbox for simulation of hybrid systems in Matlab/Simulink: Hybrid equations (HyEQ) toolbox," in *Proc. Int. Conf. Hybrid Syst.: Comp. Control*, 2013, pp. 101–106.

Cooperative Model-Based Reinforcement Learning for Approximate Optimal Tracking

Max L. Greene, Zachary I. Bell, Scott A. Nivison, Jonathan P. How, Warren E. Dixon

Abstract—This paper provides an approximate online adaptive solution to the infinite-horizon optimal tracking problem for a set of agents with homogeneous dynamics and common tracking objectives. Model-based reinforcement learning is implemented by simultaneously evaluating the Bellman error (BE) at the state of each agent and on nearby off-trajectory points, as needed, throughout the state space. Each agent will calculate and share their respective on and off-trajectory BE information with a centralized estimator, which computes updates for the approximate solution to the infinite-horizon optimal tracking problem and shares the estimate with the agents. In doing so, the computational burden associated with BE extrapolation is shared between the agents and a centralized updating resource. Edge computing is leveraged to share the computational load between the agents and a centralized resource. Uniformly ultimately bounded tracking of each agent's state to the desired state and convergence of the control policy to the neighborhood of the optimal policy is proven via a Lyapunov-like stability analysis.

I. INTRODUCTION

Reinforcement learning (RL)-based methods have been used to obtain approximate solutions to optimal control problems for systems with finite state spaces and stationary environments [1] and [2]. Numerous RL-based learning strategies have been used to solve optimal control problems such as those in [3]–[9].

When formulating optimal control problems, the Hamilton-Jacobi-Bellman (HJB) equation provides the optimality condition. The designed optimal control policy depends on the value function [10] and [11]; however, it is generally difficult to solve the HJB equation due to system uncertainties and nonlinearities. Approximate dynamic programming (ADP) is a strategy to approximate the value function online [12] and [13]. By obtaining an approximate value function, a stabilizing and approximately optimal control policy can be developed.

In ADP, the Bellman error (BE) is used as a performance metric, which is an indirect measure of the estimation of

the value function along the system trajectory. If the system dynamics are known or successfully estimated, then the BE can be evaluated at a number of arbitrary points in a system's state space with a process called BE extrapolation [8] and [14].

Results such as [9] and [15] perform BE extrapolation in the neighborhood of the current state. Hence, these results approximate the value function only within that neighborhood. As shown in [9], obtaining local approximations are more computationally efficient than methods such as [8] and [14]. Generally, the obtained value function estimate is not valid over the entire state space (i.e., globally). If obtaining an approximation over an arbitrarily large region of the state space is desired, then BE extrapolation must similarly be performed over an arbitrarily large region of the entire state space. If the methods in [8] and [14] are used, then this may be computationally taxing. To obtain a value function estimate that is less computationally taxing and valid over a smaller subset of the entire state space, BE extrapolation must be performed within the operating domain [8], [14], and [16]. This method is called regional model-based reinforcement learning (R-MBRL). Building upon the method in [9], which uses one agent and locally approximates the value function, a regional value function approximation can be achieved by using multiple agents and BE extrapolation points that are distributed throughout the state space. Using multiple agents in RL-based problems has been shown to accelerate learning across the entire team of agents [17]. The method in [9] uses BE data from one on-trajectory point and several off-trajectory points in the neighborhood of the current state, whereas the developed method uses multiple on-trajectory points (one from each agent) and several off-trajectory points that are widely distributed throughout the state space through shared learning.

R-MBRL is computationally taxing, as BE extrapolation is performed at a large number of off-trajectory points. Furthermore, the computations are centralized, which causes a large computational load on the central resource. To combat the curse of dimensionality, the result in [18] analyzes a coordinated team of homogeneous agents, which share information, to solve multiagent RL problems. A key property of homogeneous agents is interchangeability, meaning that all homogeneous agents have identical optimal policies and value functions [18] and [19]. In the aforementioned method, each agent explores a subset of the entire state space instead of individually exploring the entire state space. Drawing inspiration from [18], the developed method assigns a subset of the selected BE extrapolation points to an agent

Max L. Greene and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA. Email: {maxgreene12, wdixon}@ufl.edu. Zachary I. Bell and Scott Nivison are with the Munitions Directorate, Air Force Research Laboratory, Eglin AFB, FL, USA. Email: {zachary.bell.10, scott.nivison}@us.af.mil. Jonathan P. How is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA. Email: jhow@mit.edu.

This research is supported in part by Office of Naval Research Grant N00014-13-1-0151, A Task Order contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, AFOSR award FA9550-18-1-0109, AFOSR award FA9550-19-1-0169, and AFOSR MURI FA9550-19-1-0386. 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.

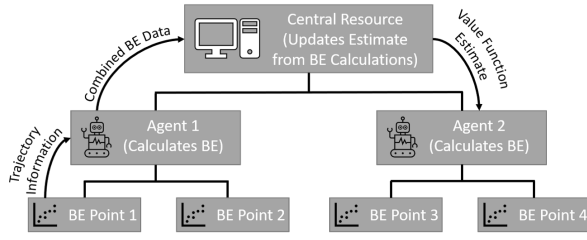


Fig. 1. Diagram of the computing network with two agents and four total extrapolation points. Every agent in the system collects trajectory information for their assigned BE extrapolation points and computes the BE. Next, every agent sends their combined BE extrapolation data to a central resource, which updates the value function estimate. The approximate optimal control policy is determined from the value function estimate (with additional computation).

to decrease the computational burden of R-MBRL. Using onboard computing, each agent calculates the total BE from the agent's current state and subset of extrapolation points. The BE from each agent and its extrapolation points will be passed to a centralized resource on the edge of the network, which updates the value function estimate and relays that information to the agents. This process is an example of edge computing.

In edge computing, the computational load is reduced via edge nodes, which handle computation tasks without transmitting all available data to a centralized resource [20]. Learning the value function online requires a large amount of information to be communicated to a centralized resource, requiring a large amount of network bandwidth [20]. In the developed method, each agent represents a unique computing node. Specifically, the BE is calculated at on and off-trajectory points by each agent. Next, the BE computation is shared with the centralized resource; hence, reducing the magnitude of information transmitted across the network. The developed network and flow of information is illustrated in Figure 1. Latency in transmitting data intensifies with larger data sets [21]. In contrast to cloud computing, edge computing is typically performed closer in vicinity to the agents to reduce latency for real-time data computations generated by the sensors of the agents.

MBRL methods use stationary BE extrapolation points that are distributed throughout the state space (e.g., [8] and [14]) or that follow the current state (e.g., [9]) to achieve sufficient exploration. As the agents converge toward their desired trajectories, the overall system may not achieve sufficient excitation. To avoid a loss of sufficient excitation, the developed framework uses additional BE extrapolation points, which may be added online. Doing so introduces discontinuities into the update laws. The results in [16] present a Lyapunov-based framework that enables switching between sets of active BE extrapolation points in the update laws. Leveraging these results, tracking of each agent's desired trajectory and convergence of the developed control policy to a neighborhood of the optimal control policy are established via a Lyapunov-like stability analysis.

The developed method is constrained to consider homo-

geneous agents with common desired trajectories. While the instantaneous desired trajectories need not be the same for each agent, the overall desired trajectory must be identical for each agent. Identical overall trajectories are required to develop identical error dynamics for all agents, which is leveraged to estimate a value function that is valid for all agents.

In this paper, a method is developed in which each agent collects data from several on and off-trajectory points. The BE is calculated at each point. All BE data from each agent is communicated to a centralized resource that updates the value function estimates and, therefore, the control policies by sending that information back to each agent. In turn, this method leverages the common task between homogeneous agents to estimate one value function, obtains a regional estimate of the value function, reduces the computational complexity of the system via edge computing, and reduces network latency.

A. Notation

For notational brevity, the time-dependence is suppressed while denoting trajectories of the dynamical systems. For example, the trajectory $x(t)$, where $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$, is denoted as $x \in \mathbb{R}^n$ and referred to as x instead of $x(t)$. Unless otherwise specified, an equation of the form $f + h(y, t) = g(x)$ should be interpreted as $f(t) + h(y(t), t) = g(x(t)) \forall t \in \mathbb{R}_{\geq 0}$.

II. PROBLEM FORMULATION

Consider a class of nonlinear control-affine systems that governs $i \in \mathbb{N}$ homogeneous agents

$$\dot{x}_i = f(x_i) + g(x_i)u_i, \quad (1)$$

where $x_i \in \mathbb{R}^n$ denotes the state of the i^{th} agent, $u_i \in \mathbb{R}^m$ denotes the control input of the i^{th} agent, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the drift dynamics, and $g : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$ is the control effectiveness matrix, where $n > m$ and the pseudoinverse of $g(x)$ exists.¹ The control objective is to command each agent to track a path, which is described by a time-varying continuously differentiable signal $x_{di} \in \mathbb{R}^n$. Each x_{di} can be distinct for each agent at every time instant; however, the overall desired trajectories must be identical. To quantify the tracking objective, the tracking error is defined as $e_i \triangleq x_i - x_{di}$. The following assumptions are made to facilitate the formulation of an approximate optimal tracking controller [14].

Assumption 1. The function f is a locally Lipschitz function and $f(0) = 0$. Furthermore, $\nabla_{x_i} f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ is continuous for all i .²

Assumption 2. The function g is a locally Lipschitz function, bounded such that $\underline{g} < \|g(x_i)\| \leq \bar{g} \forall i$ where $\underline{g} \in \mathbb{R}_{>0}$ and

¹Note that the drift dynamics f and control effectiveness matrix g in (1) are identical for each agent.

²Generally, the notation ∇_x denotes the partial derivative with respect to x .

$\bar{g} \in \mathbb{R}_{>0}$ are the minimum and maximum singular values of $g(x_i)$, respectively.

Assumption 3. The desired trajectory is bounded from above by an unknown positive constant $d \in \mathbb{R}$ such that $\|x_{di}\| \leq d \forall i$.

Assumption 4. There exists a locally Lipschitz function $h_d : \mathbb{R}^n \rightarrow \mathbb{R}^n$, such that $h_d(x_{di}) \triangleq \dot{x}_{di}$ and $g(x_{di})g^+(x_{di})(h_d(x_{di}) - f(x_{di})) = h_d(x_{di}) - f(x_{di}) \forall t \in \mathbb{R}_{\geq 0}$, where $g^+ : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$ is defined as $g^+(x_i) \triangleq (g^T(x_i)g(x_i))^{-1}g^T(x_i)$.

Based on Assumptions 2-4, the control policy $u_d : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that tracks the desired trajectory (i.e., trajectory tracking component of the controller) is $u_{di} \triangleq u_d(x_{di}) \triangleq g^+(x_{di})(h_d(x_{di}) - f(x_{di}))$. Let the transient portion of the controller be defined as $\mu_i \triangleq u_i - u_{di}$. Following the development in [14], the concatenated state dynamics are written as

$$\dot{\zeta}_i = F(\zeta_i) + G(\zeta_i)\mu_i, \quad (2)$$

where $\zeta_i \in \mathbb{R}^{2n}$ is the concatenated state of Agent i $\zeta_i \triangleq [e_i^T, x_{di}^T]^T$, $F : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$ is defined as

$$F(\zeta_i) \triangleq \begin{bmatrix} f(e_i + x_{di}) - h_d(x_{di}) + g(e_i + x_{di})u_{di} \\ h_d(x_{di}) \end{bmatrix}, \quad (3)$$

and $G : \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n \times m}$ is defined as

$$G(\zeta_i) \triangleq [g^T(e_i + x_{di}), \mathbf{0}_{m \times n}]^T, \quad (4)$$

and $\mathbf{0}_{m \times n}$ denotes a matrix of zeros with size $m \times n$.

A. Control Objective

The objective is to solve the infinite-horizon optimal regulation problem, i.e. to find a control policy u that minimizes the cost functional

$$J(\zeta_i, \mu_i) = \int_0^\infty r(\zeta_i(\tau), \mu_i(\tau)) d\tau, \quad (5)$$

subject to (2) while regulating the tracking error, where $r : \mathbb{R}^{2n} \times \mathbb{R}^m \rightarrow \mathbb{R}$ is the instantaneous cost, which is defined as $r(\zeta_i, \mu_i) \triangleq \bar{Q}(\zeta_i) + \mu_i^T R \mu_i$, where $\bar{Q} : \mathbb{R}^{2n} \rightarrow \mathbb{R}_{\geq 0}$ is a positive definite (PD) function, and $R \in \mathbb{R}^{m \times m}$ is a PD matrix. To circumvent the effect of the desired trajectories in cost function, let $Q(e_i) \triangleq \bar{Q}(\zeta_i)$, where $Q : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ is a PD function, and $R \in \mathbb{R}^{m \times m}$ is a constant user-defined PD symmetric matrix.

Property 1. The PD function $\bar{Q}(\zeta_i)$ can be bounded such that $\underline{q}(\|e_i\|) \leq \bar{Q}(\zeta_i) \leq \bar{q}(\|e_i\|) \forall e_i$, where $\underline{q}, \bar{q} : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ [14].

The scalar infinite-horizon value function for the optimal solution, denoted by $V^* : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$, is given by

$$V^*(\zeta_i) = \min_{\substack{\mu_i(\tau) \in U \\ \tau \in \mathbb{R}_{\geq 0}}} \int_t^\infty r(\zeta_i(\tau), \mu_i(\tau)) d\tau, \quad (6)$$

where $U \subseteq \mathbb{R}^m$ denotes the action space. If the optimal value function is continuously differentiable, then the optimal control policy $\mu^* : \mathbb{R}^{2n} \rightarrow \mathbb{R}^m$ is the unique solution to the corresponding HJB equation

$$0 = \nabla_{\zeta_i} V^*(\zeta_i) (F(\zeta_i) + G(\zeta_i)\mu_i^*) + \bar{Q}(\zeta_i) + \mu_i^{*T} R \mu_i^*(\zeta_i) \quad (7)$$

with boundary condition $V^*(0) = 0$. Generally, the HJB equation cannot be solved analytically with the exception of a few cases (e.g., linear quadratic regulator or scalar systems).

Remark 5. Under Assumptions 1-8, the optimal value function can be shown to be the unique PD solution of the HJB equations. In this paper, the approximation of the PD solution to the HJB equation is guaranteed by appropriately selecting initial weight estimates and Lyapunov-based update laws [22].

B. Value Function Approximation

The HJB equation in (7) requires knowledge of the optimal value function. Parametric methods can be used to approximate the value function over a compact domain. Let $\Omega \subset \mathbb{R}^{2n}$ be a compact set. The universal function approximation property of NNs (see [23] and [24]) is used to represent the value function as

$$V^*(\zeta_i) = W^T \sigma(\zeta_i) + \varepsilon(\zeta_i), \quad (8)$$

where $W \in \mathbb{R}^L$ is an unknown bounded weight, $\sigma : \mathbb{R}^{2n} \rightarrow \mathbb{R}^L$ is a user-defined vector of activation functions, and $\varepsilon : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ is the bounded function approximation error.

Assumption 6. There exist constants $\bar{W}, \bar{\sigma}, \bar{\nabla}_{\zeta_i} \sigma, \bar{\varepsilon}, \bar{\nabla}_{\zeta_i} \varepsilon \in \mathbb{R}_{>0}$ such that the unknown weight W , user-defined activation functions σ , and function approximation error ε , can be bounded such that $\|W\| \leq \bar{W}$, $\sup_{\zeta_i \in \Omega} \|\sigma\| \leq \bar{\sigma}$, $\sup_{\zeta_i \in \Omega} \|\nabla_{\zeta_i} \sigma\| \leq \bar{\nabla}_{\zeta_i} \sigma$, $\sup_{\zeta_i \in \Omega} \|\varepsilon\| \leq \bar{\varepsilon}$, and $\sup_{\zeta_i \in \Omega} \|\nabla_{\zeta_i} \varepsilon\| \leq \bar{\nabla}_{\zeta_i} \varepsilon$ (see [5], [12], and [25]).

Solving (7) for μ^* and using (8), the approximate optimal control policy $\mu^* : \mathbb{R}^{2n} \rightarrow \mathbb{R}^m$ is

$$\mu^*(\zeta_i) = -\frac{1}{2} R^{-1} G^T(\zeta_i) (\nabla_{\zeta_i} \sigma^T(\zeta_i) W + \nabla_{\zeta_i} \varepsilon^T(\zeta_i)). \quad (9)$$

Since the ideal weight vector W in (8) is unknown, a critic weight estimate vector $\hat{W}_c \in \mathbb{R}^L$ is substituted to estimate the value function $\hat{V} : \mathbb{R}^{2n} \times \mathbb{R}^L \rightarrow \mathbb{R}$ denoted as

$$\hat{V}(\zeta_i, \hat{W}_c) = \hat{W}_c^T \sigma(\zeta_i). \quad (10)$$

Since the ideal weight vector W in (9) is unknown, an actor weight estimate vector $\hat{W}_a \in \mathbb{R}^L$ is substituted to estimate the optimal control policy $\hat{\mu} : \mathbb{R}^{2n} \times \mathbb{R}^L \rightarrow \mathbb{R}$ defined by

$$\hat{\mu}(\zeta_i, \hat{W}_a) = -\frac{1}{2} R^{-1} G^T(\zeta_i) (\nabla_{\zeta_i} \sigma^T(\zeta_i) \hat{W}_a). \quad (11)$$

III. BELLMAN ERROR

The HJB equation in (7) is equal to zero under optimal conditions; however, substituting (10) and (11) into (7) results in a residual term, $\delta : \mathbb{R}^{2n} \times \mathbb{R}^L \times \mathbb{R}^L \rightarrow \mathbb{R}$, which is referred to as the BE, defined as

$$\begin{aligned} \delta(\zeta_k, \hat{W}_c, \hat{W}_a) &\triangleq \hat{\mu}(\zeta_k, \hat{W}_a)^T R \hat{\mu}(\zeta_k, \hat{W}_a) + \bar{Q}(\zeta_k) \\ &\quad + \nabla_{\zeta_k} \hat{V}(\zeta_k, \hat{W}_c) (F(\zeta_k) \\ &\quad + G(\zeta_k) \hat{\mu}(\zeta_k, \hat{W}_a)), \end{aligned} \quad (12)$$

where $\zeta_k \in \Omega$ denotes an on or off-trajectory point in the state space. The BE in (12) indicates how close the actor and critic weight estimates are to their respective ideal weights. The mismatch between the estimates and their ideal values are defined as $\tilde{W}_c \triangleq W - \hat{W}_c$ and $\tilde{W}_a \triangleq W - \hat{W}_a$, where $\tilde{W}_c, \tilde{W}_a \in \mathbb{R}^L$. Substituting (8) and (9) into (7), and subtracting from (12) yields the analytical form of the BE, given by

$$\delta(\zeta_k, \hat{W}_c, \hat{W}_a) = \frac{1}{4} \tilde{W}_a^T G_{\sigma k} \tilde{W}_a - \omega^T(\zeta_k, \hat{W}_a) \tilde{W}_c + \Delta_k, \quad (13)$$

where $\omega : \mathbb{R}^{2n} \times \mathbb{R}^L \rightarrow \mathbb{R}^{2n}$ is defined as

$$\omega(\zeta_k, \hat{W}_a) \triangleq \nabla_{\zeta_k} \sigma(\zeta_k) (F(\zeta_k) + G(\zeta_k) \hat{\mu}(\zeta_k, \hat{W}_a)), \quad (14)$$

and $\Delta_k \triangleq \frac{1}{2} W^T \nabla_{\zeta_k} \sigma(\zeta_k) G_{Rk} \nabla_{\zeta_k} \varepsilon^T(\zeta_k) + \frac{1}{4} G_{\varepsilon k} - \nabla_{\zeta_k} \varepsilon(\zeta_k) F(\zeta_k)$.³ The objective is to find the actor and critic weight estimates \hat{W}_a and \hat{W}_c , respectively, such that $\delta \rightarrow 0$ as $t \rightarrow \infty$.

A. Edge-Based BE Extrapolation

At each time instant $t \in \mathbb{R}_{\geq 0}$, the current agent state, critic weight estimate, and actor weight estimate are used to estimate the on-trajectory BE in (12) and approximate optimal control policy in (11) denoted by $\delta_i \triangleq \delta(\zeta_i, \hat{W}_c, \hat{W}_a)$ and $\hat{\mu}_i \triangleq \hat{\mu}(\zeta_i, \hat{W}_a)$, respectively. However, using only the on-trajectory BE and approximate optimal control policy requires the traditional persistence of excitation condition to be satisfied to show exponential convergence.

The BE can also be calculated at additional off-trajectory points, denoted by $\delta_j \triangleq \delta(\zeta_j, \hat{W}_c, \hat{W}_a)$. This process is referred to as BE extrapolation. Selecting additional BE extrapolation points provides simulation of experience and can lead to faster system convergence [8], [9], [14], and [16].

However, every BE extrapolation point requires additional algebraic computations. Hence, selecting an arbitrarily large number of extrapolation points a priori is computationally

³The notation G_{Rk} , $G_{\sigma k}$, and $G_{\varepsilon k}$ is defined as $G_{Rk} = G_R(\zeta_k) \triangleq G(\zeta_k) R^{-1} G(\zeta_k)^T$, $G_{\sigma k} = G_{\sigma}(\zeta_k) \triangleq \nabla_{\zeta_k} \sigma(\zeta_k) G_R(\zeta_k) \nabla_{\zeta_k} \sigma(\zeta_k)^T$, and $G_{\varepsilon k} = G_{\varepsilon}(\zeta_k) \triangleq \nabla_{\zeta_k} \varepsilon(\zeta_k) G(\zeta_k) \nabla_{\zeta_k} \varepsilon(\zeta_k)^T$, respectively. The matrices G_{Rk} and $G_{\sigma k}$ can be bounded as $\sup_{\zeta_k \in \Omega} \|G\| \leq \lambda_{\max}\{R^{-1}\} \bar{G}^2$ and $\sup_{\zeta_k \in \Omega} \|G_{\sigma k}\| \leq (\bar{\nabla}_{\zeta_k} \sigma \bar{G})^2 \lambda_{\max}\{R^{-1}\}$, respectively, where $\lambda_{\max}\{\cdot\}$ denotes the maximum eigenvalue.

expensive. In the developed method, each agent will calculate its on-trajectory BE and a subset of off-trajectory BEs, and then share that data with a central resource, which will update the actor and critic estimates. The on-policy trajectories $\{\zeta_i : \zeta_i \in \Omega\}_{i=1}^{N_o}$ and off-policy trajectories $\{\zeta_j : \zeta_j \in \Omega\}_{j=1}^{N_e}$ are shared with the central resource, where $N_o \in \mathbb{N}$ denotes the total number of agents in Ω and $N_e \in \mathbb{N}$ denotes the total number of extrapolation points selected in the system. Using all of the extrapolated trajectories $\zeta_k \in \Omega$, the tuple $(\Sigma_c, \Sigma_a, \Sigma_\Gamma)$ is defined as the extrapolation stacks corresponding to Ω such that

$$\Sigma_c \triangleq \frac{1}{N_o + N_e} \sum_{k=1}^{N_o + N_e} \frac{\omega_k}{\rho_k} \delta_k, \quad (15)$$

$$\Sigma_a \triangleq \frac{1}{N_o + N_e} \sum_{k=1}^{N_o + N_e} \frac{G_{\sigma k}^T \hat{W}_a \omega_k^T}{4\rho_k}, \quad (16)$$

$$\Sigma_\Gamma \triangleq \frac{1}{N_o + N_e} \sum_{k=1}^{N_o + N_e} \frac{\omega_k \omega_k^T}{\rho_k^2}, \quad (17)$$

where $\omega_k \triangleq \omega(\zeta_k, \hat{W}_a) = \nabla_{\zeta_k} \sigma(\zeta_k) (F(\zeta_k) + G(\zeta_k) \hat{\mu}(\zeta_k, \hat{W}_a))$, $\rho_k \triangleq \rho(\zeta_k, \hat{W}_a) = 1 + \nu \omega_k^T \Gamma \omega_k$, $\Gamma \in \mathbb{R}^{L \times L}$ is a user-initialized learning gain, and $\nu \in \mathbb{R}_{>0}$ is a user-defined constant.

The computational benefit from edge computing comes from assigning a subset of the total extrapolation points $\{\zeta_j\}_{j=1}^{N_e}$ to each agent. The number of off-trajectory extrapolation points that are assigned to agent i for computation is $N_{ai} \in \mathbb{N}$, where $N_{ai} \leq N_e$. Therefore, each agent i computes the BE corresponding to the trajectories $\{\zeta_i, \zeta_j\}_{j=1}^{N_{ai}}$ (i.e., at its current state ζ_i and N_{ai} off-trajectory points ζ_j). Specifically, the value of δ_k in (15) is computed onboard agent i and sent to the central resource. Calculating δ_k onboard significantly reduces the number of algebraic computations performed by the central resource.

Remark 7. Furthermore, depending on the computational abilities of the agents, additional extrapolation terms, such as $\frac{\omega_k}{\rho_k} \delta_k$, $\frac{G_{\sigma k}^T \hat{W}_a \omega_k^T}{4\rho_k}$, or $\frac{\omega_k \omega_k^T}{\rho_k^2}$ could be computed onboard the agent to shift the computational burden away from the central resource.

Assumption 8. There exist a finite set of trajectories $\{\zeta_k : \zeta_k \in \Omega\}_{k=1}^{N_o + N_e}$ such that

$$0 < \underline{c} \triangleq \inf_{t \in \mathbb{R}_{\geq 0}} \lambda_{\min}\{\Sigma_\Gamma\}, \quad (18)$$

for all $t \in \mathbb{R}_{\geq 0}$, where $\lambda_{\min}\{\cdot\}$ is the minimum eigenvalue, and Σ_Γ was defined in (17).

Remark 9. Assumption 8 is made in existing model-based ADP methods to guarantee sufficient simulation of experience. For insight into selecting a proper number of extrapolation points, see [8].

As the agents converge to their desired trajectory, more off-trajectory points may be needed to satisfy Assumption 8. Adding these additional points introduces discontinuities into the weight update laws, which introduces discontinuities in the time-derivative of the Lyapunov function, requiring a stability analysis with generalized solutions.

IV. ACTOR AND CRITIC WEIGHT UPDATE LAWS

Based on the subsequent analysis, the critic and actor weights are updated according to

$$\dot{\hat{W}}_c = -\eta_c \Gamma \Sigma_c \quad (19)$$

$$\dot{\Gamma} = (\lambda \Gamma - \eta_c \Gamma \Sigma_\Gamma \Gamma) \mathbf{1}_{\{\underline{\Gamma} \leq \|\Gamma\| \leq \bar{\Gamma}\}} \quad (20)$$

$$\dot{\hat{W}}_a = -\eta_{a1} (\hat{W}_a - \hat{W}_c) - \eta_{a2} \hat{W}_a + \eta_c \Sigma_a \hat{W}_c, \quad (21)$$

where $\Gamma \in \mathbb{R}^{L \times L}$ is a time-varying least-squares gain matrix, $\eta_c, \eta_{a1}, \eta_{a2}, \lambda \in \mathbb{R}_{>0}$ are constant learning gains, $\bar{\Gamma}$ and $\underline{\Gamma} \in \mathbb{R}_{>0}$ are upper and lower bound saturation constants, and $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function.⁴

V. STABILITY ANALYSIS

The function $\bar{Q}(\cdot)$ and, therefore, $V^*(\zeta_i)$ are positive semidefinite. Hence, $V^*(\zeta_i)$ is not a valid Lyapunov function. However, results in [26] can be used to show that a nonautonomous form of $V^*(\zeta_i)$, denoted as $V^* : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ and defined as $V^*(e_i, t) = V^*(\zeta_i)$ that is positive definite and decrescent. Hence, $V^*(0, t) = 0$ and there exist class \mathcal{K}_∞ functions $\underline{v}, \bar{v} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ that bound $\underline{v}(\|e_i\|) \leq V^*(e_i, t) \leq \bar{v}(\|e_i\|) \forall i, e_i \in \mathbb{R}^n, t \in \mathbb{R}_{\geq 0}$. Let $V_L : \mathbb{R}^{2L+nN_o} \rightarrow \mathbb{R}$ be a candidate Lyapunov function defined as

$$V_L(Z) \triangleq \sum_{i=1}^{N_o} V^*(e_i, t) + \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} \tilde{W}_c + \frac{1}{2} \tilde{W}_a^T \tilde{W}_a, \quad (22)$$

where $Z \triangleq [\tilde{W}_c^T, \tilde{W}_a^T, e_1^T, \dots, e_{N_o}^T]^T$ is a concatenated state vector. Using the fact $V^*(e_i, t) = V^*(\zeta_i)$ and [27, Lemma 4.3], can be bounded as $\underline{v}_l(\|Z\|) \leq V_L(Z) \leq \bar{v}_l(\|Z\|)$ for class \mathcal{K}_∞ functions $\underline{v}_l, \bar{v}_l : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$. Using (20), the normalized regressor $\frac{\omega_k}{\rho_k}$ can be bounded as $\sup_{t \in \mathbb{R}_{\geq 0}} \left\| \frac{\omega_k}{\rho_k} \right\| \leq \frac{1}{2\sqrt{\nu \underline{\Gamma}}}$ for all $\zeta_k \in \Omega$.

Theorem 10. *Given the dynamic model in (1), provided Assumptions 1-8 are satisfied, and the sufficient gain conditions*

$$\eta_{a1} + \eta_{a2} > \frac{\eta_c}{\sqrt{\nu \underline{\Gamma}}} \|\bar{W}\| \|G_\sigma\|, \quad (23)$$

$$\underline{c} > \frac{3\eta_{a1}}{\eta_c} + \frac{3\eta_c \|\bar{W}\|^2 \|G_\sigma\|^2}{16\nu \underline{\Gamma} (\eta_{a1} + \eta_{a2})}, \quad (24)$$

$$v_l^{-1}(l) < \bar{v}_l^{-1}(\underline{v}_l(\|Z\|)), \quad (25)$$

⁴ $\|\Gamma\|$ is upper and lower bounded by some user-defined saturation gains, $\bar{\Gamma}$ and $\underline{\Gamma}$, respectively. Using (20) ensures that $\underline{\Gamma} \leq \|\Gamma\| \leq \bar{\Gamma}$ for all $t \in \mathbb{R}_{>0}$, where $\underline{\Gamma} \in \mathbb{R}_{>0}$.

where $l \in \mathbb{R}$ is a positive constant, hold, then the error of each agent e_1, \dots, e_{N_o} and weight estimation errors \tilde{W}_c and \tilde{W}_a are UUB, and the applied control policy $\hat{\mu}$ converges to a neighborhood of the optimal control policy μ^* .

Proof. Let $z(t)$ for $t \in \mathbb{R}_{\geq 0}$ be a Filippov solution to the differential inclusion $\dot{z} \in K[h](z)$, where $K[\cdot]$ is defined in [28] and $h : \mathbb{R}^{nN_o+2L+L^2} \rightarrow \mathbb{R}^{nN_o+2L+L^2}$ is defined as $h(z) \triangleq [\dot{e}_1^T, \dots, \dot{e}_{N_o}^T, \dot{\tilde{W}}_c^T, \dot{\tilde{W}}_a^T, \text{vec}(\dot{\Gamma}^{-1})^T]^T$. Due to the discontinuity in the update laws in (19)-(21), the time derivative of (22) exists almost everywhere (a.e., i.e., for almost all $t \in \mathbb{R}_{\geq 0}$) and $\dot{V}_L(z) \stackrel{a.e.}{\in} \dot{\tilde{V}}_L(z)$, where $\dot{\tilde{V}}_L$ is the generalized time-derivative of (22) along the Filippov trajectories of $\dot{z} = h(z)$ [29]. Using the calculus of $K[\cdot]$ from [29], and $\dot{V}^*(\zeta_i) = V^*(\zeta_i) (F(\zeta_i) + G(\zeta_i) \hat{\mu}_i)$, then substituting (19)-(21) yields

$$\begin{aligned} \dot{\tilde{V}}_L \subseteq & \sum_{i=1}^{N_o} (\nabla_{\zeta_i} V^*(\zeta_i) (F(\zeta_i) + G(\zeta_i) \hat{\mu}_i) \\ & - \tilde{W}_c^T \Gamma^{-1} (-\eta_c \Gamma K[\Sigma_c]) \\ & - \frac{1}{2} \tilde{W}_c^T \Gamma^{-1} (\lambda \Gamma - \Gamma \eta_c K[\Sigma_\Gamma] \Gamma) \Gamma^{-1} \tilde{W}_c \\ & - \tilde{W}_a^T (-\eta_{a1} (\hat{W}_a - \hat{W}_c) - \eta_{a2} \hat{W}_a) \\ & - \tilde{W}_a^T \left(\frac{\eta_c G_\sigma^T \hat{W}_a \omega^T}{4\rho} \hat{W}_c(t) + \eta_{c2} K[\Sigma_a] \hat{W}_c \right) \Big). \quad (26) \end{aligned}$$

Using (1), Assumption 8, and substituting the sufficient conditions in (23) and (24) yields

$$\dot{V}_L(Z) \stackrel{a.e.}{\leq} -v_l(\|Z\|) \forall \|Z\| > v_l^{-1}(l), \forall t \in \mathbb{R}_{\geq 0}, \quad (27)$$

where

$$\begin{aligned} v_l(\|Z\|) \leq & \frac{1}{2} \sum_{i=1}^{N_o} \underline{q}(\|e_i\|) + \frac{\eta_c}{12} \underline{c} \|\tilde{W}_c\|^2 \\ & + \frac{\eta_{a1} + \eta_{a2}}{16} \|\tilde{W}_a\|^2, \quad (28) \end{aligned}$$

and $l \in \mathbb{R}$ is a known positive constant. Since (22) is a common Lyapunov function when additional BE points are added, [27, Theorem 4.18] can be invoked to conclude that Z is UUB such that $\limsup_{t \rightarrow \infty} \|Z\| \leq \underline{v}_l^{-1}(\bar{v}_l(v_l^{-1}(l)))$. Since $Z \in \mathcal{L}_\infty$, it follows that $e_1, \dots, e_{N_o}, \tilde{W}_c, \tilde{W}_a \in \mathcal{L}_\infty$, $\hat{W}_c, \hat{W}_a \in \mathcal{L}_\infty$, and $\hat{\mu} \in \mathcal{L}_\infty$. \square

Remark 11. See [14] for insight into satisfying the conditions (23)-(25).

VI. CONCLUSION

A framework for using multiple agents with identical nonlinear control-affine dynamics and desired trajectories is developed to obtain an approximate online solution to the infinite-horizon optimal tracking problem. Each agent calculates its own on and off-trajectory BE, which is shared with a centralized resource, which updates the value function and optimal control policy estimates. By doing so, the computational burden is shifted away from the agents, and

is placed on the centralized resource. UUB tracking of each agent's state to the neighborhood of the desired state and convergence of the control policy to the neighborhood of the optimal policy are proven using a Lyapunov-like stability analysis in the presence of discontinuities.

REFERENCES

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
- [2] D. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2007, vol. 2.
- [3] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations on the Hamilton-Jacobi-Isaacs equation for H_∞ state feedback control with input saturation," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1989–1995, Dec. 2006.
- [4] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.
- [5] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, Jan. 2013.
- [6] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.
- [7] X. Yang, D. Liu, and D. Wang, "Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints," *Int. J. Control*, vol. 87, no. 3, pp. 553–566, 2014.
- [8] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.
- [9] R. Kamalapurkar, J. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, Dec. 2016.
- [10] D. Kirk, *Optimal Control Theory: An Introduction*. Mineola, NY: Dover, 2004.
- [11] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.
- [12] R. Kamalapurkar, P. S. Walters, J. A. Rosenfeld, and W. E. Dixon, *Reinforcement learning for optimal feedback control: A Lyapunov-based approach*. Springer, 2018.
- [13] F. L. Lewis and D. Liu, *Reinforcement learning and approximate dynamic programming for feedback control*. John Wiley & Sons, 2013, vol. 17.
- [14] R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for infinite-horizon approximate optimal tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 753–758, 2017.
- [15] P. Deptula, J. Rosenfeld, R. Kamalapurkar, and W. E. Dixon, "Approximate dynamic programming: Combining regional and local state following approximations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2154–2166, June 2018.
- [16] M. L. Greene, P. Deptula, S. Nivison, and W. E. Dixon, "Sparse learning-based approximate dynamic programming with barrier constraints," *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 743–748, Jul. 2020.
- [17] F. L. Da Silva, R. Glatt, and A. H. R. Costa, "Simultaneously learning and advising in multiagent reinforcement learning," in *Proc. of the 16th Conf. on Auton. Agents and Multiagent Syst.*, 2017, pp. 1100–1108.
- [18] S. Wadhwan, D.-K. Kim, S. Omidshafiei, and J. P. How, "Policy distillation and value matching in multiagent reinforcement learning," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, Macau, China, Nov. 2019.
- [19] C. Sun, M. Shen, and J. P. How, "Scaling up multiagent reinforcement learning for robotic systems: Learn an adaptive sparse communication graph," in *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2020.
- [20] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surv. Tut.*, vol. 22, no. 2, pp. 869–904, 2020.
- [21] M. Alicherry and T. V. Lakshman, "Optimizing data access latencies in cloud systems by intelligent virtual machine placement," in *Proc. IEEE INFOCOM*, 2013, pp. 647–655.
- [22] P. Deptula, Z. Bell, E. Doucette, W. J. Curtis, and W. E. Dixon, "Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with control effectiveness faults," *Automatica*, vol. 116, pp. 1–10, June 2020.
- [23] J. A. Farrell and M. M. Polycarpou, *Adaptive approximation based control: Unifying neural, fuzzy and traditional adaptive approximation approaches*, ser. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. John Wiley & Sons, 2006, vol. 48.
- [24] F. L. Lewis, R. Selmic, and J. Campos, *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2002.
- [25] D. Wang and C. Mu, *Adaptive Critic Control with Robust Stabilization for Uncertain Nonlinear Systems*. Springer, 2019.
- [26] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, Jan. 2015.
- [27] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [28] A. F. Filippov, "Differential equations with discontinuous right-hand side," in *Fifteen papers on differential equations*, ser. American Mathematical Society Translations - Series 2. American Mathematical Society, 1964, vol. 42, pp. 199–231.
- [29] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 73–82, Jan. 1987.

Robust Cadence Tracking for Switched FES-Cycling With an Unknown Time-Varying Input Delay

Brendon C. Allen¹, Christian A. Cousin², Courtney A. Rouse¹, and Warren E. Dixon¹, *Fellow, IEEE*

Abstract—For an individual affected by a neuromuscular condition (NC), functional electrical stimulation (FES)-induced cycling provides a means of functional restoration and therapeutic exercise. Although FES-cycling has been shown to have numerous benefits, there are challenges to implementing closed-loop FES control for coordinated motion. For example, there exists a potentially destabilizing input delay between the application (or removal) of stimulation and the resulting muscle force. Moreover, switching between multiple actuators (such as FES or motor control) can also be destabilizing. This brief develops delay-dependent switching conditions and a robust control method to account for an unknown time-varying input delay of a switched system. A Lyapunov-like analysis is performed to yield semiglobal exponential cadence tracking to an ultimate bound. Experiments were performed on six able-bodied participants and four participants with NCs to validate the developed controller. The proposed controller resulted in an average cadence error of 0.01 ± 2.00 revolutions per minute (RPM) for the able-bodied participants and 0.01 ± 2.72 RPM for participants with NCs.

Index Terms—Functional electrical stimulation (FES), human-robot interaction, input delay, Lyapunov methods, rehabilitation robotics, switched systems.

I. INTRODUCTION

NEUROLOGICAL conditions (NCs), such as traumatic brain injury (TBI), stroke, spinal cord injury (SCI), and Parkinson's disease (PD), among others, often result in a deterioration of quality of life for affected individuals [1]. A common rehabilitative exercise for individuals with lower limb NCs is closed-loop functional electrical stimulation (FES)-induced cycling [1]–[6]; however, there are several challenges associated with closed-loop FES control. The primary challenges are fatigue and the existence of an input delay, called the electromechanical delay (EMD), in response to the complex electrophysiological mechanism involved in FES-induced force production, which may result in instability [7]. Fatigue reduces the FES-induced muscle

force under a fixed stimulation intensity [8] and decreases the duration an exercise can be performed (e.g., the number of repetitions), which may lower rehabilitative effectiveness. A secondary effect of fatigue is that it causes the input delay to vary [7]. Additional challenges are unmodeled disturbances and uncertain parameters in the nonlinear dynamic model [9] and nonlinearity and uncertainty of the muscle activation dynamics [10], and the coordinated functional tasks (e.g., FES cycling) require control to be switched between multiple muscle groups and often a motor to help reduce fatigue [4].

Few studies have developed FES controllers to compensate for an FES-induced input delay, and these studies have focused on continuous exercises (e.g., leg extensions) with FES of a single muscle group [11]–[13]. For example, results such as [11]–[13] all consider a continuous leg extension exercise with FES of the quadriceps femoris muscle group. The prior studies on continuous exercises only considered the contraction delay between the initial application of an electrical stimulus across a muscle and the resulting muscle contraction. When a coordinated exercise is performed, such as cycling, switching is required between multiple muscle groups and the residual forces must also be considered [14]–[18]. The residual forces result from the delayed muscle response after the removal of the electrical stimulus. Special consideration is required for these residual forces so that they are not produced by antagonistic muscles, which would increase the rate of fatigue.

Although results for with FES-induced input delays are sparse, input delayed systems have been extensively studied for general systems [19]–[34]. Often results either assume exact model knowledge (see [22]–[24]) or the input delay is known (see [24]–[26]). However, there are uncertainties in many practical engineering systems, and the input delay may be unknown and potentially time-varying (e.g., an FES-induced input delay is time-varying and difficult to measure [35]). Therefore, results such as [27]–[31] have analyzed systems with an unknown input delay. In recent years, some results have begun to examine input delay compensation for switched systems [14], [15], [32]–[34]. However, the aforementioned results do not compensate for FES-specific factors such as the development of a complex state-dependent switching signal to produce effective agonist muscle contractions despite the contraction delay while simultaneously preventing or minimizing residual antagonistic forces that remain after the stimulation has ceased.

In the authors' previous results in [14] and [15], which this work is predicated upon, closed-loop FES controllers were developed for FES-cycling. The input delay was first considered to be unknown and constant in [14] and generalized to be unknown and time-varying in [15]. Building on our

Manuscript received January 23, 2021; accepted March 22, 2021. Date of publication April 9, 2021; date of current version February 10, 2022. Manuscript received in final form March 29, 2021. This work was supported in part by NSF under Award 1762829 and in part by the Air Force Office of Scientific Research (AFOSR) under Award FA9550-18-1-0109. Recommended by Associate Editor A. Behal. (Corresponding author: Brendon C. Allen.)

Brendon C. Allen, Courtney A. Rouse, and Warren E. Dixon are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: brendoncallen@ufl.edu; courtneyarouse@ufl.edu; wdixon@ufl.edu).

Christian A. Cousin is with the Department of Mechanical Engineering, The University of Alabama, Tuscaloosa, AL 35401 USA (e-mail: cacousin@eng.ua.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCST.2021.3070189>.

Digital Object Identifier 10.1109/TCST.2021.3070189

precursory results in [14] and [15], this brief includes comparative experiments on six able-bodied participants and four participants with NCs (compared to no experiments in [14] and [15], less conservative gain conditions are achieved, and an extension is provided with a modified control objective (i.e., motorized assistance is continuously provided to further align with current clinical practice in rehabilitation cycles). Furthermore, a switched cadence tracking controller is developed for an FES cycle that is robust to an unknown time-varying FES-induced input delay. Trigger conditions are developed to appropriately schedule the activation and deactivation of stimulation for each muscle group and the motor such that the muscle forces occur in desired locations and the residual muscle forces are less likely to come from antagonist muscles, which would impede cycling and increase fatigue. A Lyapunov-like switched systems stability analysis is performed to prove a cadence tracking error with semiglobal exponential convergence to a uniform ultimate bound.

To demonstrate the performance of the developed controller, in-depth experiments were performed on six able-bodied participants and four participants with different NCs (spina bifida, quadriplegia, multiple sclerosis, and cerebral palsy). The experiments compare the developed controller, the extended controller, and for comparison, a controller that was developed assuming the system has no input delay. The developed controller achieved an average cadence error of 0.01 ± 2.00 revolutions per minute (RPM) for the able-bodied participants and 0.01 ± 2.72 RPM for the participants with NCs. The experimental results validate the controller and indicate that delay compensation can result in an improved FES-cycling experience when compared to a controller of the same form, but without delay compensation.

II. MODEL

Throughout this brief, switching signals are piecewise right-continuous and delayed functions are defined as

$$h_\tau \triangleq \begin{cases} h(t - \tau(t)), & t - \tau(t) \geq t_0 \\ 0, & t - \tau(t) < t_0 \end{cases}$$

where $t, t_0 \in \mathbb{R}_{\geq 0}$, $\tau : \mathbb{R}_{\geq 0} \rightarrow \mathbb{S}$, and $\mathbb{S} \subset \mathbb{R}_{>0}$, denote the time, initial time, EMD, and set of possible delay values [36], respectively. The nonlinear, uncertain motorized cycle-rider dynamics can be modeled as¹ [14], [15]

$$\begin{aligned} M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) + P(q, \dot{q}) + b_c\dot{q} + d(t) \\ = \underbrace{B_E k_e \sigma_e(q, \dot{q}) u_e(t)}_{B_E} + \underbrace{\sum_{m \in \mathcal{M}} B_m(q, \dot{q}, t) k_m \sigma_{m, \tau} u_\tau}_{B_M^r(q, \dot{q}, \tau, t)} \end{aligned} \quad (1)$$

where $q : \mathbb{R}_{\geq 0} \rightarrow \mathcal{Q}$, $\dot{q} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, and $\ddot{q} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ denote the measurable crank angle, measurable angular velocity (cadence), and unmeasured acceleration, respectively. The set of all possible crank angles is denoted by $\mathcal{Q} \subseteq \mathbb{R}$. The inertial effects, gravitational effects, centripetal-Coriolis effects, passive viscoelastic tissue forces, disturbances, and

viscous damping coefficient are denoted by $M : \mathcal{Q} \rightarrow \mathbb{R}_{>0}$, $G : \mathcal{Q} \rightarrow \mathbb{R}$, $V : \mathcal{Q} \times \mathbb{R} \rightarrow \mathbb{R}$, $P : \mathcal{Q} \times \mathbb{R} \rightarrow \mathbb{R}$, and $d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ and $b_c \in \mathbb{R}_{>0}$, respectively. The unknown lumped motor and muscle control terms are denoted by $B_E \in \mathbb{R}_{>0}$ and $B_M^r : \mathcal{Q} \times \mathbb{R} \times \mathbb{S} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$, respectively, the unknown motor and muscle effectiveness terms are denoted by $B_e \in \mathbb{R}_{>0}$ and $B_m : \mathcal{Q} \times \mathbb{R} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0}$, $\forall m \in \mathcal{M}$, respectively, and selectable constants are denoted by $k_e, k_m \in \mathbb{R}_{>0}$, $\forall m \in \mathcal{M}$, where $m \in \mathcal{M} \triangleq \{RH, RQ, RG, LH, LQ, LG\}$ indicates the right (R) and left (L) hamstrings (H), quadriceps femoris (Q), and gluteal (G) muscle groups. The implemented motor and FES control inputs are denoted by $u_e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ and $u : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, respectively, and the delayed FES control input is denoted by $u_\tau : \mathbb{S} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$. For a given $m \in \mathcal{M}$, the delayed FES switching signal, $\sigma_{m, \tau}$, indicates whether muscle m received the FES input u_τ at $t - \tau(t)$.

The implemented switching signals for activation of the motor and FES are denoted by $\sigma_e, \sigma_m : \mathcal{Q} \times \mathbb{R} \rightarrow \{0, 1\}$, respectively, and are defined as

$$\sigma_e(q, \dot{q}) \triangleq \begin{cases} 1, & q \in \mathcal{Q}_{\text{KDZ}} \\ 1, & q \in \mathcal{Q}_{\text{FES}}, \sum_{m \in \mathcal{M}} \sigma_m(q, \dot{q}) = 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$\sigma_m(q, \dot{q}) \triangleq \begin{cases} 1, & q_\alpha(q, \dot{q}) \in \mathcal{Q}_m \\ 1, & q_\beta(q, \dot{q}) \in \mathcal{Q}_m \forall m \in \mathcal{M} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where the trigger conditions $q_\alpha, q_\beta : \mathcal{Q} \times \mathbb{R} \rightarrow \mathbb{R}$ are designed to adjust the activation/deactivation of the FES input to ensure that muscle contractions occur in desired contraction regions, defined as $\mathcal{Q}_{\text{FES}} \triangleq \bigcup_{m \in \mathcal{M}} \{\mathcal{Q}_m\}$, and to reduce/eliminate the residual torques in nondesired regions, defined as $\mathcal{Q}_{\text{KDZ}} \triangleq \mathcal{Q} \setminus \mathcal{Q}_{\text{FES}}$. To yield efficient forward pedaling (i.e., positive crank motion), each muscle's desired contraction region, denoted by $\mathcal{Q}_m \subset \mathcal{Q}$, $\forall m \in \mathcal{M}$, is defined, according to [4] as

$$\mathcal{Q}_m \triangleq \{q \in \mathcal{Q} \mid T_m(q) > \varepsilon_m\} \quad \forall m \in \mathcal{M} \quad (4)$$

where $T_m : \mathcal{Q} \rightarrow \mathbb{R}$ and $\varepsilon_m \in \mathbb{R}_{>0}$ denote a torque transfer ratio and a selectable lower threshold.

Although the parameters in (1) are unknown, the subsequently designed FES and motor controllers only require known bounds on the aforementioned parameters [4].

Property 1: The parameters in (1) can be bounded as $|d| \leq c_d$, $b_c \dot{q} \leq c_c |\dot{q}|$, $|P| \leq c_{P_1} + c_{P_2} |\dot{q}|$, $|G| \leq c_G$, $|V| \leq c_V |\dot{q}|$, and $c_m \leq M \leq c_M$, where $c_d, c_c, c_{P_1}, c_{P_2}, c_G, c_V, c_m, c_M \in \mathbb{R}_{>0}$ are known constants.

Property 2: The lumped motor (when $\sigma_e = 1$) and FES (when $\sum_{m \in \mathcal{M}} \sigma_{m, \tau} > 0$) control terms are bounded as $c_e \leq B_E \leq c_E$ and $c_b \leq B_M^r \leq c_B$, where $c_b, c_B, c_e, c_E \in \mathbb{R}_{>0}$ are known constants.

Property 3: The delay can be bounded as $\underline{\tau} \leq \tau \leq \bar{\tau}$, where $\underline{\tau}, \bar{\tau} \in \mathbb{R}_{>0}$ are known constants. The delay estimate error can be bounded such that $\hat{\tau} - \tau \leq \bar{\tau}$, where $\hat{\tau} \in \mathbb{R}_{\geq 0}$ is a constant estimate of the delay and $\bar{\tau} \in \mathbb{R}_{>0}$ is a known constant [36].

¹For notational brevity, all explicit dependence on time, t , within the terms $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$, and $\tau(t)$ is suppressed.

III. CONTROL DEVELOPMENT

The control objective is for the bicycle crank to track a smooth desired trajectory $q_d, \dot{q}_d, \ddot{q}_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ despite the presence of uncertainties in the nonlinear dynamic model and an unknown time-varying input delay. The measurable cadence tracking error, denoted by $\dot{e} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, is defined as

$$\dot{e} \triangleq \dot{q}_d - \dot{q} \quad (5)$$

where, the measurable crank position tracking error, denoted by $e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, is defined as

$$e \triangleq q_d - q. \quad (6)$$

To facilitate the subsequent analysis, a measurable auxiliary tracking error, denoted by $r : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, is defined as

$$r \triangleq \dot{e} + \alpha_1 e + \alpha_2 e_u \quad (7)$$

where $\alpha_1, \alpha_2 \in \mathbb{R}_{\geq 0}$ are selectable constants. The auxiliary error signal, denoted by $e_u : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, is designed to inject a delay-free input term into the closed-loop error system and is defined as

$$e_u \triangleq - \int_{t-\hat{\tau}}^t u(\theta) d\theta. \quad (8)$$

The open-loop error system is obtained by taking the time derivative of (7), solving (1) for \ddot{q} , using (6) and (8), and adding and subtracting $M^{-1}B_M^\tau u_{\hat{\tau}} + e$ to obtain

$$\begin{aligned} \dot{r} = & -e + \chi + M^{-1}B_M^\tau(u_{\hat{\tau}} - u_\tau) - M^{-1}B_E u_e \\ & + (\alpha_2 - M^{-1}B_M^\tau)u_{\hat{\tau}} - \alpha_2 u \end{aligned} \quad (9)$$

where the auxiliary term, denoted by $\chi : \mathcal{Q} \times \mathbb{R} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, is defined as

$$\chi \triangleq \ddot{q}_d + M^{-1}(V\dot{q} + G + P + b_c\dot{q} + d) + \alpha_1\dot{e} + e.$$

By using Properties 1–6, χ can be bounded as

$$|\chi| \leq \Phi + \rho(\|z\|)\|z\| \quad (10)$$

where $\Phi \in \mathbb{R}_{>0}$ is a known constant, $\rho(\cdot)$ is a positive, strictly increasing, and radially unbounded function, and $z \in \mathbb{R}^3$ is a composite error vector defined as

$$z \triangleq [e \quad r \quad e_u]^\top. \quad (11)$$

Based on the open-loop error system in (9) and the subsequent stability analysis, the FES and motor controller are designed, respectively, as

$$u = k_s r \quad (12)$$

$$u_e = k_1 \text{sgn}(r) + (k_2 + k_3)r \quad (13)$$

where $k_s, k_1, k_2, k_3 \in \mathbb{R}_{>0}$ are selectable constants, and $\text{sgn}(\cdot)$ denotes the signum function. Substituting (12) and (13) into (9) yields the closed-loop error system

$$\begin{aligned} \dot{r} = & -e + \chi + k_s M^{-1}B_M^\tau(r_{\hat{\tau}} - r_\tau) \\ & - M^{-1}B_E(k_1 \text{sgn}(r) + (k_2 + k_3)r) \\ & + (\alpha_2 - M^{-1}B_M^\tau)k_s r_{\hat{\tau}} - \alpha_2 k_s r. \end{aligned} \quad (14)$$

Lyapunov–Krasovskii functionals, denoted by $Q_1, Q_2 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{>0}$, are designed to facilitate the subsequent stability analysis as

$$Q_1 \triangleq \frac{1}{2}(\varepsilon_1\omega_1 + \varepsilon_3\omega_3)k_s \int_{t-\hat{\tau}}^t r(\theta)^2 d\theta \quad (15)$$

$$Q_2 \triangleq \frac{\omega_2 k_s}{\hat{\tau}} \int_{t-\hat{\tau}}^t \int_s^t r(\theta)^2 d\theta ds \quad (16)$$

where $\varepsilon_1, \varepsilon_3, \omega_1, \omega_2, \omega_3 \in \mathbb{R}_{>0}$ are selectable constants. Based on the subsequent stability analysis, auxiliary bounding constants denoted by $\beta_1, \beta_2, \delta_1, \delta_2 \in \mathbb{R}_{>0}$ are defined as

$$\beta_1 \triangleq \min\left(\alpha_1 - \frac{\varepsilon_2 \alpha_2^2}{2}, k_s \left(\frac{1}{2}\alpha_2 - \varepsilon_1\omega_1 - \varepsilon_3\omega_3 - \omega_2\right) \frac{\omega_2}{3k_s \hat{\tau}^2} - \frac{1}{2\varepsilon_2} - \frac{\omega_3 k_s}{\varepsilon_3}\right) \quad (17)$$

$$\beta_2 \triangleq \min\left(\alpha_1 - \frac{\varepsilon_2 \alpha_2^2}{2}, \frac{c_e}{c_M} k_2 - k_s(\varepsilon_3\omega_3 + \omega_2) \frac{\omega_2}{3k_s \hat{\tau}^2} - \frac{1}{2\varepsilon_2} - \frac{\omega_3 k_s}{\varepsilon_3}\right) \quad (18)$$

$$\delta_1 \triangleq \min\left(\frac{\beta_1}{2}, \frac{2\omega_2}{3\hat{\tau}(\varepsilon_1\omega_1 + \varepsilon_3\omega_3)}, \frac{1}{3\hat{\tau}}\right) \quad (19)$$

$$\delta_2 \triangleq \min\left(\frac{\beta_2}{2}, \frac{2\omega_2}{3\hat{\tau}(\varepsilon_1\omega_1 + \varepsilon_3\omega_3)}, \frac{1}{3\hat{\tau}}\right) \quad (20)$$

where $\varepsilon_2 \in \mathbb{R}_{>0}$ is a selectable constant.

IV. STABILITY ANALYSIS

In the subsequent analysis, switching times are denoted by $\{t_n^i\}$, $i \in \{m, e\}$, $n \in \{0, 1, 2, \dots\}$, which denote the instants in time when B_M^τ becomes nonzero ($i = m$) and when B_M^τ becomes zero ($i = e$). A positive definite, continuously differentiable, common Lyapunov function candidate that is defined on a domain $\mathcal{D} \subseteq \mathbb{R}^5$ and denoted by $V_L : \mathcal{D} \rightarrow \mathbb{R}_{>0}$ is defined as

$$V_L \triangleq \frac{1}{2}e^2 + \frac{1}{2}r^2 + \frac{1}{2}\omega_3 e_u^2 + Q_1 + Q_2 \quad (21)$$

which satisfies the following inequalities:

$$\lambda_1 \|y\|^2 \leq V_L \leq \lambda_2 \|y\|^2 \quad (22)$$

where $y \in \mathbb{R}^5$ is defined as

$$y \triangleq [z^\top \quad \sqrt{Q_1} \quad \sqrt{Q_2}]^\top \quad (23)$$

and $\lambda_1, \lambda_2 \in \mathbb{R}_{>0}$ are known constants defined as

$$\lambda_1 \triangleq \min\left(\frac{1}{2}, \frac{\omega_3}{2}\right), \quad \lambda_2 \triangleq \max\left(1, \frac{\omega_3}{2}\right).$$

For the subsequent stability analysis, let the set of initial conditions be defined as

$$\mathcal{S}_\mathcal{D} \triangleq \left\{ y \in \mathcal{D} \mid \|y\| < \sqrt{\frac{\lambda_1}{\lambda_2}} \gamma \right\} \quad (24)$$

where $\gamma \in \mathbb{R}_{>0}$ is a known constant and is defined as² $\gamma \triangleq \inf\{\rho^{-1}((\sqrt{\kappa}, \infty))\}$, where $\kappa \triangleq \min((1/2)\beta_1\alpha_2k_s, (2c_e/c_M)k_3\beta_2)$.

²For a set A , the inverse image is defined as $\rho^{-1}(A) \triangleq \{a \mid \rho(a) \in A\}$.

Theorem 1: The closed-loop error system in (14) is uniformly ultimately bounded in the sense that

$$\|y(t)\|^2 \leq \frac{\lambda_2}{\lambda_1} \|y(t_0)\|^2 \exp(-\lambda_3(t - t_0)) + \frac{v}{\lambda_1 \lambda_3} (1 - \exp(-\lambda_3(t - t_0))) \quad (25)$$

where $v \triangleq (1/\alpha_2 k_s)(\Phi + k_s \bar{\gamma} \Gamma(c_B/c_m))^2$, $\Gamma \in \mathbb{R}_{>0}$ is a known constant and $\lambda_3 \triangleq \lambda_2^{-1} \min(\delta_1, \delta_2)$, $\forall t \in [t_0, \infty)$, provided $y(t_0) \in S_D$, and the following gain conditions are satisfied:

$$\alpha_1 > \frac{\varepsilon_2 \alpha_2^2}{2}, \quad \alpha_2 > 2(\varepsilon_1 \omega_1 + \varepsilon_3 \omega_3 + \omega_2) \quad (26)$$

$$\omega_2 > 3k_s \hat{\tau}^2 \left(\frac{1}{2\varepsilon_2} + \frac{\omega_3 k_s}{\varepsilon_3} \right), \quad \sqrt{\lambda_1^{-1} \lambda_3^{-1} v} < \gamma \quad (27)$$

$$k_1 \geq \frac{c_M}{c_e} (\Phi + k_s \Gamma \hat{\tau} |\alpha_2 - \varepsilon_1 \omega_1|) \quad (28)$$

$$k_2 > \frac{k_s c_M}{c_e} (\varepsilon_3 \omega_3 + \omega_2), \quad k_3 > 0 \quad (29)$$

$$\max \left(\left| \alpha_2 - \frac{c_b}{c_M} \right|, \left| \alpha_2 - \frac{c_B}{c_m} \right| \right) \leq \varepsilon_1 \omega_1. \quad (30)$$

Proof: The proof follows from the development in [15]. ■

V. EXTENSION

An extension of the developed controllers in (12) and (13) to improve the gain conditions is to allow for the motor to always be activated. This change is reflected by modifying the motor switching condition from (2) to $\sigma_e(q, \dot{q}) \triangleq 1$.

For comparative purposes, an additional controller/switching signal combination can be created to compensate for the system dynamics in (1) if the FES input delay was considered to be negligible. This “delay-free” controller can be generated by removing the delay-compensating term e_u from the auxiliary tracking error system in (7), such that $r \triangleq \dot{e} + \alpha_1 e$, and using (12) and (13) with this modified error system. In addition, since the delay is assumed to be negligible, the switching signals do not need to compensate for the delay and can be defined as

$$\sigma_m(q) \triangleq \begin{cases} 1, & q \in \mathcal{Q}_m \\ 0, & \text{otherwise} \end{cases}, \quad \sigma_e(q) \triangleq \begin{cases} 1, & q \in \mathcal{Q}_{KDZ} \\ 0, & \text{otherwise} \end{cases} \quad (31)$$

such as in [4]. The stability analysis for the delay-free controller can be developed using a method similar to [4].

VI. EXPERIMENT

The performance of the developed controllers and switching signals in (2), (3), (12), and (13), henceforth collectively labeled as Controller A, was validated through experiments on both able-bodied participants and participants with NCs. To better examine the performance of Controller A compared to alternative cadence tracking controllers, the extension and delay-free controllers described in Section V were implemented, henceforth labeled as Controllers B and C, respectively. To allow for the best comparison, all three controllers were designed to have the same form and the same objective of

TABLE I
PARTICIPANT DEMOGRAPHICS

Participant	Age	Sex	Condition	Time Since Diagnosis
P1	24	M	None	- -
P2	26	M	None	- -
P3	21	F	None	- -
P4	24	M	None	- -
P5	22	F	None	- -
P6	24	M	None	- -
N1	26	M	Spina Bifida (L5-S1)	26yr
N2	58	M	Quadriplegia	5yr
N3	57	F	Multiple Sclerosis	10yr
N4	42	F	Cerebral Palsy	42yr

TABLE II
COMPARATIVE RESULTS FOR ABLE-BODIED AND NEUROLOGICAL POPULATION DURING STEADY-STATE OPERATION: REPORTED AS AVERAGE \pm STANDARD DEVIATION

Controller	Participant	Cadence Error (RPM)	Motor Input (A)*	FES Input (μ s) [†]	FES on time (%) [‡]
A	P1	0.06 \pm 2.16	1.60 \pm 0.13	60.97 \pm 3.40	63.71
	P2	0.06 \pm 2.66	1.94 \pm 0.29	78.54 \pm 6.82	65.19
	P3	0.05 \pm 2.38	1.65 \pm 0.15	76.64 \pm 3.97	65.95
	P4	-0.04 \pm 1.40	1.15 \pm 0.13	36.61 \pm 2.52	66.32
	P5	-0.02 \pm 1.86	1.50 \pm 0.12	31.85 \pm 1.58	67.91
	P6	-0.03 \pm 1.53	1.39 \pm 0.09	21.74 \pm 0.46	72.44
	Average	0.01\pm2.00	1.54\pm0.15	51.06\pm3.12	66.92
	N1	0.00 \pm 2.10	1.32 \pm 0.09	40.28 \pm 1.92	74.96
	N2	0.00 \pm 2.70	1.87 \pm 0.13	224.25 \pm 12.51	73.54
	N3	0.03 \pm 2.96	2.01 \pm 0.16	52.43 \pm 2.39	72.61
	N4	0.00 \pm 3.14	1.84 \pm 0.13	36.39 \pm 2.39	74.90
B	Average	0.01\pm2.72	1.76\pm0.13	88.34\pm4.81	74.00
	P1	0.01 \pm 1.27	1.73 \pm 0.14	51.04 \pm 3.44	64.02
	P2	0.05 \pm 1.52	2.05 \pm 0.26	63.79 \pm 6.95	66.21
	P3	0.03 \pm 1.22	1.86 \pm 0.15	64.05 \pm 4.16	66.50
	P4	-0.03 \pm 0.80	1.14 \pm 0.13	37.61 \pm 2.28	65.15
	P5	-0.01 \pm 1.25	1.62 \pm 0.15	30.36 \pm 1.57	68.09
	P6	-0.02 \pm 0.66	1.41 \pm 0.08	20.83 \pm 0.30	72.85
	Average	0.00\pm1.12	1.64\pm0.15	44.61\pm3.12	67.14
	N1	-0.03 \pm 0.87	1.30 \pm 0.08	30.98 \pm 1.12	75.43
	N2	-0.04 \pm 1.08	1.89 \pm 0.11	146.28 \pm 7.67	74.12
	N3	-0.02 \pm 0.93	2.04 \pm 0.15	39.05 \pm 2.25	73.90
	N4	-0.05 \pm 1.34	1.88 \pm 0.12	30.91 \pm 0.87	76.38
C	Average	-0.04\pm1.06	1.78\pm0.12	61.80\pm2.98	74.96
	P1	0.09 \pm 3.29	1.61 \pm 0.14	75.31 \pm 5.26	43.15
	P2	0.10 \pm 3.86	1.87 \pm 0.27	83.78 \pm 5.91	39.11
	P3	0.09 \pm 3.68	1.78 \pm 0.21	82.66 \pm 3.63	38.70
	P4	-0.03 \pm 1.93	1.04 \pm 0.11	35.91 \pm 3.08	48.65
	P5	-0.02 \pm 2.32	1.53 \pm 0.12	34.03 \pm 1.14	45.58
	P6	0.00 \pm 2.00	1.40 \pm 0.09	24.12 \pm 0.23	62.39
	Average	0.04\pm2.85	1.54\pm0.16	55.97\pm3.25	46.26
	N1	-0.02 \pm 2.70	1.27 \pm 0.10	44.87 \pm 2.13	49.17
	N2	0.23 \pm 6.72	5.55 \pm 0.93	345.80 \pm 16.83	44.23
	N3	0.08 \pm 4.11	2.05 \pm 0.26	67.09 \pm 4.91	49.30
	N4	0.08 \pm 3.97	1.87 \pm 0.16	41.57 \pm 1.47	43.05
	Average	0.09\pm4.37	2.69\pm0.36	124.83\pm6.33	46.44

*For post-processing, a single crank-cycle (a moving window of approximately 1.2 seconds) averaging filter was applied on the motor input.

[†]The average and standard deviation of the applied stimulation was calculated using the maximum stimulation delivered to each muscle group for each FES region.

[‡]This variable represents the average percentage of a single crank cycle that FES was applied to at least one of the muscle groups.

cadence tracking. By comparing the three controllers, insights are provided on the effect of delay compensation and the effect of switching the motor ON and OFF.

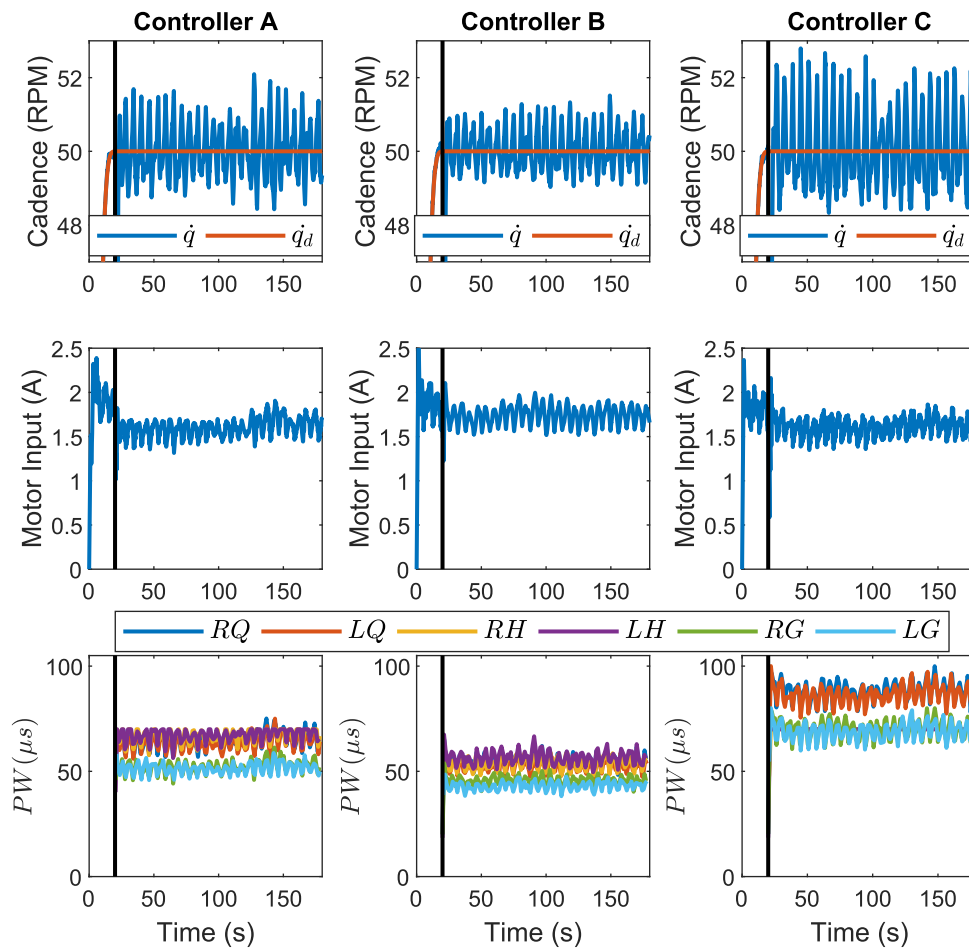


Fig. 1. Desired versus the filtered cadence (top), filtered motor input (middle), and the peak FES input PW for each FES region applied to the right (R) and left (L) quadriceps (Q), hamstring (H), and gluteal (G) (bottom) are depicted for Controllers A, B, and C from left to right for participant P1. Vertical black line indicates the time when steady-state was reached. A 1.2 s moving average filter was used on the actual cadence and the motor input for visual clarity.

A. Experimental Testbed

The experimental testbed consisted of a modified recumbent tricycle (TerraTrike Rover) mounted on a trainer and riser rings to make it stationary as described in [1] and [4]. A desktop computer executing MATLAB/Simulink/Quarc was used to interface the encoder (US Digital H1), motor (Unite Motor Company), and stimulator (Hasomed Rehasim) through a data acquisition board (Quanser Q-PiDe) at 500 Hz. Similar to [1], the current amplitude (90, 80, and 70 mA for the quadriceps, hamstrings, and gluteals, respectively) and stimulation frequency (60 Hz) of the stimulator was fixed, while the pulsewidth (PW) was used as the control input in (12).

B. Experimental Methods

An experimental protocol was performed on six able-bodied participants and four participants with NCs. The demographic information for each participant is shown in Table I.

Able-bodied participants are referred to by the letter “P” followed by their participant number, while participants with neurological conditions are referred to by the letter “N” followed by their participant number. Each participant gave written informed consent approved by the University of

Florida Institutional Review Board. During the experiment, each participant was instructed to relax and make no volitional effort to either assist or resist the FES or electric motor input (i.e., to be a passive rider, blind to the desired or actual trajectory). The experiment was repeated three times for each participant with the only change being the implemented controller. Controllers A, B, and C were implemented in a random order for the experiment. To further limit the effect of fatigue, rest of at least 5 min was provided between each experiment.

Before the experiments began, the electrodes (Axelgaard ValuTrove CF7515) were placed on each muscle group (quadriceps, hamstrings, and gluteals) and the participant was seated on the cycle with their feet secured using orthotic boots (Össur Rebound Air Tall). The seat was adjusted to ensure the participant’s comfort while cycling. Measurements as detailed in [4] were performed to determine the desired FES regions of the crank for each participant. The cycle was then run at 50 RPM, and open-loop stimulation was applied to one muscle group at a time to determine a comfort limit for each muscle, called the comfort threshold. If during an experiment, the participants comfort threshold was reached, the stimulation was saturated.

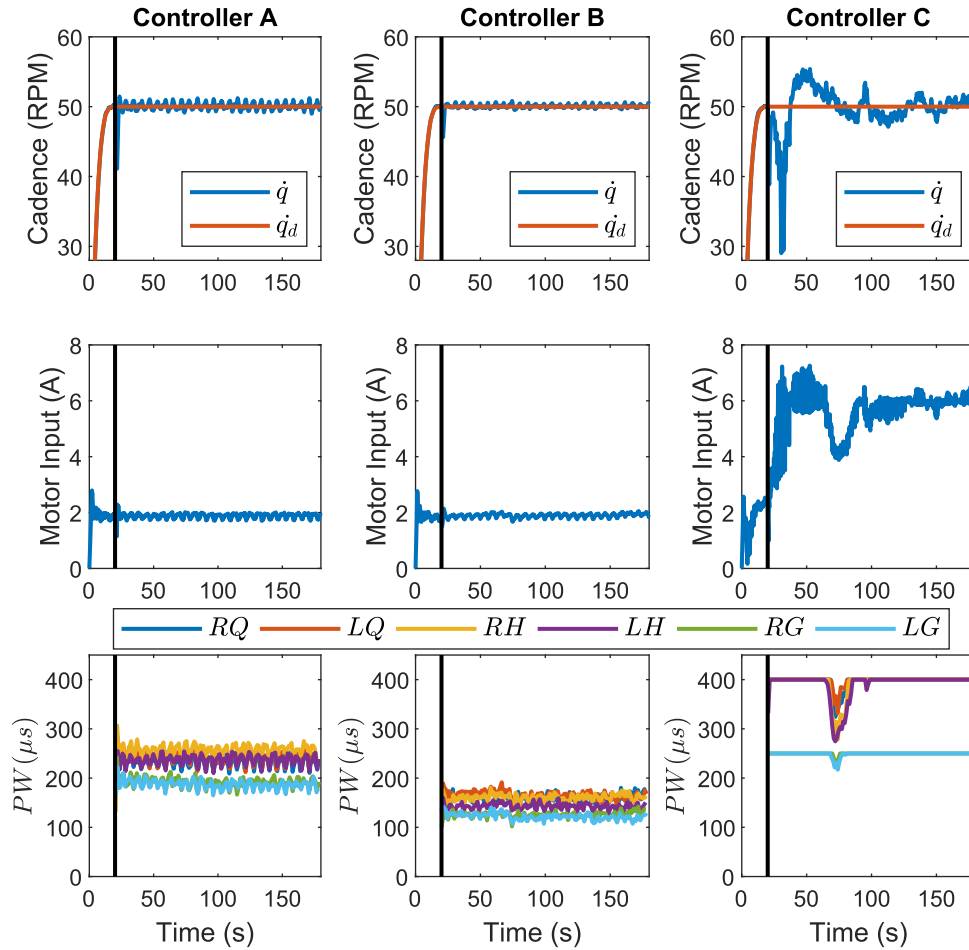


Fig. 2. Desired versus the filtered cadence (top), filtered motor input (middle), and the peak FES input PW for each FES region applied to the right (R) and left (L) quadriceps (Q), hamstring (H), and gluteal (G) (bottom) are depicted for Controllers A, B, and C from left to right for participant N2. Vertical black line indicates the time when steady state was reached. A 1.2 s moving average filter was used on the actual cadence and the motor input for visual clarity.

The experimental protocol lasted 180 s. The first 20 s consisted of the motor tracking a smooth cadence ramp from zero to $\dot{q}_d = 50$ RPM, at which point either Controller A, B, or C was implemented for the remainder of the experiment. The goal of the remaining 160 s of the experiment was to track a constant desired cadence of 50 RPM, similar to [1]. This protocol was repeated for each controller.

VII. RESULTS

Experiments were conducted using Controllers A, B, and C on both able-bodied participants and those with NCs. Each controller was implemented on each participant for a single trial. The demographics of the four neurological participants are shown in Table I. Participant N2 has quadriplegia and felt little sensation in his limbs resulting in higher stimulation thresholds. Participants N1, N3, and N4 were more sensitive to the stimulation resulting in lower stimulation thresholds, with Participant N4 being the most sensitive. Experiments were performed on participants with a range of NCs to demonstrate each controller's stability over a range of rider capabilities.

The cadence error, motor input, and FES input for both populations are shown in Table II. To highlight the performance of Controllers A, B, and C, the cadence tracking

results and control inputs over the entire experiment are shown in Figs. 1 and 2 for Participants P1 and N2, respectively. Typical results for both populations are represented by the results of Participant P1. The results of Participant N2 are also depicted because his results deviated from that of Participant P1 and because he is quadriplegic and unable to provide volitional effort; thus, any muscle produced force is caused solely due to the controllers.

VIII. DISCUSSION

The experimental results conducted on both able-bodied and neurologically impaired populations demonstrate the validity of Controller A in tracking cadence despite uncertainties in the system and an unknown time-varying input delay, as shown in Figs. 1 and 2. In fact, for the able-bodied participants, the average standard deviation of the cadence tracking error was less than 3 RPM for each controller, as shown in Table II. For the participants with NCs, Controllers A and B outperformed Controller C in cadence tracking, as shown in Table II and Fig. 2. Controller B had the best cadence tracking as shown in Table II because Controller B maintained motor control throughout the experiment. Controller B was designed to achieve the best possible tracking performance by the

controllers in (12) and (13). Controller C had the largest standard deviations for cadence tracking errors (Table II).

For Participants P1 and N2, the FES and motor inputs across the experiment for each controller are shown in Figs. 1 and 2. From Table II and Figs. 1 and 2, it is clear that Controller C required higher FES inputs than Controllers A and B. In fact, for able-bodied participants, Controllers B and C had average peak FES inputs that were 12.6% lower and 9.6% higher than Controller A, respectively. For participants with an NC, Controllers B and C had average peak FES inputs that were 30.0% lower and 41.3% higher than Controller A, respectively. Therefore, it can be noted that Controller B tends to decrease the required FES input. However, for each participant, Table II indicates that on average, the FES is on 69.8%, 70.3%, and 46.3% of the time for Controllers A, B, and C, respectively. Thus, Controller C increased the FES input, but the FES is on for a shorter amount of time. Although the FES duration is shorter for Controller C, participants commonly indicated that Controller C felt less comfortable because it resulted in higher FES inputs when compared to Controllers A and B.

For each participant, the average motor input for each controller is shown in Table II. For the able-bodied participants, the motor input was 6.5% higher for Controller B when compared to Controller A and C. For participants with an NC, the motor input was about 53% higher for Controller C than for Controllers A and B. Overall, the differences between motor input for each participant and controller were fairly similar, with the exception of the much larger input for Participant N2 and Controller C. This result is noteworthy because, although the motor is on the least for Controller C, as shown in Table II, the average motor input over one cycle was about the same for each controller. Since Participant N2 had poor cadence tracking with Controller C, the motor required much larger inputs to maintain the stability of the system.

Controller C on average had larger cadence tracking errors and higher FES inputs when compared to Controllers A and B, resulting in the worst performance. In addition, for Participant N2, Controller C resulted in the largest overall tracking errors and highest motor and FES inputs (see Table II). Controller C does not account for the delay, which results in the muscle contractions starting too late and thus occurring in less efficient regions of the crank cycle, contributing to the poor tracking performance relative to Controllers A and B. Controller B had the best tracking performance; however, in general, it caused the FES inputs to be lower than the other controllers. The challenge with Controller B is that the motor is always active, as is often the case in current clinical practice, and both the motor and FES have the same cadence tracking control objective. Therefore, it is possible that for some participants, the FES may be too low to even elicit muscle contractions since the tracking could solely be achieved by the motor. FES has been shown to be beneficial for rehabilitation and it is desired for the participant to contribute as much as possible [1], [4], [5]. Controller A, although it did not perform as well as Controller B, had much better tracking than Controller C and resulted, on average, in more FES being applied than Controller B while still being at a comfortable level. In fact,

for the participants with NCs in Table II, the standard deviation of the cadence error, average motor input, and average peak FES input is 60.7%, 52.8%, and 41.3% larger, respectively, for Controller C than Controller A.

IX. CONCLUSION

In this brief, delay-dependent switching conditions and robust cadence tracking controllers are developed for a switched uncertain nonlinear dynamic system in the presence of bounded unknown additive disturbances and an unknown time-varying input delay. A Lyapunov-like stability analysis was performed on the proposed controllers, which guarantees semiglobal exponential tracking to an ultimate bound. An extension of the proposed controller is provided to maintain motor control throughout the crank cycle (as opposed to switching the motor ON and OFF), and for comparison, a third controller was developed assuming that the system had no input delay. Experiments were performed on six able-bodied participants and four participants with NCs to compare the performance of these three controllers. The results indicate that the proposed controller exhibited the desired performance of cadence tracking with FES contributions with an average cadence error of 0.01 ± 2.00 RPM for the able-bodied participants and 0.01 ± 2.72 RPM for participants with NCs. Future work will seek to develop a dual objective control system, where the motor will track the cadence for all time (to take advantages of the motor always being ON), while power or torque will be tracked by a delay-compensating FES controller (to ensure that the muscles are contributing). Additional work includes the development of adaptive control methods to either estimate the EMD or to otherwise provide compensation for the EMD. Furthermore, clinical trials can be performed to further validate the clinical impacts of compensating for the EMD or closed-loop FES control in general.

ACKNOWLEDGMENT

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 the sponsoring agency.

REFERENCES

- [1] C. A. Cousin, C. A. Rouse, V. H. Duenas, and W. E. Dixon, "Controlling the cadence and admittance of a functional electrical stimulation cycle," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 6, pp. 1181–1192, Jun. 2019.
- [2] D. J. Pons, C. L. Vaughan, and G. G. Jaros, "Cycling device powered by the electrically stimulated muscles of paraplegics," *Med. Biol. Eng. Comput.*, vol. 27, no. 1, pp. 1–7, Jan. 1989.
- [3] L. M. Schutte, M. M. Rodgers, F. E. Zajac, and R. M. Glaser, "Improving the efficacy of electrical stimulation-induced leg cycle ergometry: An analysis based on a dynamic musculoskeletal model," *IEEE Trans. Rehabil. Eng.*, vol. 1, no. 2, pp. 109–125, Jun. 1993.
- [4] M. J. Bellman, R. J. Downey, A. Parikh, and W. E. Dixon, "Automatic control of cycling induced by functional electrical stimulation with electric motor assistance," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1225–1234, Apr. 2017.
- [5] C. A. Cousin *et al.*, "Closed-loop cadence and instantaneous power control on a motorized functional electrical stimulation cycle," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2276–2291, Nov. 2020.
- [6] C. A. Cousin, P. Deptula, C. A. Rouse, and W. E. Dixon, "A switched Lyapunov-passivity approach to motorized FES cycling using adaptive admittance control," *IEEE Trans. Control Syst. Technol.*, to be published.

- [7] R. J. Downey, M. Merad, E. J. Gonzalez, and W. E. Dixon, "The time-varying nature of electromechanical delay and muscle control effectiveness in response to stimulation-induced fatigue," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 9, pp. 1397–1408, Sep. 2017.
- [8] J. Ding, A. S. Wexler, and S. A. Binder-Macleod, "A predictive fatigue model. I. Predicting the effect of stimulation frequency and pattern on fatigue," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 1, pp. 48–58, Mar. 2002.
- [9] Z. Li, M. Hayashibe, C. Fattal, and D. Guiraud, "Muscle fatigue tracking with evoked EMG via recurrent neural network: Toward personalized neuroprosthetics," *IEEE Comput. Intell. Mag.*, vol. 9, no. 2, pp. 38–46, May 2014.
- [10] E. S. Idsø, T. Johansen, and K. J. Hunt, "Finding the metabolically optimal stimulation pattern for FES-cycling," in *Proc. Conf. Int. Funct. Electr. Stimulation Soc.*, Bournemouth, U.K., Sep. 2004, pp. 239–241.
- [11] N. Sharma, C. M. Gregory, and W. E. Dixon, "Predictor-based compensation for electromechanical delay during neuromuscular electrical stimulation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 6, pp. 601–611, Dec. 2011.
- [12] I. Karafyllis, M. Malisoff, M. D. Queiroz, M. Krstic, and R. Yang, "Predictor-based tracking for neuromuscular electrical stimulation," *Int. J. Robust Nonlinear Control*, vol. 25, no. 14, pp. 2391–2419, Sep. 2015.
- [13] S. Obuz, R. J. Downey, A. Parikh, and W. E. Dixon, "Compensating for uncertain time-varying delayed muscle response in isometric neuromuscular electrical stimulation control," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 4368–4372.
- [14] B. C. Allen, C. A. Cousin, C. A. Rouse, and W. E. Dixon, "Cadence tracking for switched FES cycling with unknown input delay," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Nice, France, Dec. 2019, pp. 60–65.
- [15] B. Allen, C. Cousin, C. Rouse, and W. E. Dixon, "Cadence tracking for switched FES-cycling with unknown time-varying input delay," in *Proc. ASME Dyn. Syst. Control Conf.*, Oct. 2019, pp. 1–7.
- [16] B. C. Allen, K. Stubbs, and W. E. Dixon, "Robust cadence tracking for switched FES-cycling with an unknown time-varying input delay using a time-varying estimate," in *Proc. IFAC World Congr.*, 2020.
- [17] B. C. Allen, K. J. Stubbs, and W. E. Dixon, "Robust power and cadence tracking on a motorized FES cycle with an unknown time-varying input delay," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Dec. 2020, pp. 3407–3412.
- [18] B. C. Allen, K. J. Stubbs, and W. E. Dixon, "Saturated control of a switched FES-cycle with an unknown time-varying input delay," in *Proc. IFAC Conf. Cyber-Phys. Hum.-Syst.*, 2020.
- [19] M. Krstic, *Delay Compensation for Nonlinear, Adaptive, and PDE Systems*. Boston, MA, USA: Springer, 2009.
- [20] I. Karafyllis and M. Krstic, *Predictor Feedback for Delay Systems: Implementations and Approximations*. Springer, 2017.
- [21] J.-P. Richard, "Time-delay systems: An overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667–1694, Oct. 2003.
- [22] N. Bekiaris-Liberis and M. Krstic, "Compensation of time-varying input and state delays for nonlinear systems," *J. Dyn. Syst., Meas., Control*, vol. 134, no. 1, Jan. 2012, Art. no. 011009.
- [23] F. Mazenc, S.-I. Niculescu, and M. Bekaik, "Stabilization of time-varying nonlinear systems with distributed input delay by feedback of plant's state," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 264–269, Jan. 2013.
- [24] M. A. Henson and D. E. Seborg, "Time delay compensation for nonlinear processes," *Ind. Eng. Chem. Res.*, vol. 33, no. 6, pp. 1493–1500, Jun. 1994.
- [25] J.-Q. Huang and F. L. Lewis, "Neural-network predictive control for nonlinear dynamic systems with time-delay," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 377–389, Mar. 2003.
- [26] I. Chakraborty, S. Obuz, R. Licitra, and W. E. Dixon, "Control of an uncertain Euler-Lagrange system with known time-varying input delay: A pde-based approach," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 4344–4349.
- [27] S. Obuz, J. R. Klotz, R. Kamalapurkar, and W. Dixon, "Unknown time-varying input delay compensation for uncertain nonlinear systems," *Automatica*, vol. 76, pp. 222–229, Feb. 2017.
- [28] D. Bresch-Pietri and M. Krstic, "Delay-adaptive control for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1203–1218, May 2014.
- [29] A. Polyakov, D. Efimov, W. Perruquetti, and J.-P. Richard, "Output stabilization of time-varying input delay systems using interval observation technique," *Automatica*, vol. 49, no. 11, pp. 3402–3410, Nov. 2013.
- [30] D. Bresch-Pietri and M. Krstic, "Adaptive trajectory tracking despite unknown input delay and plant parameters," *Automatica*, vol. 45, no. 9, pp. 2074–2081, Sep. 2009.
- [31] D. Bresch-Pietri, J. Chauvin, and N. Petit, "Adaptive control scheme for uncertain time-delay systems," *Automatica*, vol. 48, no. 8, pp. 1536–1552, Aug. 2012.
- [32] D. Enciu, I. Ursu, and G. Tecuceanu, "Dealing with input delay and switching in electrohydraulic servomechanism mathematical model," in *Proc. 5th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Apr. 2018, pp. 713–718.
- [33] Y.-E. Wang, X.-M. Sun, and F. Mazenc, "Stability of switched nonlinear systems with delay and disturbance," *Automatica*, vol. 69, pp. 78–86, Jul. 2016.
- [34] F. Mazenc, M. Malisoff, and H. Ozbay, "Stability analysis of switched systems with time-varying discontinuous delays," in *Proc. Amer. Control Conf.*, May 2017, pp. 5177–5181.
- [35] M. Merad, R. J. Downey, S. Obuz, and W. E. Dixon, "Isometric torque control for neuromuscular electrical stimulation with time-varying input delay," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 971–978, May 2016.
- [36] B. C. Allen, K. J. Stubbs, and W. E. Dixon, "Characterization of the time-varying nature of electromechanical delay during FES-cycling," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 10, pp. 2236–2245, Oct. 2020.

A Topologically Inspired Path-Following Method With Intermittent State Feedback

Sage C. Edwards^{ID}, Duc M. Le, Dan P. Guralnik^{ID}, and Warren E. Dixon^{ID}

Abstract—Autonomous systems often operate in environments where state feedback may not be available, such as in anti-access and area denial environments. In these environments, it is often required that an agent track a path, despite interruptions in state feedback. As a result, the class of *Relay-Explorer* problems has emerged, where a switched system approach is used to account for intermittent state feedback. Past work on these problems established a framework for developing dwell-time conditions for stable tracking using these methods. However, existing work only applies to a limited class of reference paths and feedback region geometries. This letter advances a topologically inspired method for guaranteeing re-acquisition of feedback for nearly arbitrary geometries in arbitrary dimensions, all while relaxing the dwell-time conditions and retaining the uniformly ultimately bounded stability result from preceding work. Numerical experiments in the plane demonstrate an increase of hundreds of percentage points—even for fairly generic geometries—in the tracking error the agent could afford, using the proposed method, without sacrificing stability.

Index Terms—Computational geometry, hybrid logical/dynamical planning and verification, motion and path planning.

I. INTRODUCTION

STATE feedback is a critical component in designing path-planning methods used for guidance, navigation, and control of autonomous vehicles. Factors such as task definition, operating environment, sensor modality, and adversarial effects may result in intermittent state feedback, inhibiting a system's ability to achieve its task. For mobile platforms, one prominent approach to overcoming difficulties with obtaining state feedback is to design controllers and planners that enable computation of the state from available observations (see [1] in the context of visual servoing, and [2] in a SLAM setting). Other approaches arise in network control, where limitations on communication and communication delays affect the stability of the system (see [3] in the context of time-varying connectivity and [4] for an event triggered approach).

An emerging body of work embraces these difficulties by considering them as a class of *Relay-Explorer* problems. Using

a switched system approach, control laws are constructed which use state feedback for navigation when available (stabilizable mode), and open-loop state estimates otherwise (unstable mode). For example, in a single agent setting [5], to ensure stability, the agent has to repeatedly reacquire feedback by entering a fixed region where feedback is available. A time-varying feedback region may naturally arise in multi-agent settings. For example in [6], some agents may have the capability of relaying feedback information to feedback-denied members of their team by moving into communication radius.

Similar to the framework in [5], this letter considers a single agent tasked with tracking a desired path that may lie outside a known feedback region. State feedback is available when the agent is inside this region, and unavailable otherwise. Since the desired path may lie outside of the feedback region, the agent dead-reckons when feedback is not available. To achieve the task, a path-planning algorithm is designed to generate an auxiliary trajectory for the agent to track. The instabilities inherent to dead-reckoning impede the agent's ability to return to the feedback region. This forces the auxiliary trajectory to alternate between following the desired path and returning to the feedback region. This framework relies on a Lyapunov-based switched system analysis [7] to derive dwell-time conditions dictating the duration a system can remain in each operating mode while ensuring stability.

The predominant factor affecting dwell-time conditions in this setting is the need to guarantee re-acquisition of state feedback. To furnish this guarantee, results in [5] and [8] require the agent to dead-reckon to a point where the region of state uncertainty is contained within the feedback region—the *Inscribed Ball Criterion* (IBC). Crucially, these results only consider circular feedback regions where the dwell-time conditions only depend on the distance of the agent to the feedback region. However, applying the same approach to more general geometries—the expected norm in real-life applications—results in unnecessarily conservative bounds. For example, when an agent moves towards a long and narrow rectangular feedback region, Fig. 1 shows that a better strategy is to aim at a point beyond the region, affording a larger error margin at the target point. Intuitively, small perturbations in the shape of this region should not result in a change of strategy, motivating a topological approach. Moreover, it is clear that the preceding considerations could not be easily replicated for an arbitrary geometry of the feedback region, because of complex interactions between local properties (e.g., curvature at nearby boundary points) and global properties (e.g., concavities, spirals, etc.).

This letter addresses the need for generating re-entry guarantees for arbitrary geometries of the feedback region. To this end,

Manuscript received August 21, 2020; accepted February 20, 2021. Date of publication March 22, 2021; date of current version April 12, 2021. This letter was recommended for publication by Associate Editor N. Amato and Editor F. van der Stappen upon evaluation of the reviewers' comments. This work is supported in part by AFOSR Award Numbers FA9550-19-1-0169, FA9550-18-1-0109, NEEC Award Number N00174-18-1-0003, and in part by Office of Naval Research Grant N00014-13-1-0151. (Corresponding author: Sage Edwards.)

The authors are with the Department of Mechanical and Aerospace Engineering, University of Florida Gainesville, Gainesville, Florida 32611 USA (e-mail: sageedwards@ufl.edu; ledan50@ufl.edu; danguralnik@ufl.edu; wdixon@ufl.edu).

Digital Object Identifier 10.1109/LRA.2021.3067295

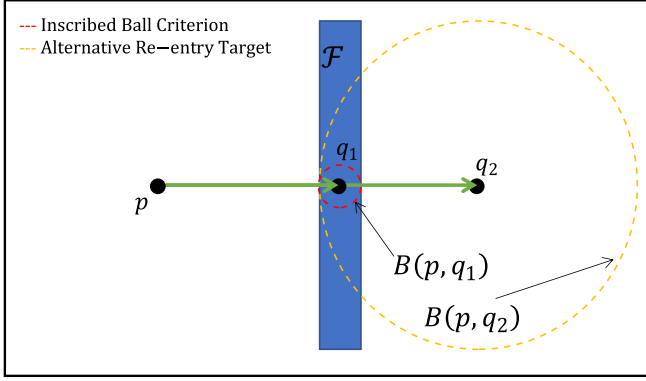


Fig. 1. Given a feedback region \mathcal{F} , an initial starting point p , and a target point $q_1 \in \mathcal{F}$, the ball $B(p, q_1)$ about q_1 inscribed in \mathcal{F} represents the maximum allowed region of uncertainty based on IBC. For this geometry, a much larger region of uncertainty may be allowed for an overwhelming majority of starting points p , provided one does not insist on q_1 to lie in \mathcal{F} , replacing q_1 with q_2 , for example.

separation properties of embedded spheres in \mathbb{R}^D , $D \geq 2$ are deployed (Sections IV-A–IV-B). The resulting *Topological Re-entry Criterion* is applied to increase the maximum dwell-time by raising the upper bound on the allowed error growth, in a state-dependent fashion, while being cognizant of the geometry of the feedback region in the agent's vicinity (Sections IV-C–IV-E). The proposed framework generalizes the results in [5] from circular geometries in \mathbb{R}^2 to nearly arbitrary contractible geometries in \mathbb{R}^D of the feedback region, by casting the problem of constructing auxiliary trajectories as an optimization problem (Section IV-D): selecting an optimal auxiliary trajectory is tantamount to computing, at every point of the desired path, the largest tracking error the agent can afford without losing the guarantee of return (the agent's *uncertainty budget* at that point). In particular, this framework guarantees uniformly ultimately bounded errors for repeated optimal choices of the auxiliary trajectories (Theorem 4). To assess the degree of improvement offered by the topological re-entry criterion over IBC an algorithm for computing uncertainty budgets is given in Section V and numerical studies in Section VI.

II. PROBLEM STATEMENT

A. Notation

For any integer $D \geq 1$, the Euclidean distance between $p, q \in \mathbb{R}^D$ is denoted by $\text{dist}(p, q) \triangleq \|p - q\|_2$. Let \mathbb{S}^{D-1} and \mathbb{B}^D denote the Euclidean unit sphere and closed unit ball in \mathbb{R}^D , respectively. Given $p \in \mathbb{R}^D$ and $r > 0$, the open ball $B_r(p)$ of radius r about p is the set of $q \in \mathbb{R}^D$ with $\text{dist}(p, q) < r$. The shortest distance from a point to a set is $\text{dist}(p, A) \triangleq \inf\{\text{dist}(p, q) : q \in A\}$. The diameter of $A \subset \mathbb{R}^D$ is $\text{diam}(A) \triangleq \sup\{\text{dist}(p, q) : p, q \in A\}$. The closure of $A \subseteq \mathbb{R}^D$ is denoted by $\text{cl}(A)$. A point is an interior point of a set if $B_r(p) \subseteq A$ for some $r > 0$. The set of interior points of $A \subseteq \mathbb{R}^D$ is denoted by A° . A different notion of interior is associated with embeddings of \mathbb{S}^{D-1} in \mathbb{R}^D , see (11).

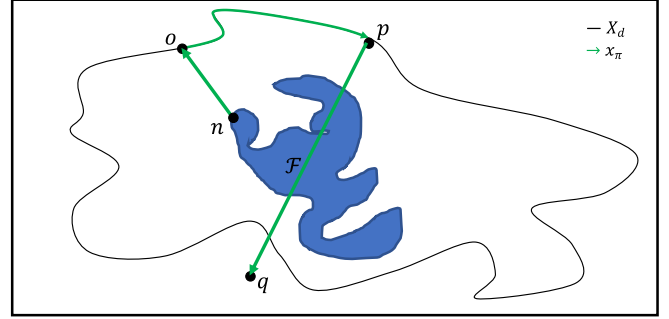


Fig. 2. The auxiliary trajectory x_π , defined by a path plan $\pi = (n, o, p, q)$, is superimposed over the desired path X_d , for a generic feedback region \mathcal{F} (see Definition 1). Note that the point q need not lie in \mathcal{F} .

B. Agent Dynamics

This letter considers an agent with dynamics modeled by

$$\dot{x} = f(x, t) + \zeta(t) + d(t), \quad (1)$$

where $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$, $D \geq 2$, denotes the state; $f : \mathbb{R}^D \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$ denotes locally Lipschitz drift dynamics; $d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$ denotes an exogenous disturbance; and $\zeta : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$ denotes a control input. The following assumption is used in the subsequent development.

Assumption 1: The exogenous disturbance d satisfies $\|d(t)\| \leq \bar{d}$ for all $t \in \mathbb{R}_{\geq 0}$, where $\bar{d} \in \mathbb{R}_{>0}$ is known. ■

C. Control Objective

Let $\mathcal{F} \subset \mathbb{R}^D$ denote a known region where state feedback is available, i.e., state feedback is available to the agent if and only if $x \in \mathcal{F}$. The *feedback region* \mathcal{F} is modeled as the closure of the interior region of a polyhedral sphere C . Equivalently, \mathcal{F} is the image of an embedding of \mathbb{B}^D in \mathbb{R}^D (see Section IV). The *feedback-denied region*, $\mathcal{F}^c \triangleq \mathbb{R}^D \setminus \mathcal{F}$, is the set of states where feedback is not available.

The agent is tasked with following a desired polygonal path X_d , which is provided as a sequence of way points $(\mathcal{P}_0, \dots, \mathcal{P}_M)$ in \mathbb{R}^D , some of which may lie outside of \mathcal{F} . Repeated dead-reckoning along the sequence of way points is inherently unstable outside of \mathcal{F} . This motivates an approach where the agent follows a sequence of *auxiliary trajectories*, relaying between the desired path and the feedback region (see Fig. 2), to ensure the error system remains bounded.

Definition 1 (Auxiliary Trajectory): There are two types of auxiliary trajectories. The auxiliary trajectory $x_\pi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$ with path plan $\pi = (n, o, p, q)$, is defined as the concatenation of three trajectories determined by four way points: n is the point of departure from the feedback region; o is the first point along a segment of X_d the agent selects to follow; p is the point of departure from X_d ; and q is the target point for the return trajectory to the feedback region. From n to o , and from p to q , x_π restricts to straight line trajectories¹ with constant speed v_0 . From o to p , x_π coincides with the desired path X_d , with a piecewise linear parametrization of constant speed v_0 . For the second type of auxiliary trajectory—with plan $\pi = (p, q)$ —set

¹Curved trajectories may improve performance, but outside our scope.

x_π to coincide with a linearly parametrized line segment from p to q with constant speed v_0 .

Remark 1: Plans $\pi = (p, q)$ are used for acquisition of feedback from points $p \notin \mathcal{F}$, while plans $\pi = (n, o, p, q)$ are used for tracking X_d and reacquiring feedback. A plan terminates the moment feedback is reacquired.

Assumption 2: It is assumed that feedback acquisition is instantaneous upon re-entry into the feedback region. ■

Remark 2: Between the executions of two plans, the agent travels through \mathcal{F} , taking advantage of the available feedback. Two cases may occur. If the next point \mathcal{P}_m along X_d lies in \mathcal{F} , there is no need for auxiliary planning. Otherwise, the agent travels to the point $n \in \mathcal{F}$ closest to \mathcal{P}_m .

To quantify the tracking objectives, define

$$e \triangleq x - x_\pi, \quad \hat{e} \triangleq \hat{x} - x_\pi, \quad \tilde{e} \triangleq x - \hat{x}, \quad (2)$$

where $\hat{x} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$ is the agent's open-loop state estimate, and $e, \hat{e}, \tilde{e} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$ are the *actual tracking error*, *estimated tracking error*, and *state estimation error*, respectively. The challenge is to regulate the norm of the actual tracking error to remain, eventually, below a prescribed bound.

D. Switched Controller

Let $\mathcal{S} \triangleq \{a, u\}$ be the set of indices denoting the operating modes, where a and u correspond to the modes where feedback is available and unavailable, respectively. Mode a is active when $x \in \mathcal{F}^\circ$. Mode u is active otherwise. Let $\sigma(x) \in \mathcal{S}$ denote the switching signal indicating the active subsystem. The control input takes the form $\zeta \triangleq \zeta_a(x, t)$ when $\sigma(x) = a$, and $\zeta \triangleq \zeta_u(\hat{x}, t)$ when $\sigma(x) = u$, where $\zeta_a, \zeta_u : \mathbb{R}^D \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^D$ are the control inputs when feedback is *available* and *unavailable*, respectively. The closed loop error system for controllers of this kind were studied in [5], Section IV.

III. ERROR BOUNDS

To facilitate the analysis, the i^{th} instant when σ switches from u to a is denoted by $t_i^a \in \mathbb{R}_{\geq 0}$ for all $i \in \mathbb{Z}_{>0}$, i.e., the instant the agent enters the interior of the feedback region. The i^{th} instant when σ switches from a to u is denoted by $t_i^u \in \mathbb{R}_{\geq 0}$, i.e., the instant the agent exits \mathcal{F}° . Based on the switching instants, dwell-times of the i^{th} activation of the subsystems a and u are defined as $\Delta t_i^a \triangleq t_i^u - t_i^a$ and $\Delta t_i^u \triangleq t_{i+1}^a - t_i^u$, respectively.

To analyze the switched system, candidate Lyapunov-like functions are defined as

$$V_e \triangleq \frac{1}{2} \|e\|^2, \quad V_{\hat{e}} \triangleq \frac{1}{2} \|\hat{e}\|^2, \quad V_{\tilde{e}} \triangleq \frac{1}{2} \|\tilde{e}\|^2, \quad (3)$$

where $V_e, V_{\hat{e}}, V_{\tilde{e}} : \mathbb{R}^D \rightarrow \mathbb{R}_{\geq 0}$. To ensure a bound on the error system, the following assumption is made.

Assumption 3: Based on the design of the control input in Section II-D, it is assumed that the time derivatives of (3) yields

$$\dot{V}_e \leq -2\lambda_s V_e, \quad \sigma = a, \quad (4)$$

$$\dot{V}_{\hat{e}} \leq -2\lambda_s V_{\hat{e}}, \quad \sigma = u, \quad (5)$$

$$\dot{V}_{\tilde{e}} \leq \begin{cases} -2\lambda_{\tilde{e}} V_{\tilde{e}}, & \sigma = a, \\ 2\lambda_u V_{\tilde{e}} + \delta, & \sigma = u, \end{cases} \quad (6)$$

where $\lambda_s, \lambda_{\tilde{e}}, \lambda_u, \delta \in \mathbb{R}_{>0}$ are known constants.² ■

While the agent is in the feedback-denied region (i.e., $\sigma = u$), solving the ordinary differential inequalities in (4)–(6) and substituting in (2) and (3) yields

$$\|\hat{e}(t)\| \leq \|\hat{e}(t_i^u)\| e^{-\lambda_s \Delta t}, \quad (7)$$

$$\|\tilde{e}(t)\|^2 \leq \|\tilde{e}(t_i^u)\|^2 e^{2\lambda_u \Delta t} + \frac{\delta}{\lambda_u} [e^{2\lambda_u \Delta t} - 1], \quad (8)$$

for all $t \in [t_i^u, t_{i+1}^a)$, where $\Delta t \triangleq t - t_i^u$. Since $e = \hat{e} + \tilde{e}$, and using the reset maps³ from [5], the bounds in (7) and (8) yield

$$\|e(t)\| \leq \rho(t - t_i^u), \quad t \in [t_i^u, t_{i+1}^a), \quad (9)$$

where

$$(\rho(\Delta t))^2 \triangleq \frac{\delta}{\lambda_u} [e^{2\lambda_u \Delta t} - 1], \quad (10)$$

and $\rho(\Delta t)$ is referred to as the *radius of uncertainty* at time t .

This bound holds for *any* trajectory under *any* controller satisfying Assumption 3, enabling the development of a dwell-time condition (Theorem 3).

IV. CRITERIA FOR GUARANTEED RE-ENTRY

A. Preliminaries: Embedded Spheres in Euclidean Space

An embedded sphere in \mathbb{R}^D is defined to be the image $C \triangleq \gamma(\mathbb{S}^{D-1})$ of an injective continuous map $\gamma : \mathbb{S}^{D-1} \rightarrow \mathbb{R}^D$. For $D = 2$, the following classical result may be applied.

Theorem 1 (Jordan-Schönflies, see [9], Thm. E1): An injective continuous map $\gamma : \mathbb{S}^1 \rightarrow \mathbb{R}^2$ extends to a homeomorphism $\Gamma : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, a continuous map with continuous inverse satisfying $\Gamma(p) = \gamma(p)$ for all $p \in \mathbb{S}^1$. The mapping Γ is called a Schönflies extension of γ .

For higher dimensions $D > 2$, Schönflies extensions exist under the additional condition that the embedding γ is *collared* [10], [11]. The embedding γ is collared if there is an injective continuous map $\tilde{\gamma} : \mathbb{S}^{D-1} \times [-1, 1] \rightarrow \mathbb{R}^D$ such that $\tilde{\gamma}(p, 0) = \gamma(p)$ for all $p \in \mathbb{S}^{D-1}$. It is well-known that polyhedral—and, more generally, piecewise-regular—maps γ (with finitely many faces) are collared (see §11 in [12]).

The interior and exterior regions of a collared embedded sphere C are defined as

$$\text{int}(C) \triangleq \Gamma((\mathbb{B}^\circ)^D), \quad \text{ext}(C) \triangleq \Gamma(\mathbb{R}^D \setminus \mathbb{B}^D), \quad (11)$$

where Γ is any Schönflies extension of γ . It is important to note $\text{int}(C)$ and $\text{ext}(C)$ do not depend on the choice of map γ —only on its image, C . They are also independent of the choice of the Schönflies extension Γ . Moreover, the boundary sets $\partial \text{int}(C)$ and $\partial \text{ext}(C)$ coincide with C , and C separates every point $p \in \text{int}(C)$ from every point $q \in \text{ext}(C)$ in the sense that

²An example of a controller satisfying the assumptions made thus far is given in Section VI of [5] where values for $\lambda_s, \lambda_{\tilde{e}}, \lambda_u$, and δ are shown to be functions of the agent dynamics and the disturbance bound \bar{d} .

³Upon each re-entry into \mathcal{F} , it is possible to reset the auxiliary path x_π to a new path and have $\hat{x}(t_{i+1}^a) = x(t_{i+1}^a)$ at t_{i+1}^a . In other words, the auxiliary path is updated based on the re-entry location of the agent instead of having to travel to the desired re-entry location before returning to the desired path to follow. Resetting the errors to zero results in the elimination of the minimum dwell-time condition and the vanishing of the initial conditions from the maximum dwell-time condition.

any continuous curve from p to q must intersect C . For the rest of this letter we make the following assumption.

Assumption 4: The feedback region \mathcal{F} is the closure of $\text{int}(C)$, where C is a collared embedded sphere in \mathbb{R}^D . ■

B. Construction of a “target Region”

In the notation of Section III, given t_i^u , consider a plan $\pi = (n, o, p, q)$. Alternatively, consider $\pi = (p, q)$, while setting $t_i^u = 0$. Let x_π be the associated auxiliary trajectory. Let $p = x_\pi(t_i^p)$ and $q = x_\pi(t_i^q)$, then the *initial and final uncertainty radii* for the plan π are defined as

$$\rho_{init}(\pi) \triangleq \rho(t_i^p - t_i^u), \quad \rho_{fin}(\pi) \triangleq \rho(t_i^q - t_i^u), \quad (12)$$

where ρ is defined in (10). The *regions of uncertainty* are

$$U_{init}(\pi) \triangleq B_{\rho_{init}(\pi)}(p), \quad U_{fin}(\pi) \triangleq B_{\rho_{fin}(\pi)}(q). \quad (13)$$

In [5] and [8], re-entry is guaranteed by selecting $q \in \mathcal{F}$ so that $U_{fin}(\pi) \subset \mathcal{F}$. This method is referred to as the *Inscribed Ball Criterion* (IBC). Treating this inclusion as a constraint results in a bound on $\rho_{fin}(\pi)$, and hence also on $\rho_{init}(\pi)$, since the function ρ is known. The example in Fig. 1 illustrates the need for a target region much larger than \mathcal{F} , in which to fit $U_{fin}(\pi)$, to obtain less conservative bounds. In this section such target regions are introduced.

Recall that $\mathcal{F}^\circ = \text{int}(C)$. Given a point $p \in \text{ext}(C)$ and a closed, connected region $R \subset \mathbb{R}^D$ with $p \in R^\circ$, define $T_{C,R}(p)$ to be the collection of all points $q \in R^\circ$ for which any smooth curve $\gamma : [0, 1] \rightarrow R^\circ$ from p to q must pass through a point of C . Let $E_{C,R}(p)$ denote the set of all $y \in C$ such that, for some $q \in T_{C,R}(p)$ there is a smooth curve $\gamma : [0, 1] \rightarrow R^\circ$ from p to q that crosses C exactly once at y .

Remark 3: In the above, one is allowed to restrict attention to smooth curves γ which intersect C transversely, i.e., their tangent line at any point of intersection with C and the tangent to C at that point together span \mathbb{R}^D (see §10 of [12]).

Lemma 1: If $q \in T_{C,R}(p)$, then any curve in R from p to q must pass through a point of $E_{C,R}(p)$.

Proof: Suppose a curve $\gamma : [0, 1] \rightarrow R^\circ$ starts at point $p = \gamma(0)$ and terminates at the point $q = \gamma(1)$. Let $t' \triangleq \inf\{t \in [0, 1] : \gamma(t) \in C\}$ be the first time γ crosses C . By Remark 3, one may assume γ only crosses C transversely. Set $q' \triangleq \gamma(t')$, and let \mathcal{U} be a neighborhood of q' not containing any other intersection point of γ and C such that $C \cap \mathcal{U}$ is a single interval. Find $\Delta t > 0$ such that $\gamma([t', t' + \Delta t]) \subset \mathcal{U}$ and now set $q'' \triangleq \gamma(t' + \frac{\Delta t}{2})$. Since the curve $\gamma' \triangleq \gamma|_{[0, t' + \Delta t/2]}$ crosses C exactly once, one finds that $q'' \in \text{int}(C)$ and $q' \in E_{C,R}(p)$, as required (see Fig. 3(left)).

C. Guarantee of Re-Entry Into the Feedback Region

Assume once more the agent is executing a plan $\pi = (n, o, p, q)$ over the time period $t \in [t_i^u, t_{i+1}^a]$. At time t_i^p , the agent departs in the direction of \mathcal{F} by dead-reckoning to a point $q \in \mathbb{R}^D$ (note that it is possible for q to not lie in \mathcal{F}). For all $t \in [t_i^p, t_i^q]$, $t_i^q - t_i^p = \frac{\|q-p\|}{v_0}$ one has $x_\pi(t) = p + (t - t_i^p)v$, where $v = v_0 \frac{q-p}{\|q-p\|}$. The true position $x(t)$ is guaranteed to lie in the ball $B_{\rho(\Delta t)}(x_\pi(t))$, where Δt and ρ are defined in (7) and (10), respectively.

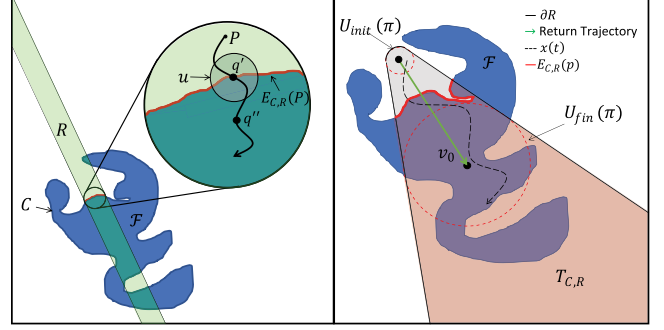


Fig. 3. Illustration of the proofs of Lemma 1 (left) and Theorem 2 (right).

Theorem 2: Suppose $U(\pi) \triangleq \bigcup_{t \in [t_i^p, t_i^q]} B_{\rho(t-t_i^u)}(x_\pi(t))$ is contained in the interior of a region R . If $U_{fin}(\pi) \subset T_{C,R}(p)$ then there exists $t \in [t_i^p, t_i^q]$ with $x(t) \in \mathcal{F}$.

Proof: If $U_{fin}(\pi) \subset T_{C,R}(p)$, then $q = x(t_i^q) \in T_{C,R}(p)$. Apply Lemma 1 to this point and the curve $\gamma : [t_i^p, t_i^q] \rightarrow R$, concluding that γ had to pass through a point of $E_{C,R}(p)$. Thus, $x(t)$ entered $\text{int}(C)$ at some earlier time $t \in (t_i^p, t_i^q)$ (see Fig. 3 (right)).

Remark 4: If R is selected as a convex region, then ensuring $U_{init}(\pi) \subset R$ and $U_{fin}(\pi) \subset T_{C,R}(p)$ suffices for meeting the requirements of Theorem 2.

D. Path-Planning With Infinite Cylinders

Theorem 2 states region R should be selected so the error growth in (9) of the agent is accounted for in R up until re-entry can be guaranteed. Since an upper bound on the error growth rate is given in (9), R may be designed to contain all possible trajectories of $x(t)$, given a plan $\pi = (n, o, p, q)$. In principle, given the reference point of departure $p = x_\pi(t_i^p)$ and a reference target velocity vector v with $\|v\| = v_0$, R could always be selected to be the (unbounded) *cone of uncertainty* $\bigcup_{t \in [t_i^p, \infty)} B_{\rho(t-t_i^p)}(p + (t - t_i^p)v)$ in that direction. However, this choice is challenging from a computational perspective, because of the non-linearity of ρ .

Given any point $p \in \mathcal{F}^c$, the planner needs to select q and a region R best suited for guaranteeing the agent's return to \mathcal{F} . In general, to apply Theorem 2 a rich collection $\mathcal{R}(p)$ of regions R satisfying $p \in R^\circ$ with “sufficiently large” $T_{C,R}(p)$ for some $R \in \mathcal{R}(p)$ must be designed.

Fig. 4 presents two alternatives to the approach using cones of uncertainty: the cylindrical geometry in Fig. 4(A) and a conical geometry in Fig. 4(B). The subsequent development only considers the cylindrical geometry, in the interest of reducing computational cost. Specifically, consider

$$\mathcal{R}(p) \triangleq \{R_{p,v,w} : v \in \mathbb{R}^D, \|v\| = v_0, w > 0\}, \quad (14)$$

where $R_{p,v,w}$ is the solid cylinder of radius w whose axis passes through p and has direction v (see Fig. 4(A)),

$$R_{p,v,w} \triangleq \left\{ z \in \mathbb{R}^D : \left\| (z-p) - \frac{(z-p) \cdot v}{v_0^2} v \right\| \leq w \right\}. \quad (15)$$

A pair (v, w) will be referred to as *return parameters*. For $R = R_{p,v,w}$, the following observations are made.

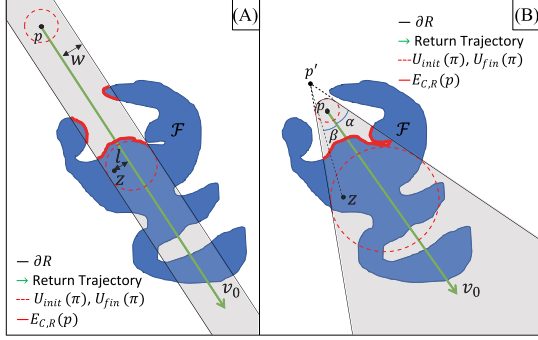


Fig. 4. Illustrations of possible target regions R for use with the planner. The “cylinder” geometry implemented in this letter is shown in A. More general geometries are possible (for example the “conical” region shown in B), but are not discussed here due to higher computational overhead.

- For excessively large values of w (e.g., w big enough for $C \subset R$) one has $T_{C,R}(p) = \mathcal{F}$. Therefore, any ball contained in $T_{C,R}(p)$ provides little improvement upon IBC of guaranteeing return.
- More generally, note that $R \setminus C$ has no more than two unbounded components. When there is only one such component, the existence of a ball of radius w contained in $T_{C,R}(p)$ cannot be guaranteed.
- In all other cases, the collection of balls of radius w contained in $T_{C,R}(p)$ is non-empty: a point of the form $q = p + \alpha v$ always exists with $\alpha > 0$ and $B_w(q) \subset T_{C,R}(p)$ (see Fig. 4(A)).

The developed planner seeks to maximize the arc length from o to p along X_d subject to the constraint of re-entry being guaranteed by $\mathcal{R}(p)$. As a result, maximizing $t_i^p - t_i^u$ subject to the existence of a point $q = p + \alpha v$ with $B_w(q) \subset T_{C,R}(p)$ and $R = R_{p,v,w}$ provides a *lower bound* on the time the agent could spend tracking X_d , which the planner seeks to maximize.

$$\Delta t^* \triangleq \max_{p,v,w,\alpha} (\rho^{-1}(w) - \alpha) \quad \text{s.t.} \begin{cases} B_w(p + \alpha v) \subset T_{C,R}(p), \\ R = R_{p,v,w} \in \mathcal{R}(p), \end{cases} \quad (16)$$

given a ball $B_w(q) \subset T_{C,R}(p)$. To guarantee re-entry, it is required that $x(t_i^q) \in B_w(q)$ and $\rho(t_i^q - t_i^u) \leq w$. It then follows that $t_i^p - t_i^u \leq \rho^{-1}(w) - \alpha$ because $\alpha = t_i^q - t_i^p$.

For each $R = R_{p,v,w} \in \mathcal{R}(p)$, the *Maximum Allowed Uncertainty Radius (MAUR)* is defined as

$$\text{MAUR}(p, v, w) \triangleq \max_{\alpha > 0} \rho(\rho^{-1}(w) - \alpha) \quad \text{s.t.} \quad B_w(p + \alpha v) \subset T_{C,R}(p), \quad (17)$$

representing the agent’s budget of uncertainty at the point p for the return parameters v, w .

In practice, assuming the distance between any two consecutive way points along X_d does not exceed some small $\epsilon > 0$, the optimization in (16) ranges over all $p = \mathcal{P}_{m'}$, $m' \geq m$, where m satisfies $o = \mathcal{P}_m$. Note that it is possible for the optimal p to satisfy $p = o$. Hence, an optimal path plan, $\pi^* = (n^*, o, p^*, q^*)$,

will be given by

$$n^* \triangleq \arg \min_n \{\text{dist}(n, o) : n \in \mathcal{F}\}, \quad (18)$$

$$(p^*, v^*, w^*, \alpha^*) \triangleq \arg \max_{p,v,w,\alpha} (\rho^{-1}(w) - \alpha), \quad (19)$$

$$q^* \triangleq p^* + \alpha^* v^*, \quad (20)$$

subject to the same constraints as (16), and noting that $p = x_{\pi^*}(\Delta t^*)$ is an implicit constraint on π^* .

E. Dwell-Time Analysis

Definition 2 (Feasible Region): For every point $p \in \mathcal{F}^c$, let the *feasible region* \mathcal{G} be defined by

$$\mathcal{G} \triangleq \mathcal{F} \cup \{p \in \mathcal{F}^c : v_0^{-1} \text{dist}(p, \mathcal{F}) < \tau(p)\}, \quad (21)$$

where $\tau(p) \triangleq \max_{v,w,\alpha} \{\rho^{-1}(w) - \alpha\}$, subject to $B_w(p + \alpha v) \subset T_{C,R}(p)$ and $R = R_{p,v,w}$. The *feasible region for initialization* is defined by

$$\mathcal{G}_0 \triangleq \mathcal{F} \cup \{p \in \mathcal{F}^c : \tau(p) > 0\}. \quad (22)$$

Theorem 3: Suppose an agent is given whose motion is governed by (1) and the controller from Section II-D. Moreover, suppose Assumptions 1–3 are satisfied. Let \mathcal{F} be a region satisfying Assumption 4. Also suppose that, given $x(0) \in \mathcal{G}_0$ and $e(0) = \hat{e}(0) = \tilde{e}(0) = 0$, the agent executes a sequence of plans as follows: (a) if $x(0) \in \mathcal{F}$ then let $\pi_i = (n_i, o_i, p_i, q_i)$, $i \geq 1$ each with an associated region $R_i \in \mathcal{R}(p_i)$ as defined in (14); (b) otherwise, let $\pi_0 = (p_0, q_0)$ with $p_0 = x(0)$ and an associated region $R_0 \in \mathcal{R}(p_0)$, followed by a sequence of plans as in (a). If, between plan executions the agent is confined to \mathcal{F} , and all the plans satisfy the re-entry condition of Theorem 2—that is $U_{fin}(\pi_i) \subset T_{C,R_i}(p_i)$ for all i —then the actual tracking error is ultimately bounded, uniformly over \mathcal{G}_0 and the set of plans, provided the switching signal satisfies the maximum dwell-time condition $\rho(\Delta t_i^u) \leq \rho_{fin}(\pi_i)$, written explicitly as

$$\Delta t_i^u \leq \frac{1}{2\lambda_u} \ln \left(\frac{\lambda_u (\rho_{fin}(\pi_i))^2}{\delta} + 1 \right), \quad (23)$$

where $\rho_{fin}(\pi_i)$ is defined in (12).

Proof: For $x(0) \in \mathcal{G}_0 \setminus \mathcal{F}$ it takes at most $\rho^{-1}(\text{diam}(\mathcal{F}))$ time to acquire feedback. Therefore, without loss of generality assume $x(0) \in \mathcal{F}$. With Assumptions 1–3 satisfied, suppose that plan π_i also satisfies $U_{fin}(\pi_i) \subset T_{C,R_i}(p_i)$. By Theorem 2, x is guaranteed to re-enter into the feedback region \mathcal{F} , while $\rho_{fin}(\pi_i)$ bounds the error growth from above. One can then apply the proof of Theorem 1 in [5] to deduce that the tracking error is ultimately bounded uniformly over \mathcal{G}_0 and the set of plans, since it satisfies the dwell-time condition (23).

Theorem 4: Suppose X_d is a polygonal curve contained in the feasible region \mathcal{G} . Then X_d can be tracked with a guarantee of re-entry, for any initial condition $x(0) \in \mathcal{G}_0$, provided the error system is initialized with $e(0) = \hat{e}(0) = \tilde{e}(0) = 0$.

Proof: Suppose X_d is a polygonal curve contained in the feasible region \mathcal{G} . For each m , let $\nu(m)$ denote the smallest $m' > m$ such that $\mathcal{P}_{m'} \notin \mathcal{F}$. The proof proceeds by induction. For the base step, at time $t = 0$, if $x(0) \in \mathcal{F}$ then set $t_1^a = 0$;

otherwise, the agent executes $\pi_0 = (p_0, q_0)$ with $p_0 = x(0)$ and $q_0 = x(0) + \alpha_0 v_0$, where (v_0, w_0, α_0) realizes the maximum in the definition of $\tau(x(0))$. By Theorem 2, entry into \mathcal{F} is guaranteed, resulting in $t_1^a \in [0, \alpha_0]$. For the induction hypothesis, assume plans $\pi_i = (n_i, o_i, p_i, q_i = p_i + \alpha_i v_i)$, $i = 1, \dots, k$ have been constructed so that (a) $o_1 = \mathcal{P}_0$; (b) $p_i = \mathcal{P}_{m_i}$ with $1 < m_1 < \dots < m_k$; (c) $o_{i+1} = \mathcal{P}_{\nu(m_i)}$ for $i = 1, \dots, k-1$; and (d) the conditions of Theorem 3 hold.

Let $x : [0, t_{k+1}^a] \rightarrow \mathbb{R}^D$ be an execution of these plans and note $x(t_{k+1}^a) \in \mathcal{F}$. For the induction step, compute a point $n_{k+1} \in \mathcal{F}$ at minimum distance to $o_{k+1} \triangleq \mathcal{P}_{\nu(m_k)}$. Select π_{k+1} to be the optimal plan $\pi^* = (n^*, o, p^*, q^* = p^* + \alpha^* v^*)$ from Section IV-D, for a choice of $n^* = n_{k+1}$ and $o = o_{k+1}$. Set $p_{k+1} = p^*$, $\alpha_{k+1} = \alpha^*$, $v_{k+1} = v^*$, and $q_{k+1} = q^*$.

Now extend x as follows: first, the agent proceeds from $x(t_{k+1}^a)$ to n_{k+1} through \mathcal{F} —while tracking X_d , if $\nu(m_k) > m_k + 1$ —defining the behavior of x over the time interval $[t_{k+1}^a, t_{k+1}^u]$; next, the agent executes the plan π_{k+1} , which guarantees re-entry into \mathcal{F} at some time $t_{k+2}^a \leq t_{k+2}^u$. Now it is required to show that the dwell-time condition $\rho(\Delta t_{k+1}^u) \leq \rho_{fin}(\pi_{k+1})$ of Theorem 3 is satisfied.

Since ρ is strictly increasing, this is equivalent to requiring $\Delta t_{k+1}^u \leq \rho^{-1}(\rho_{fin}(\pi_{k+1})) = t_{k+1}^q - t_{k+1}^u$. Recalling that $\Delta t_{k+1}^u = t_{k+2}^a - t_{k+1}^u$ finishes the argument.

Remark 5: Note the path plan π_{k+1} in the proof covers the range of consecutive way points $o_{k+1} = \mathcal{P}_{\nu(m_k)}, \dots, \mathcal{P}_{m_{k+1}} = p_{k+1}$ along X_d , $\nu(m_k) \geq m_k + 1$, while the preceding plan π_k covered the range $o_k = \mathcal{P}_{\nu(m_{k-1})}, \dots, \mathcal{P}_{m_k} = p_k \subset X_d$. Hence, progress will be made along X_d by any execution of π_{k+1} . In particular, there are no Zeno executions.

V. PRECOMPUTATION AND PLAN GENERATION

Since the geometry of the feedback region and the evolution of the region of uncertainty are known, a brute-force algorithm can be used to obtain all of the needed information for future path-planning, provided the way points \mathcal{P}_m , $m = 0, \dots, M-1$, form a subdivision of X_d of sufficiently fine mesh⁴ $\epsilon > 0$. For each m , an approximation to the solution of the optimization problem (16) for \mathcal{P}_m is obtained by solving for MAUR(\mathcal{P}_m, v, w)—see (17)—over a sufficiently dense range of possible return parameters (v, w) , and storing the solution in a data-set. The plans π_i constructed in the preceding section are then selected based on the information in the *data-set* to satisfy the mission objectives. To select a return trajectory, the agent solves the optimization problem in (19) by executing a search over the data-set.

To construct the data-set, for each \mathcal{P}_m , $m = 0, \dots, M-1$, return parameters (v_j, w_k) , $v_j \triangleq v_0[\cos(\theta_j), \sin(\theta_j)]$ are used, $\Theta \triangleq \{\theta_j \triangleq 2\pi j/J : j = 0, \dots, J-1\}$, and $W \triangleq \{w_0, \dots, w_{K-1}\} \subset \mathbb{R}_{\geq 0}$. Given the feedback region \mathcal{F} , Algorithm 1 computes $T_{C,R}(\mathcal{P}_m)$ for each $R = R_{\mathcal{P}_m, v_j, w_k}$. Next, the nearest return target and associated MAUR are computed:

$$q_{m,j,k} \triangleq \mathcal{P}_m + \alpha_{m,j,k} v_j, \quad (24)$$

⁴Recall that a subdivision S of X_d has mesh $\mu(S) \leq \epsilon$ if $\text{dist}(p, q) \leq \epsilon$ for every pair of consecutive points p, q in the subdivision.

Algorithm 1: Data-Set for R .

Require: X_d as a list of points $\mathcal{P}_0, \dots, \mathcal{P}_{M-1} \in \mathbb{R}^D$
Require: $\mathcal{F} = \text{cl}(\text{int}(C))$, C given as a sequence of vertices

- 1: **for** $m := 0$ to $M-1$ **do**
- 2: $p \leftarrow \mathcal{P}_m$
- 3: **for** $j := 0$ to $J-1$ **do**
- 4: $v \leftarrow v_0 \cdot [\cos \frac{2\pi j}{J}, \sin \frac{2\pi j}{J}]$
- 5: **for** $k := 0$ to $K-1$ **do**
- 6: $w \leftarrow w_k$
- 7: $R \leftarrow R_{p,v,w}$
- 8: $T_{C,R}(p) \leftarrow \text{FINDTARGETREGION}(p, R, v, w)$
- 9: $q_{m,j,k} \leftarrow \text{FINDTARGETPOINT}(p, v, w, T_{C,R}(p))$
- 10: $\rho_{m,j,k} \leftarrow \text{Equation (26)}$
- 11: **return** $(q_{m,j,k}, \rho_{m,j,k})_{m=0,j=0,k=0}^{M-1,J-1,K-1}$
- 12: **function** $\text{FINDTARGETREGION}(p, R, v, w)$
- 13: $A \leftarrow R \setminus \mathcal{F}$
- 14: **for all** $a \in \text{REGIONS } A$ **do**
- 15: **if** $p \in a$ **then**
- 16: $B \leftarrow a$
- 17: **Exit**
- 18: **return** $R \setminus B$
- 19: **function** $\text{FindTargetPoint}(p, v, w, t)$
- 20: $\alpha^* \leftarrow \min\{\alpha > 0 : B_w(p + \alpha v) \subseteq T\}$
- 21: **return** $p + \alpha^* v$
- 22: **function** $\text{Regions } A$
- 23: **return** the list of connected components of A

$$\alpha_{m,j,k} \triangleq \min\{\alpha > 0 : B_{w_k}(q_{m,j,k}) \subset T_{C,R}(\mathcal{P}_m)\}, \quad (25)$$

$$\rho_{m,j,k} \triangleq \rho(\rho^{-1}(w_k) - \alpha_{m,j,k}). \quad (26)$$

Note that $\rho^{-1}(w_k) - \alpha_{m,j,k} \leq 0$ means $\mathcal{P}_m \notin \mathcal{G}$, in which case X_d cannot be tracked with a guarantee of re-entry using this method. Also note that $v_0 \alpha_{m,j,k} = \text{dist}(\mathcal{P}_m, q_{m,j,k})$.

Upon termination, the result is a three dimensional array denoted by $\rho_{m,j,k} \in \mathbb{R}^{M \times J \times K}$, where each element of the array is the MAUR at point $\mathcal{P}_m \in X_d$, associated with a return trajectory $\theta_j \in \Theta$, and a width $w_k \in W$.

Remark 6: In some applications, it may be necessary to further restrict the MAUR to ensure a certain degree of accuracy while tracking X_d . If this is the case, the user may use this predetermined desired upper bound as long as it is less than the MAUR at the point of departure.

A. Algorithm Simplification (Algorithm 2)

The size of the data-set, $M \times J \times K$, can easily become prohibitive for optimization by brute force search. To reduce the search space, some of the iterations for the different $R \in \mathcal{R}(\mathcal{P}_m)$ may be bypassed by selecting a single return trajectory and/or single width. In the numerical experiments in Section VI, a single pair of return parameters, (v, w) is assigned to each \mathcal{P}_m . Specifically, v is set to equal the vector which bisects the smallest sector emanating from \mathcal{P}_m and containing \mathcal{F} ; next, w is selected as the largest $w_k \in W$ such that $R = R_{\mathcal{P}_m, v, w_k}$ satisfies $T_{C,R}(p) \neq \mathcal{F}$. This results in the data-set having size

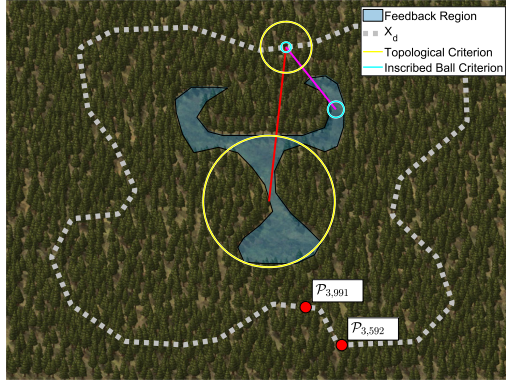


Fig. 5. Comparison of criteria for guaranteed re-entry at a point. The MAUR for the topological criterion is larger than that of IBC. Hence, the agent can afford more uncertainty about its position while guaranteeing re-entry along a different path (red instead of magenta). Note the IBC return trajectory (magenta) is shorter, indicating a possible trade-off between maintaining tracking uncertainty budgets and minimizing deviation from the desired path—a topic for future study.

M. Algorithm 2 assigns the largest possible radius of uncertainty ρ_m to the points q_m , but it may fail to account for the effects of the distance traveled from \mathcal{P}_m to q_m . In (26), maximizing w_k may come at the expense of α_m having to become large, as well. Ultimately, this simplification trades geometric information for computational efficiency.

VI. NUMERICAL EXPERIMENTS

Experiments were conducted in MATLAB to investigate different geometries for the feedback region. In each experiment the MAURs ρ_m generated by the Algorithm 2 (Section V-A) are compared⁵ with the corresponding MAURs—denoted by ρ'_m —generated by IBC.

Experiment, A Generic Example. The task space is a rectangular region in the plane with unitless dimensions $2,048 \times 1,536$. Generic polygonal feedback region \mathcal{F} and desired path X_d were hand-drawn (see Fig. 5). X_d was then subdivided to ensure a mesh size at most 1, resulting in 8,449 way points. The parameters are $\lambda_u = 3$, $\delta = 5$, and $v_0 = 2,000$, to ensure $\rho'_m > 0$ exists for all m .

Fig. 6 plots the percent increase $\mu_m \triangleq \frac{\rho_m - \rho'_m}{\rho'_m} * 100$ as a function of m . The plot indicates that Algorithm 2 provides a significant improvement over IBC for this specific geometry. However, IBC outperforms Algorithm 2 ($\mu_m < 0$) for points $\mathcal{P}_{3,592}$ to $\mathcal{P}_{3,991}$ (see Fig. 6; also see the points marked in Fig. 5). This reduction in the performance of Algorithm 2 results from the MAUR being generated from only one pair of return parameters. Had Algorithm 1 been deployed instead of Algorithm 2, the list of possible return trajectories would have included the one suggested by IBC, guaranteeing $\mu_m \geq 0$ at every point. Despite its sub-optimal performance, for this geometry, Algorithm 2 provides an average improvement of 209% over IBC, with the largest value of μ_m being 656%.

⁵The typical run time for Algorithm 2 in the experiments presented is about an hour. Consequently, running Algorithm 1 on all the experiments would require weeks of computation time, motivating Algorithm 2.

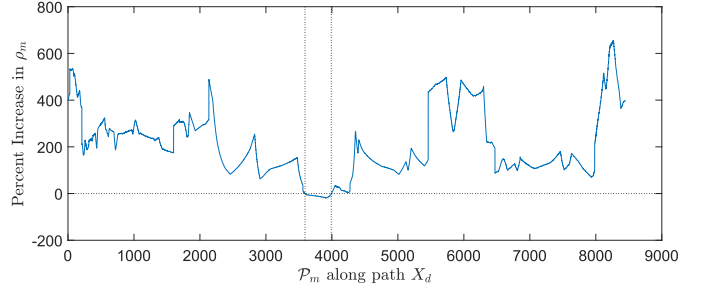


Fig. 6. Percent increase, μ_m , in the MAUR at the point of departure $\mathcal{P}_m \in X_d$ using the proposed criterion for guaranteed re-entry. Note $\mu_m < 0$ between $\mathcal{P}_{3,592}$ and $\mathcal{P}_{3,991}$ (vertical dashed lines).

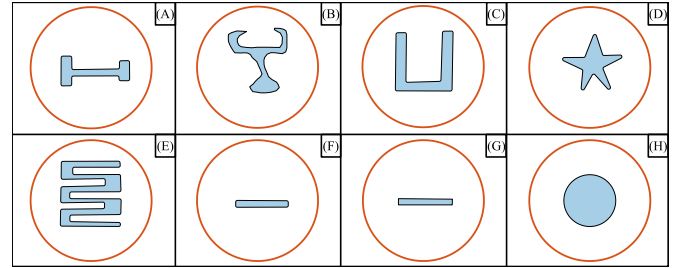


Fig. 7. Geometries: (A) Dumbbell, (B) Generic, (C) Horseshoe, (D) Star, (E) Switchback, (F) Rounded Rectangle, (G) Standard Rectangle, (H) Circle.

Experiment: Sample Geometries. To gain more insight into the interaction between the uncertainty radii and the geometry of the feedback region, a select number of example feedback regions was considered (see Fig. 7). In geometries A–G, the polygons were hand drawn, and the circular region H was approximated by a regular polygon with 100 vertices.

The same parameters were used in these experiments that were used in Section VI. A polygonal approximation of a circle is used as the desired path X_d . To ensure the same mesh requirement, 6,401 vertices was used. The initial point \mathcal{P}_1 along X_d is the rightmost point of the circle, with X_d oriented counter-clockwise.

Baseline Experiment, Concentric Circles. As a baseline for comparing the two methods, we replicate the settings of [5]. C and X_d were taken to be (approximate) concentric circles (see Fig. 7(H)). The two methods render near identical results: the value of μ_m fluctuates at high frequency between -0.21% and 2.74% . The fluctuations are caused by the discrete approximation of the circular boundary of the feedback region C by a regular polygon.

Baseline Experiment, Rectangles. Fig. 8 plots the functions μ_m for the two rectangular geometries (see Fig. 7(F,G)) and confirms the intuitive argument, presented in the introduction (Fig. 1), behind Algorithms 1 and 2.

Also, note how the sharp corners of the standard rectangle cause ρ'_m to approach zero, resulting in the spikes in μ_m visible in the figure. This spiking is due to the fact that, for points \mathcal{P}_m whose nearest point projection to \mathcal{F} results in a return trajectory collinear with an edge of the rectangle, no inscribed ball can be constructed. Fig. 8 demonstrates how rounding the corners mitigates this problem.

Comparing Multiple Geometries. Fig. 9 presents a comparison of the functions μ_m for the geometries in Fig. 7(A–F).

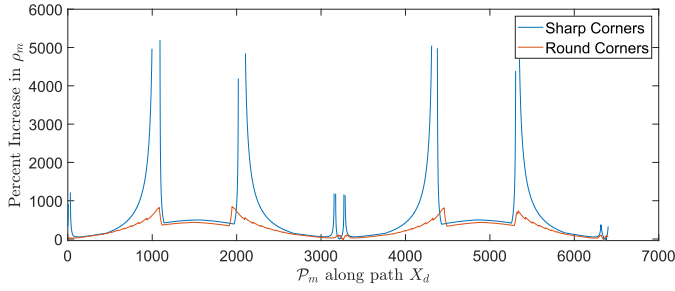


Fig. 8. Percent increase, μ_m , in the MAUR for two types of rectangular feedback regions. One with sharp corners and one with rounded corners.

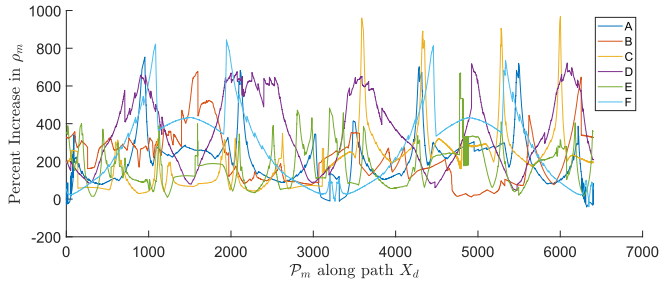


Fig. 9. Percent increase, μ_m , in the MAUR at the point of departure from X_d for a variety of geometries.

The observation made in Section VI, motivated “rounding” the corners of all the feedback region shapes.

Given these experiments (see Fig. 9), with a total of 46,855 data points (including the generic example), the mean-average increase in the maximum allowed radius of uncertainty was found to be 233%. The largest mean-average for a single geometry was 354%. This was observed in the “star” geometry (see Fig. 7(D)). The smallest mean-average for a single geometry was 168%. This was observed in the “switchback” geometry (see Fig. 7(E)). The largest increase at a single point was found to be 969%. This was observed in the “horse-shoe” geometry (see Fig. 7(C)). The smallest was found to be -42%. This was observed in the “dumbbell” geometry (see Fig. 7(A)).

Overall, it was shown that, even across a range of unique geometries, the topological criterion for guaranteed re-entry in Theorem 2 is superior to IBC, in terms of MAUR. The topological criterion was shown to render large improvements even though the sub-optimal Algorithm 2 was used instead of Algorithm 1. It is expected that Algorithm 1, with a much richer data-set, would generate larger values for the optimal MAUR. Moreover, Algorithm 1 is guaranteed to remove any instances where IBC outperforms the topological method. This guarantee is due to the fact the richer data-set would include the suggested return trajectories generated by IBC.

VII. CONCLUSION

Given an autonomous agent in Euclidean space, a topologically motivated method for guaranteeing re-entry of the agent into a feedback region was developed, with the aim of

extending the reach of existing methods to include arbitrary geometries of the feedback region. This method was integrated into an existing framework for developing dwell-times for an autonomous system tasked with following a desired path in the presence of intermittent state feedback. A path-planning algorithm leveraging the new topological re-entry criterion is presented and evaluated against the same planner using IBC for synthetic and generic example geometries. The new topological method was shown to increase the time an agent is able to safely operate in a feedback-denied region, for a variety of geometries. A simplified and sub-optimal implementation of the new method yields improvements in the allowed error growth by hundreds of percentage points. This outcome, as well as computational inefficiencies in the current algorithm, motivate future investigation along several lines of inquiry. Among these, more efficient methods for computing MAURs and optimizing the choice of return trajectories, as well as a method for implementing the “cone of uncertainty”—rather than an infinite strip—as a target region (Section IV-D), are of great interest to the future development of an optimal implementation. Future research will also focus on applying the topological criterion for guaranteed re-entry to the task of optimizing traditional long-term tracking objectives such as total time to task completion, rather than merely maximizing each individual segment.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are of the author(s), and do not necessarily reflect the views of the sponsoring agency.

REFERENCES

- [1] G. Palmieri, M. Palpacelli, M. Battistelli, and M. Callegari, “A comparison between position-based and image-based dynamic visual servoings in the control of a translating parallel manipulator,” *J. Robot.*, vol. 2012, 2012, Art. no. 103954.
- [2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
- [3] N. E. Leonard and A. Olshevsky, “Cooperative learning in multi-agent systems from intermittent measurements,” in *Proc. IEEE Conf. Decis. Control*, Florence, Italy, Dec. 2013, pp. 7492–7497.
- [4] E. Garcia and P. J. Antsaklis, “Model-based event-triggered control for systems with quantization and time-varying network delays,” *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 422–434, Feb. 2013.
- [5] H.-Y. Chen, Z. I. Bell, P. Deptula, and W. E. Dixon, “A switched systems framework for path following with intermittent state feedback,” *IEEE Control Syst. Lett.*, vol. 2, no. 4, pp. 749–754, Oct. 2018.
- [6] F. Zegers, H.-Y. Chen, P. Deptula, and W. E. Dixon, “Distributed Coordination of Multiple Unknown Euler-Lagrange Systems,” in *Proc. IEEE Trans. Cont. Network Syst.*, vol. 5, no. 1, 2018, pp. 55–66.
- [7] D. Liberzon, *Switching in Systems and Control*. Cambridge, MA, USA: Birkhauser, 2003.
- [8] D. Le, H.-Y. Chen, A. R. Teel, and W. E. Dixon, “Path following with stable and unstable modes subject to time-varying dwell-time conditions,” *IFAC World Congr.*, Berlin, Germany, July, 2020.
- [9] J. Gallier and D. Xu, *A Guide to the Classification Theorem for Compact Surfaces*. Berlin, Germany: Springer, ch. Appendix E., pp. 157–163, 2012.
- [10] A. Hatcher, *Algebraic Topology*. Cambridge Univ. Press, Cambridge, 2002, [Online]. Available: <http://www.math.cornell.edu/hatcher/AT/ATpage.html>
- [11] M. Brown, “A proof of the generalized schoenflies theorem,” *Bull. Amer. Math. Soc.*, vol. 66, no. 2, pp. 74–76, 1960.
- [12] R. Burns, B. Dubrovnik, A. Fomenko, and S. Novikov, *Modern Geometry-Methods and Applications: Part II: The Geometry and Topology of Manifolds*. Berlin, Germany: Springer, 2012.

Event-Triggered Formation Control and Leader Tracking With Resilience to Byzantine Adversaries: A Reputation-Based Approach

Federico M. Zegers , Matthew T. Hale , *Member, IEEE*, John M. Shea , *Member, IEEE*, and Warren E. Dixon , *Fellow, IEEE*

Abstract—A distributed event-triggered controller is developed for formation control and leader tracking (FCLT) with robustness to adversarial Byzantine agents for a class of heterogeneous multi-agent systems (MASs). Assuming each agent can accurately measure the state of a neighbor whenever the neighbor broadcasts its state, a reputation-based strategy is developed for each agent to detect Byzantine agent behaviors within their neighbor set and then selectively disregard Byzantine state information. Selectively ignoring Byzantine agents results in a time-varying graph topology. Nonsmooth dynamics also result from intermittent communication due to an event-triggered strategy, which facilitates the efficient use of resources. Nonsmooth Lyapunov methods are used to prove stability and FCLT of the MAS consisting of the remaining cooperative agents.

Index Terms—Decentralized control, fault tolerant control, multi-agent systems, networked control systems.

I. INTRODUCTION

EVENT-TRIGGERED control (ETC) is an intermittent state feedback strategy motivated by advantages such as the efficient use of resources, like communication energy and bandwidth, through trigger-based sensing, actuation, and/or communication (see [1]–[5]) and the ability to design the trigger condition to ensure stability properties [6]. ETC has been applied for the coordination of multi-agent systems (MASs), especially in mobile network applications, given the limited energy in

portable power supplies and constrained networks available in operational scenarios (see [2], [3], [7]). Results such as [1], [3], [5], [8], and [9], among many others, are developed under the implicit assumption that the MASs operate in a benign environment that is free from adversaries (i.e., all agents are cooperative in the sense that they communicate true state information and follow the network objective).

With respect to ETC, if the update of state variables is delayed and/or updated with incorrect information, the resulting system performance may degrade, especially if state updates are infrequent. Because the communication timing conditions of ETC methods create potential vulnerabilities, resilient strategies are motivated for assured coordination in contested environments. Common threats in contested environments include: Denial-of-service (DoS) attacks, time-delay switch (TDS) attacks, and Byzantine attacks. A DoS attack occurs when an adversary interrupts communication within a network [10]; a TDS attack occurs when an adversary imparts time delays on communication within a network [11]; and a Byzantine attack is a more general threat where communication can be delayed, corrupted, and/or interrupted arbitrarily [12].

In this article, we focus on Byzantine threats since they are a generalization of DoS and TDS attacks. As in [13], we consider two types of Byzantine behavior. A Type I Byzantine agent remains in the mobile network, where it can halt, delay, or corrupt information communicated to its neighbors temporarily or indefinitely. A Type II Byzantine agent abandons the mobile network while communicating true or no state information about itself temporarily or indefinitely. These designations are not fixed for all time, and any adversary can be categorized as either type at any time.

Results, such as [14]–[16], attain consensus in the presence of Type I Byzantine adversaries. However, they are not able to identify Byzantine threats nor can they alter the communication network to stop data sharing between the cooperative and Byzantine agents as in [13]. In [13], Byzantine adversaries are identified through a Lyapunov-based detector that compares communicated state information to worst-case state estimates that are based on accurate past neighbor state information. While such a strategy enables Byzantine agent detection, it is limited in the sense that the detector requires an upper bound on the control of each neighboring agent, which may be unknown *a*

Manuscript received September 7, 2020; revised September 15, 2020 and January 22, 2021; accepted February 21, 2021. Date of publication March 23, 2021; date of current version September 17, 2021. This work was supported in part by a Task Order contract with the Air Force Research Laboratory, Munitions Directorate at Eglin AFB, Office of Naval Research under Grant N00014-13-1-0151, in part by NEEC Grant N00174-18-1-0003, and in part by AFOSR Grant FA9550-18-1-0109 and Grant FA9550-19-1-0169. Recommended by Associate Editor I. Lestas. (Corresponding author: Federico Zegers.)

Federico M. Zegers and Matthew T. Hale are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: fredzeg@ufl.edu; matthewhale@ufl.edu).

John M. Shea is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: jshea@ece.ufl.edu).

Warren E. Dixon is with the Department of Mechanical and Aerospace Engineering and also with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6250 USA (e-mail: wdixon@ufl.edu).

Digital Object Identifier 10.1109/TCNS.2021.3068348

priori. Moreover, once an agent is categorized as a Byzantine adversary, it cannot be reincorporated back into the cooperative neighbor set even if it becomes cooperative. Such a scenario can occur when the communication links between a group of cooperative agents are temporarily jammed by an adversary. These limitations can potentially be circumvented through the use of a reputation algorithm [17], which does not require exact model knowledge of each neighbor's dynamics, does not require bounds on neighbor quantities such as control, and enables the reintegration of rehabilitated agents, i.e., agents that convert from Byzantine to cooperative.

In this work, we expand our precursory result in [13] and investigate formation control and leader tracking (FCLT), also referred to as leader–follower formation control in [18], in the presence of Byzantine adversaries. Specifically, FCLT refers to a set of follower agents tracking the trajectory of a leader agent while the follower agents preserve a predefined formation. The Byzantine adversaries are assumed to operate independently, i.e., Byzantine agents do not work together against the cooperative followers. Since only the cooperative followers must collaborate to perform secure FCLT, a traditional unsigned graph is used to model the network of cooperative followers. Moreover, an ETC method is developed that facilitates assured FCLT while promoting the efficient use of resources and providing resilience to Byzantine adversaries. While the coordination strategy in [13] relies on exact model knowledge, this result enables coordination for agents modeled with uncertain nonlinear dynamics subject to an exogenous disturbance.

Inspired by the reputation-based method for adjusting the network edge weights in [17], we develop a reputation-based Byzantine detection strategy that enables coordination between cooperative agents only, while enabling the capability of reintegrating rehabilitated agents. The reputation strategy also enables malfunctioning agents to be isolated from the cooperative MAS to ensure safety and enable the remaining agents to achieve the objective. A malfunctioning agent can be reintegrated into the network once functional operational control can be established. By coordinating only with cooperative neighbor information, the influence from each Byzantine adversary is cut out from the network consisting of only the cooperative followers. The Byzantine detection strategy is based on two-point authentication, e.g., comparing communicated and sensed state information, where the redundancy in state information enables Byzantine agent detection. Stability of the ETC strategy is examined through nonsmooth Lyapunov analysis.

In our preliminary result in [19], we established the concept of using a zero-order hold to enable formation control and leader tracking with intermittent communication between the followers only and the use of trust and reputation models to impart resilience to Byzantine adversaries. In this article, a more complete development of the ideas in [19] is provided by: 1) A revised, more detailed, and more rigorous narrative and presentation; 2) a generalization of the mathematics to relax the previous requirement of continuous communication with the leader to allow only intermittent feedback; 3) a proof of the existence of a positive uniform lower bound for the difference between consecutive broadcast events for each agent,

including the leader; and 4) simulation results that demonstrate the performance of the developed methods.

II. PRELIMINARIES

A. Notation

Let \mathbb{R} and \mathbb{Z} denote the sets of real numbers and integers, respectively, where $\mathbb{R}_{\geq 0} \triangleq [0, \infty)$, $\mathbb{R}_{> 0} \triangleq (0, \infty)$, $\mathbb{Z}_{\geq 0} \triangleq \mathbb{R}_{\geq 0} \cap \mathbb{Z}$, and $\mathbb{Z}_{> 0} \triangleq \mathbb{R}_{> 0} \cap \mathbb{Z}$. Let $p, q, n \in \mathbb{Z}_{> 0}$. The $p \times q$ zero matrix and the $p \times 1$ zero column vector are denoted by $0_{p \times q}$ and 0_p , respectively. The $p \times p$ identity matrix and the $p \times 1$ ones column vector are denoted by I_p and 1_p , respectively. The Euclidean norm of a vector $r \in \mathbb{R}^p$ is denoted by $\|r\| \triangleq \sqrt{r^\top r}$. The Frobenius norm of $A \in \mathbb{R}^{p \times q}$ is denoted by $\|A\|_F \triangleq \sqrt{1_p^\top (A \odot A) 1_q}$, where \odot denotes the Hadamard product. The block diagonal matrix, whose diagonal blocks consist of $G_1, G_2, \dots, G_n \in \mathbb{R}^{p \times q}$, is denoted by $\text{diag}(G_1, G_2, \dots, G_n) \in \mathbb{R}^{np \times nq}$. Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$, $\lambda_{\min}(A)$, $\lambda_{\max}(A)$, and $\lambda_i(A)$ denote the minimum, maximum, and i th eigenvalue of A , respectively. The class C^1 refers to the set of continuously differentiable functions. Let f be an essentially bounded measurable function. Then, $f \in \mathcal{L}_\infty$ if and only if $\inf\{C \geq 0 : |f(x)| \leq C \text{ for almost every } x\} \in \mathbb{R}_{\geq 0}$. The Kronecker product of $A \in \mathbb{R}^{p \times q}$ and $B \in \mathbb{R}^{u \times v}$ is denoted by $(A \otimes B) \in \mathbb{R}^{pu \times qv}$. The complement and power set of the set S are denoted by S^C and 2^S , respectively.

B. Algebraic Graph Properties

Let $\mathcal{G}(t) \triangleq (\mathcal{V}, \mathcal{E}(t), \mathcal{A}(t))$ be a time-varying, weighted, and undirected graph with node set $\mathcal{V} \triangleq \{1, 2, \dots, N\}$, for $N \in \mathbb{Z}_{> 0}$, edge mapping $\mathcal{E} : [0, \infty) \rightarrow 2^{\mathcal{V} \times \mathcal{V}}$, and weighted adjacency mapping $\mathcal{A} : [0, \infty) \rightarrow \mathbb{R}^{N \times N}$, where $\mathcal{A}(t) \triangleq [a_{ij}(t)]$, such that $a_{ij} : [0, \infty) \rightarrow [0, 1]$. Within the context of this work, no self-loops are considered and, therefore, $a_{ii}(t) \triangleq 0$ for all $i \in \mathcal{V}$ and for all $t \geq 0$. An undirected edge is defined as an ordered pair (j, i) , where $(j, i) \in \mathcal{E}(t)$ if and only if $(i, j) \in \mathcal{E}(t)$. Note that $(j, i) \in \mathcal{E}(t)$ implies agent i can obtain information from agent j and vice versa. An undirected path is a sequence of undirected edges in $\mathcal{E}(t)$. An undirected graph is called connected if and only if there exists an undirected path between any two distinct nodes. The time-varying neighbor set of node i is defined by $\mathcal{N}_i : [0, \infty) \rightarrow 2^{\mathcal{V}}$, where $j \in \mathcal{N}_i(t)$ if and only if $(j, i) \in \mathcal{E}(t)$. The weighted degree matrix of the undirected graph $\mathcal{G}(t)$ is defined by $\Delta : [0, \infty) \rightarrow \mathbb{R}^{N \times N}$, such that $\Delta(t) \triangleq [\Delta_{ij}(t)]$, where $\Delta_{ij}(t) \triangleq 0$ for all $i \neq j$ and $\Delta_{ii}(t) \triangleq \sum_{j \in \mathcal{V}} a_{ij}(t)$.

The weighted graph Laplacian $L : [0, \infty) \rightarrow \mathbb{R}^{N \times N}$ of the undirected graph $\mathcal{G}(t)$ is defined by $L(t) \triangleq \Delta(t) - \mathcal{A}(t)$. Let node 0, where $0 \notin \mathcal{V}$, be independent of the graph structure and $b_{\max} \in \mathbb{R}_{> 0}$. Let the diagonal (pinning) matrix encoding the edge weights between node 0 and node $i \in \mathcal{V}$ be defined by $B : [0, \infty) \rightarrow \mathbb{R}^{N \times N}$, where $B(t) \triangleq [b_{ij}(t)]$, such that $b_i \triangleq b_{ii} : [0, \infty) \rightarrow [0, b_{\max}]$ for all $i \in \mathcal{V}$ and $b_{ij}(t) \triangleq 0$ for all $i \neq j$. If $b_i(t) > 0$, then node i can receive information from node 0. The time-varying weighted connectivity matrix $H : [0, \infty) \rightarrow$

$\mathbb{R}^{N \times N}$, encoding the flow of information between all nodes in $\mathcal{V} \cup \{0\}$, is defined by $H(t) \triangleq L(t) + B(t)$.

III. AGENT DYNAMICS AND NETWORK TOPOLOGY

Consider a heterogeneous MAS consisting of a single leader agent indexed by 0 and a set of $N \in \mathbb{Z}_{>0}$ follower agents indexed by \mathcal{V} . The uncertain nonlinear model for agent $i \in \mathcal{V} \cup \{0\}$ is

$$\dot{x}_i(t) \triangleq f_i(x_i(t)) + g_i(x_i(t))u_i(t) + d_i(t) \quad (1)$$

where $x_i : [0, \infty) \rightarrow \mathbb{R}^n$ denotes the position, $\dot{x}_i : [0, \infty) \rightarrow \mathbb{R}^n$ denotes the velocity, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes the uncertain drift dynamics, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ denotes the known control effectiveness matrix, $u_i : [0, \infty) \rightarrow \mathbb{R}^m$ denotes the control input, and $d_i : [0, \infty) \rightarrow \mathbb{R}^n$ denotes an exogenous disturbance for agent i . Let $\mathcal{B} : [0, \infty) \rightarrow 2^{\mathcal{V}}$ define the time-varying set of Byzantine agents and $\mathcal{C} : [0, \infty) \rightarrow 2^{\mathcal{V}}$ define the time-varying set of cooperative agents, where $\mathcal{B}(t) \cap \mathcal{C}(t) = \emptyset$ and $\mathcal{B}(t) \cup \mathcal{C}(t) = \mathcal{V}$ for all $t \geq 0$. The following assumptions are made to facilitate the subsequent analysis.

Assumption 1: For each $i \in \mathcal{V} \cup \{0\}$, the uncertain drift dynamics f_i is class C^1 and bounded given a bounded argument, i.e., if $\|x(t)\| \leq \bar{c}_1$ for some $\bar{c}_1 \in \mathbb{R}_{>0}$, then $\|f_i(x(t))\| \leq \bar{c}_2$ for some $\bar{c}_2 \in \mathbb{R}_{>0}$.

Assumption 2: The control effectiveness matrix g_i is C^1 , bounded given a bounded argument, and full-row rank for all $i \in \mathcal{V} \cup \{0\}$. Moreover, the right pseudoinverse of g_i is denoted by $g_i^+ : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$, where $g_i^+(\cdot) \triangleq g_i^\top(\cdot)(g_i(\cdot)g_i^\top(\cdot))^{-1}$ is bounded given a bounded argument for each $i \in \mathcal{V} \cup \{0\}$.¹

Assumption 3: The exogenous disturbance d_i is continuous and bounded in the sense that $\|d_i(t)\| \leq d_{i,\max}$ for all $t \geq 0$ and $i \in \mathcal{V} \cup \{0\}$, where $d_{i,\max} \in \mathbb{R}_{>0}$ is a known bounding constant.

Assumption 4: The leader is cooperative for all $t \geq 0$.²

Assumption 5: Agent i is capable of measuring its own position $x_i(t)$ for all $t \geq 0$ and all $i \in \mathcal{V} \cup \{0\}$.

Assumption 6: [17] The control and position of the leader are bounded, i.e., there exist $u_{0,\max}, x_{0,\max} \in \mathbb{R}_{>0}$, such that $\|u_0(t)\| \leq u_{0,\max}$ and $\|x_0(t)\| \leq x_{0,\max}$ for all $t \geq 0$.

Assumption 7: For each instant $t \geq 0$ that follower $j \in \mathcal{N}_i(t)$ broadcasts its state to follower i , follower i can accurately measure the state of follower j .

Let $x_{ij} : [0, \infty) \rightarrow \mathbb{R}_{\geq 0}$ be defined as $x_{ij}(t) \triangleq \|x_i(t) - x_j(t)\|$. Agent i can broadcast information to agent j if and only if $x_{ij}(t) \leq R_{C,i}$, where $R_{C,i} \in \mathbb{R}_{>0}$ denotes the communication radius of agent i . Similarly, agent i can sense agent j if and only if $x_{ij}(t) \leq R_{S,i}$, where $R_{S,i} \in \mathbb{R}_{>0}$ denotes the sensing radius of agent i . Without loss of generality, let $R \triangleq \min_{i \in \mathcal{V}} \{R_{C,i}, R_{S,i}\} \in \mathbb{R}_{>0}$, where R is defined as the interaction radius of all agents in the MAS. The neighbor set of follower

i is given by $\mathcal{N}_i(t) \triangleq \{j \in \mathcal{V} : x_{ij}(t) \leq R\}$, where followers i and $j \in \mathcal{N}_i(t)$ can both broadcast information to and sense each other. Followers i and j are said to be paired if and only if $i \in \mathcal{N}_j(t)$ and $j \in \mathcal{N}_i(t)$. Similarly, followers i and j are said to be connected if and only if $a_{ij}(t) \neq 0$ and $a_{ji}(t) \neq 0$.

Observe that follower i can be influenced by follower j if and only if $a_{ij}(t) \neq 0$. The influence relationships between the followers of the MAS are modeled by a time-varying, weighted, and undirected graph $\mathcal{G}(t) \triangleq (\mathcal{V}, \mathcal{E}(t), \mathcal{A}(t))$. Let $\mathcal{E}_C(t)$ denote the undirected edge set and $\mathcal{A}_C(t)$ denote the weighted adjacency matrix associated with all cooperative followers in $\mathcal{C}(t)$. Moreover, the sub-MAS consisting of only the cooperative followers is modeled by the time-varying, weighted, and undirected graph $\mathcal{G}_C(t) \triangleq (\mathcal{C}(t), \mathcal{E}_C(t), \mathcal{A}_C(t))$ and is referred to as the cooperative MAS (CMAS).

Assumption 8: The graph $\mathcal{G}_C(t)$ is connected for all $t \geq 0$, and $b_i(t) > 0$ for some $i \in \mathcal{C}(t)$ for all $t \geq 0$.³

Remark 1: Assumptions 4 and 8 ensure that each cooperative agent has at least one cooperative neighbor for all $t \geq 0$, even in the presence of a DoS attack. Moreover, Assumptions 4 and 8 ensure that the Byzantine agents cannot enter the MAS in a manner that partitions $\mathcal{G}_C(t)$ for any $t \geq 0$.

IV. OBJECTIVES

The goal is to design distributed controllers for all followers in the MAS that maneuver the followers to a desired formation while tracking the leader. However, as FCLT is taking place, some followers may transform into Byzantine agents as a result of operating within a contested environment, e.g., if they suffer cyber-attacks. The objective is to design distributed controllers for all followers $i \in \mathcal{V}$ governed by (1) that enable the cooperative followers to achieve FCLT while identifying Byzantine agents and removing all Byzantine influence from the CMAS. The distributed controllers are event-triggered to promote the efficient use of communication and sensing resources. Cooperative and Byzantine agents are managed through the edge weight policy, which is based on a reputation algorithm. The policy enables all agents to differentiate between cooperative and Byzantine neighbors, coordinate their motion by using only information from cooperative neighbors, and reintegrate agents into the CMAS once an agent converts from Byzantine to cooperative. The separation between communication and influence is made to enable the reintegration of remediated followers, which requires communication between cooperative-Byzantine pairs. The ability to reintegrate cooperative agents is key for defense against adversarial behaviors such as mobile jammers that can temporarily affect agents before moving on to jam other agents. To quantify the objective, let the FCLT error $e_{1,i} : [0, \infty) \rightarrow \mathbb{R}^n$ be defined as

$$e_{1,i}(t) \triangleq x_i(t) - x_0(t) - v_i \quad (2)$$

where $v_i \in \mathbb{R}^n$ denotes the desired relative position between follower i and the leader.

¹The assumption of a full-row rank control effectiveness matrix is potentially restrictive for some applications (e.g., underactuated systems) and is a topic for future investigation. For LTI systems with a full-column rank control effectiveness matrix, the algebraic Riccati equation or linear matrix inequalities can potentially be used to develop stabilizing controllers.

²In the absence of a manned leader, multiple leaders can be added to the MAS through the pinning matrix strategy to impart additional resilience to Byzantine adversaries. Assumption 4 can then be reduced to requiring that at least one leader is cooperative for all time.

³An alternative to Assumption 8 is to assume $\mathcal{G}_C(t)$ is connected for all time, upper bound the number of Byzantine adversaries within a network by $f \in \mathbb{Z}_{>0}$, use more than f leaders, and employ connectivity models like the $2f + 1$ model described in [14].

Assumption 9: The relative position vector v_i is fixed for all $i \in \mathcal{V}$. Moreover, each follower knows v_i for all $i \in \mathcal{V}$, i.e., each follower knows the entire formation.

By allowing each follower to know v_i for all $i \in \mathcal{V}$, any rehabilitated agent can be reintegrated into any available formation vacancy, if there are multiple options. The FCLT problem can be converted into a leader–follower consensus problem provided $v_i \triangleq 0_n$ for all $i \in \mathcal{V}$. The use of ETC also motivates the development of an estimator to provide continuous state estimates between communication events. The state estimation error of follower i is defined by $e_{2,i} : [0, \infty) \rightarrow \mathbb{R}^n$, where

$$e_{2,i}(t) \triangleq \hat{x}_i(t) - x_i(t) \quad (3)$$

such that $\hat{x}_i : [0, \infty) \rightarrow \mathbb{R}^n$ denotes the state estimate of x_i .

V. CONTROLLER DEVELOPMENT

A. Trust Model

As Byzantine agents emerge in the MAS, the remaining cooperative followers require a method to identify their cooperative neighbors. Let $\tau_{ij} : [0, \infty) \rightarrow [0, 1]$ denote the piecewise constant trust that follower i has in follower $j \in \mathcal{N}_i(t)$, where 0 and 1 represent no trust and maximum trust, respectively. Each follower i can obtain state information from any neighbor j through communication and sensing, where the redundancy in state information is used to compute τ_{ij} . Let $x_{i,1} : [0, \infty) \rightarrow \mathbb{R}^n$ and $x_{i,2} : [0, \infty) \rightarrow \mathbb{R}^n$ denote the communicated and sensed, i.e., measured, state of follower i , respectively. Subscripts 1 and 2 denote the type of data, i.e., 1 refers to communicated data and 2 refers to sensed data, where both types of data describe the same quantity.

Let $\{t_k^j\}_{k=0}^\infty \subset \mathbb{R}_{\geq 0}$ be an increasing sequence of event-times determined by the event-trigger mechanism of follower j , where the event-time t_k^j denotes the k th instance of follower j broadcasting its state information to its neighbors, all of which are received simultaneously. Let $t_{\text{reset}} \in \mathbb{R}_{>0}$ be a user-defined parameter that denotes the length of time over which trust is determined. The trust follower i has in neighbor j is determined by (see the motivating result in [17])

$$\tau_{ij}(t) \triangleq \begin{cases} 1, & |S_j| = 0 \\ \frac{1}{|S_j|} \sum_{t_k^j \in S_j} e^{-s_1 \Psi_{ij}(t_k^j)}, & |S_j| \neq 0 \end{cases} \quad (4)$$

$$\Psi_{ij}(t_k^j) \triangleq \|x_{j,1}(t_k^j) - x_{j,2}(t_k^j)\|$$

where $S_j \triangleq \{t_k^j \in \mathbb{R}_{\geq 0} : t - t_{\text{reset}} \leq t_k^j < t\}$, $x_{j,1}(t_k^j)$ and $x_{j,2}(t_k^j)$ denote the communicated and sensed version of the state of follower j at event-time t_k^j , respectively, and $s_1 \in \mathbb{R}_{>0}$ is a user-defined parameter that determines how fast trust decreases. Note that $\Psi_{ij}(t_k^j)$ measures the discrepancy in the state information follower i has about follower $j \in \mathcal{N}_i(t)$ at time t_k^j . Other trust models can be used instead of (4) provided agreement and disagreement between the communicated and sensed version of the state of follower j results in high and low trust, respectively. In (4), all agents begin with maximum trust. However, as discrepancies in the two-point authentication of follower j grow,

the trust value of follower j decreases to zero. Conversely, the trust of follower j may increase given the discrepancies in the two-point authentication of follower j are negligible for each $t_k^j \in S_j$, i.e., if $\Psi_{ij}(t_k^j) \approx 0$ for each $t_k^j \in S_j$, then $\tau_{ij}(t) \approx 1$. In the event that follower $j \in \mathcal{N}_i(t)$ does not provide state information to follower i when required, i.e., $\Delta t_k^j \triangleq t_k^j - t_{k-1}^j > \Delta_j$, then $\Psi_{ij}(t_k^j) = \vartheta$, where $\Delta_j \in \mathbb{R}_{>0}$ is a user-defined parameter based on either a simulation/experimental study or an analysis-based derivation, and $\vartheta \in \mathbb{R}_{>0}$ is a user-defined penalty. Similarly, if the distance between follower $j \in \mathcal{N}_i(t)$ and follower i is beyond a user-defined threshold a time t_k^j , i.e., $r < \omega_{ij}(t_k^j) \triangleq \|x_i(t_k^j) - x_{j,2}(t_k^j)\| \leq R$, for $r \in (0, R)$, then $\Psi_{ij}(t_k^j) = \vartheta$.

Remark 2: Assumption 7 affords each agent access to ground-truth state information for each of its neighbors, where comparisons between the communicated and sensed states enable Type I Byzantine agent detection. Moreover, Type II Byzantine agents can abandon the MAS while potentially communicating true state information that could pull the remaining agents with them in their attempt to maintain connectivity. Such a scenario may perturb and destabilize the MAS. Therefore, agent i requires access to accurate state information for each $j \in \mathcal{N}_i(t)$ that is r -close to $x_i(t)$ to ensure Type I and Type II Byzantine agent detection for each $i \in \mathcal{V}$.

Remark 3: By Assumption 7, follower i is able to measure the state of follower $j \in \mathcal{N}_i(t)$ each time follower j broadcasts its state. Therefore, Assumption 7 implies that the broadcast state of follower j is synchronized with the measured state of follower j . In practice, achieving synchronization between the broadcast and sensed states of follower j may be unattainable, where the sensed state may be obtained $\delta t > 0$ time units after the broadcast state is received. However, the s_1 parameter in (4) can be tuned to account for the asynchronous state information provided $\delta t > 0$ is small enough, such that $\|x_{j,1}(t_k^j) - x_{j,2}(t_k^j + \delta t)\| \leq \epsilon(\delta t)$ for small $\epsilon(\delta t) > 0$. Future works aim at developing trust models that enable Byzantine agent detection through the use of asynchronous state information, where [20] and [21] serve as potential inroads.

B. Reputation Model

Because a Byzantine agent can provide different state information to each of its neighbors, each neighbor may have a different trust value for the same Byzantine agent. However, multiple trust values for a common neighbor can be consolidated into an overall reputation for the common neighbor. Let $\mathcal{N}_{ij}(t) \triangleq \mathcal{N}_i(t) \cap \mathcal{N}_j(t)$ denote the set of common neighbors shared between followers i and j . Motivated by [17], the continuous reputation $\zeta_{ij} : [0, \infty) \rightarrow \mathbb{R}_{\geq 0}$ follower i has for follower $j \in \mathcal{N}_i(t)$ is

$$\dot{\zeta}_{ij}(t) \triangleq \text{proj} \left(\eta_\tau (\tau_{ij}(t) - \zeta_{ij}(t)) + \sum_{n \in \mathcal{N}_{ij}(t)} \eta_\zeta \zeta_{in}(t) (\zeta_{nj}(t_k^n) - \zeta_{ij}(t)) \right) \quad (5)$$

where $\zeta_{ij}(0) = 1$ and $\text{proj}(\cdot)$ denotes the continuous projection operator defined in [22] that is used to ensure $\zeta_{ij}(t) \in [0, 1]$ for all $t \geq 0$. In (5), the parameters $\eta_\tau \in \mathbb{R}_{>0}$ and $\eta_\zeta \in \mathbb{R}_{>0}$ allow the user to select whether the reputation model places more emphasis on measured information, i.e., $\tau_{ij}(t) - \zeta_{ij}(t)$; observed information, i.e., $\zeta_{in}(t)(\zeta_{nj}(t_k^n) - \zeta_{ij}(t))$; or weighs both measured and observed information equally.

Like the trust model in (4), reputation values of 0 and 1 correspond to no and maximum reliability, respectively. At event-time t_k^n , follower i receives reputation values held by follower $n \in \mathcal{N}_i(t)$ for all followers $j \in \mathcal{N}_n(t)$, i.e., $\zeta_{nj}(t_k^n)$, where follower i computes $\zeta_{in}(t)(\zeta_{nj}(t_k^n) - \zeta_{ij}(t))$ over $n \in \mathcal{N}_{ij}(t)$. The measured information in (5) contributes toward the reputation held by follower i for follower j based on the trust measurements follower i has of follower j . The observed information in (5) contributes to the reputation held by follower i for follower j based on the reputation of follower j held by common neighbors $n \in \mathcal{N}_{ij}(t)$, which is weighted based on the corresponding reputation of follower $n \in \mathcal{N}_{ij}(t)$. Hence, a common neighbor $n_1 \in \mathcal{N}_{ij}(t)$ with a low reputation has less influence on the reputation of neighbor j than common neighbor $n_2 \in \mathcal{N}_{ij}(t) \setminus \{n_1\}$ with a higher reputation.

C. Edge Weight Policy

The edge weights of $\mathcal{G}(t)$ encode the degree of influence each neighbor $j \in \mathcal{N}_i(t)$ has on follower i . Since the objective is to achieve FCLT by the cooperative followers, and the reputation model captures the degree of reliability of each follower, the edge weights can be continuously updated according to the reputation model. The edge weight $a_{ij}(t)$ is defined by

$$a_{ij}(t) \triangleq \begin{cases} \zeta_{ij}(t), & \zeta_{ij}(t) \geq \zeta_{\min} \text{ and } j \in \mathcal{N}_i(t) \\ 0, & \zeta_{ij}(t) < \zeta_{\min} \text{ or } j \notin \mathcal{N}_i(t) \end{cases}, \quad (6)$$

where $\zeta_{\min} \in [0, 1]$ is a user-defined parameter that determines whether follower i categorizes follower $j \in \mathcal{N}_i(t)$ as cooperative or Byzantine.

The set of cooperative and Byzantine neighbors of follower i at time t are given by $\mathcal{C}_i(t) \triangleq \{j \in \mathcal{N}_i(t) : a_{ij}(t) \neq 0\}$ and $\mathcal{B}_i(t) \triangleq \mathcal{N}_i(t) \setminus \mathcal{C}_i(t)$, respectively. Remark 4 explains the time-varying nature of the cooperative and Byzantine neighbor sets. Furthermore, $\mathcal{B}(t) \triangleq \{j \in \mathcal{V} : j \in \mathcal{B}_i(t) \text{ for some } i \in \mathcal{V}\}$ and $\mathcal{C}(t) \triangleq \mathcal{V} \setminus \mathcal{B}(t)$. From (6), the edge weight $a_{ij}(t)$ is positive if follower j is a cooperative neighbor of follower i . Conversely, edge weight $a_{ij}(t)$ is zero if followers i and j are not neighbors or if follower j is a Byzantine neighbor of follower i . Note that if $j \in \mathcal{B}_i(t)$, then follower j cannot influence follower i . However, follower i can still compute trust and reputation for follower $j \in \mathcal{N}_i(t)$, where follower j can be reintegrated into $\mathcal{C}_i(t)$ once $\zeta_{ij}(t) \geq \zeta_{\min}$ provided $j \in \mathcal{N}_i(t)$. This enables a remediated Byzantine agent to enter the CMAS and become cooperative neighbors with any cooperative agent. Hence, the information exchange and influence between agents are decoupled. Furthermore, if follower $j \in \mathcal{B}_i(t)$, then follower i will not communicate any true state information about itself to follower j until follower j becomes a cooperative neighbor of follower i . Note that the leader will only communicate state information to

its cooperative neighbors by also using (6). Hence, cooperative state information is only communicated between cooperative agents.

Remark 4: The cooperative and Byzantine neighbor sets are time-varying because cooperative agents may be attacked within contested environments and converted into Byzantine agents. Moreover, it may be possible for operators to employ countermeasures to convert Byzantine agents back into cooperative agents. Hence, a follower may be initiated as cooperative, eventually become Byzantine, and then eventually become cooperative again. While the Byzantine neighbor set of a follower may be empty at some time, Assumption 8 ensures the cooperative neighbor set of each cooperative follower is never empty. This also implies that each cooperative agent cannot have all neighbors be Byzantine if there are at least two cooperative followers in the network. Relaxing Assumption 8 is the subject of future work.

D. Event-Triggered Control Development

The state estimate of agent $i \in \mathcal{V} \cup \{0\}$, which is synchronized among all agents $j \in \mathcal{N}_i(t) \cup \{i\}$, is generated by the zero-order hold

$$\hat{x}_i(t) \triangleq x_{i,1}(t_k^i), t \in [t_k^i, t_{k+1}^i). \quad (7)$$

According to (7), agent i samples its position at time t_k^i and broadcasts it to all agents $j \in \mathcal{N}_i(t)$. Each agent $j \in \mathcal{N}_i(t) \cup \{i\}$ equates the state estimate of agent i , i.e., $\hat{x}_i(t)$, to $x_i(t_k^i)$ for all time until the next broadcast event of agent i . Recall that $x_{i,1}(t)$ denotes the broadcast state of agent i at time t , which cooperative agents communicate accurately, i.e., $x_{i,1}(t_k^i) = x_i(t_k^i)$. Based on the subsequent stability analysis, the controller for follower $i \in \mathcal{V}$ is

$$u_i(t) \triangleq g_i^+(x_i(t))(k_1 z_i(t) + k_2 e_{2,i}(t)) \quad (8)$$

$$z_i(t) \triangleq \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)(\hat{x}_j(t) - \hat{x}_i(t) - v_j + v_i) + b_i(t)(v_i + \hat{x}_0(t) - \hat{x}_i(t)), \quad (9)$$

where

$$k_1 \triangleq \frac{1}{\Lambda_{\min}} \left(k_{1,1} + \frac{\rho_1^2}{\delta_1} \right) \in \mathbb{R}_{>0},$$

$k_2 \triangleq k_{2,1} + \frac{\rho_2^2}{\delta_2} \in \mathbb{R}_{>0}$, $k_{1,1} \triangleq k_{1,2} + k_{1,3} \in \mathbb{R}_{>0}$, and $k_{1,2}, k_{1,3}, k_{2,1}, \rho_1, \rho_2, \delta_1, \delta_2 \in \mathbb{R}_{>0}$ are parameters defined in Theorem 1. Note that $\Lambda_{\min} \in \mathbb{R}_{>0}$ is a parameter defined in Lemma 2, and $z_i : [0, \infty) \rightarrow \mathbb{R}^n$ is the estimate-based distributed FCLT control effort. The stacked form of (9) is defined by $Z \triangleq [z_1^\top(t), z_2^\top(t), \dots, z_N^\top(t)]^\top \in \mathbb{R}^{nN}$.

The stacked error systems for the leader-follower relative position error in (2) and state estimation error in (3) are $E_1 \triangleq [e_{1,1}^\top(t), e_{1,2}^\top(t), \dots, e_{1,N}^\top(t)]^\top \in \mathbb{R}^{nN}$ and $E_2 \triangleq [e_{2,1}^\top(t), e_{2,2}^\top(t), \dots, e_{2,N}^\top(t)]^\top \in \mathbb{R}^{nN}$, respectively. Substituting (1)–(3), (8), and (9) into the time-derivative of (2) yields

$$\dot{e}_{1,i}(t) = f_i(x_i(t)) + k_1 \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)(e_{2,j}(t) - e_{2,i}(t))$$

$$\begin{aligned}
& + k_1 \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (e_{1,j}(t) - e_{1,i}(t)) + k_1 b_i(t) e_{2,0}(t) \\
& - k_1 b_i(t) e_{1,i}(t) - k_1 b_i(t) e_{2,i}(t) + k_2 e_{2,i}(t) + d_i(t) \\
& - \dot{x}_0(t).
\end{aligned} \tag{10}$$

Substituting (1)–(3), and (7)–(9) into the time-derivative of (3) yields

$$\begin{aligned}
\dot{e}_{2,i}(t) & = -f_i(x_i(t)) - k_1 \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (e_{2,j}(t) - e_{2,i}(t)) \\
& - k_1 \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t) (e_{1,j}(t) - e_{1,i}(t)) - k_1 b_i(t) e_{2,0}(t) \\
& + k_1 b_i(t) e_{1,i}(t) + k_1 b_i(t) e_{2,i}(t) - k_2 e_{2,i}(t) - d_i(t).
\end{aligned} \tag{11}$$

Substituting (10) and (11) into the time-derivative of E_1 and E_2 , respectively, and compactly expressing the results with the Kronecker product yields

$$\begin{aligned}
\dot{E}_1 & = \tilde{N} + N_d - k_1 (H(t) \otimes I_n) E_2 - k_1 (H(t) \otimes I_n) E_1 \\
& + k_1 (B(t) 1_N \otimes e_{2,0}(t)) + k_2 E_2
\end{aligned} \tag{12}$$

$$\begin{aligned}
\dot{E}_2 & = -F(X) + k_1 (H(t) \otimes I_n) E_2 + k_1 (H(t) \otimes I_n) E_1 \\
& - k_1 (B(t) 1_N \otimes e_{2,0}(t)) - k_2 E_2 - D,
\end{aligned} \tag{13}$$

where $\tilde{N} \triangleq F(X) - F(X_0) \in \mathbb{R}^{nN}$, $N_d \triangleq F(X_0) + D - \dot{X}_0 \in \mathbb{R}^{nN}$, $F(X) \triangleq [f_1^\top(x_1(t)), f_2^\top(x_2(t)), \dots, f_N^\top(x_N(t))]^\top \in \mathbb{R}^{nN}$, $F(X_0) \triangleq [f_1^\top(x_0(t)), f_2^\top(x_0(t)), \dots, f_N^\top(x_0(t))]^\top \in \mathbb{R}^{nN}$, $D \triangleq [d_1^\top(t), d_2^\top(t), \dots, d_N^\top(t)]^\top \in \mathbb{R}^{nN}$, $X_0 \triangleq 1_N \otimes x_0(t) \in \mathbb{R}^{nN}$, $X \triangleq [x_1^\top(t), x_2^\top(t), \dots, x_N^\top(t)]^\top \in \mathbb{R}^{nN}$, and $\mathbf{V} \triangleq [v_1^\top, v_2^\top, \dots, v_N^\top]^\top \in \mathbb{R}^{nN}$. Given the dynamics in (1) and Assumptions 1–3, and 6, there exists a $c_1 \in \mathbb{R}_{>0}$, such that $\|N_d\| \leq c_1$. By Assumption 3, there exists a $c_2 \in \mathbb{R}_{>0}$, such that $\|D\| \leq c_2$. By Assumptions 1 and 6, there exists a $c_3 \in \mathbb{R}_{>0}$, such that $\|F(X_0)\| \leq c_3$. Using Assumption 9, there exists a $c_4 \in \mathbb{R}_{>0}$, such that $\|\mathbf{V}\| \leq c_4$. Using [23, Lemma 5], we can bound \tilde{N} as $\|\tilde{N}\| \leq \mu(\|E\|)\|E\| + c_5$, where $E(t) \triangleq [E_1^\top, E_2^\top]^\top \in \mathbb{R}^{2nN}$ denotes the MAS error, $\mu: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a positive, nondecreasing, and radially unbounded function, and $c_5 \in \mathbb{R}_{>0}$ is a bounding constant. Note that $E(0) = [E_1^\top(0), E_2^\top(0)]^\top$, where $E_k(0) = [e_{k,1}^\top(0), e_{k,2}^\top(0), \dots, e_{k,N}^\top(0)]^\top$ for $k \in \{1, 2\}$.

Recall that the objective is to minimize $e_{1,i}(t)$ as given by (2) for each $i \in \mathcal{C}(t)$. However, E_1 and E_2 may contain error systems belonging to Byzantine agents, which cannot be controlled, may be unbounded, and may prevent the objective. Therefore, the FCLT error and the state estimation error are set to zero for all Byzantine agents, i.e., $e_{1,i}(t) \triangleq 0_n$ and $e_{2,i}(t) \triangleq 0_n$ for all $i \in \mathcal{B}(t)$, which allows the objective to apply only to the cooperative followers.

VI. STABILITY ANALYSIS

To facilitate the subsequent stability analysis, consider the following lemmas.

Lemma 1: There exists a bounding constant $\Lambda_{\max} \in \mathbb{R}_{>0}$, such that $\|H(t) \otimes I_n\| \leq \Lambda_{\max}$ for all $t \geq 0$.

Proof: See Appendix A.

Lemma 2: If Assumptions 4, 7, and 8 are satisfied for all $t \geq 0$, then a bounding constant $\Lambda_{\min} \in \mathbb{R}_{>0}$ exists, such that $E_1^\top(H(t) \otimes I_n)E_1 \geq \Lambda_{\min}\|E_1\|^2$ for all $t \geq 0$.⁴

Proof: See Appendix B.

Furthermore, consider the following. Substituting (2), (3), and (9) for all $i \in \mathcal{V}$ into Z yields

$$\begin{aligned}
Z & = -(H(t) \otimes I_n) E_1 - (H(t) \otimes I_n) E_2 \\
& + (B(t) 1_N \otimes e_{2,0}(t)).
\end{aligned} \tag{14}$$

Using Lemma 1, (14), and Young's inequality, it follows that:

$$\begin{aligned}
-k_{1,3} \|E_1\|^2 & \leq k_{1,3} \|E_2\|^2 - \frac{k_{1,3}}{(2\Lambda_{\max}^2 + \Lambda_{\max})} \|Z\|^2 \\
& + \frac{k_{1,3}}{\Lambda_{\max}} \|B(t) 1_N \otimes e_{2,0}(t)\|^2.
\end{aligned} \tag{15}$$

Note that (15) is a useful inequality that facilitates the development of the event-trigger mechanisms for the leader and the followers. Moreover, observe that

$$\|B(t) 1_N \otimes e_{2,0}(t)\|^2 \leq N b_{\max}^2 \|e_{2,0}(t)\|^2 \tag{16}$$

since $\|B(t) 1_N \otimes e_{2,0}(t)\|^2 = \sum_{i \in \mathcal{V}} b_i^2(t) \|e_{2,0}(t)\|^2$ and $b_i(t) \in [0, b_{\max}]$ for all $t \geq 0$ and each $i \in \mathcal{V}$ by construction. The subsequent stability analysis uses several auxiliary parameters. Let

$$\begin{aligned}
\phi_1 & \triangleq \left(1 - \frac{1}{\kappa} \left(\frac{2\Lambda_{\max} + 1}{2\Lambda_{\min}}\right)\right) k_{1,2} - \frac{1}{2} \\
& - \frac{1}{\kappa} \left(k_{1,3} + \frac{\rho_1^2}{\delta_1}\right) \left(\frac{2\Lambda_{\max} + 1}{2\Lambda_{\min}}\right) - \frac{k_2}{2\kappa}, \\
\phi_2 & \triangleq k_{2,1} - \frac{1}{2}, \quad \phi_3 \triangleq \frac{k_1}{2} + \frac{\kappa k_1}{2} + \frac{k_{1,3}}{\Lambda_{\max}}, \\
\phi_4 & \triangleq \frac{k_1}{2} + k_{1,3} + k_1 \Lambda_{\max} + \frac{\kappa(2k_1 \Lambda_{\max} + k_2)}{2}, \\
\phi_5 & \triangleq \frac{k_{1,3}}{(2\Lambda_{\max}^2 + \Lambda_{\max})}, \quad \phi_6 \triangleq \min\{\phi_1, \phi_2\}, \\
\delta^* & \triangleq \delta_1 + \delta_2 + c_0 + \varepsilon.
\end{aligned} \tag{17}$$

Note that κ , c_0 , and ε are defined in Theorem 1. The set over which the stability analysis is performed is

$$\mathcal{D} \triangleq \left\{ \xi \in \mathbb{R}^{2nN} : \|\xi\| < \inf \mu^{-1} \left(\left[\sqrt{\phi_6/4}, \infty \right) \right) \right\},$$

where, given a set $\Omega \subset \mathbb{R}$, the preimage $\mu^{-1}(\Omega) \subset \mathbb{R}$ is defined as $\mu^{-1}(\Omega) \triangleq \{\omega \in \mathbb{R} : \mu(\omega) \in \Omega\}$. The admissible set of initial conditions is

$$\mathcal{S}_{\mathcal{D}} \triangleq \left\{ \xi \in \mathbb{R}^{2nN} : \|\xi\| < \frac{\sqrt{2}}{2} \inf \mu^{-1} \left(\left[\sqrt{\phi_6/4}, \infty \right) \right) \right\}.$$

⁴The use of Assumption 7 the trust model in (4), reputation model in (5), and edge weight policy in (6) ensure Type I and Type II Byzantine agents are detected and removed from the CMAS.

Let $E_{\max} \triangleq \frac{\sqrt{2}}{2} \inf \mu^{-1}([\sqrt{\phi_6/4}, \infty))$, and recall that μ is a nondecreasing function. If $\|E\| < E_{\max}$, then $\mu(\|E\|) \leq \sqrt{\phi_6/4}$. In the Appendix, we present an algorithm that summarizes the control strategy used by each agent to achieve the objective. The algorithm is expressed with respect to follower i , and a similar algorithm follows for the leader. Recall that $\omega_{ij}(t_k^j) = \|x_i(t_k^j) - x_{j,2}(t_k^j)\|$ and $\Delta t_k^j = t_k^j - t_{k-1}^j$ as defined in Section V-A.

Theorem 1: The trust model in (4), reputation model in (5), edge weight policy in (6), estimator in (7), and controller in (8) and (9) ensure the MAS error E is uniformly ultimately bounded (UUB) in the sense that

$$\limsup_{t \rightarrow \infty} \|E\| \leq 2\sqrt{\frac{4c_5^2 + 2\delta^*}{\phi_6}} \quad (18)$$

provided the leader broadcasts its state as dictated by the event-trigger mechanism in

$$t_{k+1}^0 \triangleq \inf \left\{ t > t_k^0 : Nb_{\max}^2 \phi_3 \|e_{2,0}(t)\|^2 \geq c_0 \right\}, \quad (19)$$

each follower $i \in \mathcal{V}$ broadcasts its state as dictated by the event-trigger mechanism in

$$t_{k+1}^i \triangleq \inf \left\{ t > t_k^i : \phi_4 \|e_{2,i}(t)\|^2 \geq \phi_5 \|z_i(t)\|^2 + \frac{\varepsilon}{N} \right\}, \quad (20)$$

Assumptions 1–9 are satisfied, the initial condition of the system is selected, such that $E(0) \in \mathcal{S}_D$, and the following sufficient user-defined parameter conditions are satisfied:

$$\kappa > \frac{2\Lambda_{\max} + 1}{2\Lambda_{\min}}, \quad k_{1,3} > 0, \quad k_{2,1} > \frac{1}{2}, \quad \rho_1 \geq c_1,$$

$$\rho_2 \geq c_2 + c_3, \quad c_0 > 0, \quad \delta_1 > 0, \quad \delta_2 > 0, \quad \varepsilon > 0,$$

$$k_{1,2} > \frac{2\kappa\Lambda_{\min}}{2(\kappa\Lambda_{\min} - \Lambda_{\max}) - 1} \left(\frac{1}{2} + \frac{k_2}{2\kappa} \right) + \frac{1}{\kappa} \left(k_{1,3} + \frac{\rho_1^2}{\delta_1} \right) \left(\frac{2\Lambda_{\max} + 1}{2\Lambda_{\min}} \right),$$

$$\sqrt{(8c_5^2 + 4\delta^*)/\phi_6} < \frac{\sqrt{2}}{2} \inf \mu^{-1}([\sqrt{\phi_6/4}, \infty)). \quad (21)$$

Proof: Consider the candidate Lyapunov function $V_1 : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ defined as

$$V_1(E(t)) \triangleq \frac{1}{2} E_1^\top E_1 + \frac{1}{2} E_2^\top E_2, \quad (22)$$

which can be bounded as

$$\alpha_1(\|E\|) \leq V_1(E(t)) \leq \alpha_2(\|E\|), \quad (23)$$

where $\alpha_1, \alpha_2 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ are user-defined class \mathcal{K} functions. Without loss of generality, let $\alpha_1(\|E\|) \triangleq \frac{1}{2}\|E\|^2$ and $\alpha_2(\|E\|) \triangleq \|E\|^2$. Suppose $g : [0, \infty) \rightarrow \mathbb{R}^{2nN}$ is a Filippov solution to the differential inclusion $\dot{g}(t) \in K[h](g(t))$, where $g(t) = E(t)$, the mapping $K[\cdot]$ provides a calculus for computing Filippov's differential inclusion as defined in [24], and $h : \mathbb{R}^{2nN} \rightarrow \mathbb{R}^{2nN}$ is defined as $h(g(t)) = [\dot{E}_1^\top, \dot{E}_2^\top]^\top$. The

time-derivative of V_1 exists almost everywhere (a.e.), i.e., for almost all $t \in [0, \infty)$, and

$$\dot{V}_1(g(t)) \stackrel{a.e.}{\in} \dot{\tilde{V}}_1(g(t)), \quad (24)$$

where $\dot{\tilde{V}}_1(g(t))$ is the generalized time-derivative of V_1 along the Filippov trajectories of $\dot{g}(t) = h(g(t))$. By [25, eq. 13], $\dot{\tilde{V}}_1(g(t)) \triangleq \bigcap_{\xi \in \partial V_1(g(t))} \xi^\top [K[h]^\top(g(t)), 1]^\top$, where $\partial V_1(g(t))$ denotes the Clarke generalized gradient of $V_1(g(t))$. Since $V_1(g(t))$ is continuously differentiable in $g(t)$, $\partial V_1(g(t)) = \{\nabla V_1(g(t))\}$, where ∇ denotes the gradient operator. The generalized time-derivative of (22) is

$$\dot{\tilde{V}}_1(g(t)) \subseteq E^\top(t) K[h](g(t)). \quad (25)$$

Using the calculus of $K[\cdot]$ from [24], (25), and simplifying the substitution of (12) and (13) into the generalized time-derivative of (22) yields

$$\begin{aligned} \dot{\tilde{V}}_1(g(t)) \subseteq & \left\{ E_1^\top \tilde{N} + E_1^\top N_d - E_2^\top F(X) - E_2^\top D \right\} \\ & + k_1 E_2^\top K[(H(t) \otimes I_n) E_2] + k_2 E_1^\top K[E_2] \\ & - k_1 E_1^\top K[(H(t) \otimes I_n) E_2] - k_2 E_2^\top K[E_2] \\ & + k_1 E_2^\top K[(H(t) \otimes I_n)] E_1 \\ & - k_1 E_1^\top K[(H(t) \otimes I_n)] E_1 \\ & + k_1 E_1^\top K[(B(t) 1_N \otimes e_{2,0}(t))] \\ & - k_1 E_2^\top K[(B(t) 1_N \otimes e_{2,0}(t))], \end{aligned} \quad (26)$$

where set addition is defined by the Minkowski sum. Adding and subtracting $E_2^\top F(X_0)$ and using (24), Lemma 1, Lemma 2, $\|N_d\| \leq c_1$, $\|D\| \leq c_2$, $\|F(X_0)\| \leq c_3$, $\|V\| \leq c_4$, $\|\tilde{N}\| \leq \mu(\|E\|)\|E\| + c_5$, and Young's inequality, (26) can be upper bounded as

$$\begin{aligned} \dot{V}_1(E(t)) & \stackrel{a.e.}{\leq} \frac{1}{2} \|E_1\|^2 + 2\mu^2(\|E\|) \|E\|^2 + 2c_5^2 \\ & + c_1 \|E_1\| + 2k_1 \Lambda_{\max} \|E_1\| \|E_2\| - k_1 \Lambda_{\min} \|E_1\|^2 \\ & + k_1 \|E_1\| \|B(t) 1_N \otimes e_{2,0}(t)\| + k_2 \|E_1\| \|E_2\| \\ & + c_3 \|E_2\| + \frac{1}{2} \|E_2\|^2 + k_1 \Lambda_{\max} \|E_2\|^2 \\ & + k_1 \|E_2\| \|B(t) 1_N \otimes e_{2,0}(t)\| - k_2 \|E_2\|^2 \\ & + c_2 \|E_2\|. \end{aligned} \quad (27)$$

Since $\rho_1 \geq c_1$ and $\rho_2 \geq c_2 + c_3$ by the hypothesis of Theorem 1, we then see that $(c_1 - \frac{\rho_1^2}{\delta_1} \|E_1\|) \|E_1\| \leq \delta_1$ and $(c_2 + c_3 - \frac{\rho_2^2}{\delta_2} \|E_2\|) \|E_2\| \leq \delta_2$. Using these bounds, $k_1 = \frac{1}{\Lambda_{\min}}(k_{1,1} + \frac{\rho_1^2}{\delta_1})$, $k_{1,1} = k_{1,2} + k_{1,3}$, $k_2 = k_{2,1} + \frac{\rho_2^2}{\delta_2}$, (15), Young's inequality, the inequality in (16), and the auxiliary parameters in (17) and (27) can be upper bounded by

$$\begin{aligned} \dot{V}_1(E(t)) & \stackrel{a.e.}{\leq} -\frac{\phi_6}{2} \|E\|^2 - \left(\frac{\phi_6}{2} - 2\mu^2(\|E\|) \right) \|E\|^2 \\ & + \sum_{i \in \mathcal{V}} \left(\phi_4 \|e_{2,i}(t)\|^2 - \phi_5 \|z_i(t)\|^2 - \frac{\varepsilon}{N} \right) \\ & + 2c_5^2 + Nb_{\max}^2 \phi_3 \|e_{2,0}(t)\|^2 - c_0 + \delta^*, \end{aligned} \quad (28)$$

where satisfying the parameter conditions in (21) ensures $\phi_i > 0 \forall i \in \{1, 2, \dots, 6\}$. Based on (28), the event-trigger mechanism for the leader is given by (19), and the event-trigger mechanism for each follower $i \in \mathcal{V}$ is given by (20). Since each agent provides state feedback according to the event-trigger mechanisms in (19) and (20), (28) can be upper bounded as

$$\begin{aligned} \dot{V}_1(E(t)) &\stackrel{a.e.}{\leq} -\frac{\phi_6}{2} \|E\|^2 - \left(\frac{\phi_6}{2} - 2\mu^2(\|E\|) \right) \|E\|^2 \\ &\quad + 2c_5^2 + \delta^*. \end{aligned} \quad (29)$$

Using (23), we see that $\|E\| \leq \alpha_1^{-1}(\alpha_2(\|E\|))$ and $\alpha_2^{-1}(\alpha_1(\|E\|)) \leq \|E\|$, where $\alpha_1^{-1}(\alpha_2(\|E\|)) = \sqrt{2}\|E\|$ and $\alpha_2^{-1}(\alpha_1(\|E\|)) = \frac{\sqrt{2}}{2}\|E\|$ given the selected class \mathcal{K} functions. Note that $\phi_6/2 - 2\mu^2(\|E\|) > 0$ provided $E_{\max} = \frac{\sqrt{2}}{2} \inf \mu^{-1}([\sqrt{\phi_6/4}, \infty)) > \|E\|$. Moreover, $-(\phi_6/4)\|E\|^2 + 2c_5^2 + \delta^* \leq 0$ provided $\|E\| \geq \sqrt{(8c_5^2 + 4\delta^*)/\phi_6}$. It then follows that (29) can be upper bounded as $\dot{V}_1(E(t)) \stackrel{a.e.}{\leq} -\frac{\phi_6}{4}\|E\|^2$ for all $E_{\max} > \|E\| \geq \sqrt{(8c_5^2 + 4\delta^*)/\phi_6}$. Let $\mathcal{Z} \triangleq \{\xi \in \mathbb{R}^{2nN} : \|\xi\| \geq \sqrt{(8c_5^2 + 4\delta^*)/\phi_6}\}$. Since $\sqrt{(8c_5^2 + 4\delta^*)/\phi_6} < \frac{\sqrt{2}}{2} \inf \mu^{-1}([\sqrt{\phi_6/4}, \infty))$, we then see that $\dot{V}_1(E(t)) \stackrel{a.e.}{<} 0$ for all $E \in \mathcal{S}_D \cap \mathcal{Z}$. If $E \in \mathcal{S}_D \cap \mathcal{Z}^C$, then $\dot{V}_1(E(t)) \stackrel{a.e.}{\leq} -\frac{\phi_6}{4}\|E\|^2 + 2c_5^2 + \delta^*$, which may allow $V_1(E(t))$ to grow. However, E will exit \mathcal{Z}^C before exiting \mathcal{S}_D , and, therefore, flow into $\mathcal{S}_D \cap \mathcal{Z}$. It then follows that \mathcal{S}_D is forward invariant, where initializing the MAS, such that $E(0) \in \mathcal{S}_D$ ensures E is uniformly ultimately bounded with the ultimate bound presented in (18).

We now show that the state, state estimate, control signal, FCLT error, and state estimation error are bounded for each agent. Since $E \in \mathcal{L}_\infty$, it follows that $E_1 \in \mathcal{L}_\infty$ and $E_2 \in \mathcal{L}_\infty$ given the definition of E . Since $E_1 \in \mathcal{L}_\infty$, $e_{1,i}(t) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$ given the definition of E_1 . From Assumption 6, $x_0(t) \in \mathcal{L}_\infty$. Since $e_{1,i}(t) \in \mathcal{L}_\infty$, $v_i \in \mathcal{L}_\infty$, and $x_0(t) \in \mathcal{L}_\infty$, (2) implies $x_i(t) \in \mathcal{L}_\infty$ for each $i \in \mathcal{V}$.

Since $E_2 \in \mathcal{L}_\infty$, we also see that $e_{2,i}(t) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$ given the definition of E_2 . Since $x_i(t) \in \mathcal{L}_\infty$ and $e_{2,i}(t) \in \mathcal{L}_\infty$, (3) implies $\hat{x}_i(t) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$. Since $x_0(t) \in \mathcal{L}_\infty$ by Assumption 6, (7) implies $\hat{x}_0(t) \in \mathcal{L}_\infty$. Since $\hat{x}_i(t) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V} \cup \{0\}$, $a_{ij}(t) \in [0, 1]$ for all $t \geq 0$ and each $i, j \in \mathcal{V}$ by construction, $v_i \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$ by design, and $b_i(t) \in [0, b_{\max}]$ for all $t \geq 0$ and each $i \in \mathcal{V}$ by construction, (9) implies $z_i(t) \in \mathcal{L}_\infty$ for each $i \in \mathcal{V}$. Since $z_i(t) \in \mathcal{L}_\infty$, $e_{2,i}(t) \in \mathcal{L}_\infty$, $x_i(t) \in \mathcal{L}_\infty$, and $g_i^+(x_i(t)) \in \mathcal{L}_\infty$ by Assumption 2, it follows that (8) implies $u_i(t) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$. ■

Remark 5: With respect to (21), c_0 is a user-defined parameter that determines the rate at which the leader broadcasts its state to its neighbors. Moreover, c_0 is used to uniformly lower bound the difference between consecutive broadcast events performed by the leader away from zero. Similarly, ε is a user-defined parameter used to uniformly lower bound the difference between consecutive broadcast events for each follower $i \in \mathcal{V}$ away from zero.

Remark 6: Based on the definition of ϕ_1 and ϕ_2 in (17), ϕ_6 can be increased by increasing $k_{1,2}$ and $k_{2,1}$ provided $k_{1,2}$ and

κ are selected according to (21). Given $E(0)$, select ϕ_6 , such that $E(0) \in \mathcal{S}_D$. Observe that $\sqrt{(8c_5^2 + 4\delta^*)/\phi_6}$ decreases with increasing ϕ_6 , where c_5 and δ^* are fixed. Furthermore, since μ is a nondecreasing function, it follows that $\mu^{-1}([\sqrt{\phi_6/4}, \infty))$ is nondecreasing with respect to ϕ_6 . Hence, $\sqrt{(8c_5^2 + 4\delta^*)/\phi_6} < \frac{\sqrt{2}}{2} \inf \mu^{-1}([\sqrt{\phi_6/4}, \infty))$ can be satisfied for some ϕ_6 .

We now show the event-trigger mechanisms in (19) and (20) are free from Zeno behavior.

Theorem 2: The difference between consecutive broadcast times generated by the event-trigger mechanism of the leader in (19) is uniformly lower bounded by

$$t_{k+1}^0 - t_k^0 \geq \frac{1}{b_{\max}\theta_{0,\max}} \sqrt{\frac{c_0}{N\phi_3}} \quad (30)$$

for all $k \in \mathbb{Z}_{\geq 0}$, where $\theta_{0,\max} \in \mathbb{R}_{>0}$ is a user-defined parameter selected, such that $\|f_0(x_0(t))\| + \|g_0(x_0(t))\| \|u_0(t)\| + \|d_0(t)\| \leq \theta_{0,\max}$.

Proof: See Appendix C. ■

Theorem 3: The difference between consecutive broadcast times generated by the event-trigger mechanism of follower $i \in \mathcal{V}$ in (20) is uniformly lower bounded by

$$t_{k+1}^i - t_k^i \geq \frac{1}{k_2} \ln \left(\frac{k_2}{\theta_{i,\max}} \sqrt{\frac{\varepsilon}{N\phi_4}} + 1 \right) \quad (31)$$

for all $k \in \mathbb{Z}_{\geq 0}$, where $\theta_{i,\max} \in \mathbb{R}_{>0}$ is a user-defined parameter selected, such that $\|f_i(x_i(t))\| + k_1\|z_i(t)\| + \|d_i(t)\| \leq \theta_{i,\max}$.

Proof: See Appendix D. ■

VII. SIMULATION STUDY

A simulation study is included to validate the developed approach. The simulated MAS consists of five follower agents and a single leader agent. The initial positions of each agent are $x_0(0) = [500, 10]^\top$, $x_1(0) = [465, -51]^\top$, $x_2(0) = [566, -52]^\top$, $x_3(0) = [417, -103]^\top$, $x_4(0) = [518, -104]^\top$, and $x_5(0) = [619, -105]^\top$. The uncertain drift dynamics⁵ and known control effectiveness matrix of agent i are $f_i(x_i(t)) \triangleq [\bar{a}_{1i}\psi(x_{1i}(t)) + \bar{a}_{2i}, \bar{a}_{3i} + \bar{a}_{4i}\psi(x_{2i}(t))]^\top \in \mathbb{R}^2$ and

$$g_i(x_i(t)) \triangleq \begin{bmatrix} \cos(\varphi_i(t)) & -\sin(\varphi_i(t)) \\ \sin(\varphi_i(t)) & \cos(\varphi_i(t)) \end{bmatrix} \in \mathbb{R}^{2 \times 2},$$

respectively, where $x_i(t) \triangleq [x_{1i}(t), x_{2i}(t)]^\top \in \mathbb{R}^2$, $\bar{a}_i \triangleq [\bar{a}_{1i}, \bar{a}_{2i}, \bar{a}_{3i}, \bar{a}_{4i}]^\top \in \mathbb{R}^4$,

$$\psi(x) \triangleq \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \in \mathbb{R}_{>0},$$

and

$$\varphi_i(t) \triangleq \arctan\left(\frac{x_{2i}(t)}{x_{1i}(t)}\right) \in \mathbb{R},$$

such that $\arctan(\cdot)$ is the four quadrant inverse tangent, i.e., $\text{atan2}(\cdot)$ with respect to MATLAB. The uncertain drift

⁵The leader knows its drift dynamics while the followers do not know their drift dynamics.

dynamics coefficients for each agent are $\bar{a}_0 \triangleq [1, 1, 1]^\top$, $\bar{a}_1 \triangleq [1, 1.5, 3, 2]^\top$, $\bar{a}_2 \triangleq [0.5, 0.5, 1.9, 0.7]^\top$, $\bar{a}_3 \triangleq [1.5, 2.1, 1.2, 0.5]^\top$, $\bar{a}_4 \triangleq [3, 1.75, 1.15, 3]^\top$, and $\bar{a}_5 \triangleq [2, 1, 1, 1.6]^\top$. The exogenous disturbance acting on all agents is random, drawn from a normal distribution, and scaled by $d_{\text{mag}} \in \mathbb{R}_{>0}$, which is subsequently defined. The relative position vectors defining the desired formation are $v_1 \triangleq [-50, -50]^\top$, $v_2 \triangleq [50, -50]^\top$, $v_3 \triangleq [-100, -100]^\top$, $v_4 \triangleq [0, -100]^\top$, and $v_5 \triangleq [100, -100]^\top$. The known desired trajectory $x_d: [0, \infty) \rightarrow \mathbb{R}^2$ of the leader is $x_d(t) \triangleq 500[\cos(2\pi 10^{-2}t), \sin(2\pi 10^{-2}t)]^\top$, while the leader's trajectory tracking error $e_0: [0, \infty) \rightarrow \mathbb{R}^2$ is defined as $e_0(t) \triangleq x_d(t) - x_0(t)$. The leader's tracking error can be globally exponentially regulated using the following controller: $u_0(t) \triangleq g_0^+(x_0(t))(-f_0(x_0(t)) + \dot{x}_d(t) + k_{0,1}e_0(t) + k_{0,2}\text{sgn}(e_0(t)))$, where $k_{0,1} \in \mathbb{R}_{>0}$ and $k_{0,2} \in \mathbb{R}_{>0}$ are user-defined parameters. The simulation is 50 time units long and uses an integration time-step of 1.00×10^{-3} time units. The following parameters are used to generate the simulation results: $b_{\text{max}} = 1.25$, $\vartheta = 10$, $\sigma = 100$, $R = 110$, $k_{0,1} = 1$, $k_{0,2} = 0.25$, $d_{\text{mag}} = 0.03$, $c_1 = 3$, $c_2 = 1$, $c_3 = 0.008$, $\Lambda_{\text{min}} = 1$, $\Lambda_{\text{max}} = 10$, $\rho_1 = 4$, $\delta_1 = 0.25$, $\rho_2 = 2.008$, $\delta_2 = 0.25$, $\kappa = 105$, $k_{1,3} = 1$, $k_{2,1} = 1$, $k_{1,2} = 15.74$, $\varepsilon = 10^6$, $c_0 = 10^4$, $N = 5$, $s_1 = 5$, $t_{\text{reset}} = 1$, $\Delta_i = 0.125$ for each $i \in \{1, 2, \dots, 5\}$, $\eta_\tau = 70$, $\eta_\zeta = 70$, and $\zeta_{\text{min}} = 0.95$. The adjacency matrix of the communication graph of the followers and the leader pinning matrix are

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

and $B(t) = b_{\text{max}} \cdot \text{diag}(1, 1, 0, 0, 0)$, respectively.

Figs. 1 and 2 display the simulation results. The simulation subjected the MAS to at most two Byzantine agents, where Follower 3 was converted to a Type II Byzantine agent for $t \in [10, 40]$ time units, and Follower 5 was converted to a Type I Byzantine agent for $t \in [13, 40]$ time units. Follower 3 does not communicate with its neighbors during $t \in [10, 40]$. For $t \in [10, 30]$, Follower 3 was first maneuvered to $[225, 150]$ in an attempt to strain the network and destabilize the CMAS. For $t \in [30, 40]$, Follower 3 was maneuvered toward the CMAS, where it was converted back to a cooperative follower for $t > 40$ time units.⁶

The controller used to maneuver Follower 3 to $[225, 150]$ and then back to the CMAS is identical in form to that of the leader, where exact model knowledge was used only for the purpose of moving the follower away from the CMAS and simulating unanticipated behavior of an initial member of the

⁶An adversary may corrupt a cooperative agent and cause it to abandon the CMAS. However, it may be possible to execute countermeasures to convert the Byzantine agent back into a cooperative agent. In such a case, it may be desirable to maneuver the cooperative agent back into the formation formed by the remaining cooperative agents.

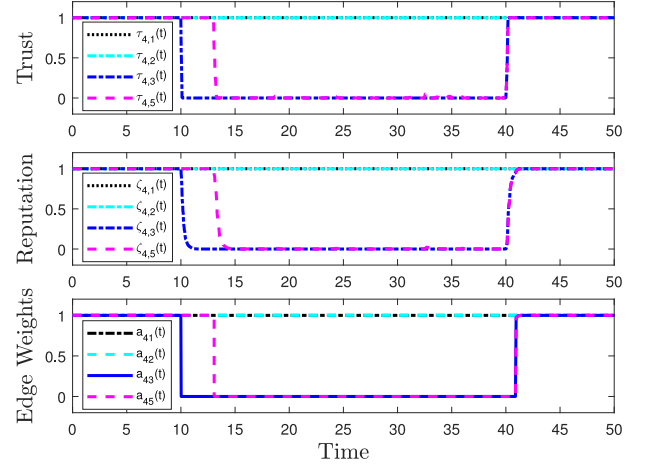


Fig. 1. Trust, reputation, and edge weight values that Follower 4 has for its neighbors. Since Followers 1 and 2 are cooperative agents for all time, they communicate true state information about themselves to Follower 4, which results in maximum trust, reputation, and edge weight values for all time. Conversely, Followers 3 and 5 are Byzantine for $t \in [10, 40]$ and $t \in [13, 40]$, respectively, which results in their zero trust, reputation, and edge weight values with respect to Follower 4 during their Byzantine status. The trust, reputation, and edge weight values that Follower 4 has in Followers 3 and 5 are quickly restored to the maximum once Followers 3 and 5 become cooperative for $t > 40$ and $t > 40$, respectively.

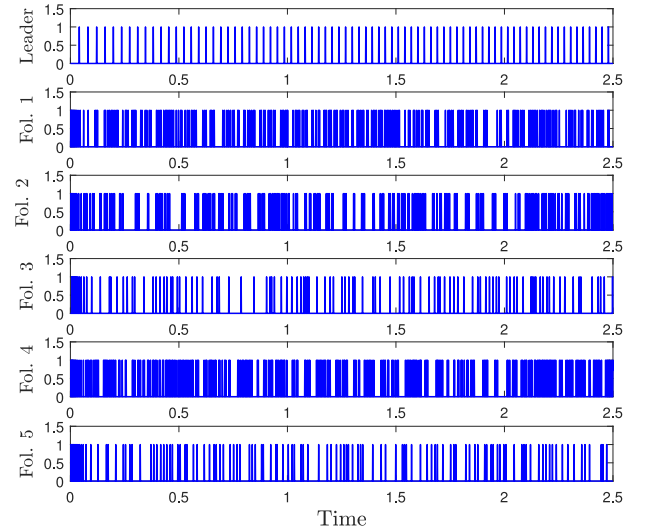


Fig. 2. Illustration of event-times for the leader and each follower during the first 2.5 time units of the simulation. A 0 or white space, denotes no communication and 1, or blue line, denotes a communication event. The first 2.5 time units of the simulation are shown, rather than entire simulation, to better exhibit the intermittency in communication.

CMAS. The communication protocol used by Follower 5 during $t \in [13, 40]$ was $x_{5,1}(t) = -0.1 \cdot x_5(t)$, i.e., the communicated information was negative one-tenth the true state of Follower 5. Since Follower 5 remained with the CMAS for $t \in [13, 40]$, its tracking error $e_{1,5}(t)$ is similar to that of the cooperative followers.

The cooperative followers, i.e., Followers 1, 2, and 4, satisfied the objective for all time, even in the presence of the Byzantine

adversaries. Followers 3 and 5 also satisfied the objective during their periods of cooperation. Fig. 1 depicts the trust, reputation, and edge weights of the neighbors of Follower 4, which illustrates the Byzantine behavior of Followers 3 and 5. The trust, reputation, and edge weight figures for Followers 1, 2, 3, and 5 are omitted since they are similar to those of Follower 4. Fig. 1 shows that Follower 4 detected the Byzantine behavior of Followers 3 and 5 at $t = 10$ and $t = 13$, respectively. As a result of the detected Byzantine behavior, the trust that Follower 4 had in Followers 3 and 5 decreased to 0, which caused the corresponding reputation values and edge weights to decrease to 0. Fig. 1 shows that Follower 4 detected cooperative behavior from Followers 3 and 5 at $t > 40$ and $t > 40$, respectively, which caused the trust, reputation, and edge weights of Followers 3 and 5 with respect to Follower 4 to increase to 1.

Fig. 2 depicts the event-times for the leader and each follower for the first 2.5 time units of the simulation. The average differences between consecutive event-times for the entire simulation for the leader and Followers 1–5 are 0.0282, 0.0144, 0.0136, 0.0016, 0.0141, and 0.0197 time units, respectively. The minimum differences between consecutive event-times for the leader and Followers 1–5 are 0.028, 0.002, 0.002, 0.001, 0.003, and 0.003 time units, respectively.

VIII. CONCLUSION

This work examines the formation control while leader tracking problem for a heterogeneous MAS consisting of agents with uncertain nonlinear dynamics. A distributed event-triggered controller is developed along with a reputation-based detection method that enables each follower to discern between cooperative and Byzantine neighbors. The edge weight policy alters the interaction among agents to enable the cooperative followers to achieve the objective. Future efforts will focus on developing controllers capable of ensuring the connectivity of the CMAS and collision avoidance while also considering time-varying formations. Moreover, more sophisticated Byzantine agent detectors that relax Assumption 7 and provide detection guarantees are motivated. While we study the fundamental case of multiple Byzantine adversaries that act independently within this work, future efforts are motivated to examine the case, where multiple Byzantine adversaries collaborate between each other to thwart the objective. Several open questions still remain. For example, how can accurate state information be determined in the absence of a ground-truth? Moreover, for agents with multiple common neighbors and no available ground-truth information, how can an agent determine its cooperative neighbor set if both the true cooperative neighbor set and the true Byzantine neighbor set seem cooperative?

APPENDIX

A. Proof of Lemma 1

Proof: Let $\mathcal{H} \triangleq \{\|H(t) \otimes I_n\| : t \geq 0\}$. Fix $t_1 \geq 0$, and suppose $\mathcal{C}(t_1) = \mathcal{V}$, $\mathcal{G}(t_1)$ is complete, and $B(t) = b_{\max} \cdot I_N$. Then, $\|H(t) \otimes I_n\|_F$ is maximum at t_1 , where $\Lambda_{\max} \triangleq \sqrt{n(N(N-1+b_{\max})^2 + N^2 - N)} = \|H(t_1) \otimes I_n\|_F$.

Since $\|H(t) \otimes I_n\| \leq \|H(t) \otimes I_n\|_F \leq \Lambda_{\max}$ for all $t \geq 0$, \mathcal{H} is a nonempty set that is bounded above. Therefore, $\Lambda_{\max} \geq \sup\{\mathcal{H}\}$.

B. Proof of Lemma 2

Proof: Fix $t_2 \geq 0$, and suppose that at time t_2 the MAS consists of $N_C(t_2) \in \mathbb{Z}_{\geq 0}$ cooperative followers and $N_B(t_2) \in \mathbb{Z}_{\geq 0}$ Byzantine adversaries, where $N_C(t_2) + N_B(t_2) = N$. Using Assumptions 4 and 7, the trust model in (4), the reputation model in (5), and the edge weight policy in (6), the connectivity matrix $\bar{H}(t_2)$ can be expressed as the block matrix

$$\bar{H}(t_2) = \begin{bmatrix} H_{CC}(t_2) & H_{CB}(t_2) \\ H_{BC}(t_2) & H_{BB}(t_2) \end{bmatrix},$$

where $H_{CC}(t_2) \in \mathbb{R}^{N_C(t_2) \times N_C(t_2)}$ is a diagonally dominant matrix with positive diagonal entries by construction, $H_{BB}(t_2) \in \mathbb{R}^{N_B(t_2) \times N_B(t_2)}$, $H_{BC}(t_2) \in \mathbb{R}^{N_B(t_2) \times N_C(t_2)}$, and $H_{CB}(t_2) \in \mathbb{R}^{N_C(t_2) \times N_B(t_2)}$. Note that $\bar{H}(t_2) = P(t_2)H(t_2)P^{-1}(t_2)$, where $\bar{H}(t_2)$ is permutation-similar to $H(t_2)$, and $P(t_2)$ is an orthogonal permutation matrix. Since $H_{CC}(t_2)$ is irreducibly diagonally dominant by Assumption 8, $H_{CC}(t_2)$ is nonsingular, i.e., $H_{CC}(t_2)$ is invertible [26, Corollary 6.2.27].⁷

Next, we show that the eigenvalues of $H_{CC}(t_2)$ have positive real parts. Since $P(t_2)$ is orthogonal, $P(t_2)$ is invertible, where $P^{-1}(t_2) = P^T(t_2)$. Let $\bar{E}_1 \triangleq (P(t_2) \otimes I_n)E_1 \in \mathbb{R}^{nN}$, where $\bar{E}_1 \triangleq [E_{1,C}^T, E_{1,B}^T]^T \in \mathbb{R}^{nN}$, such that $E_{1,C} \triangleq [e_{1,1}^T(t_2), e_{1,2}^T(t_2), \dots, e_{1,N_C(t_2)}^T(t_2)]^T \in \mathbb{R}^{nN_C(t_2)}$ and $E_{1,B} \triangleq [e_{1,N_C(t_2)+1}^T(t_2), e_{1,N_C(t_2)+2}^T(t_2), \dots, e_{1,N}^T(t_2)]^T = 0_{nN_B(t_2)}$ by convention. Substituting $\bar{H}(t_2)$ and \bar{E}_1 into $E_1^T(H(t_2) \otimes I_n)E_1$ yields $E_1^T(H(t_2) \otimes I_n)E_1 = E_{1,C}^T(H_{CC}(t_2) \otimes I_n)E_{1,C}$. Let $H_{\text{sym}}(t_2) \triangleq \frac{1}{2}(H_{CC}(t_2) + H_{CC}^T(t_2))$ and $H_{\text{skew}}(t_2) \triangleq \frac{1}{2}(H_{CC}(t_2) - H_{CC}^T(t_2))$, where $H_{\text{sym}}(t_2)$ and $H_{\text{skew}}(t_2)$ are symmetric and skew-symmetric matrices, respectively, by construction. Moreover, $H_{CC}(t_2) = H_{\text{sym}}(t_2) + H_{\text{skew}}(t_2)$. Since $H_{\text{skew}} \otimes I_n$ is a skew-symmetric matrix, $E_{1,C}^T(H_{\text{skew}}(t_2) \otimes I_n)E_{1,C} = 0$ and $E_{1,C}^T(H_{CC}(t_2) \otimes I_n)E_{1,C} = E_{1,C}^T(H_{\text{sym}}(t_2) \otimes I_n)E_{1,C}$. Since $H_{CC}(t_2)$ is a diagonally dominant matrix with positive diagonal entries, the real part of the eigenvalues of $H_{CC}(t_2)$ are nonnegative by the Gershgorin disk theorem in [27, The. 3.9.]. Moreover, since the real part of the eigenvalues of $H_{CC}(t_2)$ are nonnegative and $H_{CC}(t_2)$ is invertible, the real part of the eigenvalues of $H_{CC}(t_2)$ are positive.

We next show that the symmetric component of $H_{CC}(t_2)$ is positive definite. Since $H_{CC}(t_2)$ is invertible, for all nonzero $w \in \mathbb{R}^{N_C(t_2)}$ $H_{CC}(t_2)w \neq 0_{N_C(t_2)}$ and $H_{CC}^T(t_2)w \neq 0_{N_C(t_2)}$. Moreover, $H_{CC}(t_2)w + H_{CC}^T(t_2)w \neq 0_{N_C(t_2)}$ since $H_{CC}(t_2) + H_{CC}^T(t_2) \neq 0_{N_C(t_2) \times N_C(t_2)}$, i.e.,

⁷Every graph Laplacian is diagonally dominant because it has zero row sums. For connected graphs, their Laplacians are irreducible. Adding the Laplacian of a connected graph to a diagonal matrix with at least one positive entry makes at least one row strictly diagonally dominant. Hence, summing a connected graph Laplacian with the leader pinning matrix yields an irreducibly diagonally dominant matrix.

$H_{CC}(t_2)$ is not skew-symmetric. It then follows that $H_{\text{sym}}(t_2)w = \frac{1}{2}(H_{CC}(t_2)w + H_{CC}^\top(t_2)w) \neq 0_{N_C(t_2)}$ for all nonzero $w \in \mathbb{R}^{N_C(t_2)}$. Hence, $H_{\text{sym}}(t_2)$ has the trivial null space and 0 fails to be an eigenvalue of $H_{\text{sym}}(t_2)$. Furthermore, $H_{\text{sym}}(t_2)$ is a diagonally dominant matrix by construction with nonnegative eigenvalues by the Gershgorin disk theorem. Since $H_{\text{sym}}(t_2)$ is a symmetric matrix with positive real eigenvalues, $H_{\text{sym}}(t_2)$ is positive definite.

We next show $E_1^\top(H(t) \otimes I_n)E_1 \geq \Lambda_{\min}\|E_1\|^2$ for all $t \geq 0$, where $\Lambda_{\min} \in \mathbb{R}_{>0}$. By the Rayleigh quotient, it follows that $E_{1,C}^\top(H_{CC}(t_2) \otimes I_n)E_{1,C} \geq \lambda_{\min}(H_{\text{sym}}(t_2) \otimes I_n)E_{1,C}^\top E_{1,C}$. Since $E_{1,B}^\top E_{1,B} = 0$, $E_{1,C}^\top(H_{CC}(t_2) \otimes I_n)E_{1,C} \geq \lambda_{\min}(H_{\text{sym}}(t_2) \otimes I_n)\bar{E}_1^\top \bar{E}_1$. Since $E_1^\top(H(t_2) \otimes I_n)E_1 = E_{1,C}^\top(H_{CC}(t_2) \otimes I_n)E_{1,C}$, $E_{1,C}^\top(H_{CC}(t_2) \otimes I_n)E_{1,C} \geq \lambda_{\min}(H_{\text{sym}}(t_2) \otimes I_n)\bar{E}_1^\top \bar{E}_1$, and $\bar{E}_1^\top \bar{E}_1 = E_1^\top E_1$, $E_1^\top(H(t_2) \otimes I_n)E_1 \geq \lambda_{\min}(H_{\text{sym}}(t_2) \otimes I_n)E_1^\top E_1$. Since t_2 was arbitrary, $E_1^\top(H(t) \otimes I_n)E_1 \geq \lambda_{\min}(H_{\text{sym}}(t) \otimes I_n)\|E_1\|^2$ for all $t \geq 0$. Let $\lambda_{\min}(\mathcal{H}_{\text{sym}}) \triangleq \{\lambda_{\min}(H_{\text{sym}}(t) \otimes I_n) : t \geq 0\} \subset (0, \infty)$. Since $\lambda_{\min}(H_{\text{sym}}(t) \otimes I_n) > 0$ for all $t \geq 0$ and $\lambda_{\min}(\mathcal{H}_{\text{sym}}) \neq \emptyset$, $\lambda_{\min}(\mathcal{H}_{\text{sym}})$ is a nonempty set that is bounded below. Hence, $\Lambda_{\min} \triangleq \inf(\lambda_{\min}(\mathcal{H}_{\text{sym}})) \in \mathbb{R}_{>0}$.

C. Proof of Theorem 2

Proof: Similar argument to proof of Theorem 3.

D. Proof of Theorem 3

Proof: Let $t \geq t_k^i \geq 0$ and $i \in \mathcal{V}$. Substituting (1), (7), and (8) into the time-derivative of (3) yields $\dot{e}_{2,i}(t) \stackrel{a.e.}{=} -f_i(x_i(t)) - k_1 z_i(t) - k_2 e_{2,i}(t) - d_i(t)$. By Assumption 3, $\|d_i(t)\| \leq d_{i,\max}$. Recall that $x_i(t) \in \mathcal{L}_\infty$ and $z_i(t) \in \mathcal{L}_\infty$ from the proof of Theorem 1. Therefore, there exists $x_{i,\max} \in \mathbb{R}_{>0}$ and $z_{i,\max} \in \mathbb{R}_{>0}$, such that $\|x_i(t)\| \leq x_{i,\max}$ and $\|z_i(t)\| \leq z_{i,\max}$, respectively. Since $\|x_i(t)\| \leq x_{i,\max}$, Assumption 1 implies that $\|f_i(x_i(t))\| \leq f_{i,\max}$ for some $f_{i,\max} \in \mathbb{R}_{>0}$. It then follows that $\|\dot{e}_{2,i}(t)\| \leq k_2\|e_{2,i}(t)\| + \theta_{i,\max}$, where $\theta_{i,\max} \geq f_{i,\max} + k_1 z_{i,\max} + d_{i,\max} \in \mathbb{R}_{>0}$. Let $v_i : [t_k^i, \infty) \rightarrow \mathbb{R}_{\geq 0}$ satisfy $\dot{v}_i(t) = k_2 v_i(t) + \theta_{i,\max}$ with initial condition $v_i(t_k^i) = \|e_{2,i}(t_k^i)\|$. Then, $v_i(t_k^i) = 0$ and

$$v_i(t) = \frac{\theta_{i,\max}}{k_2} (e^{k_2(t-t_k^i)} - 1).$$

Observe that

$$\frac{d}{dt}\|e_{2,i}(t)\| = \frac{e_{2,i}^\top(t)\dot{e}_{2,i}(t)}{\|e_{2,i}(t)\|} \stackrel{a.e.}{\leq} \|\dot{e}_{2,i}(t)\|.$$

Since $\frac{d}{dt}\|e_{2,i}(t)\| \leq \|\dot{e}_{2,i}(t)\|$ and $\|\dot{e}_{2,i}(t)\| \leq k_2\|e_{2,i}(t)\| + \theta_{i,\max}$, it follows that $\frac{d}{dt}\|e_{2,i}(t)\| \stackrel{a.e.}{\leq} k_2\|e_{2,i}(t)\| + \theta_{i,\max}$. Using the solution of $\frac{d}{dt}\|e_{2,i}(t)\| \stackrel{a.e.}{\leq} k_2\|e_{2,i}(t)\| + \theta_{i,\max}$ and $v_i(t) = \frac{\theta_{i,\max}}{k_2} (e^{k_2(t-t_k^i)} - 1)$, we see that $\|e_{2,i}(t)\| \leq v_i(t)$ for all $t \in [t_k^i, \infty)$. Since $\|e_{2,i}(t)\| \leq v_i(t)$ and $v_i(t) = \frac{\theta_{i,\max}}{k_2} (e^{k_2(t-t_k^i)} - 1)$, equation (20) yields (31).

E. Control Algorithm

Algorithm 1: Control Algorithm for Follower i .

- 1: Select $\Lambda_{\max} > 0$ and $\Lambda_{\min} > 0$ according to Lemmas 1 and 2, respectively.
- 2: Select $x_0(0)$, $x_i(0)$, $\hat{x}_i(0)$, and v_i for all $i \in \mathcal{V}$.
- 3: Set $N = |\mathcal{V}|$. Select $b_{\max} > 0$.
- 4: Select κ , c_0 , ε , $k_{1,3}$, $k_{2,1}$, ρ_1 , ρ_2 , δ_1 , δ_2 , and $k_{1,2}$ according to (21) and $E(0) \in \mathcal{S}_D$.
- 5: Compute $k_1 = \frac{1}{\Lambda_{\min}}(k_{1,1} + \frac{\rho_1^2}{\delta_1})$, $k_2 = k_{2,1} + \frac{\rho_2^2}{\delta_2}$, and $k_{1,1} = k_{1,2} + k_{1,3}$.
- 6: Compute ϕ_1 through ϕ_6 according to (17).
- 7: Set $\tau_{ij}(0) = 1$ and $\zeta_{ij}(0) = 1$ for all $i, j \in \mathcal{V}$.
- 8: Select $s_1 > 0$, $\zeta_{\min} \in [0, 1]$, $t_{\text{reset}} > 0$, $\eta_\tau > 0$, and $\eta_\zeta > 0$.
- 9: Select $\vartheta > 0$, $r \in (0, R)$, and $\Delta_i > 0$ for all $i \in \mathcal{V}$.
- 10: Set $k(i) = 0$ and $t_0^i = 0$ for all $i \in \mathcal{V} \cup \{0\}$.
- 11: Set $a_{ij}(t) = 1$ for all $j \in \mathcal{N}_i(0)$.
- 12: Set $\hat{x}_j(t) = x_j(t_{k(j)}^j)$ for all $j \in \mathcal{N}_i(0) \cup \{i\}$.
- 13: **if** $0 \in \mathcal{N}_i(0)$ **then**
- 14: Set $\hat{x}_0(t) = x_0(t_{k(0)}^0)$. Set $b_i(t) = b_{\max}$.
- 15: **end if**
- 16: **while** true **do**
- 17: Compute $\zeta_{ij}(t)$ according to (5) for each $j \in \mathcal{N}_i(t)$.
- 18: Compute $a_{ij}(t)$ according to (6) for each $j \in \mathcal{N}_i(t)$.
- 19: Compute $z_i(t)$ according to (9).
- 20: Compute $e_{2,i}(t) = \hat{x}_i(t) - x_i(t)$.
- 21: Compute $u_i(t) = g_i^+(x_i(t))(k_1 z_i(t) + k_2 e_{2,i}(t))$.
- 22: **if** $\phi_4\|e_{2,i}(t)\|^2 \geq \phi_5\|z_i(t)\|^2 + \frac{\varepsilon}{N}$ **then**
- 23: Set $k(i) = k(i) + 1$. Set $t_{k(i)}^i = t$.
- 24: Set $\hat{x}_i(t) = x_i(t_{k(i)}^i)$.
- 25: Broadcast $x_i(t_{k(i)}^i)$ to all $j \in \mathcal{N}_i(t)$.
- 26: Broadcast $\zeta_{ij}(t_{k(i)}^i)$ to all $j \in \mathcal{N}_i(t)$.
- 27: **end if**
- 28: **if** agent $j \in \mathcal{N}_i(t) \cup \{0\}$ broadcasts **then**
- 29: Set $k(j) = k(j) + 1$. Set $t_{k(j)}^j = t$.
- 30: Receive $x_{j,1}(t_{k(j)}^j)$. Set $\hat{x}_j(t) = x_{j,1}(t_{k(j)}^j)$.
- 31: Measure $x_{j,2}(t_{k(j)}^j)$.
- 32: Receive $\zeta_{jp}(t_{k(j)}^j)$ for each $p \in \mathcal{N}_j(t)$.
- 33: **if** $\omega_{ij}(t_{k(j)}^j) \leq r$ and $\Delta t_{k(j)}^j \leq \Delta_j$ **then**
- 34: Compute $\Psi_{ij}(t_{k(j)}^j)$ according to (4) for each $t_{k(j)}^j \in S_j$.
- 35: **else**
- 36: Set $\Psi_{ij}(t_{k(j)}^j) = \vartheta$.
- 37: **end if**
- 38: Set $S_j = \{t_{k(j)}^j : t - t_{\text{reset}} \leq t_{k(j)}^j < t\}$.
- 39: Store $\{\Psi_{ij}(t_{k(j)}^j) : t_{k(j)}^j \in S_j\}$.
- 40: Compute $\tau_{ij}(t) = \frac{1}{|S_j|} \sum_{t_{k(j)}^j \in S_j} e^{-s_1 \Psi_{ij}(t_{k(j)}^j)}$.
- 41: **end if**
- 42: **end while**

REFERENCES

- [1] A. Adaldo *et al.*, "Event-triggered pinning control of switching networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 204–213, Jun. 2015.
- [2] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.
- [3] T. H. Cheng, Z. Kan, J. R. Klotz, J. M. Shea, and W. E. Dixon, "Event-triggered control of multi-agent systems for fixed and time-varying network topologies," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5365–5371, Oct. 2017.
- [4] X. Dong and G. Hu, "Time-varying formation control for general linear multi-agent systems with switching directed topologies," *Automatica*, vol. 73, pp. 47–55, 2016.
- [5] C. Nowzari and J. Cortés, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.
- [6] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.
- [7] W. Heemels, K. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE Conf. Decis. Control*, Dec. 2012, pp. 3270–3285.
- [8] E. Garcia, Y. Cao, and D. W. Casbeer, "Decentralized event-triggered consensus with general linear dynamics," *Automatica*, vol. 50, no. 10, pp. 2633–2640, 2014.
- [9] Y. Fan, L. Liu, G. Feng, and Y. Wang, "Self-triggered consensus for multi-agent systems with zeno-free triggers," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2779–2784, Oct. 2015.
- [10] Z. Feng and G. Hu, "Secure cooperative event-triggered control of linear multi-agent systems under DoS attacks," *IEEE Trans. Control Syst. Tech.*, vol. 28, no. 3, pp. 741–752, May 2020.
- [11] A. Sargolzaei, K. K. Yen, and M. N. Abdelghani, "Preventing time-delay switch attack on load frequency control in distributed power systems," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 1176–1185, Mar. 2015.
- [12] N. H. Vaidya, L. Tseng, and G. Liang, "Iterative approximate byzantine consensus in arbitrary directed graphs," in *Proc. ACM Symp. Prin. Distrib. Comput.*, 2012, pp. 365–374.
- [13] F. M. Zegers, P. Deptula, J. M. Shea, and W. E. Dixon, "Event-triggered approximate leader-follower consensus with resilience to byzantine adversaries," in *Proc. IEEE Conf. Decis. Control*, Nice, France, Dec. 2019, pp. 6412–6417.
- [14] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.
- [15] J. Usevitch and D. Panagou, "Resilient leader-follower consensus to arbitrary reference values," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 1292–1298.
- [16] S. M. Dibaji and H. Ishii, "Resilient consensus of second-order agent networks: Asynchronous update rules with delays," *Automatica*, vol. 81, pp. 123–132, 2017.
- [17] J. R. Klotz, A. Parikh, T.-H. Cheng, and W. E. Dixon, "Decentralized synchronization of uncertain nonlinear systems with a reputation algorithm," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 434–445, Mar. 2018.
- [18] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, pp. 1343–1349, 2008.
- [19] F. Zegers, M. Hale, J. M. Shea, and W. E. Dixon, "Reputation-based event-triggered formation control and leader tracking with resilience to byzantine adversaries," in *Proc. Annu. Amer. Control Conf.*, 2020, pp. 761–766.
- [20] A. V. Proskurnikov, A. S. Matveev, and M. Cao, "Opinion dynamics in social networks with hostile camps: Consensus vs. polarization," *IEEE Trans. Autom. Control*, vol. 61, no. 6, pp. 1524–1536, Jun. 2015.
- [21] W. Xia, M. Cao, and K. H. Johansson, "Structural balance and opinion separation in trust-mistrust social networks," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 46–56, Mar. 2016.
- [22] Z. Cai, M. S. de Queiroz, and D. M. Dawson, "A sufficiently smooth projection operator," *IEEE Trans. Autom. Control*, vol. 51, volume. 1, pp. 135–139, Jan. 2006.
- [23] R. Kamalapurkar, J. A. Rosenfeld, J. Klotz, R. J. Downey, and W. E. Dixon, "Supporting lemmas for rise-based control methods," 2014, *arXiv:1306.3432*.
- [24] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 73–82, Jan. 1987.
- [25] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, Sep. 1994.
- [26] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.
- [27] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in multi-agent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.



Federico M. Zegers received the B.S. degree in mechanical engineering and the B.S. degree in mathematics, and the M.S. degree in mechanical engineering from the University of Florida, Gainesville, FL, USA, in 2016, and 2019, respectively. He is currently working towards the Ph.D. degree in mechanical engineering under the supervision of Dr. Warren Dixon at the University of Florida.

His research interests include Lyapunov-based nonlinear and adaptive control, multi-agent systems, switched/hybrid systems theory, and robotics.



Matthew T. Hale (Member, IEEE) received the B.S.E. degree in electrical engineering *summa cum laude* from the University of Pennsylvania, Philadelphia, PA, USA, in 2012, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, in Atlanta, GA, USA, 2015 and 2017, respectively.

He is an Assistant Professor of Mechanical and Aerospace Engineering with the University of Florida, Gainesville, FL, USA. He directs the Control, Optimization, and Robotics Engineering (CORE) Lab, at the University of Florida. His research interests include multi-agent systems, mobile robotics, privacy in control, distributed optimization, and graph theory.

Dr. Hale was the recipient of the Teacher of the Year award in the UF Department of Mechanical and Aerospace Engineering for the 2018–2019 school year, and the recipient of the NSF CAREER Award in 2020 for his work on privacy in control systems.



John M. Shea (Member, IEEE) received the B.S. degree in computer engineering, and the M.S. and Ph.D. degrees in electrical engineering from Clemson University, Clemson, SC, USA, in 1993, 1995 and 1998, respectively.

In 1999, he joined as the faculty with the Department of Electrical and Computer Engineering, at the University of Florida, Gainesville, FL, USA, where he was promoted to the rank of Professor, in 2015. His research interests include wireless and military communications and networked autonomous systems.

Dr. Shea was the recipient of the Technical Achievement Award for contributions to military communications from the IEEE Military Communications Conference (MILCOM), in 2012, the Ellersick Award from the IEEE Communications Society for the Best Paper in the Unclassified Program of MILCOM, in 1996 and 2013, and the Outstanding Young Alumni award from the Clemson University College of Engineering and Science, in 2011. He was selected as a Finalist for the 2004 Eta Kappa Nu Outstanding Young Electrical Engineer. He has been an Editor for IEEE WIRELESS COMMUNICATIONS magazine, since 2010. He was an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, from 2008 to 2012, and was an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, from 2002 to 2007. He was the Technical Program Chair for the Unclassified Program of MILCOM, in 2010.



Warren E. Dixon (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering from Clemson University, Clemson, SC, USA, in 2000.

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 Mechanical and Aerospace Engineering Department, the University of Florida, Gainesville, FL, USA, where he currently is the Newton C.

Ebaugh Professor and Department Chair. His research interests include the development and application of Lyapunov-based control techniques for uncertain nonlinear systems.

Dr. Dixon is an ASME and IEEE Fellow for contributions to adaptive control of uncertain nonlinear systems. He was the recipient of various career and best paper awards.