DATA-BASED LEARNING FOR UNCERTAIN NONLINEAR SYSTEMS WITH INTERMITTENT STATE FEEDBACK

By

RUNHAN SUN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2022

To Gailong Sun, Yanhua Wang, and Bingyu Li

ACKNOWLEDGEMENTS

I thank my research advisor, Dr. Warren Dixon, for his consistent patience, guidance, and encouragement throughout my Ph.D. career. Dr. Dixon not only spends time and efforts in training me to become a researcher in controls, but also provides me many suggestions in making career path and life decisions. I thank my committee members Dr. Matthew Hale, Dr. Carl Crane, and Dr. Dapeng Wu for their guidance and oversight. I thank the Nonlinear Controls and Robotics group for all the joys and unforgettable memories.

I would also like to thank the National Science Foundation, the Office of Naval Research, the Air Force Office of Scientific Research, and the Naval Engineering Education Consortium. This dissertation is supported in part by NSF under award numbers 1509516 and 1762829, ONR under grant numbers N00014-21-1-2481 and N00014-13-1-0151, AFOSR under award numbers FA9550-18-1-0109 and FA9550-19-1-0169, NEEC under award number N00174-18-1-0003.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figures page

LIST OF ABBREVIATIONS

a.e.	Almost Everywhere
AI	Artificial Intelligence
AtLoc	Attention guided camera Localization
CL	Concurrent Learning
CNN	Convolutional Neural Network
D-MRAC	Deep Model Reference Adaptive Control
DNN	Deep Neural Network
FE	Finite Excitation
FOV	Field-of-View
GPS	Global Positioning System
GUUB	Globally Uniformly Ultimately Bounded
ICL	Integral Concurrent Learning
IMU	Inertial Measurement Unit
LSTM	Long Short-Term Memory
MAS	Multi-Agent System
ML	Machine Learning
MSE	Mean Squared Error
MVT	Mean Value Theorem
PE	Persistent Excitation
ResNet	Residual Network
RMSE	Root Mean Squared Error
SGD	Stochastic Gradient Descent

SLAM Simultaneous Localization and Mapping

UAV Unmanned Aerial Vehicle

UUV Unmanned Underwater Vehicle

WMR Wheeled Mobile Robot

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

DATA-BASED LEARNING FOR UNCERTAIN NONLINEAR SYSTEMS WITH INTERMITTENT STATE FEEDBACK

By

Runhan Sun

May 2022

Chair: Warren E. Dixon
Major: Mechanical Engineering

In many autonomous applications, nonlinear dynamical systems experience intermittent state information due to various reasons, e.g., the operating environment, task definition, system constraint. To achieve the autonomous control objective, various switched system approaches have been investigated under intermittent state information. A generalized switched systems consists of feedback available and feedback unavailable subsystems. The autonomous agent's state can be switched between these subsystems. While state information is available, a controller can be developed using true state information. While state information is unavailable, state predictors or observers can be designed to predict the state using estimated information. When the agent's state is switched from a feedback unavailable to a feedback available subsystem, the estimated state will be reset to the true state to compensate for the state estimation error. By using a switched systems approach to design controllers, observers, and adaptation laws, the autonomous agent is enable to achieve the control objective under intermittent state information.

Various control applications may require an autonomous system to operate in completely unknown or adversarial environments. Recent advancements in machine learning (ML) and artificial intelligence (AI) can potentially address the challenges posed by unknown and adversarial environments. Specifically, ML/AI-based tools such as neural networks (NNs) are useful for learning and adapting to unknown environments. NNs are universal function approximators that are capable of approximating continuous functions. Therefore, the universal

approximation property of NNs is typically used to approximate functions representing the dynamics of an autonomous system. Recent advancements in ML/AI show that depth of the network structure provides significant improvement in the function approximation performance of a NN. Although deep neural network (DNN)-based control methods facilitate task execution in unknown environments, the learning schemes for ML/AI augmented autonomous control systems are often implemented in an open-loop manner, based on pre-trained networks that may not represent the operational environment. It is well-known that open-loop approaches lack stability, robustness, and convergence guarantees that are afforded by closed-loop feedback control schemes. For safety-critical applications, there is a need to balance adaptation, safety, robustness, and stability.

Chapter 2 provides a generalized switched systems technique called the relay-explorer approach, to allow a relay agent to intermittently provide navigational feedback information to an explorer leader for nonlinear multi-agent systems (MASs). A distributed controller is developed for formation control and leader tracking for the explorer followers, enabling a MAS to explore an unknown environment indefinitely. To compensate for the lack or inability to use navigational sensors, state observers are used to propagate state estimates for the relay and explorer agents (e.g., in Global Positioning System (GPS)-denied regions). Stabilizing dwell-time conditions are determined via a Lyapunov-based switched systems approach to ensure the trajectory tracking errors are globally uniformly ultimately bounded (GUUB) defined by user-defined thresholds. Using the developed approach, a MAS's trajectory tracking error remains bounded, enabling the exploration of GPS-denied regions for a predetermined period of time, before acquiring navigational feedback from a relay agent.

Chapter 3 provides a method to enable a nonholonomic agent to explore an unknown environment with intermittent state feedback. A maximum stabilizing dwell-time condition is determined via a Lyapunov-based switched systems approach to maintain overall system stability despite the intermittent loss of state feedback and the presence of external disturbances. A minimum stabilizing dwell-time condition is determined via a Lyapunov-based switched systems

approach to ensure the tracking error converges within a desired neighborhood of the desired trajectory. Using the developed stabilizing maximum and minimum dwell-time conditions, a nonholonomic agent's tracking error is proven to be GUUB, enabling the exploration of the feedback-denied region for a predetermined period of time, before acquiring state feedback.

Chapter 4 provides a real-time DNN adaptive control architecture for general uncertain nonlinear dynamical systems to track a desired time-varying trajectory. A Lyapunov-based method is leveraged to develop adaptation laws for the output-layer weights of a DNN model in real-time while a data-driven supervised learning algorithm is used to update the inner-layer weights of the DNN. Specifically, the output-layer weights of the DNN are estimated using an unsupervised learning algorithm to provide responsiveness and guaranteed tracking performance with real-time feedback. The inner-layer weights of the DNN are trained with collected data sets to increase performance, and the adaptation laws are updated once a sufficient amount of data is collected. Building on the results in [1] and [2], which focus on deep model reference adaptive control (D-MRAC) for linear systems with known drift dynamics and control effectiveness matrices, this chapter considers general control-affine uncertain nonlinear systems. The real-time controller and adaptation laws enable the system to track a desired time-varying trajectory while compensating for the unknown drift dynamics and parameter uncertainties in the control effectiveness. A nonsmooth Lyapunov-based analysis is used to prove semi-global asymptotic tracking of the desired trajectory.

Chapter 5 concludes the dissertation and presents potential extensions to the work presented in the previous chapters.

CHAPTER 1
INTRODUCTION

1.1 Background

Robotic systems are often challenged by the temporary or permanent loss of navigational feedback, which can arise due to environmental and physical constraints or from the task definition. For instance, unmanned underwater vehicles (UUVs) are often used to conduct surveying or reconnaissance operations, requiring the UUVs to navigate using inertial measurement units (IMUs) because absolute position and orientation sensing is unavailable underwater. However, disturbances in the dynamics and measurement noise from the ambient environment can cause IMU-based state estimates to drift over time. To compensate for the accumulated error, UUVs are required to intermittently surface to obtain absolute position and orientation information, increasing the time and energy needed to accomplish the operational objectives.

Numerous results have been developed to provide robustness to intermittent feedback [3]-[15]. For example, [8]-[11] used event-triggered approaches to allow the agents to limit communication. However, these approaches determine when to communicate based on triggering conditions which is not possible while exploring an unknown environment due to additional feedback constraints determined by the agents' position in the environment. Specifically, state feedback is only available when an agent is contained in a feedback region, and hence, the agent must travel to designated feedback regions periodically to acquire state feedback while exploring the unknown environment.

In lieu of requiring the UUVs to intermittently surface to obtain absolute position and orientation information in the previous example, a network of UUVs, consisting of relay and explorer agents, can be used to address this issue. Specifically, a relay agent can be tasked with surfacing to acquire absolute position and orientation information, then traveling back to the explorer network to provide estimated navigational feedback. For some applications, the navigational sensors can be replaced with low cost or less capable sensors or even removed. Such scenarios motivated the development of a class of relay-explorer problems.

Relay-explorer problems indicate that one dynamic system (i.e., relay agent) is

intermittently providing state feedback to another dynamic system (i.e., explorer agent). Specifically, the results in [16]-[18] develop a control method that is robust to intermittent loss of state feedback, guaranteeing that an agent will return to a previously occupied feedback region, after operating in a sensors feedback-denied region. The result in [16] yields trajectory tracking for a nonholonomic system with intermittent state feedback, by developing a set of dwell-time conditions via a Lyapunov-based switched systems approach. Specifically, the developed maximum dwell-time condition dictates the maximum time the agent can operate without state feedback, while a minimum dwell-time condition dictates the minimum time the agent must dwell within a feedback region to compensate for the accumulated error. By satisfying the developed dwell-time conditions in [16], the tracking error is proven to remain globally uniformly ultimately bounded (GUUB). In [17], the same objective of following a desired trajectory is achieved for a holonomic system with intermittent state feedback. Both results require using an observer adaptation law with a high-frequency sliding-mode term. A reset map is introduced in [18], eliminating the need for an observer that requires continuous state feedback for a holonomic system. Specifically, the state estimate is reset to the true state whenever the agent has feedback, allowing the holonomic agent to operate in a sensors feedback-denied region for an extended period of time.

In [19], a nonholonomic agent is allowed to temporarily navigate outside of a set of feedback regions and explore an unknown environment, provided the agent tracks an auxiliary trajectory that enables the agent to satisfy dwell-time conditions. The dwell-time conditions are obtained from a Lyapunov-based switched systems approach, without the requirement of using an observer that requires continuous state feedback. With the availability of multiple feedback regions, the nonholonomic agent can further explore the sensors feedback-denied region while satisfying the dwell-time condition.

MASs are motivated to explore an unknown environment more rapidly, but coordinating the agents with intermittent feedback is challenging. In [20], a method for regulating the position of multiple follower agents is developed for operating within a sensors feedback-denied region,

where a leader agent with global feedback sensors, visits each of the follower agents to update the agent's state estimate. However, this approach requires the leader to know the full state feedback of itself and requires the leader to communicate with each individual follower.

In [21], a relay agent is tasked with switching between feedback available and feedback-denied regions, while intermittently providing state estimates to an explorer leader. This strategy allows the explorer network to operate indefinitely in the sensors feedback-denied region, where a distributed formation controller regulates the relative distances between the explorer leader and follower agents, improving coverage of the unknown environment. The design is complicated by the fact that the relay agent does not have navigational-feedback once it enters the sensors feedback-denied region, and hence, the relay agent has to rely on its state estimate to reach the explorer leader and return to the feedback region before the accumulated error exceeds a user-defined threshold. Another challenge is the relay agent only communicates with one exploring agent, i.e., the explorer leader. Thus, a decentralized leader-follower formation controller is required to maintain the formation of the explorer agents. To ensure the stability of the overall system, a Lyapunov-based switched systems approach is used to develop dwell-time conditions to determine the maximum time the relay agent can remain in the sensors feedback-denied region and the maximum time the explorer agents are allowed to operate without a state estimate update. A distributed controller for the explorer followers is developed, ensuring the explorer agents maintain a formation while exploring the unknown environment.

In addition to holonomic systems, regulation and tracking control of nonholonomic systems have been heavily studied (cf., [22]-[24]) due to their wide range of applications (e.g., exploration of unknown environment and autonomous driving). In many cases, nonholonomic constraints, the operating environment, or task definition could result in temporary loss of state feedback. For example, cameras are a popular and extensively studied sensor capable of providing state feedback. Various results (cf., [25]-[33]) have investigated methods to measure distance and viewing angles by keeping the feature points of landmarks in the field-of-view (FOV) while an agent performs a task. Keeping landmarks in the FOV of systems with nonholonomic constraints

is challenging and may result in limited, sharp-angled, or non-smooth trajectories as illustrated in [34] and [35]. Furthermore, in many scenarios, the desired trajectory may not be in a region where a specific landmark is located (e.g., unknown environment exploration), motivating the development of methods that are robust to intermittent loss of feature-based state feedback.

Previous results have investigated the guidance, navigation, and control of nonholonomic systems under intermittent state feedback. Results such as [36] employ a daisy-chaining approach that utilizes multiple landmarks within the FOV to establish a relationship between the moving camera frame and the landmark frame via coordinate transformations, which allows the previous landmark to leave the FOV but maintain estimated state information to expand the operating domain. However, the estimation error accumulates over time due to disturbances in the dynamics and measurement noise from the ambient environment. Without error corrections, the agent will deviate from a desired trajectory. Similarly, numerous simultaneous localization and mapping (SLAM) approaches (cf., [37]-[41]) assume an environment is feature-rich, and typically generate relationships between landmarks to estimate the state (e.g., position and orientation) of the system. However, SLAM algorithms also accumulate error caused by measurement noise over time unless a loop-closure can compensate for the accumulated error. Additionally, returning to a landmark before the state error exceeds a threshold is not guaranteed, implying the agent may still diverge from a desired trajectory.

To ensure loop-closures are achievable, the results in [16]-[18] guarantee an agent can return to a single feedback region after entering the feedback-denied region. In comparison to the previous methods, [19] enables a nonholonomic agent to temporarily navigate outside of a set of feedback regions and explore an unknown environment provided the agent tracks a desired trajectory that enables the agent to satisfy dwell-time conditions, which are obtained from a Lyapunov-based switched systems approach, without the requirement of using an observer that requires continuous state feedback. While the trajectory design in [18] is limited to a holonomic vehicle and a single feedback zone, in [19], multiple feedback regions are established for the nonholonomic vehicle to receive feedback and a reset map is utilized to reset the state estimation

error. With the availability of multiple feedback regions, the nonholonomic vehicle can further explore the feedback-denied region while satisfying the dwell-time conditions. In addition, the development in [19] includes timing conditions from a switched systems approach that ensures the nonholonomic vehicle can reach one of the feedback regions ensuring the agent achieves loop-closure.

Various control applications may require an autonomous system to operate in completely unknown or adversarial environments. The universal approximation property of neural networks (NNs) is typically used to approximate functions representing the dynamics of an autonomous system. Deep neural network (DNN) methods can capture complex features of the dynamics by using back-propagation algorithms that indicate how to update the inner-layer weights [42]. In results such as [42] and [43], the emergence of DNN models with more complex structures improve function approximation performance. Although DNN function approximation methods show improved performance empirically, these methods typically lack performance guarantees because the accuracy of the outputs are probabilistic. As a result, DNN-based methods may have limited adoption for safety-critical applications.

Motivated to ensure performance guarantees, early works in [44]-[47] use Lyapunov-based methods for NN-based adaptive control of unknown nonlinear systems. In [44]-[46], NNs are trained with a gradient descent-based adaptive update law and used as a feedforward control term. Since the update laws are derived from a stability analysis and the NN weights are embedded inside activation functions, it is challenging to derive adaptation laws from a stability analysis beyond a single-hidden-layer.

In [48] and [49], the authors developed a data-driven adaptive learning method called concurrent learning (CL) to increase performance of parameter estimation. CL leverages recorded input and output data concurrent to real-time execution to apply batch-like updates to adaptive update laws, and has been extended to works in [50]-[52]. Results in [1], [2] and [53] leverage CL to develop a deep model reference adaptive control (D-MRAC). Specifically, a gradient descent-based adaptive update law is used to estimate the ideal output-layer weights of a DNN in

real-time online, and an offline data-driven method is used to apply batch updates to the inner-layer weights of the DNN for linear systems with known system matrices. The methods were tested on quadrotors and demonstrated that DNN-based adaptive control can significantly improve learning performance [2, 53]. The authors demonstrated that DNN enabled model reference adaptive control (MRAC) outperforms shallow NN MRAC, and also showed that the DNN weights cluster in different regions in different operating envelopes of the quadrotor, clearly establishing the learning performance of DNNs [53]. However, the D-MRAC development is specific to linear systems with known system A, B matrices with matched system uncertainty $\Delta(x(t))$, i.e., $\dot{x}(t) = Ax(t) + B(u(t) + \Delta(x(t)))$.

Building on the output-layer weight adjustment strategy in [1] and [2], a new control design and stability analysis method is developed for general uncertain nonlinear systems [54]. A Lyapunov-based adaptive control law is developed to estimate the unknown output-layer weights of the DNN using real-time state feedback. Concurrent to real-time execution, data is collected and an offline function approximation method is used to update the estimates of the inner-layer DNN weights. Moreover, control-affine dynamics with uncertain state-dependent control effectiveness matrices are considered. To compensate for the uncertain control effectiveness, a novel adaptive update law is developed that has internal feedback. Specifically, the adaptive update law depends on the control input, and hence, is a function of both the input uncertainty estimates and the DNN weight estimates. To account for switching from iterative updates of the DNN weights, a nonsmooth Lyapunov-based analysis is performed to ensure asymptotic tracking of the desired trajectory.

1.2 Outline of the Dissertation

Chapter 2 develops a method for MAS to allow a relay agent to intermittently provide navigational feedback to an explorer leader. The developed approach enables the network of explorer agents to explore an unknown environment where navigational feedback is unavailable. The explorer agents follow the explorer leader, which is following a desired trajectory through the unknown environment, and intermittently receiving state estimate information from the relay

agent. The relay agent is intermittently receiving state information from regions where navigational-feedback is available and then communicating with the leader of the explorer agents to provide state estimate information when the explorer leader is within the communication radius of the relay agent.

The contributions in Chapter 2 include developing a method for MAS to explore a sensors feedback-denied environment. Specifically, agents can follow the desired trajectory within the feedback-denied environment by only using state estimates. To compensate the state estimation error of the explorer agents, a switched system approach is used to enable a relay agent to provide state estimates to an explorer leader, which prevents the explorer leader deviating away from the desired trajectory. A distributed formation controller using state estimates is developed to enable the explorer follower agents to maintain a formation with respect to the explorer leader while exploring the environment, which significantly increases the exploring coverage compared to a single agent. To ensure the objective is achievable, a switched systems approach is also used for the relay agent. The relay agent can travel between a feedback available and feedback-denied regions, and the state estimate can be updated with the true state while it is within a feedback available region. With the developed method, an MAS can explore a feedback-denied environment indefinitely.

Chapter 3 enables a nonholonomic agent to explore an unknown environment, where feedback is unavailable by intermittently following a trajectory through the unknown environment and intermittently into regions, where feedback is available. A maximum stabilizing dwell-time condition is determined via a Lyapunov-based switched systems approach to maintain overall system stability despite the intermittent loss of state feedback and the presence of external disturbances. A minimum stabilizing dwell-time condition is determined via a Lyapunov-based switched systems approach to ensure the tracking error converges within a desired neighborhood of the desired trajectory. Using the proposed maximum and minimum dwell-time conditions, a nonholonomic agent's tracking error is GUUB, enabling the exploration of the feedback-denied region for a predetermined period of time, before acquiring state feedback.

The contributions in Chapter 3 include developing a set of stabilizing dwell-time conditions using a Lyapunov-based switched systems approach to ensure stability of the system while the nonholonomic agent is exploring the unknown environment. Specifically, a maximum dwell-time condition is developed to allow the nonholonomic agent to explore the feedback-denied environment for a predetermined period of time, and a minimum dwell-time condition is developed to compensate for the state estimation error to ensure the trajectory tracking error converges within a desired neighborhood of the desired trajectory. By satisfying these stabilizing dwell-time conditions, the trajectory tracking error of the nonholonomic agent is shown to be bounded by a user-defined threshold despite the intermittent loss of state feedback and the presence of external disturbances.

In Chapter 4, a real-time DNN adaptive control architecture is developed for general uncertain nonlinear dynamical systems to track a desired time-varying trajectory. A Lyapunov-based method is leveraged to develop adaptation laws for the output-layer weights of a DNN model in real-time while a data-driven supervised learning algorithm is used to update the inner-layer weights of the DNN. Specifically, the output-layer weights of the DNN are estimated using an unsupervised learning algorithm to provide responsiveness and guaranteed tracking performance with real-time feedback. The inner-layer weights of the DNN are trained with collected data sets to increase performance, and the adaptation laws are updated once a sufficient amount of data is collected. The real-time controller and adaptation laws enable the system to track a desired time-varying trajectory while compensating for the unknown drift dynamics and parameter uncertainties in the control effectiveness.

The contributions in Chapter 4 include developing a multiple timescale learning DNN adaptive control architecture for general uncertain nonlinear dynamical systems. Specifically, the unknown drift dynamics are approximated using a universal function approximator (i.e., a feedforward fully connected DNN). A Lyapunov-based real-time adaptation law is developed to update the output-layer weights of the DNN, and a batch optimization (i.e., minimize the mean squared error (MSE)) is used to periodically update the inner-layer weights of the DNN. The

output-layer weights are updated continuously to ensure system stability, while the inner-layer weights are updated when a sufficient number of data are collected and trained to improve the approximation of the DNN. The multiple timescale learning adaptive control architecture enables the general uncertain nonlinear dynamical systems to track a desired trajectory, while using a DNN to iteratively improve the control performance.

Chapter 5 concludes the dissertation and presents potential extensions to the work presented in the previous chapters.

1.3 Preliminaries

1.3.1 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 $p \in \mathbb{Z}_{>0}$. The $p \times p$ identity matrix is denoted by I_p. The Euclidean norm of a vector $m \in \mathbb{R}^p$ is denoted by $\|m\| \triangleq \sqrt{m^{\mathrm{T}} m}$, and the absolute-value of a scalar $n \in \mathbb{R}$ is denoted by $|n|$. The notation $\lambda_{\min}\{\cdot\}$ and $\lambda_{\max}\{\cdot\}$ denote the minimum and maximum eigenvalues of $\{\cdot\}$, respectively. The symbol $\mathcal{L}_\infty$ denotes the set of essentially bounded measurable functions, i.e., given the Lebesgue measurable function $f : \mathbb{R} \to \mathbb{R}$, $f \in \mathcal{L}_\infty$ if and only if $\inf\{C \geq 0 : |f(x)| \leq C \text{ for almost every } x \in \mathbb{R}\} \in \mathbb{R}_{\geq 0}$.

1.3.2 Graphs

Let $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ be an undirected graph with node set $\mathcal{V} \triangleq \{1, 2, ..., N\}$ and undirected edge set $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$, where $N \in \mathbb{Z}_{>0}$. The neighbor set of node i is denoted by $\mathcal{N}_i \subset \mathcal{V}$. An undirected edge is defined as an unordered pair (i, j) where $(i, j) \in \mathcal{E}$. Note that $(i, j) \in \mathcal{E}$ implies agent j can obtain information from agent i. An undirected path is a sequence of undirected edges in $\mathcal{E}$. An undirected graph is called connected if and only if there exists an undirected path between any two distinct nodes. There are no self-loops or repeated edges in graph $\mathcal{G}$. The adjacency matrix is defined as $\mathcal{A} \triangleq [a_{ij}] \in \mathbb{R}_{\geq 0}^{N \times N}$, where $a_{ij} \triangleq a_{ji} \geq 0$ for all $i \neq j$ and $a_{ii} \triangleq 0$. The degree matrix of the undirected graph $\mathcal{G}$ is defined as a diagonal matrix such that $\Delta \triangleq [\Delta_{ij}] \in \mathbb{R}_{\geq 0}^{N \times N}$, where $\Delta_{ij} \triangleq 0$ for all $i \neq j$ and $\Delta_{ii} \triangleq \sum_{j=1}^{N} a_{ij}$. The graph Laplacian $L \in \mathbb{R}^{N \times N}$ of the undirected graph $\mathcal{G}$ is defined as $L \triangleq \Delta - \mathcal{A}$.

CHAPTER 2
RELAY-EXPLORER APPROACH FOR MULTI-AGENT EXPLORATION OF AN UNKNOWN ENVIRONMENT WITH INTERMITTENT COMMUNICATION

A relay-explorer control method for nonlinear MASs developed in [21] is described in this chapter to allow a relay agent to intermittently provide navigational feedback to an explorer leader. The contributions in this chapter include developing a method for MAS to explore a sensors feedback-denied environment. Specifically, agents can follow the desired trajectory within the feedback-denied environment by only using state estimates. To compensate for the state estimation error of the explorer agents, a switched system approach is used to enable a relay agent to provide state estimates to an explorer leader, which prevents the explorer leader from deviating away from the desired trajectory. A distributed formation controller using state estimates is developed to enable the explorer follower agents to maintain a formation with respect to the explorer leader while exploring the environment, which significantly increases the exploring coverage compared to a single agent. To ensure the objective is achievable, a switched systems approach is also used for the relay agent. The relay agent can travel between a feedback available and feedback-denied regions, and the state estimate can be updated with the true state while it is within a feedback available region. To compensate for the lack or inability to use navigational sensors, state observers are used to propagate state estimates for the relay and explorer agents (e.g., in sensors feedback-denied regions). Stabilizing dwell-time conditions are determined via a Lyapunov-based switched systems approach to ensure the trajectory tracking errors are GUUB defined by user-defined thresholds. Using the developed approach, a MAS's trajectory tracking error remains bounded, enabling the exploration of a sensors feedback-denied region for a predetermined period of time, before acquiring navigational feedback from a relay agent.

2.1 System Model

Consider a MAS with one relay agent and $N+1$ explorer agents, where $N \in \mathbb{Z}_{>0}$.

2.1.1 Relay Agent

A control affine nonlinear dynamic system for the relay agent is described as

$$\dot{x}_r(t) \triangleq f_r(x_r(t)) + v_r(x_r(t)) + d_r(t), \tag{2-1}$$

where $t_0 \in \mathbb{R}_{\geq 0}$ is the initial time. In (2-1), $x_r, \dot{x}_r : [t_0, \infty) \to \mathbb{R}^n$ denote the generalized state and its first order time-derivative with $n \in \mathbb{Z}_{>0}$, $f_r : \mathbb{R}^n \to \mathbb{R}^n$ denotes the drift dynamics, $v_r : \mathbb{R}^n \to \mathbb{R}^n$ is the control input, and $d_r : [t_0, \infty) \to \mathbb{R}^n$ is an exogenous disturbance acting on the relay agent.

2.1.2 Explorer Agents

The subscript 0 will be used to denote the leader, which is considered to be independent of the follower graph structure. The followers that are connected to the leader can continuously receive information from the leader without latency.

Consider a multi-agent explorer system consisting of the single leader indexed by 0 and the set of N follower agents contained in $\mathcal{V}$. The nonlinear dynamic system for agent $i \in \mathcal{V} \bigcup \{0\}$ is described as

$$\dot{x}_i(t) \triangleq f_e(x_i(t)) + v_i(x_i(t)) + d_i(t), \tag{2-2}$$

where $x_i, \dot{x}_i : [t_0, \infty) \to \mathbb{R}^n$ denote the generalized state of the i^{th} explorer agent and its first order time-derivative, respectively, $f_e : \mathbb{R}^n \to \mathbb{R}^n$ denotes the explorer drift dynamics, $v_i : \mathbb{R}^n \to \mathbb{R}^n$ is the control input, and $d_i : [t_0, \infty) \to \mathbb{R}^n$ is an exogenous disturbance acting on the i^{th} explorer agent.

Assumption 2.1. The graph $\mathcal{G}$ is connected for all $t \in [t_0, \infty)$.

Assumption 2.2. The exogenous disturbances acting on the relay and explorer agents are upper bounded by known positive constants, i.e., $\|d_r(t)\| \leq \bar{d}_r \in \mathbb{R}_{>0}$ and $\|d_i(t)\| \leq \bar{d}_i \in \mathbb{R}_{>0}$ for all $t \in [t_0, \infty)$.

Assumption 2.3. The drift dynamics $f_r(x_r(t))$ of the relay agent defined in (2-1) and the drift dynamics $f_e(x_i(t))$ of the i^{th} explorer agent defined in (2-2) are known, locally Lipschitz, and bounded given a bounded argument (see [55]).

Assumption 2.4. The communication radius of the relay agent is defined as $R_{\text{com}} \in \mathbb{R}_{>0}$. The relay and explorer leader can communicate when the measurable relative distance, defined as $x_{r0}(t) \triangleq x_0(t) - x_r(t)$, is within communication range, i.e., $\|x_{r0}(t)\| \leq R_{\text{com}}$.

2.2 Control Design

2.2.1 Control Objective

The objective is to enable the network of explorer agents to explore an unknown environment where navigational-feedback is unavailable. The explorer agents follow the explorer leader, which is following a desired trajectory through the unknown environment, and intermittently receiving state estimate information from the relay agent. The relay agent is intermittently receiving state information from regions where navigational-feedback is available and then communicating with the leader of the explorer agents to provide state estimate information when $\|x_{r0}(t)\| \leq R_{\text{com}}$. Once the explorer leader receives state estimate information from the relay agent, the follower agent's state estimate is updated by the explorer leader because the network is connected.

Assumption 2.5. There exists multiple feedback regions, located outside of the unknown environment, which are known and stationary. Each of the feedback regions is defined by a compact set $\mathcal{F}_k \subset \mathbb{R}^n$, $k \in \mathcal{O} \triangleq \{1,2,...,O\}$, where $O \in \mathbb{Z}_{>0}$ denotes the number of regions.

The zone outside of all the feedback regions is defined as $\mathcal{F}^u \subset \mathbb{R}^n$, where $\mathcal{F}^u \bigcap \mathcal{F} = \{\emptyset\}$, and the union of the feedback regions is defined as $\mathcal{F} \triangleq \bigcup_{k \in \mathcal{O}} \mathcal{F}_k$. State feedback is available if and only if $x_r(t) \in \mathcal{F}$.

Assumption 2.6. The relay agent is initialized in a feedback region, i.e., $x_r(t_0) \in \mathcal{F}$, while the explorer agents may be initialized outside of all the feedback regions, i.e., $x_i(t_0) \in \mathcal{F}^u$, with $x_i(t_0)$ available to explorer agent i.

Assumption 2.7. The control and estimated position of the explorer leader are bounded, i.e., $\|v_0(\hat{x}_0(t))\| \leq \bar{v}_0 \in \mathbb{R}_{>0}$ and $\|\hat{x}_0(t)\| \leq \bar{\hat{x}}_0 \in \mathbb{R}_{>0}$ for all $t \geq 0$ (see [55]).

Definition 2.1. Let $\phi_r : [t_0, \infty) \to \{a_r, u_r\}$ be a piece-wise constant switching signal for the relay agent, where a_r indicates $x_r(t) \in \mathcal{F}$ and u_r indicates $x_r(t) \in \mathcal{F}^u$. Let $t_l^a \in \mathbb{R}_{\geq t_0}$ denote the l^{th} time instance when $\phi_r(t) = a_r$, where $l \in \mathbb{Z}_{\geq 0}$. Let $t_l^u \in \mathbb{R}_{>t_0}$ denote the l^{th} time instance when

$\phi_r(t) = u_r$. The dwell-time of the l^{th} instance of each subsystem is defined as $\Delta t_l^a \triangleq t_l^u - t_l^a \in \mathbb{R}_{>0}$, and $\Delta t_l^u \triangleq t_{l+1}^a - t_l^u \in \mathbb{R}_{>0}$, respectively.

Definition 2.2. Let $\phi_e : [t_0, \infty) \to \{a_e, u_e\}$ be a piece-wise constant switching signal for the explorer leader, where a_e indicates that the explorer leader has received state estimate information from the relay agent, and u_e indicates that communication between the explorer leader and the relay agent has been lost. Let $t_m^a \in \mathbb{R}_{>t_0}$ denote the m^{th} time instance when $\phi_e(t) = a_e$, where $m \in \mathbb{Z}_{\geq 0}$. Let $t_m^u \in \mathbb{R}_{>t_0}$ denote the m^{th} time instance when $\phi_e(t) = u_e$. The dwell-time of the m^{th} instance of each subsystem is defined as $\Delta t_m^a \triangleq t_m^u - t_m^a \in \mathbb{R}_{>0}$, and $\Delta t_m^u \triangleq t_{m+1}^a - t_m^u \in \mathbb{R}_{>0}$, respectively.

Based on the control objective, three error systems are defined for the relay agent as [18]

$$e_r(t) \triangleq x_r(t) - x_{rd}(t), \tag{2-3}$$

$$\hat{e}_r(t) \triangleq \hat{x}_r(t) - x_{rd}(t), \tag{2-4}$$

$$\tilde{e}_r(t) \triangleq x_r(t) - \hat{x}_r(t), \tag{2-5}$$

where $x_{rd} : [t_0, \infty) \to \mathbb{R}^n$ denotes the desired trajectory for the relay agent, $\hat{x}_r : [t_0, \infty) \to \mathbb{R}^n$ denotes the state estimate of $x_r(t)$, $e_r : [t_0, \infty) \to \mathbb{R}^n$ is the trajectory tracking error, $\hat{e}_r : [t_0, \infty) \to \mathbb{R}^n$ is the estimated tracking error, and $\tilde{e}_r : [t_0, \infty) \to \mathbb{R}^n$ is the state estimation error of the relay agent, respectively. When $\phi_r(t) = u_r$, state feedback for the relay agent is unavailable, and an observer is introduced based on (2-1) as

$$\dot{\hat{x}}_r(t) \triangleq f_r(\hat{x}_r(t)) + v_r(\hat{x}_r(t)), \tag{2-6}$$

where $\dot{\hat{x}}_r : [t_0, \infty) \to \mathbb{R}^n$ denotes the estimate of $\dot{x}_r(t)$.

Similarly, three error systems are defined for the explorer leader as

$$e_0(t) \triangleq x_0(t) - x_{0d}(t), \tag{2-7}$$

$$\hat{e}_0(t) \triangleq \hat{x}_0(t) - x_{0d}(t), \tag{2-8}$$

$$\tilde{e}_0(t) \triangleq x_0(t) - \hat{x}_0(t), \tag{2-9}$$

where $x_0 : [t_0, \infty) \to \mathbb{R}^n$ denotes the generalized state for the explorer leader, $\hat{x}_0 : [t_0, \infty) \to \mathbb{R}^n$ denotes the state estimate of $x_0(t)$, $x_{0d} : [t_0, \infty) \to \mathbb{R}^n$ denotes the desired trajectory for the explorer leader, $e_0 : [t_0, \infty) \to \mathbb{R}^n$ is the trajectory tracking error, $\hat{e}_0 : [t_0, \infty) \to \mathbb{R}^n$ is the estimated tracking error, and $\tilde{e}_0 : [t_0, \infty) \to \mathbb{R}^n$ is the state estimation error of the explorer leader, respectively. When $\phi_e(t) = u_e$, state information for the explorer leader from the relay agent is unavailable, and an observer is introduced based on (2-2) as

$$\dot{\hat{x}}_i(t) \triangleq f_e(\hat{x}_i(t)) + v_i(\hat{x}_i(t)), \tag{2-10}$$

where $\dot{\hat{x}}_i : [t_0, \infty) \to \mathbb{R}^n$ denotes the estimate of $\dot{x}_i(t)$ and $\hat{x}_i : [t_0, \infty) \to \mathbb{R}^n$ denotes the estimate of $x_i(t)$. Similarly, an observer for the explorer leader is defined as

$$\dot{\hat{x}}_0(t) \triangleq f_e(\hat{x}_0(t)) + v_0(\hat{x}_0(t)). \tag{2-11}$$

To achieve the formation control and leader tracking objective for the explorer agents, an auxiliary estimate, $\hat{e}_i : [t_0, \infty) \to \mathbb{R}^n$ is defined for the i^{th} explorer agent as

$$\hat{e}_i(t) \triangleq \hat{x}_i(t) - \hat{x}_0(t) - p_i, \tag{2-12}$$

where $p_i \in \mathbb{R}^n$ denotes the desired relative position between the explorer leader and the i^{th} explorer follower. Each p_i is fixed and each explorer follower knows p_i for all $i \in \mathcal{V}$, i.e., each follower knows the desired formation. The stacked form of (2-12) is defined as $\widehat{E} \triangleq \left[\hat{e}_1^{\mathrm{T}}(t), \hat{e}_2^{\mathrm{T}}(t), ..., \hat{e}_N^{\mathrm{T}}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{nN}$.

Assumption 2.8. The signals $x_{rd}(t)$, $\dot{x}_{rd}(t)$, $x_{0d}(t)$, and $\dot{x}_{0d}(t)$ are bounded such that $\|x_{rd}(t)\| \leq \bar{x}_{rd} \in \mathbb{R}_{>0}$, $\|\dot{x}_{rd}(t)\| \leq \bar{\dot{x}}_{rd} \in \mathbb{R}_{>0}$, $\|x_{0d}(t)\| \leq \bar{x}_{0d} \in \mathbb{R}_{>0}$, and $\|\dot{x}_{0d}(t)\| \leq \bar{\dot{x}}_{0d} \in \mathbb{R}_{>0}$ for all $t \in [t_0, \infty)$, where $\bar{x}_{rd}, \bar{\dot{x}}_{rd}, \bar{x}_{0d}, \bar{\dot{x}}_{0d}$ are known positive constants.

2.2.2 Control Development

2.2.2.1 Relay Agent

To facilitate the subsequent stability analysis, the relay agent's control inputs, i.e., $v_r(x_r(t))$ when $\phi_r(t) = a_r$ and $v_r(\hat{x}_r(t))$ when $\phi_r(t) = u_r$, are designed as

$$v_r(x_r(t)) \triangleq -k_r e_r(t) - f_r(x_r(t)) - \bar{d}_r \text{sgn}(e_r(t)) + \dot{x}_{rd}(t), \tag{2-13}$$

$$v_r(\hat{x}_r(t)) \triangleq -k_{\hat{r}} \hat{e}_r(t) - f_r(\hat{x}_r(t)) + \dot{x}_{rd}(t), \tag{2-14}$$

respectively, where $k_r, k_{\hat{r}} \in \mathbb{R}_{>0}$ are constant control gains, $\bar{d}_r$ is a known positive constant defined in Assumption 2.2, and $\text{sgn}(\cdot)$ is the signum function. Substituting (2-1) and (2-13) into the time-derivative of (2-3), and substituting (2-6) and (2-14) into the time-derivative of (2-4) yields the closed-loop error systems for the relay agent as

$$\dot{e}_r(t) = -k_r e_r(t) - \bar{d}_r \text{sgn}(e_r(t)) + d_r(t), \tag{2-15}$$

$$\dot{\hat{e}}_r(t) = -k_{\hat{r}} \hat{e}_r(t), \tag{2-16}$$

respectively.

2.2.2.2 Explorer Agents

Similarly, the control input for the explorer leader, i.e., $v_0(\hat{x}_0(t))$, is designed as

$$v_0(\hat{x}_0(t)) \triangleq -k_{\hat{e},0} \hat{e}_0(t) - f_e(\hat{x}_0(t)) + \dot{x}_{0d}(t), \tag{2-17}$$

where $k_{\hat{e},0} \in \mathbb{R}_{>0}$ is a positive control gain. Substituting (2-11) and (2-17) into the time-derivative of (2-8) yields the closed-loop error system for the explorer leader as

$$\dot{\hat{e}}_0(t) = -k_{\hat{e},0} \hat{e}_0(t). \tag{2-18}$$

To achieve the formation control and leader tracking objective, a distributed controller for explorer follower $i \in \mathcal{V}$ is designed as

$$v_i(\hat{x}_i(t)) \triangleq k_{\hat{e},f} \sum_{j \in \mathcal{N}_i, j \neq 0} \left(\hat{x}_j(t) - \hat{x}_i(t) - p_j + p_i \right) + k_{\hat{e},f} \left(p_i + \hat{x}_0(t) - \hat{x}_i(t) \right), \tag{2-19}$$

where $k_{\hat{e},f} \in \mathbb{R}_{>0}$ is a positive control gain for the i^{th} explorer follower agent, $\hat{x}_j : [t_0, \infty) \to \mathbb{R}^n$ denotes the state estimate of $x_j(t)$, and $p_j \in \mathbb{R}^n$ denotes the desired relative position between the explorer leader and the j^{th} explorer follower. Substituting (2-10) and (2-19) into the time-derivative of (2-12) yields the closed-loop error systems for the i^{th} explorer follower agent as

$$\dot{\hat{e}}_i(t) = f_e(\hat{x}_i(t)) + k_{\hat{e},f}\left(\sum_{j \in \mathcal{N}_i, j \neq 0} \left(\hat{e}_j(t) - \hat{e}_i(t)\right)\right) - k_{\hat{e},f}\hat{e}_i(t) - \dot{\hat{x}}_0(t). \tag{2-20}$$

Substituting (2-20) into the time-derivative of $\widehat{E}$ and compactly expressing the results with the Kronecker product yields

$$\dot{\widehat{E}} = \widetilde{N} + N_d - k_{\hat{e},f}(L \otimes I_n)\widehat{E} - k_{\hat{e},f}\widehat{E}, \tag{2-21}$$

where $\widetilde{N} \triangleq F\left(\widehat{X}\right) - F\left(\widehat{X}_0\right) \in \mathbb{R}^{nN}$, $N_d \triangleq F\left(\widehat{X}_0\right) - \dot{\widehat{X}}_0 \in \mathbb{R}^{nN}$, $F\left(\widehat{X}\right) \triangleq \left[f_e^{\mathrm{T}}(\hat{x}_1(t)), f_e^{\mathrm{T}}(\hat{x}_2(t)), ..., f_e^{\mathrm{T}}(\hat{x}_N(t))\right]^{\mathrm{T}} \in \mathbb{R}^{nN}$, $F\left(\widehat{X}_0\right) \triangleq [f_e^{\mathrm{T}}(\hat{x}_0(t)), f_e^{\mathrm{T}}(\hat{x}_0(t)), ..., f_e^{\mathrm{T}}(\hat{x}_0(t))]^{\mathrm{T}} \in \mathbb{R}^{nN}$, $\widehat{X}_0 \triangleq \left[\hat{x}_0^{\mathrm{T}}(t), \hat{x}_0^{\mathrm{T}}(t), ..., \hat{x}_0^{\mathrm{T}}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{nN}$.

To facilitate the subsequent stability analysis, some terms in (2-21) can be upper bounded. Specifically, given Assumption 2.3, the Mean Value Theorem (MVT) is invoked to conclude that $\left\|\widetilde{N}\right\| \leq c_1 \left\|\widehat{E}\right\| + c_1 \|P\|$, where $c_1 \in \mathbb{R}_{>0}$ and $P \triangleq \left[p_1^T, p_2^T, ..., p_N^T\right]^T \in \mathbb{R}^{nN}$. By Assumptions 2.3 and 2.7, if $\|\hat{x}_0(t)\| \leq \bar{\hat{x}}_0$, then $\|f_e(\hat{x}_0(t))\| \leq \bar{f}_{e,0}$ for some $\bar{f}_{e,0} \in \mathbb{R}_{>0}$. Since $\|f_e(\hat{x}_0(t))\| \leq \bar{f}_{e,0}$, $\|v_0(\hat{x}_0(t))\| \leq \bar{v}_0$, then $\left\|\dot{\hat{x}}_0(t)\right\| \leq \bar{\dot{\hat{x}}}_0$ for some $\bar{\dot{\hat{x}}}_0 \in \mathbb{R}_{>0}$ using (2-11). Therefore, $\|N_d\| \leq c_2$, where $c_2 \in \mathbb{R}_{>0}$. In addition, there exists a known bounding constant $\Lambda_{\min} \in \mathbb{R}_{>0}$ such that $\|L \otimes I_n\| \geq \Lambda_{\min}$ for all $t \in [t_0, \infty)$.

2.2.3 State Estimate

When $t = t_l^a$, a reset map is used to update the estimated state to the true state for the relay agent, i.e., $\hat{x}_r(t) = x_r(t)$. Similarly, when $t = t_m^a$, $\hat{x}_0(t) = \hat{x}_r(t) + x_{r0}(t)$ for the explorer leader, where the relative distance $x_{r0}(t)$ is measurable by Assumption 2.4. However, when state information is not available, i.e., $\phi_r = u_r$ or $\phi_e = u_e$, the state is estimated using the observer in (2-6) or (2-10), and the following closed-loop error systems must be used in the subsequent

stability analysis to develop the maximum dwell-time conditions.[1]

To develop the maximum dwell-time conditions, the time-derivative of the error in (2-5) must be used in addition to (2-15) and (2-16) to determine the overall state estimation error. Taking the time-derivative of (2-5) and substituting for the dynamics in (2-1) and the observer in (2-6) yields

$$\dot{\tilde{e}}_r(t) = f_r(x_r(t)) - f_r(\hat{x}_r(t)) + v_r(x_r(t)) - v_r(\hat{x}_r(t)) + d_r(t). \tag{2-22}$$

Similarly, taking the time-derivative of (2-9) and substituting in the dynamics in (2-2) for $i = 0$, and the observer in (2-11) for the explorer leader yields

$$\dot{\tilde{e}}_0(t) = f_e(x_0(t)) - f_e(\hat{x}_0(t)) + v_0(x_0(t)) - v_0(\hat{x}_0(t)) + d_0(t). \tag{2-23}$$

2.3 Stability Analysis

To ensure the overall system stability, six theorems are provided in this section.

2.3.1 Relay Agent

Three theorems are provided in this subsection to analyze the trajectory tracking error of the relay agent when feedback is available and when feedback is unavailable. Theorem 2.1 shows the trajectory tracking error of the relay agent $e_r(t)$ is bounded for all $t \in \left[t_l^a, t_l^u\right)$ when state feedback is available, then Theorem 2.2 and 2.3 show the trajectory tracking error of the relay agent $e_r(t)$ is bounded for all $t \in \left[t_l^u, t_{l+1}^a\right)$ when state feedback is unavailable, provided the maximum dwell-time condition is satisfied for the relay agent.

Theorem 2.1. *When state feedback for the relay agent is available, i.e., $\phi_r = a_r$ for all $t \in \left[t_l^a, t_l^u\right)$, the control law given in (2-13) ensures exponential stability of the trajectory tracking error defined in (2-3) in the sense that*

$$\|e_r(t)\| \leq \|e_r(t_l^a)\| \exp\left(-\frac{\lambda_r}{2}(t - t_l^a)\right), \tag{2-24}$$

where $\lambda_r \triangleq 2k_r \in \mathbb{R}_{>0}$ is a known constant.

[1] Since the state information from the relay agent is estimated (i.e., operating in a GPS-denied region), even when the reset map is used to update the explorer leader's state, all the explorers can only use state estimates instead of their true states.

Proof. Let $V_r^a : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a candidate Lyapunov functional defined as

$$V_r^a\left(e_r\left(t\right)\right) \triangleq \frac{1}{2} e_r^{\mathrm{T}}\left(t\right) e_r\left(t\right). \tag{2-25}$$

While $\phi_r = a_r$, i.e., $t \in \left[t_l^a, t_l^u\right)$, substituting (2-15) into the time-derivative of (2-25) yields

$$\dot{V}_r^a\left(e_r\left(t\right)\right) \leq -\lambda_r V_r^a\left(e_r\left(t\right)\right). \tag{2-26}$$

Invoking the Comparison Lemma in [56, Lemma 3.4] for (2-26) yields

$$V_r^a\left(e_r\left(t\right)\right) \leq V_r^a\left(e_r\left(t_l^a\right)\right) \exp\left(-\lambda_r\left(t - t_l^a\right)\right). \tag{2-27}$$

Substituting (2-25) into (2-27) yields (2-24), which implies $e_r\left(t\right) \in \mathcal{L}_\infty$ for all $t \in \left[t_l^a, t_l^u\right)$. By Assumption 2.2, $d_r\left(t\right) \in \mathcal{L}_\infty$, and $e_r\left(t\right) \in \mathcal{L}_\infty$, and hence $\dot{e}_r\left(t\right) \in \mathcal{L}_\infty$ for all $t \in \left[t_l^a, t_l^u\right)$ by (2-15). Additionally, from the time-derivative of (2-3) and the fact that $\dot{e}_r\left(t\right), \dot{x}_{rd}\left(t\right) \in \mathcal{L}_\infty$, then $\dot{x}_r\left(t\right) \in \mathcal{L}_\infty$ for all $t \in \left[t_l^a, t_l^u\right)$. Since $x_r\left(t\right) \in \mathcal{L}_\infty$, by Assumption 2.3, $f_r\left(x_r\left(t\right)\right) \in \mathcal{L}_\infty$ for all $t \in \left[t_l^a, t_l^u\right)$. From the relay agent dynamics described in (2-1), $v_r\left(x_r\left(t\right)\right) \in \mathcal{L}_\infty$ for all $t \in \left[t_l^a, t_l^u\right)$. □

The following two theorems are provided to show that the trajectory tracking error of the relay agent $e_r\left(t\right)$ is bounded for all $t \in \left[t_l^u, t_{l+1}^a\right)$ when state feedback is unavailable, provided the maximum dwell-time condition is satisfied.

Theorem 2.2. *When state feedback for the relay agent is unavailable, i.e., $\phi_r = u_r$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$, the controller in (2-14) ensures exponential stability of the estimated tracking error defined in (2-4) in the sense that*

$$\left\|\hat{e}_r\left(t\right)\right\| \leq \left\|\hat{e}_r\left(t_l^u\right)\right\| \exp\left(-\frac{\lambda_{\hat{r}}}{2}\left(t - t_l^u\right)\right), \tag{2-28}$$

where $\lambda_{\hat{r}} \triangleq 2k_{\hat{r}} \in \mathbb{R}_{>0}$ is a known constant.

Proof. Following the same approach in Theorem 2.1 with $V_{\hat{r}}^u : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ defined as

$$V_{\hat{r}}^u\left(\hat{e}_r\left(t\right)\right) \triangleq \frac{1}{2} \hat{e}_r^{\mathrm{T}}\left(t\right) \hat{e}_r\left(t\right), \tag{2-29}$$

while $\phi_r = u_r$, i.e., $t \in \left[t_l^u, t_{l+1}^a\right)$, then

$$V_{\hat{r}}^u\left(\hat{e}_r(t)\right) \leq V_{\hat{r}}^u\left(\hat{e}_r\left(t_l^u\right)\right) \exp\left(-\lambda_{\hat{r}}\left(t - t_l^u\right)\right). \tag{2-30}$$

Substituting (2-29) into (2-30) yields (2-28). Likewise, using similar arguments from the proof of Theorem 2.1, $\hat{e}_r(t), \dot{\hat{e}}_r(t), \dot{\hat{x}}_r(t), f_r(\hat{x}_r(t)), v_r(\hat{x}_r(t)) \in \mathcal{L}_\infty$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$. □

Based on Theorem 2.2, let $\hat{e}_{M,r} \in \mathbb{R}_{\geq 0}$ denote the upper bound for $\left\|\hat{e}_r\left(t_l^u\right)\right\|$, i.e., $\left\|\hat{e}_r\left(t_l^u\right)\right\| \leq \hat{e}_{M,r}$. To ensure the trajectory tracking error for the relay agent $e_r(t)$ is bounded, a user-defined maximum bound $e_{M,r} \in \mathbb{R}_{>0}$ can be selected to ensure $\limsup_{t \to \left(t_{l+1}^a\right)^-} \|e_r(t)\| \leq e_{M,r}$.

When state feedback for the relay agent is unavailable, the trajectory tracking error described in (2-3) will increase over time. However, Theorem 2.2 shows that the estimated trajectory tracking error in (2-4) exponentially converges when $\phi_r = u_r$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$. To guarantee (2-3) does not exceed the user-defined threshold $e_{M,r}$ when $\phi_r = u_r$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$, the error growth for (2-5) and the relationship $e_r(t) = \hat{e}_r(t) + \tilde{e}_r(t)$ are considered in Theorem 2.3.

Theorem 2.3. *For the trajectories of the switched system generated by the switching signal ϕ_r, (2-3)-(2-5) are GUUB provided the switching signal satisfies the maximum loss of feedback dwell-time condition*

$$\Delta t_l^u \leq \frac{1}{\lambda_{\tilde{r}}} \ln\left(\frac{\lambda_{\tilde{r}}}{\bar{d}_r^2}\left(e_{M,r} - \hat{e}_{M,r}\right)^2 + 1\right), \tag{2-31}$$

where $\lambda_{\tilde{r}} \triangleq 2c_r + 1 \in \mathbb{R}_{>0}$ is a known constant and $c_r \in \mathbb{R}_{>0}$ is a Lipschitz constant.

Proof. Let $V_{\tilde{r}}^u : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a candidate Lyapunov functional defined as

$$V_{\tilde{r}}^u\left(\tilde{e}_r(t)\right) \triangleq \frac{1}{2}\tilde{e}_r^{\mathrm{T}}(t)\,\tilde{e}_r(t). \tag{2-32}$$

While $\phi_r = u_r$, i.e., $t \in \left[t_l^u, t_{l+1}^a\right)$, substituting (2-22) into the time-derivative of (2-32) and invoking the Comparison Lemma yields

$$V_{\tilde{r}}^u\left(\tilde{e}_r(t)\right) \leq \frac{\bar{d}_r^2}{2\lambda_{\tilde{r}}}\left(\exp\left(\lambda_{\tilde{r}}\left(t - t_l^u\right)\right) - 1\right). \tag{2-33}$$

Substituting (2-32) into (2-33) yields

$$\|\tilde{e}_r(t)\| \leq \sqrt{\frac{\bar{d}_r^2}{\lambda_{\tilde{r}}}\left(\exp\left(\lambda_{\tilde{r}}\left(t-t_l^u\right)\right)-1\right)}. \tag{2-34}$$

Using (2-28) and (2-34) in $\left\|e_r\left(t_{l+1}^a\right)\right\| \leq \left\|\hat{e}_r\left(t_{l+1}^a\right)\right\| + \left\|\tilde{e}_r\left(t_{l+1}^a\right)\right\| \leq e_{M,r}$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$, the following inequality can be obtained

$$\hat{e}_{M,r} + \sqrt{\frac{\bar{d}_r^2}{\lambda_{\tilde{r}}}\left(\exp\left(\lambda_{\tilde{r}}\Delta t_l^u\right)-1\right)} \leq e_{M,r}. \tag{2-35}$$

When (2-35) is satisfied, $\left\|\hat{e}_r\left(t_{l+1}^a\right)\right\| + \left\|\tilde{e}_r\left(t_{l+1}^a\right)\right\| \leq e_{M,r}$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$, and since $\left\|e_r\left(t_{l+1}^a\right)\right\| \leq \left\|\hat{e}_r\left(t_{l+1}^a\right)\right\| + \left\|\tilde{e}_r\left(t_{l+1}^a\right)\right\|$, then $\left\|e_r\left(t_{l+1}^a\right)\right\| \leq e_{M,r}$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$. Using (2-35), the maximum Δt_l^u can be determined yielding (2-31). □

At the time instant $t = t_l^u$, $\hat{x}_r\left(t_l^u\right) = x_r\left(t_l^u\right)$ by using a reset map, implying $\hat{e}_r\left(t_l^u\right) = e_r\left(t_l^u\right)$ and $\tilde{e}_r\left(t_l^u\right) = 0$; therefore, the switched system is continuous at $t = t_l^u$ because $e_r\left(t_l^u\right) = \hat{e}_r\left(t_l^u\right) + \tilde{e}_r\left(t_l^u\right)$. At the time instant $t = t_{l+1}^a$, given $e_r(t) = \hat{e}_r(t) + \tilde{e}_r(t)$, utilizing (2-35), then $\lim_{t \to \left(t_{l+1}^a\right)^-} \sup \|e_r(t)\| \leq e_{M,r}$; therefore, the trajectory tracking error $e_r(t)$ is finite.

Theorem 2.1 shows the trajectory tracking error is bounded when $\phi_r = a_r$ for all $t \in \left[t_l^a, t_l^u\right)$, i.e., $\|e_r(t)\| \leq \left\|e_r\left(t_l^a\right)\right\| \exp\left(-\frac{\lambda_r}{2}\left(t-t_l^a\right)\right)$. Theorem 2.2 and 2.3 show the trajectory tracking error is bounded when $\phi_r = u_r$ for all $t \in \left[t_l^u, t_{l+1}^a\right)$, i.e., $\|e_r(t)\| \leq e_{M,r}$, by satisfying the maximum dwell-time condition in (2-31).

Zeno behavior occurs when the difference between $t_{l+1}^a - t_l^a$ is arbitrarily small, which implies the relay agent requires continuous state feedback. Since the objective is to ensure the system is stable while exploring the GPS-denied region, then it is critical to show that the difference between consecutive return times, i.e., $t_{l+1}^a - t_l^a$ is lower bounded by a finite positive constant. While $\phi_r = u_r$, let $t_r^{\text{travel}} \in \left[t_l^u, t_{l+1}^a\right)$ represent the minimum time it would take the relay agent to travel between consecutive feedback regions. Therefore, the maximum dwell-time condition has a lower constant bound, i.e., $t_r^{\text{travel}} \leq \Delta t_l^u$, where $t_r^{\text{travel}} = \frac{d_r^{\text{travel}}}{\bar{v}_{\text{vel},r}}$, $d_r^{\text{travel}} \in \mathbb{R}_{>0}$ denotes the actual distance the relay agent travels, $\bar{v}_{\text{vel},r} \in \mathbb{R}_{>0}$ denotes the maximum velocity of the relay

agent. By the definition of Δt_l^a and Δt_l^u, $t_{l+1}^a - t_l^a = \Delta t_l^u + \Delta t_l^a$. Since $t_r^{\text{travel}} \leq \Delta t_l^u$, $t_{l+1}^a - t_l^a \geq t_r^{\text{travel}}$. Therefore, Zeno behavior is excluded.

Based on Theorem 2.3, let $\tilde{e}_{M,r} \in \mathbb{R}_{\geq 0}$ denote the upper bound for $\left\|\tilde{e}_r\left(t_l^a\right)\right\|$, i.e., when $x_r(t) \in \mathcal{F}$. The selection of $\tilde{e}_{M,r}$ dictates how large the relay agent's state estimation error may grow prior to returning to the feedback region, and $\tilde{e}_{M,r}$ will be used in Theorem 2.5 to determine the maximum dwell-time of the explorer leader.

2.3.2 Explorer Agents

Three theorems are provided in this subsection to analyze the trajectory tracking error and the estimated tracking error of the explorer leader and the formation tracking error of the explorer followers for all time. Theorem 2.4 shows the estimated tracking error of the explorer leader $\hat{e}_0(t)$ is bounded for all $t \in \left[t_m^a, t_{m+1}^a\right)$, and Theorem 2.5 shows the trajectory tracking error of the explorer leader $\tilde{e}_0(t)$ is bounded for all $t \in \left[t_m^a, t_{m+1}^a\right)$, provided the maximum dwell-time condition is satisfied for the explorer leader. Additionally, Theorem 2.6 shows the explorer agents achieved formation control and leader tracking with the distributed controller.

Theorem 2.4. *When the explorer leader communicates with the relay agent, i.e., $\phi_e = a_e$ for all $t \in \left[t_m^a, t_m^u\right)$, and when the explorer leader does not communicate with the relay agent, i.e., $\phi_e = u_e$ for all $t \in \left[t_m^u, t_{m+1}^a\right)$, the control law given in (2-17) ensures exponential stability of the estimated tracking error defined in (2-8) in the sense that*

$$\left\|\hat{e}_0(t)\right\| \leq \left\|\hat{e}_0\left(t_m^a\right)\right\| \exp\left(-\frac{\lambda_{\hat{e},0}}{2}\left(t - t_m^a\right)\right) \tag{2-36}$$

for all $t \in \left[t_m^a, t_{m+1}^a\right)$, where $\lambda_{\hat{e},0} \triangleq 2k_{\hat{e},0} \in \mathbb{R}_{>0}$ is a known constant.

Proof. Following the same approach in Theorem 2.1 with $V_{\hat{e}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ defined as

$$V_{\hat{e}}\left(\hat{e}_0(t)\right) \triangleq \frac{1}{2}\hat{e}_0^{\mathrm{T}}(t)\,\hat{e}_0(t), \tag{2-37}$$

while $\phi_e = a_e$, i.e., $t \in \left[t_m^a, t_m^u\right)$, then

$$V_{\hat{e}}\left(\hat{e}_0(t)\right) \leq V_{\hat{e}}\left(\hat{e}_0\left(t_m^a\right)\right) \exp\left(-\lambda_{\hat{e},0}\left(t - t_m^a\right)\right). \tag{2-38}$$

Substituting (2-37) into (2-38) yields

$$\|\hat{e}_0(t)\| \leq \|\hat{e}_0(t_m^a)\| \exp\left(-\frac{\lambda_{\hat{e},0}}{2}(t-t_m^a)\right) \tag{2-39}$$

for all $t \in [t_m^a, t_m^u)$. Likewise, using similar arguments from the proof of Theorem 2.1, $\hat{e}_0(t), \dot{\hat{e}}_0(t), \dot{\hat{x}}_0(t), f_i(\hat{x}_0(t)), v_i(\hat{x}_0(t)) \in \mathcal{L}_\infty$ for all $t \in [t_m^a, t_m^u)$. Similarly, using the same Lyapunov functional (2-37) while $\phi_e = u_e$, i.e., $t \in \left[t_m^u, t_{m+1}^a\right)$, then

$$V_{\hat{e}}(\hat{e}_0(t)) \leq V_{\hat{e}}(\hat{e}_0(t_m^u)) \exp\left(-\lambda_{\hat{e},0}(t-t_m^u)\right). \tag{2-40}$$

Substituting (2-37) into (2-40) yields

$$\|\hat{e}_0(t)\| \leq \|\hat{e}_0(t_m^u)\| \exp\left(-\frac{\lambda_{\hat{e},0}}{2}(t-t_m^u)\right) \tag{2-41}$$

for all $t \in \left[t_m^u, t_{m+1}^a\right)$. Additionally, $\hat{e}_0(t), \dot{\hat{e}}_0(t), \dot{\hat{x}}_0(t), f_i(\hat{x}_0(t)), v_i(\hat{x}_0(t)) \in \mathcal{L}_\infty$ for all $t \in \left[t_m^u, t_{m+1}^a\right)$. Using (2-39) and (2-41) yields (2-36). □

Based on Theorem 2.4, let $\hat{e}_{M,e} \in \mathbb{R}_{\geq 0}$ denote the upper bound for $\|\hat{e}_0(t_m^u)\|$, i.e., $\|\hat{e}_0(t_m^u)\| \leq \hat{e}_{M,e}$. The value of $\hat{e}_{M,e}$ dictates how large the explorer leader's estimated tracking error may grow prior to communicating with the relay agent, which will be used in Theorem 2.5. When the state estimate for the explorer leader from the relay agent is unavailable, the trajectory tracking error described in (2-7) will increase over time. However, Theorem 2.4 shows that the estimated trajectory tracking error in (2-8) exponentially converges when $t \in \left[t_m^a, t_{m+1}^a\right)$. To guarantee (2-7) does not exceed the user-defined threshold for all $t \in \left[t_m^a, t_{m+1}^a\right)$, the error growth for (2-9) and the relationship $e_0(t) = \hat{e}_0(t) + \tilde{e}_0(t)$ are considered in Theorem 2.5.

Remark 2.1. Using the reset map at time $t = t_m^a$ may result in an instantaneous growth of (2-37), e.g., an instantaneous growth of $\hat{e}_0(t)$ at t_m^a, and therefore $V_{\hat{e}}(\hat{e}_0(t))$ by (2-37). However, the jump is finite using (2-41) and the constraint $\|\hat{e}_0(t_m^u)\| \leq \hat{e}_{M,e}$ in Theorem 2.4 implies $\left\|\hat{e}_0\left(t_{m+1}^a\right)\right\| \leq \hat{e}_{M,e}$. The value of $V_{\hat{e}}(\hat{e}_0(t_m^a))$ at the beginning of the current cycle is greater or equal to the value of $V_{\hat{e}}\left(\hat{e}_0\left(t_{m+1}^a\right)\right)$ at the beginning of the next cycle. Therefore, using the reset map will not lead to instability.

To show the trajectory tracking error of the explorer leader $\tilde{e}_0(t)$ is bounded for all $t \in \left[t_m^a, t_{m+1}^a\right)$, the following theorem is provided when the maximum dwell-time condition is satisfied for the explorer leader.

Theorem 2.5. *For the trajectories of the switched system generated by the switching signal ϕ_e, (2-7)-(2-9) are GUUB provided the switching signal satisfies the maximum loss of feedback dwell-time condition*

$$\Delta t_m^u \leq \frac{1}{\lambda_{\tilde{e}}} \ln\left(\frac{\tilde{e}_{M,e}^2 + \frac{\bar{d}_0^2}{\lambda_{\tilde{e}}}}{\tilde{e}_{m,e}^2 + \frac{\bar{d}_0^2}{\lambda_{\tilde{e}}}}\right), \tag{2-42}$$

where $\lambda_{\tilde{e}} \triangleq 2c_e + 1 \in \mathbb{R}_{>0}$ is a known constant, $c_e \in \mathbb{R}_{>0}$ is a Lipschitz constant, and $\tilde{e}_{M,e} \in \mathbb{R}_{>0}$, $\tilde{e}_{m,e} \in \mathbb{R}_{>0}$ are user-defined parameters with $\tilde{e}_{M,e} > \tilde{e}_{m,e}$.

Proof. Let $V_{\tilde{e}} : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be a candidate Lyapunov functional defined as

$$V_{\tilde{e}}\left(\tilde{e}_0(t)\right) \triangleq \frac{1}{2}\tilde{e}_0^{\mathrm{T}}(t)\,\tilde{e}_0(t), \tag{2-43}$$

while $\phi_e = u_e$, i.e., $t \in \left[t_m^u, t_{m+1}^a\right)$, substituting (2-23) into the time-derivative of (2-43) and invoking the Comparison Lemma yields

$$V_{\tilde{e}}\left(\tilde{e}_0(t)\right) \leq V_{\tilde{e}}\left(\tilde{e}_0(t_m^u)\right)\exp\left(\lambda_{\tilde{e}}(t - t_m^u)\right) + \frac{\bar{d}_0^2}{2\lambda_{\tilde{e}}}\left(\exp\left(\lambda_{\tilde{e}}(t - t_m^u)\right) - 1\right). \tag{2-44}$$

Substituting (2-43) into (2-44) yields

$$\left\|\tilde{e}_0(t)\right\| \leq \sqrt{\left(\left\|\tilde{e}_0(t_m^u)\right\|^2 + \frac{\bar{d}_0^2}{\lambda_{\tilde{e}}}\right)\exp\left(\lambda_{\tilde{e}}(t - t_m^u)\right) - \frac{\bar{d}_0^2}{\lambda_{\tilde{e}}}} \tag{2-45}$$

for all $t \in \left[t_m^u, t_{m+1}^a\right)$. By enforcing the constraints $\left\|\tilde{e}_0\left(t_{m+1}^a\right)\right\| \leq \tilde{e}_{M,e}$ and $\left\|\tilde{e}_0(t_m^u)\right\| \leq \tilde{e}_{m,e}$, the maximum dwell-time condition in (2-42) can be obtained. When the stabilizing condition described in (2-42) is satisfied, the error defined in (2-9) remains bounded for all $t \in \left[t_m^u, t_{m+1}^a\right)$. Note that the value of $\tilde{e}_{M,e}$ is selected to be larger than that of $\tilde{e}_{M,r}$, i.e., $\tilde{e}_{M,e} > \tilde{e}_{M,r}$, and this allows the relay agent to provide a better state estimate to the explorer leader. Substituting (2-41)

and (2-45) in $\left\|e_0\left(t_{m+1}^a\right)\right\| \leq \left\|\hat{e}_0\left(t_{m+1}^a\right)\right\| + \left\|\tilde{e}_0\left(t_{m+1}^a\right)\right\|$, yields

$$\left\|e_0\left(t_{m+1}^a\right)\right\| \leq \hat{e}_{M,e} + \tilde{e}_{M,e} \tag{2-46}$$

for all $t \in \left[t_m^u, t_{m+1}^a\right)$.

Similarly, while $\phi_e = a_e$, i.e., $t \in [t_m^a, t_m^u)$, substituting (2-23) into the time-derivative of (2-43) and invoking the Comparison Lemma yields

$$V_{\tilde{e}}\left(\tilde{e}_0\left(t\right)\right) \leq V_{\tilde{e}}\left(\tilde{e}_0\left(t_m^a\right)\right)\exp\left(\lambda_{\tilde{e}}\left(t-t_m^a\right)\right) + \frac{\bar{d}_0^2}{2\lambda_{\tilde{e}}}\left(\exp\left(\lambda_{\tilde{e}}\left(t-t_m^a\right)\right)-1\right). \tag{2-47}$$

Substituting (2-43) into (2-47) yields

$$\left\|\tilde{e}_0\left(t\right)\right\| \leq \sqrt{\left(\left\|\tilde{e}_0\left(t_m^a\right)\right\|^2 + \frac{\bar{d}_0^2}{\lambda_{\tilde{e}}}\right)\exp\left(\lambda_{\tilde{e}}\left(t-t_m^a\right)\right) - \frac{\bar{d}_0^2}{\lambda_{\tilde{e}}}} \tag{2-48}$$

for all $t \in [t_m^a, t_m^u)$. By enforcing the constraint $\left\|\tilde{e}_0\left(t_m^u\right)\right\| \leq \tilde{e}_{m,e}$, (2-48) shows the error defined in (2-9) remains bounded for all $t \in [t_m^a, t_m^u)$. Substituting (2-39) and (2-48) in $\left\|e_0\left(t_m^u\right)\right\| \leq \left\|\hat{e}_0\left(t_m^u\right)\right\| + \left\|\tilde{e}_0\left(t_m^u\right)\right\|$, yields

$$\left\|e_0\left(t_m^u\right)\right\| \leq \hat{e}_{M,e} + \tilde{e}_{m,e} \tag{2-49}$$

for all $t \in [t_m^a, t_m^u)$. Note the upper bounds for the trajectory tracking error of the explorer leader at t_{m+1}^a and t_m^u are shown in (2-46) and (2-49), respectively. By satisfying the maximum dwell-time in (2-42), i.e., $\tilde{e}_{M,e} > \tilde{e}_{m,e}$, therefore the upper bound of $e_0\left(t_{m+1}^a\right)$ is larger than that of $e_0\left(t_m^u\right)$, and this condition allows the explorer leader to navigate the GPS-denied region without communicating with the relay agent for Δt_m^u. Additionally, by satisfying (2-42), and using (2-46), (2-49) implies the trajectory tracking error $e_0\left(t\right)$ is bounded for all $t \in \left[t_m^a, t_{m+1}^a\right)$. □

Remark 2.2. To ensure the relay agent can communicate with the explorer leader, the sum of the state estimation errors for both the relay and the explorer leader must be selected so that the communication radius, i.e., $\tilde{e}_{M,r} + \tilde{e}_{M,e} \leq R_{\text{com}}$.

To show the explorer agents achieved formation control and leader tracking with the distributed controller, the following theorem is provided.

Theorem 2.6. *The controller in (2-19) ensures the stacked leader-follower error $\widehat{E}$ is GUUB in the sense that*

$$\left\|\widehat{E}\right\| \leq \sqrt{\left\|\widehat{E}\left(t_0\right)\right\|^2 \exp\left(-2\phi\left(t-t_0\right)\right)+\frac{\varepsilon}{\phi}}, \tag{2-50}$$

where $\widehat{E}\left(t_0\right) \triangleq \left[\hat{e}_1^T\left(t_0\right), \hat{e}_2^T\left(t_0\right), \ldots, \hat{e}_N^T\left(t_0\right)\right]^T \in \mathbb{R}^{nN}$, $\phi \triangleq k_{\hat{e},f}\Lambda_{\min} - c_1 \in \mathbb{R}_{>0}$, $\varepsilon \triangleq \frac{(\delta+c_2)^2}{4k_{\hat{e},f}} \in \mathbb{R}_{>0}$ *are known constants, and* $\delta \triangleq c_1 \|P\| \in \mathbb{R}_{>0}$.

Proof. Let $V_{\widehat{E}} : \mathbb{R}^{nN} \to \mathbb{R}_{\geq 0}$ be a candidate Lyapunov functional defined as

$$V_{\widehat{E}}\left(\widehat{E}\right) \triangleq \frac{1}{2}\widehat{E}^{\mathrm{T}}\widehat{E}. \tag{2-51}$$

Substituting (2-21) into the time-derivative of (2-51) yields

$$\dot{V}_{\widehat{E}}\left(\widehat{E}\right) = \widehat{E}^{\mathrm{T}}\widetilde{N} + \widehat{E}^{\mathrm{T}}N_d - k_{\hat{e},f}\widehat{E}^{\mathrm{T}}\left(L\otimes I_n\right)\widehat{E} - k_{\hat{e},f}\widehat{E}^{\mathrm{T}}\widehat{E}, \tag{2-52}$$

where (2-52) can be upper bounded as

$$\dot{V}_{\widehat{E}}\left(\widehat{E}\right) \leq \left\|\widehat{E}\right\|\left\|\widetilde{N}\right\| + \left\|\widehat{E}\right\|\|N_d\| - k_{\hat{e},f}\widehat{E}^{\mathrm{T}}\left(L\otimes I_n\right)\widehat{E} - k_{\hat{e},f}\left\|\widehat{E}\right\|^2. \tag{2-53}$$

Through the use of the previous bounds at the end of Section 2.2.2.2, (2-53) can be further upper bounded as

$$\dot{V}_{\widehat{E}}\left(\widehat{E}\right) \leq -\phi\left\|\widehat{E}\right\|^2 + \varepsilon, \tag{2-54}$$

where ϕ and ε are defined in (2-50). Note that the selection of $k_{\hat{e},f}$ needs to satisfy $k_{\hat{e},f} > \frac{c_1}{\Lambda_{\min}}$ for ϕ to be positive. Substituting (2-51) into (2-54) and invoking the Comparison Lemma yields

$$V_{\widehat{E}}\left(\widehat{E}\right) \leq V_{\widehat{E}}\left(\widehat{E}\left(t_0\right)\right)\exp\left(-2\phi\left(t-t_0\right)\right) + \frac{\varepsilon}{2\phi}. \tag{2-55}$$

Substituting (2-51) into (2-55) yields (2-50), which implies $\widehat{E} \in \mathcal{L}_\infty$. Since the stack error $\widehat{E} \in \mathcal{L}_\infty$, $\hat{e}_i\left(t\right) \in \mathcal{L}_\infty$. By (2-12), $\hat{x}_i\left(t\right), \hat{x}_0\left(t\right) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$ when $t \in \left[t_0, \infty\right)$. Since $\hat{x}_i\left(t\right), \hat{x}_0\left(t\right) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$, the distributed controller for explorer follower $v_i\left(\hat{x}_i\left(t\right)\right) \in \mathcal{L}_\infty$ for all $i \in \mathcal{V}$. □

2.4 Simulation

To demonstrate the performance of the developed method, a simulation example is performed using a robotics simulator called Gazebo. A quadrotor is used to model the relay agent, and three wheeled mobile robots (WMRs) are used to model the explorer agents. The environmental setup is depicted in Figure 2-1. Section 2.4.1 shows the simulation results of the errors and trajectories of the relay agent while switching between feedback available region and feedback-denied region, and Section 2.4.2 shows the simulation results of the errors and trajectories of the explorer agents while navigating the feedback-denied region under intermittent communication with the relay agent.

The dynamics and controllers used for the relay agent are

$$\dot{x}_r(t) = v_r(x_r(t))$$

and

$$\begin{aligned} v_r(x_r(t)) &= -k_r e_r(t) - \bar{d}_r \operatorname{sgn}(e_r(t)) + \dot{x}_{rd}(t),\ \phi_r = a_r, \\ v_r(\hat{x}_r(t)) &= -k_{\hat{r}} \hat{e}_r(t) + \dot{x}_{rd}(t),\ \phi_r = u_r, \end{aligned}$$

respectively. The desired velocities for the relay agent are designed as

$$\dot{x}_{rd}(t) = v_{r,\max} \frac{\dot{\hat{x}}_0(t) + k_{vrd}(\hat{x}_0(t) - x_{rd}(t))}{\left\| \dot{\hat{x}}_0(t) + k_{vrd}(\hat{x}_0(t) - x_{rd}(t)) \right\|}, \tag{2-56}$$

$$\dot{x}_{rd}(t) = v_{r,\max} \frac{k_{vrd}\left(x_{\text{goal}} - x_{rd}(t)\right)}{\left\| k_{vrd}\left(x_{\text{goal}} - x_{rd}(t)\right) \right\|}, \tag{2-57}$$

where $k_{vrd} \in \mathbb{R}_{>0}$ is a control gain, $v_{r,\max} \in \mathbb{R}_{>0}$ is the maximum velocity of the relay agent, and the goal location is selected as $x_{\text{goal}} = [5.5, -5.5]^{\mathrm{T}}$. The expression (2-56) represents the desired velocity of the relay agent navigates to the explorer leader, and (2-57) represents the desired velocity of the relay agent travels to the center of the feedback available region. The Euler's method is used to integrate the desired velocities designed in (2-56) and (2-57) to get the desired position for the relay agent. When the relay agent reaches the explorer leader and enters the

feedback available region, the switching is triggered to set the estimated state to the desired state for the change of traveling directions, i.e., $x_{rd}\left(t_{\text{switch},k}\right) = \hat{x}_r\left(t_{\text{switch},k}\right)$, where $t_{\text{switch},k} \in \mathbb{R}$ denotes the kth switching instances. The dynamics and controller used for the explorer leader are

$$\dot{\hat{x}}_0(t) = \hat{x}_0^2(t) + v_0(\hat{x}_0(t))$$

and

$$v_0(\hat{x}_0(t)) = -k_{\hat{e},0}\hat{e}_0(t) - \hat{x}_0^2(t) + \dot{x}_{0d}(t),$$

respectively. The desired trajectory $x_{0d}(t)$ is selected as

$$\begin{aligned} x_{0d,x} &= a\cos(\omega t), \\ x_{0d,y} &= a\sin(\omega t), \end{aligned}$$

where $a = 1$ m is the radius, and $\omega = 0.1$ rad/s is the cycle frequency. The dynamics and controllers used for the explorer followers are

$$\dot{\hat{x}}_i(t) = v_i(\hat{x}_i(t)),\ i = 1,2,$$

$$v_i(\hat{x}_i(t)) = k_{\hat{e},f} \sum_{j\in\mathcal{N}_i, j\neq 0} \left(\hat{x}_j(t) - \hat{x}_i(t) - p_j + p_i\right) + k_{\hat{e},f}\left(p_i + \hat{x}_0(t) - \hat{x}_i(t)\right),$$

where the desired relative positions of the explorer followers with respect to the explorer leader are selected as $p_1 = [-1.5, -0.86]^{\mathrm{T}}$ and $p_2 = [-1.5, 0.86]^{\mathrm{T}}$ represented in the body coordinate frame, respectively.

2.4.1 Relay Agent

The relay agent is tasked with servicing the explorer leader agent when the state feedback for the relay agent is unavailable, so the state estimator developed in [57] is used to update the state estimate of the relay agent in the feedback-denied region. As shown in Figure 2-2, the estimated trajectory is depicted with the solid blue line. When the distance between the estimated state of the relay agent and the estimated state of the explorer leader is within the communication

range, the relay agent delivers its state estimate to the explorer leader, and the communications are represented by the green solid lines. After servicing the explorer leader, the relay agent travels towards the center of feedback available region represented by a dashed red circle centered at $[5.5, -5.5]^{\mathrm{T}}$ with a radius of 1.5 m. When the estimated trajectory of the relay agent is within the feedback region, true state information is used to update the estimated state of the relay agent.

Figure 2-3 shows the trajectory tracking error $e_r(t)$, the estimated trajectory tracking error $\hat{e}_r(t)$, and the state estimation error $\tilde{e}_r(t)$ of the relay agent are bounded over the simulation time period from $t = 0$ to $t = 64s$. The vertical dashed cyan lines denote the instances at $t = 0s, 19s, 34s, 49s$, where the estimated trajectory of the relay agent enters the feedback region, and all the errors are reset using true sensor feedback at these instances. The vertical dashed black lines denote the instances at $t = 8s, 25s, 40s, 57s$, where the relay agent services the explorer leader, and the errors of the explorer agents are improved as shown in Figures 2-5-2-7. The simulation parameters are selected as shown in Table 2-1.

2.4.2 Explorer Agents

The explorer leader is tasked with tracking a given desired trajectory (i.e., depicted using the dashed circle with a radius of 1 m in Figure 2-4) in the feedback-denied region, while the explorer followers maintain a fixed formation with respect to the explorer leader (i.e., $p_1 = [-1.5, -0.86]^{\mathrm{T}}$, $p_2 = [-1.5, 0.86]^{\mathrm{T}}$) by using the distributed controllers designed in (2-19).

Figure 2-5 shows the trajectory tracking error $e_0(t)$, the estimated trajectory tracking error $\hat{e}_0(t)$, and the state estimation error $\tilde{e}_0(t)$ of the explorer leader are bounded. The vertical dashed black lines denote the servicing instances at $t = 8s, 25s, 40s, 57s$, where the explorer leader gets state estimate from the relay agent, and the state estimation error is dropped after the servicing instances. Similar to the errors of the explorer leader, the errors of the explorer follower 1 and explorer follower 2 are shown in Figures 2-6 and 2-7, the errors of the corresponding agent are bounded, and therefore, the stacked error $\widehat{E}$ defined in (2-12) is bounded. This simulation example shows that the developed relay-explorer approach enables multiple explorer agents to track a given desired trajectory in a feedback-denied environment indefinitely while intermittently

communicating with a relay agent.

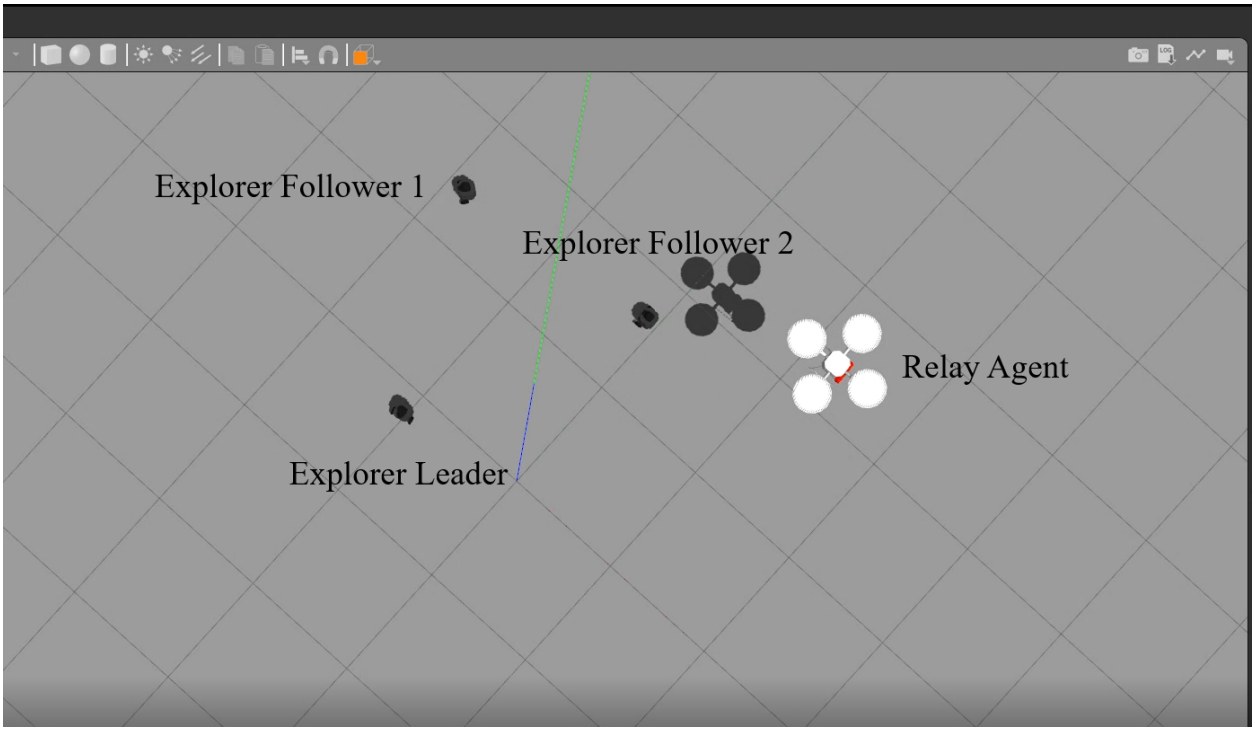


Figure 2-1. The world environmental setup in Gazebo. One quadrotor and three WMRs are used to represent the relay agent and explorer agents, respectively.

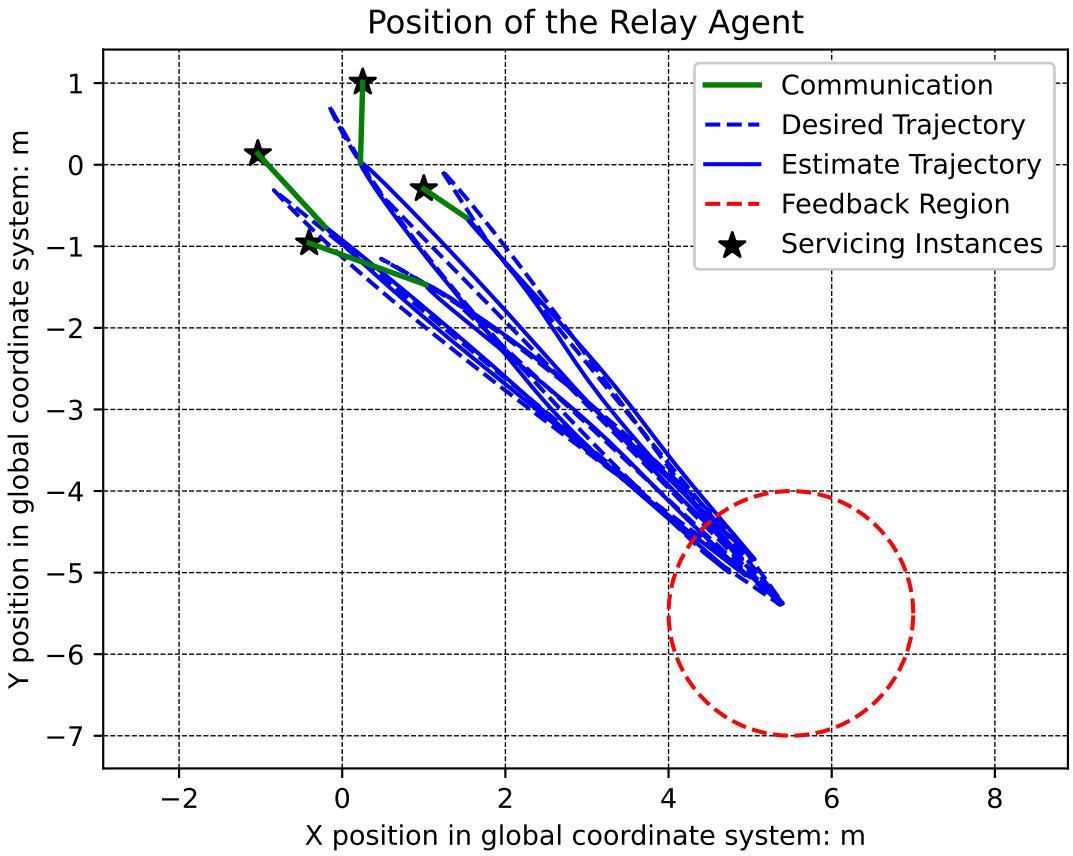


Figure 2-2. The estimated and the desired trajectories of the relay agent switching between feedback available and feedback-denied regions.

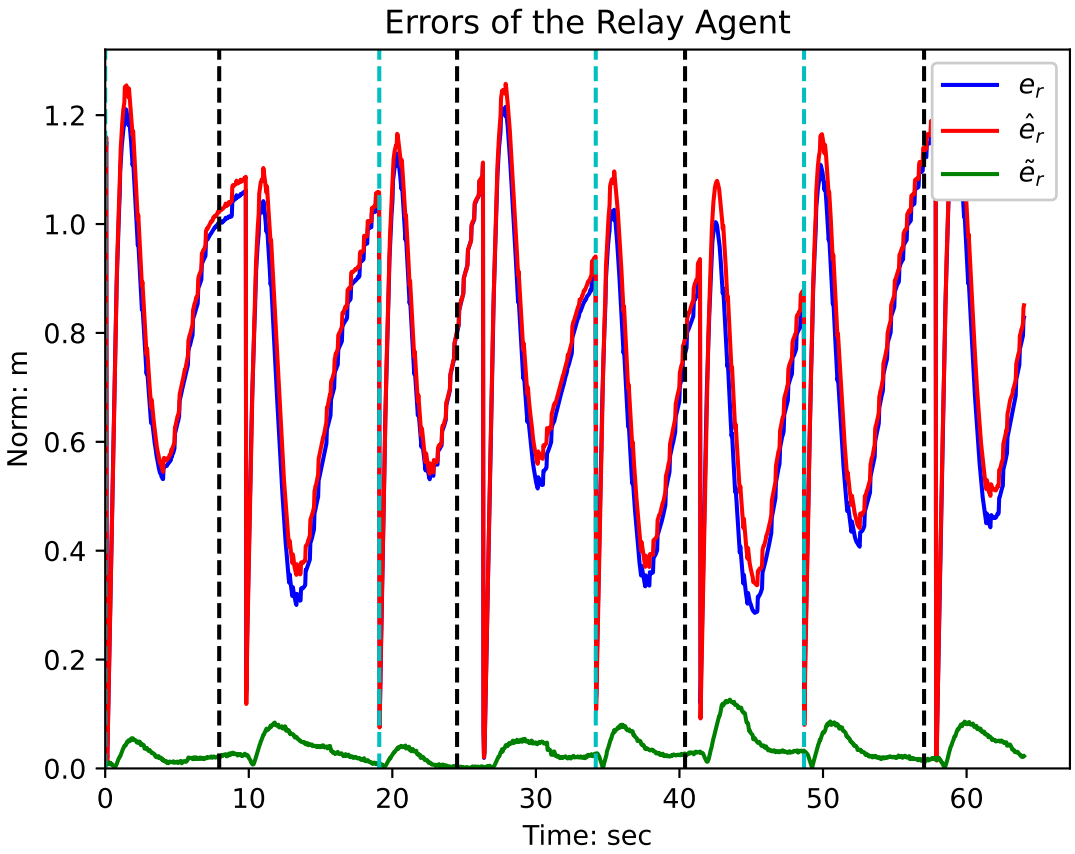


Figure 2-3. The trajectory tracking error $e_r(t)$, the estimated trajectory tracking error $\hat{e}_r(t)$, and the state estimation error $\tilde{e}_r(t)$ of the relay agent over the simulation time period from $t = 0$ to $t = 64s$. The vertical dashed cyan lines denote the instances at $t = 0s, 19s, 34s, 49s$, where the estimated trajectory of the relay agent enters the feedback region, and all the errors are reset using true sensor feedback at these instances. The vertical dashed black lines denote the instances at $t = 8s, 25s, 40s, 57s$, where the relay agent services the explorer leader.

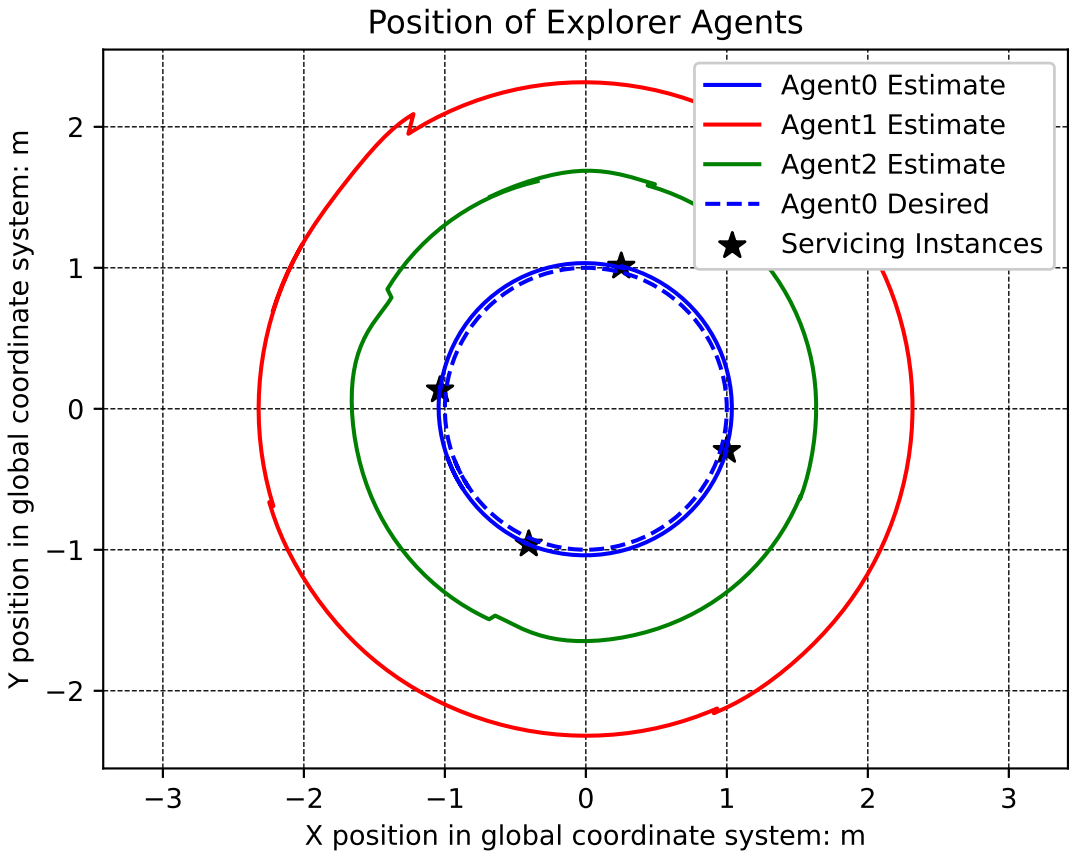


Figure 2-4. The desired and the estimated trajectories of the explorer agents in the feedback-denied region.

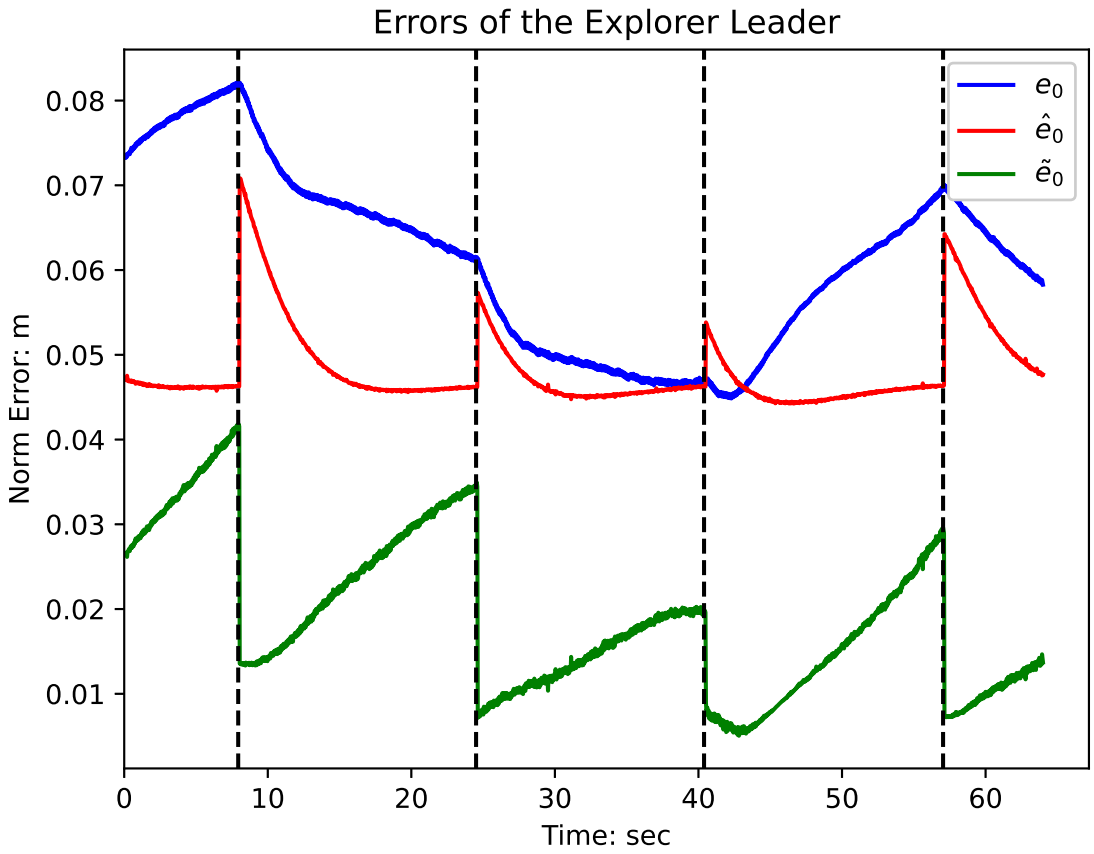


Figure 2-5. The trajectory tracking error $e_0(t)$, the estimated trajectory tracking error $\hat{e}_0(t)$, and the state estimation error $\tilde{e}_0(t)$ of the explorer leader.

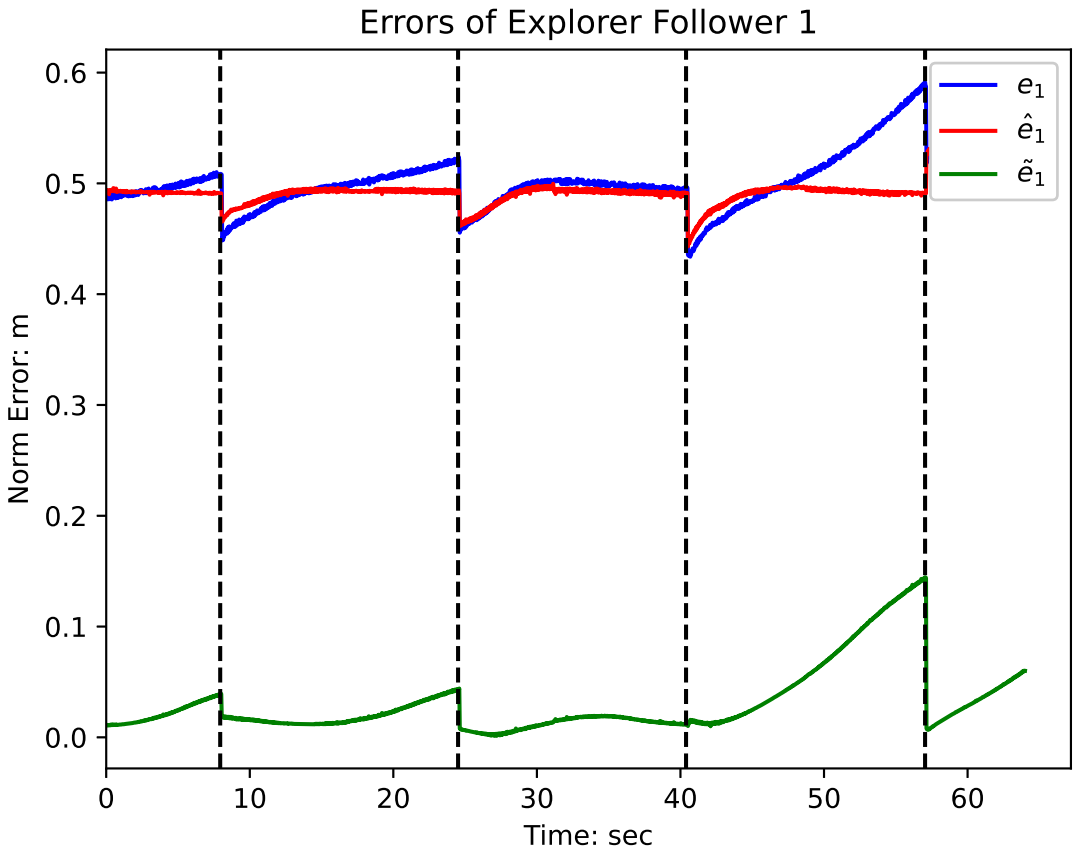


Figure 2-6. The trajectory tracking error $e_1(t)$, the estimated trajectory tracking error $\hat{e}_1(t)$, and the state estimation error $\tilde{e}_1(t)$ of the explorer follower 1.

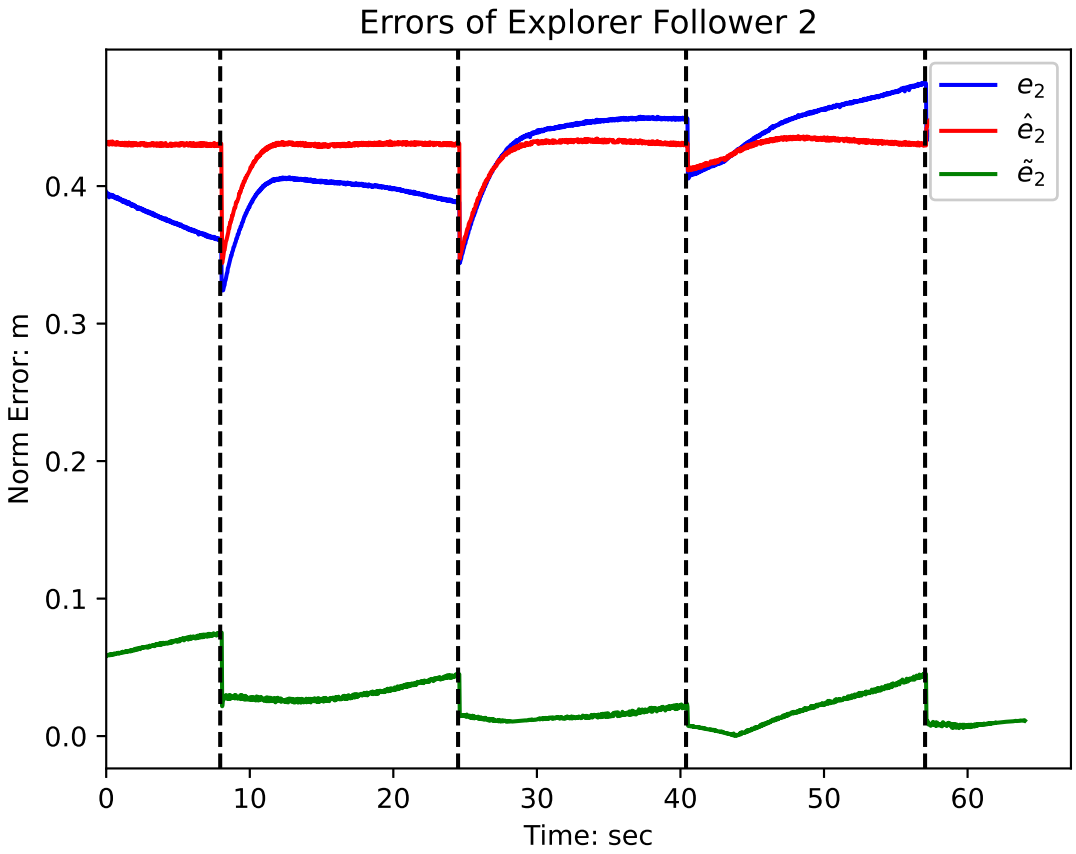


Figure 2-7. The trajectory tracking error $e_2(t)$, the estimated trajectory tracking error $\hat{e}_2(t)$, and the state estimation error $\tilde{e}_2(t)$ of the explorer follower 2.

Table 2-1. Simulation parameters.

Relay Agent			Explorer Agents		
$v_{r,\max} = 0.75\ \frac{\text{m}}{\text{s}}$	$R_{\text{com}} = 1.5$ m	$k_r = 0.3$	$v_{e,\max} = 0.4\ \frac{\text{m}}{\text{s}}$	$k_{\hat{e},0} = 0.2$	$k_{\hat{e},f} = 0.4$
$k_{\hat{r}} = 0.5$	$\bar{d}_r = 0.001$	$e_{M,r} = 1.5$ m	$\hat{e}_{M,e} = 0.08$ m	$\tilde{e}_{M,e} = 0.05$ m	$\tilde{e}_{m,e} = 0.05$ m
$\hat{e}_{M,r} = 1.2$ m	$\tilde{e}_{M,r} = 0.13$ m				

CHAPTER 3
A SWITCHED SYSTEMS APPROACH TO UNKNOWN ENVIRONMENT EXPLORATION WITH INTERMITTENT STATE FEEDBACK FOR NONHOLONOMIC SYSTEMS

A method developed in [19] is provided in this chapter to enable a nonholonomic vehicle to explore an unknown environment with intermittent state feedback. The contributions include developing a set of stabilizing dwell-time conditions using a Lyapunov-based switched systems approach to ensure stability of the system while the nonholonomic agent is exploring the unknown environment. Specifically, a maximum dwell-time condition is developed to allow the nonholonomic agent to explore the feedback-denied environment for a predetermined period of time, and a minimum dwell-time condition is developed to compensate for the state estimation error to ensure the trajectory tracking error converges within a desired neighborhood of the desired trajectory. Using the proposed maximum and minimum dwell-time conditions, a nonholonomic vehicle's tracking error remains GUUB, enabling the exploration of the feedback-denied region for a predetermined period of time, before acquiring state feedback.

3.1 System Model

The kinematic model for a unicycle with an exogenous disturbance from [24] is

$$\dot{q}(t) = S(q(t))\, v(t) + d(t), \tag{3-1}$$

where $t \in \mathbb{R}_{\geq t_0}$ is the time and $\mathbb{R}_{\geq t_0} \triangleq [t_0, \infty)$. Let $t_0 \in \mathbb{R}_{\geq 0}$ denote the initial time and $q : [t_0, \infty) \to \mathbb{R}^3$ denote the state, which is defined as

$$q(t) \triangleq \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^{\mathrm{T}}, \tag{3-2}$$

where $x(t), y(t)$ denote the planar position and $\theta(t)$ denotes the orientation of the nonholonomic vehicle. The matrix $S : \mathbb{R}^3 \to \mathbb{R}^{3\times 2}$ is defined as

$$S(q(t)) \triangleq \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \\ 0 & 1 \end{bmatrix}, \tag{3-3}$$

and the input velocity vector $v : [t_0, \infty) \to \mathbb{R}^2$ is defined as

$$v(t) \triangleq \begin{bmatrix} v_1(t) & v_2(t) \end{bmatrix}^{\mathrm{T}}, \tag{3-4}$$

where $v_1(t), v_2(t)$ denote the linear and angular velocity inputs of the nonholonomic vehicle, respectively, and $v_2(t) \triangleq \dot{\theta}(t)$. The exogenous disturbance $d : [t_0, \infty) \to \mathbb{R}^3$ is defined as

$$d(t) \triangleq \begin{bmatrix} d_1(t) & d_2(t) & d_3(t) \end{bmatrix}^{\mathrm{T}}, \tag{3-5}$$

where $d_1, d_2, d_3 : [t_0, \infty) \to \mathbb{R}$.

Assumption 3.1. The system disturbances are upper bounded by known positive constants $\bar{d}_1, \bar{d}_2, \bar{d}_3, \bar{d}_4, \bar{d} \in \mathbb{R}_{>0}$, where $\mathbb{R}_{>0} \triangleq (0, \infty)$, such that $\|d_1(t)\| \leq \bar{d}_1$, $\|d_2(t)\| \leq \bar{d}_2$, $\|d_3(t)\| \leq \bar{d}_3$, $\|d(t)\| = \sqrt{d_1^2 + d_2^2 + d_3^2} \leq \bar{d}$ for all $t \geq t_0$. In addition, $\bar{d}_4$ satisfies $\bar{d}_4 > \bar{d}_1 + \bar{d}_2$ (see (3-48)).

3.2 Control Design

3.2.1 Control Objective

The control objective is to enable a nonholonomic vehicle to explore an unknown environment, where feedback is unavailable by intermittently following a trajectory through the unknown environment and intermittently into regions, where feedback is available.

Assumption 3.2. There are multiple stationary known feedback zones in the unknown environment. Each of the feedback regions is defined by a compact set $\mathcal{F}_j \subset \mathbb{R}^3$, $j \in \mathcal{N} \triangleq \{1, 2, ..., N\}$, where N denotes the number of regions and $N \in \mathbb{Z}_{>0}$.

Assumption 3.3. The nonholonomic vehicle is initialized in a feedback region, i.e., $q(t_0) \in \bigcup_{j \in \mathcal{N}} \mathcal{F}_j$.

Assumption 3.4. The feedback regions are adequately spaced so that a vehicle can move between the regions while satisfying the subsequently defined dwell-time conditions. The zone outside all the feedback regions is defined as $\mathcal{F}^u \triangleq \mathbb{R}^3 \backslash \left(\bigcup_{j \in \mathcal{N}} \mathcal{F}_j \right)$. Feedback is available to the nonholonomic agent if and only if $q \in \bigcup_{j \in \mathcal{N}} \mathcal{F}_j$. If $q \in \mathcal{F}^u$, then the nonholonomic agent does not have access to feedback.

Definition 3.1. Let $\phi : [t_0, \infty) \to \{a, u\}$ be a switching signal, where a indicates $q \in \bigcup_{j \in \mathcal{N}} \mathcal{F}_j$ and u indicates $q \in \mathcal{F}^u$. Let $t_i^a \in \mathbb{R}_{\geq t_0}$ denote the time of the i^{th} instance when $\phi(t) = a$, i.e., $t_i^a \triangleq \inf\left\{t > t_{i-1}^a | \phi(t) = a\right\}$, where $i \in \mathbb{Z}_{\geq 0}$. Let $t_i^u \in \mathbb{R}_{>t_0}$ denote the time of the i^{th} instance when $\phi(t) = u$, i.e., $t_i^u \triangleq \inf\left\{t > t_{i-1}^u | \phi(t) = u\right\}$. The dwell-time of the i^{th} instance of each subsystem is defined as $\Delta t_i^a \triangleq t_i^u - t_i^a \in \mathbb{R}_{>0}$ and $\Delta t_i^u \triangleq t_{i+1}^a - t_i^u \in \mathbb{R}_{>0}$.

A reference trajectory for the nonholonomic vehicle is generated by

$$\dot{q}_d(t) = S(q_d(t)) v_d(t), \tag{3-6}$$

where $q_d : [t_0, \infty) \to \mathbb{R}^3$ denotes the desired position and orientation, and is defined as

$$q_d(t) \triangleq \begin{bmatrix} x_d(t) & y_d(t) & \theta_d(t) \end{bmatrix}^{\mathrm{T}}. \tag{3-7}$$

Let $x_d, y_d, \theta_d : [t_0, \infty) \to \mathbb{R}$ and the matrix $S : \mathbb{R}^3 \to \mathbb{R}^{3\times 2}$ be defined as

$$S(q_d(t)) \triangleq \begin{bmatrix} \cos(\theta_d(t)) & 0 \\ \sin(\theta_d(t)) & 0 \\ 0 & 1 \end{bmatrix}. \tag{3-8}$$

The term $v_d : [t_0, \infty) \to \mathbb{R}^2$ denotes the desired linear and angular velocity defined as

$$v_d(t) \triangleq \begin{bmatrix} v_{d1}(t) & v_{d2}(t) \end{bmatrix}^{\mathrm{T}}, \tag{3-9}$$

where $v_{d1}, v_{d2} : [t_0, \infty) \to \mathbb{R}$.

Remark 3.1. The design of the desired trajectory could include various survey/exploration trajectories generated in a variety of ways. Specifics of such trajectory designs do not impact the development in this chapter provided the trajectories are sufficiently smooth (see Assumption 3.5) and that the trajectory satisfies the subsequent developed dwell-time conditions (i.e., the trajectory visits a known landmark location within the maximum dwell-time).

Assumption 3.5. The signals $v_d(t), \dot{v}_d(t), q_d(t)$, and $\dot{q}_d(t)$ are upper bounded for all time such that $\|v_d(t)\| \leq \bar{v}_d \in \mathbb{R}_{>0}$, $\|\dot{v}_d(t)\| \leq \bar{\dot{v}}_d \in \mathbb{R}_{>0}$, $\|q_d(t)\| \leq \bar{q}_d \in \mathbb{R}_{>0}$, and $\|\dot{q}_d(t)\| \leq \bar{\dot{q}}_d \in \mathbb{R}_{>0}$ for

all $t \geq t_0$, where $\bar{v}_d, \bar{\dot{v}}_d, \bar{q}_d, \bar{\dot{q}}_d$ are known positive constants.

Assumption 3.6. The maximum velocity of the nonholonomic agent can be upper bounded by a known positive constant $v(t) \leq \bar{v}_{\max} \in \mathbb{R}_{>0}$.

Since feedback is available while $\phi(t) = a$, a trajectory tracking error $e : [t_0, \infty) \to \mathbb{R}^3$, is defined as

$$e(t) \triangleq \begin{bmatrix} x_e(t) & y_e(t) & \theta_e(t) \end{bmatrix}^{\mathrm{T}}, \tag{3-10}$$

where $e(t) \triangleq q(t) - q_d(t)$. When $\phi(t) = u$, feedback is unavailable, and a predictor model is introduced based on (3-1) as

$$\dot{\hat{q}}(t) \triangleq \operatorname{proj}\left(S\left(\hat{q}(t)\right) v(t)\right), \tag{3-11}$$

where $\operatorname{proj}(\cdot)$ denotes the continuous projection algorithm defined in [58, Section 4] and $\hat{q} : [t_0, \infty) \to \mathbb{R}^3$ denotes the position and orientation estimates, which is defined as

$$\hat{q}(t) \triangleq \begin{bmatrix} \hat{x}(t) & \hat{y}(t) & \hat{\theta}(t) \end{bmatrix}^{\mathrm{T}}. \tag{3-12}$$

The matrix $S : \mathbb{R}^3 \to \mathbb{R}^{3\times 2}$ is defined as

$$S\left(\hat{q}(t)\right) \triangleq \begin{bmatrix} \cos\left(\hat{\theta}(t)\right) & 0 \\ \sin\left(\hat{\theta}(t)\right) & 0 \\ 0 & 1 \end{bmatrix}. \tag{3-13}$$

While $q_d(t) \in \mathcal{F}^u$, the estimated trajectory tracking error $\hat{e} : [t_0, \infty) \to \mathbb{R}^3$ is used to track a desired trajectory and is defined as

$$\hat{e}(t) \triangleq \begin{bmatrix} x_{\hat{e}}(t) & y_{\hat{e}}(t) & \theta_{\hat{e}}(t) \end{bmatrix}^{\mathrm{T}}, \tag{3-14}$$

where $\hat{e}(t) \triangleq \hat{q}(t) - q_d(t)$. The state estimation error $\tilde{e} : [t_0, \infty) \to \mathbb{R}^3$ is defined as

$$\tilde{e}(t) \triangleq \begin{bmatrix} x_{\tilde{e}}(t) & y_{\tilde{e}}(t) & \theta_{\tilde{e}}(t) \end{bmatrix}^{\mathrm{T}}, \tag{3-15}$$

where $\tilde{e}(t) \triangleq q(t) - \hat{q}(t)$.[1]

[1] Unless otherwise specified, time dependence is suppressed in equations and definitions.

To facilitate the development, state transformations for (3-10) and (3-14) are defined according to [24] as

$$\begin{bmatrix} w_e \\ z_{1e} \\ z_{2e} \end{bmatrix} \triangleq \begin{bmatrix} -\theta_e c_\theta + 2s_\theta & -\theta_e s_\theta - 2c_\theta & 0 \\ 0 & 0 & 1 \\ c_\theta & s_\theta & 0 \end{bmatrix} e, \tag{3-16}$$

$$\begin{bmatrix} w_{\hat{e}} \\ z_{1\hat{e}} \\ z_{2\hat{e}} \end{bmatrix} \triangleq \begin{bmatrix} -\theta_{\hat{e}} c_{\hat{\theta}} + 2s_{\hat{\theta}} & -\theta_{\hat{e}} s_{\hat{\theta}} - 2c_{\hat{\theta}} & 0 \\ 0 & 0 & 1 \\ c_{\hat{\theta}} & s_{\hat{\theta}} & 0 \end{bmatrix} \hat{e}, \tag{3-17}$$

respectively, where $w_e, w_{\hat{e}}, z_{1e}, z_{2e}, z_{1\hat{e}}, z_{2\hat{e}} : [t_0, \infty) \to \mathbb{R}$, $s_\theta \triangleq \sin(\theta)$, and $c_\theta \triangleq \cos(\theta)$ for all $\theta \in \mathbb{R}$. Let the auxiliary tracking errors $z_e, z_{\hat{e}} : [t_0, \infty) \to \mathbb{R}^2$ be defined as $z_e \triangleq \begin{bmatrix} z_{1e} & z_{2e} \end{bmatrix}^{\mathrm{T}}$ and $z_{\hat{e}} \triangleq \begin{bmatrix} z_{1\hat{e}} & z_{2\hat{e}} \end{bmatrix}^{\mathrm{T}}$, respectively.

3.2.2 Control Development

Substituting (3-1)-(3-10) into the time-derivative of (3-16), and substituting (3-4), (3-6)-(3-9), and (3-11)-(3-14) into the time-derivative of (3-17), the open-loop system of the transformed state errors can be obtained as

$$\begin{bmatrix} \dot{w}_e \\ \dot{z}_e \end{bmatrix} = \begin{bmatrix} u_e^{\mathrm{T}} J^{\mathrm{T}} z_e + f_e + \chi_1 \\ u_e + \chi_2 \end{bmatrix}, \tag{3-18}$$

$$\begin{bmatrix} \dot{w}_{\hat{e}} \\ \dot{z}_{\hat{e}} \end{bmatrix} = \begin{bmatrix} u_{\hat{e}}^{\mathrm{T}} J^{\mathrm{T}} z_{\hat{e}} + f_{\hat{e}} \\ u_{\hat{e}} \end{bmatrix}. \tag{3-19}$$

In (3-18) and (3-19), the matrix $J \in \mathbb{R}^{2\times 2}$ is defined as

$$J \triangleq \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \tag{3-20}$$

which satisfies $J^{\mathrm{T}} = -J$, $J^{\mathrm{T}}J = I_{2\times 2}$. Moreover, $f_e, f_{\hat{e}} : [t_0, \infty) \to \mathbb{R}$ are defined as

$$f_e \triangleq 2\left(v_{d2}z_{2e} - v_{d1}s_{z_{1e}}\right), \tag{3-21}$$

$$f_{\hat{e}} \triangleq 2\left(v_{d2}z_{2\hat{e}} - v_{d1}s_{z_{1\hat{e}}}\right), \tag{3-22}$$

respectively. In (3-18), $\chi_1 : [t_0, \infty) \to \mathbb{R}$, $\chi_2 : [t_0, \infty) \to \mathbb{R}^2$ are auxiliary signals defined as

$$\chi_1 \triangleq 2\left(d_1 s_\theta - d_2 c_\theta\right) + d_3\left(z_{2e} + \frac{z_{1e}}{2}\left(w_e + z_{1e}z_{2e}\right)\right) - z_{1e}\left(d_1 c_\theta + d_2 s_\theta\right), \tag{3-23}$$

$$\chi_2 \triangleq \begin{bmatrix} d_3 \\ d_1 c_\theta + d_2 s_\theta - \frac{d_3}{2}\left(w_e + z_{1e}z_{2e}\right) \end{bmatrix}, \tag{3-24}$$

respectively. Auxiliary controllers $u_e, u_{\hat{e}} : [t_0, \infty) \to \mathbb{R}^2$ motivated by [24], are defined as

$$u_e \triangleq T_e^{-1}v_e - \begin{bmatrix} v_{d2} \\ v_{d1}c_{\theta_e} \end{bmatrix}, \tag{3-25}$$

$$u_{\hat{e}} \triangleq T_{\hat{e}}^{-1}v_{\hat{e}} - \begin{bmatrix} v_{d2} \\ v_{d1}c_{\theta_{\hat{e}}} \end{bmatrix}, \tag{3-26}$$

respectively. Note that $u_e = \begin{bmatrix} u_{1e} & u_{2e} \end{bmatrix}^{\mathrm{T}}$ and $u_{\hat{e}} = \begin{bmatrix} u_{1\hat{e}} & u_{2\hat{e}} \end{bmatrix}^{\mathrm{T}}$. The matrices $T_e, T_{\hat{e}} : [t_0, \infty) \to \mathbb{R}^{2\times 2}$ are motivated by [24], where

$$T_e \triangleq \begin{bmatrix} \left(x_e s_\theta - y_e c_\theta\right) & 1 \\ 1 & 0 \end{bmatrix}, \tag{3-27}$$

$$T_{\hat{e}} \triangleq \begin{bmatrix} \left(x_{\hat{e}} s_{\hat{\theta}} - y_{\hat{e}} c_{\hat{\theta}}\right) & 1 \\ 1 & 0 \end{bmatrix}. \tag{3-28}$$

The auxiliary signals $v_e, v_{\hat{e}} : [t_0, \infty) \to \mathbb{R}^2$ are designed to be the implementable input velocities to the system, specifically,

$$v = \begin{cases} v_e, & \phi = a, \\ v_{\hat{e}}, & \phi = u, \end{cases} \tag{3-29}$$

Substituting (3-27) into (3-25) and substituting (3-28) into (3-26) yields

$$v_e = T_e u_e + \begin{bmatrix} v_{d1} c_{\theta_e} + v_{d2} \left(x_e s_\theta - y_e c_\theta \right) \\ v_{d2} \end{bmatrix}, \tag{3-30}$$

$$v_{\hat{e}} = T_{\hat{e}} u_{\hat{e}} + \begin{bmatrix} v_{d1} c_{\theta_{\hat{e}}} + v_{d2} \left(x_{\hat{e}} s_{\hat{\theta}} - y_{\hat{e}} c_{\hat{\theta}} \right) \\ v_{d2} \end{bmatrix}, \tag{3-31}$$

respectively. To facilitate the subsequent development, let $\tilde{z}_e, \tilde{z}_{\hat{e}} : [t_0, \infty) \to \mathbb{R}^2$ be defined as

$$\tilde{z}_e \triangleq z_{de} - z_e, \tag{3-32}$$

$$\tilde{z}_{\hat{e}} \triangleq z_{d\hat{e}} - z_{\hat{e}}, \tag{3-33}$$

respectively, where auxiliary signals $z_{de}, z_{d\hat{e}} : [t_0, \infty) \to \mathbb{R}^2$ are subsequently designed. The auxiliary controllers u_e and $u_{\hat{e}}$ in (3-25) and (3-26) are designed as

$$u_e \triangleq u_{ae} - k_{2e} z_e, \tag{3-34}$$

$$u_{\hat{e}} \triangleq u_{a\hat{e}} - k_{2\hat{e}} z_{\hat{e}}, \tag{3-35}$$

respectively, where the auxiliary control terms $u_{ae}, u_{a\hat{e}} : [t_0, \infty) \to \mathbb{R}^2$ are defined as

$$u_{ae} \triangleq \left(\frac{k_{1e} w_e + f_e}{\delta_{de}^2} \right) J z_{de} + \Omega_{1e} z_{de}, \tag{3-36}$$

$$u_{a\hat{e}} \triangleq \left(\frac{k_{1\hat{e}} w_{\hat{e}} + f_{\hat{e}}}{\delta_{d\hat{e}}^2} \right) J z_{d\hat{e}} + \Omega_{1\hat{e}} z_{d\hat{e}}, \tag{3-37}$$

respectively. In (3-36) and (3-37), motivated by [24], the time-derivative of z_{de} and $z_{d\hat{e}}$ can be designed as

$$\dot{z}_{de} \triangleq \frac{\dot{\delta}_{de}}{\delta_{de}} z_{de} + \left(\frac{k_{1e} w_e + f_e}{\delta_{de}^2} + w_e \Omega_{1e} \right) J z_{de}, \tag{3-38}$$

$$\dot{z}_{d\hat{e}} \triangleq \frac{\dot{\delta}_{d\hat{e}}}{\delta_{d\hat{e}}} z_{d\hat{e}} + \left(\frac{k_{1\hat{e}} w_{\hat{e}} + f_{\hat{e}}}{\delta_{d\hat{e}}^2} + w_{\hat{e}} \Omega_{1\hat{e}} \right) J z_{d\hat{e}}, \tag{3-39}$$

respectively, where

$$z_{de}^{\mathrm{T}}(t_i^a) z_{de}(t_i^a) = \delta_{de}^2(t_i^a), z_{d\hat{e}}^{\mathrm{T}}(t_i^u) z_{d\hat{e}}(t_i^u) = \delta_{d\hat{e}}^2(t_i^u). \tag{3-40}$$

As described in [59], ${z_d}^{\mathrm{T}} z_d = \delta_d^2$, implying

$$z_{de}^{\mathrm{T}} z_{de} = \delta_{de}^2, z_{d\hat{e}}^{\mathrm{T}} z_{d\hat{e}} = \delta_{d\hat{e}}^2. \tag{3-41}$$

The auxiliary terms $\Omega_{1e}, \Omega_{1\hat{e}}, \delta_{de}, \delta_{d\hat{e}} : [t_0, \infty) \to \mathbb{R}$ in (3-36)-(3-39) are defined as

$$\Omega_{1e} \triangleq k_{2e} + \frac{\dot{\delta}_{de}}{\delta_{de}} + w_e \left(\frac{k_{1e} w_e + f_e}{\delta_{de}^2} \right), \tag{3-42}$$

$$\Omega_{1\hat{e}} \triangleq k_{2\hat{e}} + \frac{\dot{\delta}_{d\hat{e}}}{\delta_{d\hat{e}}} + w_{\hat{e}} \left(\frac{k_{1\hat{e}} w_{\hat{e}} + f_{\hat{e}}}{\delta_{d\hat{e}}^2} \right), \tag{3-43}$$

$$\delta_{de} \triangleq \alpha_0 \exp\left(-\alpha_1 \left(t - t_i^a\right)\right) + \varepsilon_1, \tag{3-44}$$

$$\delta_{d\hat{e}} \triangleq \alpha_2 \exp\left(-\alpha_3 \left(t - t_i^u\right)\right) + \varepsilon_2, \tag{3-45}$$

respectively, where $k_{1\hat{e}}, k_{2\hat{e}}, \alpha_0, \alpha_1, \alpha_2, \alpha_3, \varepsilon_1, \varepsilon_2 \in \mathbb{R}_{>0}$ are constant parameters [59]. Let $k_{1e}, k_{2e} : [t_0, \infty) \to \mathbb{R}_{>0}$ be time-varying control parameters designed as

$$k_{1e} \triangleq k_s + 2 k_{n2} \kappa_1^2, \tag{3-46}$$

$$k_{2e} \triangleq k_s + 2 k_{n2} \kappa_2^2, \tag{3-47}$$

respectively, where $k_s, k_{n2} \in \mathbb{R}_{>0}$ are constant control parameters. Let $\kappa_1, \kappa_2 : [t_0, \infty) \to \mathbb{R}_{>0}$ be bounding functions such that

$$\kappa_1 \leq 2\bar{d}_4 + \left(\bar{d}_3 + \bar{d}_4 + \frac{\bar{d}_3}{2}\|w_e\|\right)(\|z_{de}\| + \|\tilde{z}_e\|) + \frac{\bar{d}_3}{2}(\|z_{de}\| + \|\tilde{z}_e\|)^3, \tag{3-48}$$

$$\kappa_2 \leq \sqrt{\bar{d}_3^2 + \left(\bar{d}_4 + \frac{\bar{d}_3}{2}(\|z_{de}\| + \|\tilde{z}_e\|)^2 + \frac{\bar{d}_3}{2}\|w_e\|\right)^2}. \tag{3-49}$$

Based on the upper bounds in (3-23) and (3-24),

$$\|\chi_1\| \leq \kappa_1, \ \|\chi_2\| \leq \kappa_2. \tag{3-50}$$

Substituting (3-32), (3-34), (3-36), (3-38), (3-41) and (3-42) into (3-18) yields the closed-loop trajectory tracking error

$$\begin{bmatrix} \dot{w}_e \\ \dot{\tilde{z}}_e \end{bmatrix} = \begin{bmatrix} -k_{1e}w_e + u_{ae}^{\mathrm{T}} J\tilde{z}_e + \chi_1 \\ -k_{2e}\tilde{z}_e + w_e J u_{ae} - \chi_2 \end{bmatrix}. \tag{3-51}$$

Similarly, substituting (3-33), (3-35), (3-37), (3-39), (3-41) and (3-43) into (3-19) yields the closed-loop estimated trajectory tracking error

$$\begin{bmatrix} \dot{w}_{\hat{e}} \\ \dot{\tilde{z}}_{\hat{e}} \end{bmatrix} = \begin{bmatrix} -k_{1\hat{e}}w_{\hat{e}} + u_{a\hat{e}}^{\mathrm{T}} J\tilde{z}_{\hat{e}} \\ -k_{2\hat{e}}\tilde{z}_{\hat{e}} + w_{\hat{e}} J u_{a\hat{e}} \end{bmatrix}. \tag{3-52}$$

3.2.3 State Estimate

When $\phi = a$, a reset map (cf., [18, 20, 17]), $\psi : \hat{q}(t) \mapsto q(t)$, is used. During this time, the closed-loop error system in (3-51) is used for analysis given the error in (3-15) is zero. However, when $\phi = u$, the state is estimated using the predictor in (3-11) and the closed-loop error system must be used for analysis to develop the subsequent maximum dwell-time condition. To develop the maximum dwell-time condition, the time-derivative of the error in (3-15) must be used in addition to (3-52) to determine the overall error. Taking the time-derivative of (3-15) and substituting in the dynamics (3-1) and the predictor (3-11) yields

$$\dot{\tilde{e}} = S(q)\, v - \text{proj}\left(S(\hat{q})\, v\right) + d,\ \phi = u. \tag{3-53}$$

3.3 Stability Analysis

In the analysis, the stability of $e(t)$ is analyzed for all time, and $\hat{e}(t)$ and $\tilde{e}(t)$ are included to facilitate the stability analysis. The following theorem is provided to show the trajectory tracking error defined in (3-10) is bounded when feedback is available.

Theorem 3.1. *When feedback is available, i.e., $\phi = a$ for all $t \in [t_i^a, t_i^u)$, the control laws given in (3-30), (3-34), (3-36), (3-38), (3-42), (3-44), (3-46), and (3-47) ensure the trajectory tracking error defined in (3-10) is GUUB in the sense that*

$$\|e(t)\| \leq \sqrt{3}\sqrt{\beta_0\left(t_i^a\right)\exp\left(-\lambda_e\left(t - t_i^a\right)\right) + \beta_1} + \sqrt{3}\left(\beta_2\left(t_i^a\right)\exp\left(-\lambda_e\left(t - t_i^a\right)\right) + \beta_3\right), \tag{3-54}$$

where $\lambda_e \in \mathbb{R}_{>0}$ is a known constant, $\beta_0\left(t_i^a\right) \triangleq \left(\frac{3}{2} + \alpha_0 + \varepsilon_1\right)^2 \left(\left\|\rho_e\left(t_i^a\right)\right\|^2 - \frac{1}{2\lambda_e k_{n2}}\right)$,
$\beta_1 \triangleq \left(\frac{3}{2} + \alpha_0 + \varepsilon_1\right)^2 \frac{1}{2\lambda_e k_{n2}}$, $\beta_2\left(t_i^a\right) \triangleq \left(\frac{1}{2}\left\|\rho_e\left(t_i^a\right)\right\|^2 - \frac{1}{4\lambda_e k_{n2}}\right)$, and
$\beta_3 \triangleq \frac{1}{4\lambda_e k_{n2}} + \frac{1}{2}\varepsilon_1^2 + \varepsilon_1 + \left(\frac{1}{2}\alpha_0^2 + \alpha_0\varepsilon_1 + \alpha_0\right)$.

Proof. Let the candidate Lyapunov functional $V_e : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ be defined as

$$V_e\left(w_e(t), \tilde{z}_e(t)\right) \triangleq \frac{1}{2}w_e^2(t) + \frac{1}{2}\tilde{z}_e^{\mathrm{T}}(t)\,\tilde{z}_e(t). \tag{3-55}$$

While $\phi = a$, i.e., $t \in [t_i^a, t_i^u)$, taking the time-derivative of (3-55) and substituting (3-46), (3-47), (3-50) and (3-51), yields

$$\dot{V}_e(t) \leq -\lambda_e V_e(t) + \frac{1}{4k_{n2}}, \tag{3-56}$$

where $\lambda_e \triangleq 2k_s \in \mathbb{R}_{>0}$. Invoking the Comparison Lemma in [56, Lemma 3.4] on (3-56) yields

$$V_e(t) \leq V_e\left(t_i^a\right)\exp\left(-\lambda_e\left(t - t_i^a\right)\right) - \frac{1}{4\lambda_e k_{n2}}\left(\exp\left(-\lambda_e\left(t - t_i^a\right)\right) - 1\right). \tag{3-57}$$

Let $\rho_e : [t_0, \infty) \to \mathbb{R}^3$ denote a composite error for the trajectory tracking error defined as

$$\rho_e(t) \triangleq \begin{bmatrix} w_e(t) & \tilde{z}_e^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}. \tag{3-58}$$

Using (3-55) and (3-58), the inequality in (3-57) can be written as

$$\|\rho_e(t)\| \leq \sqrt{\left\|\rho_e\left(t_i^a\right)\right\|^2 \exp\left(-\lambda_e\left(t-t_i^a\right)\right) - \frac{1}{2\lambda_e k_{n2}}\left(\exp\left(-\lambda_e\left(t-t_i^a\right)\right)-1\right)}. \tag{3-59}$$

From (3-58) and (3-59), the error signals $w_e(t)$, $\tilde{z}_e(t) \in \mathcal{L}_\infty$. Since $\delta_{de}(t) \in \mathcal{L}_\infty$ by (3-44), using (3-41) leads to $z_{de}(t) \in \mathcal{L}_\infty$. Since $\tilde{z}_e(t)$, $z_{de}(t) \in \mathcal{L}_\infty$, using (3-32) implies $z_e(t) \in \mathcal{L}_\infty$. Specifically, using (3-32), (3-41), (3-44), (3-58), and (3-59) along with the triangle inequality, it follows that $z_e(t)$ defined in (3-16) is GUUB as

$$\begin{aligned}\|z_e\| &\leq \|z_{de}\| + \|\tilde{z}_e\| \\ &\leq \alpha_0 \exp\left(-\alpha_1\left(t-t_i^a\right)\right) + \varepsilon_1 + \\ &\quad \sqrt{\left\|\rho_e\left(t_i^a\right)\right\|^2 \exp\left(-\lambda_e\left(t-t_i^a\right)\right) - \frac{1}{2\lambda_e k_{n2}}\left(\exp\left(-\lambda_e\left(t-t_i^a\right)\right)-1\right)}.\end{aligned} \tag{3-60}$$

The inverse of (3-16) yields the trajectory tracking error in (3-10) as

$$e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \frac{1}{2}s_\theta & 0 & \frac{1}{2}\left(\theta_e s_\theta + 2c_\theta\right) \\ -\frac{1}{2}c_\theta & 0 & -\frac{1}{2}\left(\theta_e c_\theta - 2s_\theta\right) \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} w_e \\ z_{1e} \\ z_{2e} \end{bmatrix}. \tag{3-61}$$

Using (3-60), $z_e(t) \in \mathcal{L}_\infty$, and therefore $z_{1e}(t), z_{2e}(t) \in \mathcal{L}_\infty$, (3-61) implies $\theta_e(t) \in \mathcal{L}_\infty$. Given $\rho_e(t)$, $\theta_e(t) \in \mathcal{L}_\infty$, using (3-61) implies $x_e(t)$, $y_e(t) \in \mathcal{L}_\infty$. Specifically, using (3-58)-(3-61), each component of $e(t)$ can be bounded as

$$\|x_e(t)\|, \|y_e(t)\|, \|\theta_e(t)\| \leq \sqrt{\beta_0\left(t_i^a\right)\exp\left(-\lambda_e\left(t-t_i^a\right)\right) + \beta_1} + \beta_2\left(t_i^a\right)\exp\left(-\lambda_e\left(t-t_i^a\right)\right) + \beta_3. \tag{3-62}$$

Using (3-10) and (3-62), the result in (3-54) follows. From Assumption 3.5, $v_d(t)$, $\dot{v}_d(t)$, $q_d(t)$, $\dot{q}_d(t) \in \mathcal{L}_\infty$, so the signal $f_e(z_e, v_d) \in \mathcal{L}_\infty$. Using (3-46)-(3-49), $k_{1e}(t)$, $k_{2e}(t) \in \mathcal{L}_\infty$, and since $w_e(t)$, $\delta_{de}(t)$, $\dot{\delta}_{de}(t) \in \mathcal{L}_\infty$, then $\Omega_{1e}(t) \in \mathcal{L}_\infty$ based on (3-42). Given $k_{1e}(t)$, $w_e(t)$, $f_e(t)$, $\delta_{de}(t)$, J, $z_{de}(t)$, $\Omega_{1e}(t) \in \mathcal{L}_\infty$, $u_{ae}(t) \in \mathcal{L}_\infty$. Therefore, given $u_{ae}(t)$, $k_{2e}(t)$, $z_e(t) \in \mathcal{L}_\infty$, $u_e(t) \in \mathcal{L}_\infty$. Given $x_e(t)$, $y_e(t) \in \mathcal{L}_\infty$, (3-27) implies $T_e(t) \in \mathcal{L}_\infty$. Given $T_e(t)$, $u_e(t)$, $v_d(t)$, $x_e(t)$, $y_e(t) \in \mathcal{L}_\infty$, (3-30) implies $v_e(t) \in \mathcal{L}_\infty$. Finally, (3-29) implies $v(t) \in \mathcal{L}_\infty$ because

$v_e(t) \in \mathcal{L}_\infty$ for $\phi = a$. □

While Theorem 3.1 proves the stability of $e(t)$ when $\phi = a$, Theorems 3.2 and 3.3 prove the stability of $e(t)$ when $\phi = u$.

Theorem 3.2. *When feedback is unavailable, i.e., $\phi = u$ for all $t \in \left[t_i^u, t_{i+1}^a\right)$, the control laws given in (3-31), (3-35), (3-37), (3-39), (3-43), and (3-45) ensure the estimated trajectory tracking error defined in (3-14) is GUUB in the sense that*

$$\|\hat{e}(t)\| \leq \sqrt{3}\left(\sqrt{\beta_4\left(t_i^u\right)\exp\left(-\lambda_{\hat{e}}\left(t-t_i^u\right)\right)}\right) + \sqrt{3}\left(\beta_5\left(t_i^u\right)\exp\left(-\lambda_{\hat{e}}\left(t-t_i^u\right)\right)+\beta_6\right), \tag{3-63}$$

where $\lambda_{\hat{e}} \in \mathbb{R}_{>0}$ is a known constant, $\beta_4\left(t_i^u\right) \triangleq \left(\frac{3}{2}+\alpha_2+\varepsilon_2\right)^2 \left\|\rho_{\hat{e}}\left(t_i^u\right)\right\|^2$, $\beta_5\left(t_i^u\right) \triangleq \frac{1}{2}\left\|\rho_{\hat{e}}\left(t_i^u\right)\right\|^2$, and $\beta_6 \triangleq \frac{1}{2}\varepsilon_2^2+\varepsilon_2+\left(\frac{1}{2}\alpha_2^2+\alpha_2\varepsilon_2+\alpha_2\right)$.

Proof. Let the candidate Lyapunov functional $V_{\hat{e}} : \mathbb{R}\times\mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ be defined as

$$V_{\hat{e}}\left(w_{\hat{e}}(t), \tilde{z}_{\hat{e}}(t)\right) \triangleq \frac{1}{2}w_{\hat{e}}^2(t) + \frac{1}{2}\tilde{z}_{\hat{e}}^{\mathrm{T}}(t)\tilde{z}_{\hat{e}}(t). \tag{3-64}$$

While $\phi = u$, i.e., $t \in \left[t_i^u, t_{i+1}^a\right)$, taking the time-derivative of (3-64) then substituting (3-52), yields

$$\dot{V}_{\hat{e}}(t) \leq -\lambda_{\hat{e}} V_{\hat{e}}(t), \tag{3-65}$$

where $\lambda_{\hat{e}} \triangleq 2\min\left(k_{1\hat{e}}, k_{2\hat{e}}\right) \in \mathbb{R}_{>0}$. The solution to (3-65) is obtained by the Comparison Lemma in [56, Lemma 3.4] yielding

$$V_{\hat{e}}(t) \leq V_{\hat{e}}\left(t_i^u\right)\exp\left(-\lambda_{\hat{e}}\left(t-t_i^u\right)\right). \tag{3-66}$$

Let $\rho_{\hat{e}} : [t_0, \infty) \to \mathbb{R}^3$ denote a composite error for the estimated trajectory tracking error defined as

$$\rho_{\hat{e}}(t) \triangleq \begin{bmatrix} w_{\hat{e}}(t) & \tilde{z}_{\hat{e}}^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}. \tag{3-67}$$

Using (3-64) and (3-67), the inequality in (3-66) can be written as

$$\left\|\rho_{\hat{e}}(t)\right\| \leq \sqrt{\left\|\rho_{\hat{e}}\left(t_i^u\right)\right\|^2 \exp\left(-\lambda_{\hat{e}}\left(t-t_i^u\right)\right)}. \tag{3-68}$$

From (3-67) and (3-68), the error signals $w_{\hat{e}}(t), \tilde{z}_{\hat{e}}(t) \in \mathcal{L}_\infty$. Since $\delta_{d\hat{e}}(t) \in \mathcal{L}_\infty$ by (3-45), using (3-41) leads to $z_{d\hat{e}}(t) \in \mathcal{L}_\infty$. Since $\tilde{z}_{\hat{e}}(t), z_{d\hat{e}}(t) \in \mathcal{L}_\infty$, using (3-33) implies $z_{\hat{e}}(t) \in \mathcal{L}_\infty$. Specifically, using (3-33), (3-41), (3-45), (3-67), and (3-68) along with the triangle inequality, it follows that $z_{\hat{e}}(t)$ defined in (3-17) is GUUB as

$$\begin{aligned} \|z_{\hat{e}}\| &\leq \|z_{d\hat{e}}\| + \|\tilde{z}_{\hat{e}}\| \\ &\leq \alpha_2 \exp\left(-\alpha_3 \left(t - t_i^u\right)\right) + \varepsilon_2 + \sqrt{\left\|\rho_{\hat{e}}\left(t_i^u\right)\right\|^2 \exp\left(-\lambda_{\hat{e}}\left(t - t_i^u\right)\right)}. \end{aligned} \tag{3-69}$$

The inverse of (3-17) yields the estimated trajectory tracking error in (3-14) as

$$\hat{e} = \begin{bmatrix} x_{\hat{e}} \\ y_{\hat{e}} \\ \theta_{\hat{e}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}s_{\hat{\theta}} & 0 & \frac{1}{2}\left(\theta_{\hat{e}} s_{\hat{\theta}} + 2c_{\hat{\theta}}\right) \\ -\frac{1}{2}c_{\hat{\theta}} & 0 & -\frac{1}{2}\left(\theta_{\hat{e}} c_{\hat{\theta}} - 2s_{\hat{\theta}}\right) \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} w_{\hat{e}} \\ z_{1\hat{e}} \\ z_{2\hat{e}} \end{bmatrix}. \tag{3-70}$$

Using (3-69), $z_{\hat{e}}(t) \in \mathcal{L}_\infty$, and therefore $z_{1\hat{e}}(t), z_{2\hat{e}}(t) \in \mathcal{L}_\infty$, (3-70) implies $\theta_{\hat{e}}(t) \in \mathcal{L}_\infty$. Given $\rho_{\hat{e}}(t), \theta_{\hat{e}}(t) \in \mathcal{L}_\infty$, using (3-70) implies $x_{\hat{e}}(t), y_{\hat{e}}(t) \in \mathcal{L}_\infty$. Specifically, using (3-67)-(3-70), each component of $\hat{e}$ can be bounded by

$$\|x_{\hat{e}}(t)\|, \|y_{\hat{e}}(t)\|, \|\theta_{\hat{e}}(t)\| \leq \sqrt{\beta_4\left(t_i^u\right) \exp\left(-\lambda_{\hat{e}}\left(t - t_i^u\right)\right)} + \beta_5\left(t_i^u\right) \exp\left(-\lambda_{\hat{e}}\left(t - t_i^u\right)\right) + \beta_6. \tag{3-71}$$

Using (3-14) and (3-71), the result in (3-63) can now be obtained. Using a similar approach as in Theorem 3.1, $v_{\hat{e}}(t) \in \mathcal{L}_\infty$; therefore, $v(t) \in \mathcal{L}_\infty$ while $\phi = u$. □

Remark 3.2. While $\phi = a$, i.e., $t \in [t_i^a, t_i^u)$, the trajectory tracking error $e(t)$ is bounded as shown in (3-54), and the state estimate error $\tilde{e}(t)$ is reset to zero, i.e., $\hat{q}(t) = q(t)$. Since $e(t) = \hat{e}(t) + \tilde{e}(t)$, the estimated tracking error $\hat{e}(t)$ is bounded for $\phi = a$.

Theorem 3.3. *For the trajectories of the switched system generated by the switching signal ϕ, (3-15) is GUUB provided the switching signal satisfies the maximum loss of feedback dwell-time condition*

$$\Delta t_i^u \leq \frac{1}{\lambda_u} \ln\left(\frac{\lambda_u}{2\varepsilon}\left(e_M - \beta_7\left(t_i^u\right)\right)^2 + 1\right), \tag{3-72}$$

where $\lambda_u \in \mathbb{R}_{>0}$ is a known constant, $\varepsilon \triangleq \frac{1}{2}\bar{d}^2 \in \mathbb{R}_{>0}$, $e_M > e_T \in \mathbb{R}_{>0}$ is a user-defined parameter,

$e_T \in \mathbb{R}_{>0}$ is a subsequently defined constant (see Theorem 3.4), and
$\beta_7(t_i^u) \triangleq \sqrt{3}\left(\sqrt{\beta_4(t_i^u)} + \beta_5(t_i^u) + \beta_6\right)$.

Proof. Let the candidate Lyapunov functional $V_{\tilde{e}} : \mathbb{R}^3 \to \mathbb{R}_{\geq 0}$ be defined as

$$V_{\tilde{e}}(\tilde{e}(t)) \triangleq \frac{1}{2}\tilde{e}^{\mathrm{T}}(t)\tilde{e}(t). \tag{3-73}$$

While $\phi = u$, i.e., $t \in \left[t_i^u, t_{i+1}^a\right)$, taking the time-derivative of (3-73) and substituting (3-53) yields

$$\dot{V}_{\tilde{e}}(t) \leq \lambda_u V_{\tilde{e}}(t) + \varepsilon, \tag{3-74}$$

where $\lambda_u \triangleq 2c+1 \in \mathbb{R}_{>0}$, and $c \in \mathbb{R}_{>0}$. The solution to (3-74) is obtained by the Comparison Lemma in [56, Lemma 3.4] yielding

$$V_{\tilde{e}}(t) \leq \frac{\varepsilon}{\lambda_u}\left(\exp\left(\lambda_u\left(t - t_i^u\right)\right) - 1\right). \tag{3-75}$$

From (3-73), the inequality in (3-75) can be written as

$$\|\tilde{e}(t)\| \leq \sqrt{\frac{2\varepsilon}{\lambda_u}\left(\exp\left(\lambda_u\left(t - t_i^u\right)\right) - 1\right)}. \tag{3-76}$$

To ensure the state estimation error $e(t)$ is bounded, a user-defined maximum bound $e_M \in \mathbb{R}_{>0}$ can be selected to ensure $\left\|e\left(t_{i+1}^a\right)\right\| \leq e_M$. Using (3-63) and (3-76) in $\left\|e\left(t_{i+1}^a\right)\right\| \leq \left\|\hat{e}\left(t_{i+1}^a\right)\right\| + \left\|\tilde{e}\left(t_{i+1}^a\right)\right\|$, the following inequality can be obtained

$$\sqrt{3}\left(\sqrt{\beta_4\left(t_i^u\right)} + \beta_5\left(t_i^u\right) + \beta_6\right) + \sqrt{\frac{2\varepsilon}{\lambda_u}\left(\exp\left(\lambda_u \Delta t_i^u\right) - 1\right)} \leq e_M. \tag{3-77}$$

When (3-77) is satisfied, $\left\|\hat{e}\left(t_{i+1}^a\right)\right\| + \left\|\tilde{e}\left(t_{i+1}^a\right)\right\| \leq e_M$, and since $\left\|e\left(t_{i+1}^a\right)\right\| \leq \left\|\hat{e}\left(t_{i+1}^a\right)\right\| + \left\|\tilde{e}\left(t_{i+1}^a\right)\right\|$, therefore $\left\|e\left(t_{i+1}^a\right)\right\| \leq e_M$. Using (3-77), Δt_i^u can be determined, yielding (3-72). □

To ensure $\|e(t)\|$ does not grow beyond a user-defined parameter $e_M \in \mathbb{R}_{>0}$ when $\phi = u$, the dwell-time condition in (3-72) is developed. Since $e(t) = \hat{e}(t) + \tilde{e}(t)$, by Theorem 3.2 and 3.3, the maximum dwell-time in (3-72) ensures $e(t) \in \mathcal{L}_\infty$ for $\phi = u$.

At the time instant $t = t_i^u$, $\hat{q}(t_i^u) = q(t_i^u)$ by design, implying $\hat{e}(t_i^u) = e(t_i^u)$ and $\tilde{e}(t_i^u) = 0$.

Therefore, the switched system is continuous at $t = t_i^u$ because $e(t_i^u) = \hat{e}(t_i^u) + \tilde{e}(t_i^u)$. At the time instant $t = t_{i+1}^a$, given $e(t) = \hat{e}(t) + \tilde{e}(t)$, using (3-77), then $\limsup_{t \to t_{i+1}^a} \|e(t)\| \leq e_M$. Therefore, the trajectory tracking error $e(t)$ is finite.

Zeno behavior occurs when the difference between $t_{i+1}^a - t_i^a$ is arbitrarily small, which implies the agent requires continuous state feedback. Since the objective is to explore the feedback-denied region for a finite time period, then it is critical to show that the difference between consecutive return times, i.e., $t_{i+1}^a - t_i^a$ is lower bounded by a finite positive constant. While $\phi = u$, let $t_{\text{travel}} \in \left[t_i^u, t_{i+1}^a\right)$ represent the minimum time it would take the nonholonomic agent to travel between consecutive feedback regions. Therefore, the maximum dwell-time condition has a lower constant bound, i.e. $t_{\text{travel}} \leq \Delta t_i^u$, where $t_{\text{travel}} = \frac{d_k}{\bar{v}_{\max}}$, $d_k \in \mathbb{R}_{>0}$ denotes the actual distance the nonholonomic agent travels, and $k \in \mathbb{Z}_{\geq 0}$. By the definition of Δt_i^a and Δt_i^u, $t_{i+1}^a - t_i^a = \Delta t_i^u + \Delta t_i^a$. Since $t_{\text{travel}} \leq \Delta t_i^u$, $t_{i+1}^a - t_i^a \geq t_{\text{travel}}$. Therefore, Zeno behavior is excluded.

The following theorem is provided to ensure the trajectory tracking error converges within a desired neighborhood.

Theorem 3.4. *Given trajectories of the switched system generated by the switching signal ϕ, (3-10) is GUUB provided the switching signal satisfies the minimum feedback availability dwell-time condition given by*

$$\Delta t_i^a > \max\left\{ \frac{2}{\lambda_e} \ln\left(\frac{\sqrt{\beta_0\left(t_i^a\right)} + \beta_2\left(t_i^a\right)}{\frac{e_T}{\sqrt{3}} - \sqrt{\beta_1} - \beta_3} \right), 0 \right\}, \tag{3-78}$$

where $e_T > \sqrt{3}\left(\sqrt{\beta_1} + \beta_3\right) \in \mathbb{R}_{>0}$ is a user-defined constant.

Proof. While $\phi = a$, i.e., $t \in [t_i^a, t_i^u)$, the upper bound of the trajectory tracking error, $e(t)$, is given by (3-54). This inequality is used to ensure $e(t)$ converges within a desired neighborhood $e_T \in \mathbb{R}_{>0}$, i.e., $\|e(t)\| \leq e_T$. Using the user-defined lower threshold $e_T \in \mathbb{R}_{>0}$, Δt_i^a is determined to satisfy

$$\sqrt{3}\left(\sqrt{\beta_0\left(t_i^a\right)\exp\left(-\lambda_e \Delta t_i^a\right) + \beta_1}\right) + \sqrt{3}\left(\beta_2\left(t_i^a\right)\exp\left(-\lambda_e \Delta t_i^a\right) + \beta_3\right) \leq e_T. \tag{3-79}$$

Hence, (3-79) holds provided (3-78) is satisfied. □

As described in Assumption 3.3, the vehicle initially starts inside a feedback zone where it then starts tracking the desired trajectory to reach a different feedback zone before exceeding the maximum dwell-time described in Theorem 3.3. During exploration, without absolute feedback, i.e., dead-reckoning, the state estimate error is upper bounded as in (3-76). Once the agent reaches a feedback zone, the estimated state is instantaneously reset to the true state as described in Section 3.2.3, and the state estimate error $\tilde{e}(t)$ is reset to zero. While the agent has feedback, the relative position of the feedback zones are known. With this approach, the objective of exploring unknown regions with intermittent state feedback of a nonholonomic vehicle can be achieved provided the exploration trajectory enables the agent to satisfy the maximum and minimum dwell-time conditions.

3.4 Experiment

An experiment is performed using a WMR with Kobuki base. In lieu of using motion capture system to determine the actual pose of the agent, an attention guided camera localization (AtLoc) architecture developed in [60] is used to estimate the pose of the agent by keeping a known landmark within the FOV of an onboard monocular camera. As depicted in Figure 3-1, a ground WMR is used to model the nonholonomic kinematics described in (3-1). To acquire the pose estimates of the WMR, five customized landmarks are placed at centers of the feedback regions. The WMR tracks a circle trajectory centered at each of the feedback regions with radius of 0.8 m. A differentiable time-varying controller proposed in [61] is used to track the circle trajectory as

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -k_1 e_1 + v_{1r}\cos(e_3) \\ -v_{1r}\frac{\sin(e_3)}{e_3}e_2 - k_2 e_3 + v_{2r} \end{bmatrix}, \tag{3-80}$$

where $k_1, k_2 \in \mathbb{R}_{>0}$ are constant control gains selected as $k_1 = 0.5$ and $k_2 = 3$, $v_{1r}, v_{2r} \in \mathbb{R}$ are the desired linear and angular velocities, respectively, selected as

$$\begin{bmatrix} v_{1r} \\ v_{2r} \end{bmatrix} = \begin{bmatrix} \sqrt{(-r\sin(2\pi f)\,2\pi f)^2 + (r\cos(2\pi f)\,2\pi f)^2} \\ 2\pi f \end{bmatrix},$$

and $r, f \in \mathbb{R}_{>0}$ denote the radius and frequency of the circle trajectory, respectively, selected as $r = 0.8$ m and $f = 0.05$. The errors $e_1, e_2, e_3 \in \mathbb{R}$ in (3-80) are defined as

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta)\tilde{x} + \sin(\theta)\tilde{y} \\ -\sin(\theta)\tilde{x} + \cos(\theta)\tilde{y} \\ \tilde{\theta} \end{bmatrix},$$

where $\tilde{x}, \tilde{y}, \tilde{\theta} \in \mathbb{R}$ are defined as

$$\begin{aligned} \tilde{x} &= x - x_d \\ \tilde{y} &= y - y_d \\ \tilde{\theta} &= \theta - \theta_d. \end{aligned}$$

While the WMR is tracking the circle trajectory with respect to a landmark, the viewing direction of onboard monocular camera is set to be perpendicular to the x direction of the WMR (i.e., keeping a landmark within the FOV). To pretrain the network using AtLoc, high definition images of the landmarks, true pose of the WMR (i.e., provided by a NaturalPoint, Inc. OptiTrack motion capture system), pose of the landmarks are recorded. The recorded data is used as input to the customized AtLoc architecture for pretraining. The loss function used for the training is selected as

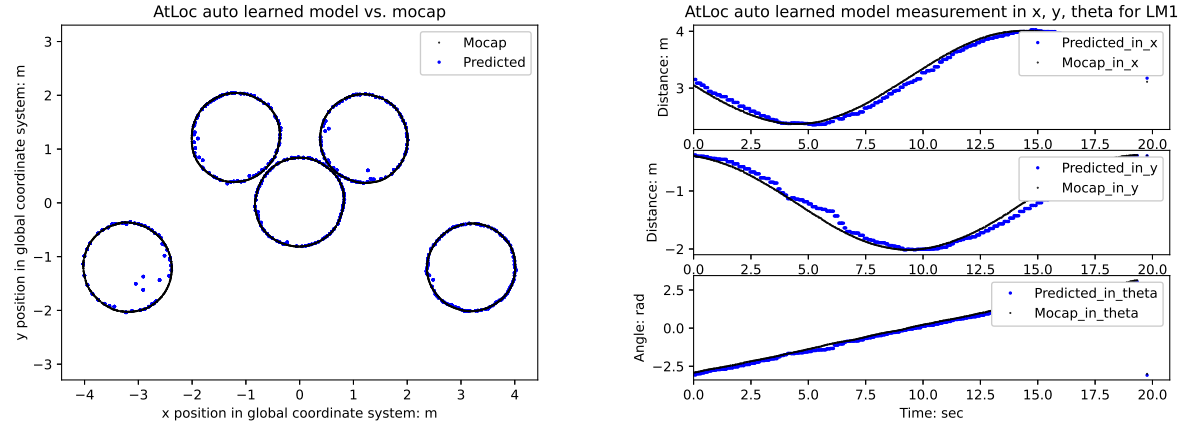
$$\mathbb{L} = e^{-\alpha} \|\hat{x} - x\| + \alpha + e^{-\beta} \|\hat{y} - y\| + \beta + e^{-\gamma} \left\|\hat{\theta} - \theta\right\| + \gamma,$$

where $\alpha, \beta, \gamma \in \mathbb{R}_{>0}$ are the training parameters initialized with $\alpha_0 = 0$, $\beta_0 = 0$, $\gamma_0 = -2$, $x, y, \theta \in \mathbb{R}$ are the actual state information provided by the motion capture system, $\hat{x}, \hat{y}, \hat{\theta}$ are the outputs from the pretrained network, and the learning rate is selected to be 0.001.

Figures 3-2a-3-4b show the results of pose estimates generated from the pretrain AtLoc model and the actual pose information provided by the motion capture system. Specifically, Figure 3-2a shows the estimated and actual pose states of the WMR while keeping one of the five landmarks within the FOV in x and y directions. Figures 3-2b-3-4b show the generated estimated pose and the actual pose of the WMR for each of the landmarks in x, y, and θ directions. Table

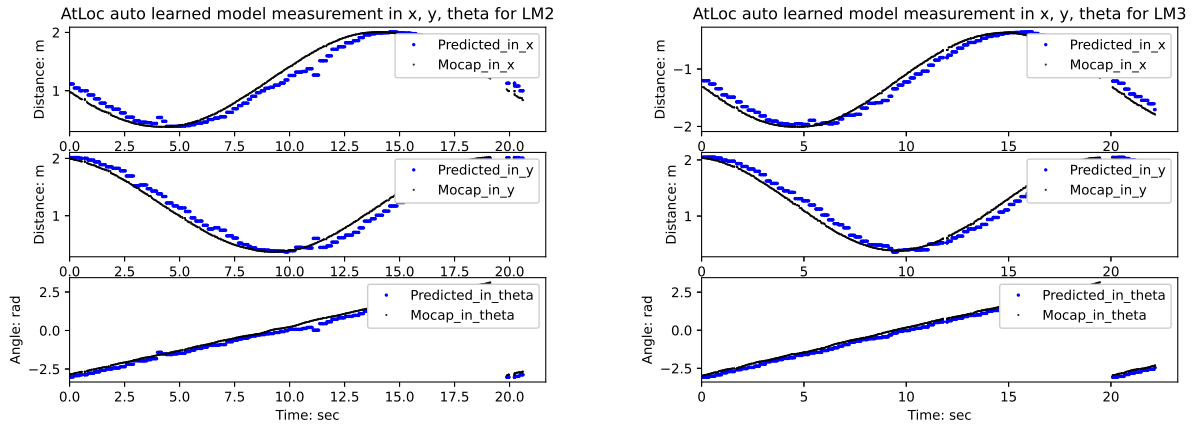
3-1 shows the comparison between the tested and AtLoc results in [60], and we achieved better performance in position estimation, but we didn't achieve better result in orientation estimation.

Figure 3-1. The experimental setup using one WMR with an onboard monocular camera and five customized landmarks. The centroid of each landmark is at the center of a feedback available region, and the radius of each feedback available region is selected as 0.8 m. The WMR can obtain pose estimates (i.e., true state information) when one of the landmarks is within the FOV of the monocular camera.



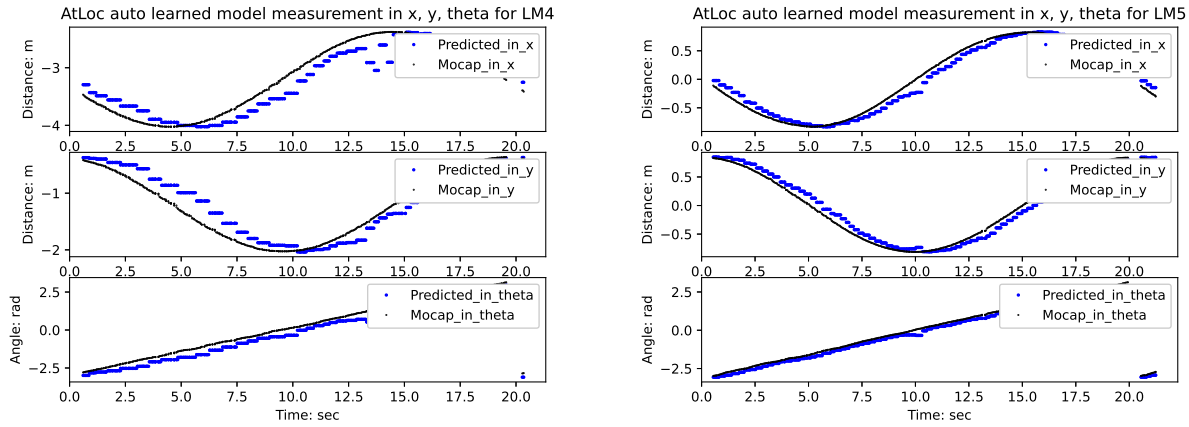
(a) (b)

Figure 3-2. Pose estimation using the AtLoc model versus the pose information provided by motion capture system for (a) five landmarks, and (b) landmark 1.



(a) (b)

Figure 3-3. Pose estimation using the AtLoc model versus the pose information provided by motion capture system for (a) landmark 2, and (b) landmark 3.



(a) (b)

Figure 3-4. Pose estimation using the AtLoc model versus the pose information provided by motion capture system for (a) landmark 4, and (b) landmark 5.

Table 3-1. Results comparison.

	AtLoc	Tested (Ours)
Median Error in Position	0.21m	**0.15m**
Median Error in Orientation	**7.56**°	10.41°

CHAPTER 4
LYAPUNOV-BASED REAL-TIME AND ITERATIVE ADJUSTMENT OF DEEP NEURAL NETWORKS

A real-time DNN adaptive control architecture developed in [54] is included in this chapter for general uncertain nonlinear dynamical systems to track a desired time-varying trajectory. The contributions include developing a multiple timescale learning DNN adaptive control architecture for general uncertain nonlinear dynamical systems. Specifically, the unknown drift dynamics are approximated using a universal function approximator (i.e., a feedforward fully connected DNN). A Lyapunov-based real-time adaptation law is developed to update the output-layer weights of the DNN, and a batch optimization (i.e., minimize the mean squared error (MSE)) is used to periodically update the inner-layer weights of the DNN. The output-layer weights are updated continuously to ensure system stability, while the inner-layer weights are updated when a sufficient number of data are collected and trained to improve the approximation of the DNN. The multiple timescale learning adaptive control architecture enables the general uncertain nonlinear dynamical systems to track a desired trajectory, while using a DNN to iteratively improve the control performance. A nonsmooth Lyapunov-based analysis is used to prove semi-global asymptotic tracking of the desired trajectory. Numerical simulation examples are included to validate the results, and the Levenberg-Marquardt algorithm is used to train the weights of the DNN. The developed DNN architecture is shown in Figure 4-1.

4.1 System Dynamics

Consider a control-affine nonlinear dynamic system modeled as

$$\dot{x}(t) = f(x(t)) + g(x(t))\,u(t), \tag{4-1}$$

where $x : [t_0, \infty) \to \mathbb{R}^n$ denotes the generalized state, $t_0 \in \mathbb{R}_{\geq 0}$ denotes the initial time, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the unknown drift dynamics, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denotes the uncertain control effectiveness matrix, and $u : [t_0, \infty) \to \mathbb{R}^m$ denotes the control input. To facilitate the control development, the following assumption is made.

Assumption 4.1. The product of the uncertain control effectiveness matrix and control input can

be linearly parametrized as

$$g(x(t))u(t) = Y(x(t),u(t),t)\theta, \tag{4-2}$$

where $Y : \mathbb{R}^n \times \mathbb{R}^m \times [t_0,\infty) \to \mathbb{R}^{n\times q}$ denotes a measurable regression matrix, and $\theta \in \mathbb{R}^q$ denotes a vector of constant unknown parameters.

4.2 DNN Approximation and Update Policy

Let $\Omega \subset \mathbb{R}^n$ be a compact simply connected set with $x(t) \in \Omega$, and define $\mathbb{S}^n(\Omega)$ as the space where $f(x(t))$ is continuous. There exists ideal weights, ideal basis functions, and an ideal pre-trained DNN such that the drift dynamics $f(x(t)) \in \mathbb{S}^n(\Omega)$ can be represented as [62]

$$f(x(t)) = W^{*\mathrm{T}}\sigma(\Phi^*(x(t))) + \varepsilon(x(t)), \tag{4-3}$$

where $W^* \in \mathbb{R}^{L\times n}$ is an unknown bounded ideal output-layer weight matrix, $\sigma : \mathbb{R}^p \to \mathbb{R}^L$ is a user-defined bounded vector of activation functions, $\Phi^* : \mathbb{R}^n \to \mathbb{R}^p$ is the ideal unknown DNN, $\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ is the bounded unknown function reconstruction error associated with the ideal weights, activation functions, and DNN. The ideal unknown DNN Φ^* can be expressed as $\Phi^*(x(t)) = \left(W_k^{\mathrm{T}}\phi_k \circ W_{k-1}^{\mathrm{T}}\phi_{k-1} \circ \ldots \circ W_1^{\mathrm{T}}\phi_1\right)(x(t))$, where $k \in \mathbb{Z}$ denotes the number of inner-layers of the DNN, the symbol $\circ$ denotes function composition, W and $\phi(\cdot)$ denote the corresponding inner-layer weights and activation functions of the DNN, respectively.

The DNN is updated using a multiple timescale approach. The DNN is trained *a priori* using data sets collected from previous experiments, simulation data, etc. Ideally, large data sets from the same dynamic system operating under the same environmental conditions will be available for training the DNN. However, the developed strategy of real-time (Lyapunov-based) adjustment of the output-layer weights provides an advantage of significant flexibility in the training data. For example, as observed in the subsequent simulation, the training data could be from a dynamic system with different parameters (i.e., transfer learning), or could also be initialized with random weights.

Based on (4-3), the DNN approximation of the drift dynamics $\hat{f}_i : \mathbb{R}^n \to \mathbb{R}^n$ can be represented as $\hat{f}_i(x(t)) = \widehat{W}^{\mathrm{T}}(t)\sigma\left(\widehat{\Phi}_i(x(t))\right)$, where $\widehat{W} : [t_0,\infty) \to \mathbb{R}^{L\times n}$ is the estimate of the

ideal output-layer weight matrix, $\widehat{\Phi}_i : \mathbb{R}^n \to \mathbb{R}^p$ is the i^{th} estimates of Φ^*, and $i \in \mathbb{N}$ is the DNN estimate update index.[1] The mismatch between the ideal output-layer weights and the weight estimates $\widetilde{W} : [t_0, \infty) \to \mathbb{R}^{L \times n}$ is defined as

$$\widetilde{W}(t) \triangleq W^* - \widehat{W}(t). \tag{4-4}$$

Assumption 4.2. Using the universal function approximation property there exists known constants $\overline{W^*}$, $\overline{\sigma}$, $\overline{\varepsilon} \in \mathbb{R}_{>0}$ such that the unknown ideal weights W^*, user-selected activation functions $\sigma(\cdot)$, the unknown ideal DNN $\Phi^*(\cdot)$, and the function reconstruction error $\varepsilon(\cdot)$ can be upper bounded such that $\sup_{x(t)\in\Omega} \|W^*\| \leq \overline{W^*}$, $\sup_{x(t)\in\Omega} \|\sigma(\cdot)\| \leq \overline{\sigma}$,[2] and $\sup_{x(t)\in\Omega} \|\varepsilon(\cdot)\| \leq \overline{\varepsilon}$.

A priori training provides $\widehat{\Phi}_1(\cdot)$ and $\widehat{W}(t_0)$. The offline DNN training can be achieved by using different techniques. For example, [1] and [2] use a Stochastic Gradient Descent (SGD) based generative network architecture to generate estimates of matched system uncertainty, and the SGD update policy depends on a stochastic estimation of the expected value of the gradient of the loss function over a training set. When the offline DNN training phase is completed, an adaptive control law will be implemented for the system described in (4-1) to generate the output-layer DNN weight estimates, i.e., $\widehat{W}(t)$ for all $t \geq t_0$. Simultaneous to the online execution, data is collected and offline function approximation methods are used to update estimates on the inner-layer DNN weights.

4.3 Control Design

4.3.1 Control Objective

The control objective is to ensure the trajectory of the system in (4-1) tracks a desired sufficiently smooth time-varying trajectory $x_d : [t_0, \infty) \to \mathbb{R}^n$. To quantify the tracking objective, a

[1] The subscript i on $\widehat{\Phi}$ represents the i^{th} training iteration of the estimated DNN. The explicit expression for $\widehat{\Phi}_i(x(t))$ can be expressed as $\widehat{\Phi}_i(x(t)) = \left(\widehat{W}_{i,k}^{\mathrm{T}} \phi_{i,k} \circ \widehat{W}_{i,k-1}^{\mathrm{T}} \phi_{i,k-1} \circ \ldots \circ \widehat{W}_{i,1}^{\mathrm{T}} \phi_{i,1}\right)(x(t))$, where $\widehat{W}$ and $\phi(\cdot)$ denote the corresponding estimated inner-layer weights and activation functions of the corresponding training iteration, respectively.

[2] For common activation functions, e.g., hyperbolic tangent function, sigmoid function, radial basis function, $\overline{\sigma} = L$.

tracking error $e : [t_0, \infty) \to \mathbb{R}^n$ is defined as

$$e(t) \triangleq x(t) - x_d(t). \tag{4-5}$$

4.3.2 Control Development

To facilitate the subsequent control development, the product of the estimated control effectiveness matrix and the control input can be written as

$$\hat{g}(x(t))\, u(t) = Y(x(t), u(t), t)\, \hat{\theta}(t), \tag{4-6}$$

where $\hat{g} : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ denotes the estimate of the control effectiveness matrix. The parameter estimation error $\tilde{\theta} : [t_0, \infty) \to \mathbb{R}^q$ is defined as

$$\tilde{\theta}(t) \triangleq \theta - \hat{\theta}(t), \tag{4-7}$$

where $\hat{\theta} : [t_0, \infty) \to \mathbb{R}^q$ denotes the parameter estimate.

Assumption 4.3. The estimate of the control effectiveness matrix $\hat{g}$ is a full-row rank matrix for $t \geq t_0$, and the right pseudo inverse of $\hat{g}(\cdot)$ is denoted by $\hat{g}^+ : \mathbb{R}^n \to \mathbb{R}^{m \times n}$, where $\hat{g}^+(\cdot) \triangleq \hat{g}^{\mathrm{T}}(\cdot)\left(\hat{g}(\cdot)\hat{g}^{\mathrm{T}}(\cdot)\right)^{-1}$ is bounded given a bounded argument.

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

$$u(t) \triangleq \hat{g}^+(x(t))\left[-ke(t) - k_s \operatorname{sgn}(e(t)) + \dot{x}_d(t) - \widehat{W}^{\mathrm{T}}(t)\,\sigma\left(\widehat{\Phi}_i(x(t))\right)\right], \tag{4-8}$$

where $k, k_s \in \mathbb{R}_{>0}$ are constant control gains, and $\operatorname{sgn}(\cdot)$ denotes the signum function. The weight estimate adaptation law is designed as

$$\dot{\widehat{W}}(t) \triangleq \Gamma_W \sigma\left(\widehat{\Phi}_i(x(t))\right) e^{\mathrm{T}}(t), \tag{4-9}$$

where $\Gamma_W \in \mathbb{R}^{L \times L}$ denotes a user-defined positive definite, diagonal control gain matrix. The adaptation law for the parameter estimate in (4-6) is designed as

$$\dot{\hat{\theta}}(t) \triangleq \Gamma_\theta Y^{\mathrm{T}}(x(t), u(t), t)\, e(t), \tag{4-10}$$

where $\Gamma_\theta \in \mathbb{R}^{q\times q}$ denotes a user-defined positive definite, diagonal control gain matrix. Taking the time-derivative of (4-5) and substituting in (4-1)-(4-3) and (4-6)-(4-8) yields the closed-loop error system

$$\begin{aligned}\dot{e}(t) &= W^{*\mathrm{T}}\sigma\left(\Phi^*(x(t))\right) - \widehat{W}^{\mathrm{T}}(t)\,\sigma\left(\widehat{\Phi}_i(x(t))\right) \\ &\quad + \varepsilon(x(t)) - ke(t) - k_s \mathrm{sgn}(e(t)) + Y(x(t),u(t),t)\,\tilde{\theta}(t). \end{aligned} \tag{4-11}$$

Recall the initially trained DNN provides initial estimates $\widehat{\Phi}_1(\cdot)$ and $\widehat{W}(t_0)$. During implementation of the real-time controller, the output-layer weights of the DNN are estimated online. Concurrently, the data generated in real-time is stored for additional DNN training. Once a sufficient amount of data (user-defined) is collected to improve the function approximation performance, the inner-layer weights of the DNN will be updated to generate $\widehat{\Phi}_{i+1}(\cdot)$ for all i, i.e., (4-8)-(4-10).

4.4 Stability Analysis

The stability of the DNN-based adaptive tracking controller is established in the following theorem.

Theorem 4.1. *Consider a general nonlinear system modeled by the dynamics in (4-1) with $x(t_0) \in \Omega$ and satisfying Assumptions 4.1-4.3. The control input in (4-8), the output-layer weight adaptation law in (4-9), and the parameter estimate adaptation law in (4-10) ensure the trajectory tracking error defined in (4-5) yields semi-global asymptotic tracking in the sense that $\lim_{t\to\infty}\|e(t)\| \to 0$, $t \geq t_0$, provided the following gain condition is satisfied*

$$k_s > 2\overline{\sigma}\overline{W^*} + \overline{\varepsilon}. \tag{4-12}$$

Proof. Consider the candidate Lyapunov-like function $V_L : \mathbb{R}^{n(L+1)+q} \times [t_0,\infty) \to \mathbb{R}_{\geq 0}$ defined as

$$V_L(z,t) \triangleq \frac{1}{2}e^{\mathrm{T}}e + \frac{1}{2}\tilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\tilde{\theta} + \frac{1}{2}\mathrm{tr}\left(\widetilde{W}^{\mathrm{T}}\Gamma_W^{-1}\widetilde{W}\right), \tag{4-13}$$

where $z : [t_0,\infty) \to \mathbb{R}^{n(L+1)+q}$ is defined as $z \triangleq \left[\begin{array}{ccc} e^{\mathrm{T}}, & \tilde{\theta}^{\mathrm{T}}, & \mathrm{vec}\left(\widetilde{W}\right)^{\mathrm{T}} \end{array}\right]^{\mathrm{T}}$ and $\mathrm{vec}(\cdot)$ denotes the vectorization operator. Let $\zeta : [t_0,\infty) \to \mathbb{R}^{n(L+1)+q}$ be a Filippov solution to the differential

inclusion $\dot{\zeta} \in K[h](\zeta)$, where $\zeta(t) = z(t)$, the calculus of $K[\cdot]$ is used to compute Filippov's differential inclusion as defined in [63], and $h : \mathbb{R}^{n(L+1)+q} \to \mathbb{R}^{n(L+1)+q}$ is defined as $h(\zeta) \triangleq \left[\begin{array}{cccc} \dot{e}^{\mathrm{T}}, & \dot{\hat{\theta}}^{\mathrm{T}}, & \mathrm{vec}\left(\dot{\widehat{W}}\right)^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}$. The time-derivative of V_L exists almost everywhere (a.e.), i.e., for almost all $t \in [0,\infty)$, $\dot{V}_L(\zeta) \overset{\text{a.e.}}{\in} \dot{\tilde{V}}_L(\zeta)$, where $\dot{\tilde{V}}_L(\zeta)$ is the generalized time-derivative of V_L along the Filippov trajectories of $\dot{\zeta} = h(\zeta)$. By [64, Equation 13], $\dot{\tilde{V}}_L(\zeta) \triangleq \bigcap_{\xi \in \partial V_L(\zeta)} \xi^{\mathrm{T}} \left[K[h]^{\mathrm{T}}(\zeta), 1 \right]^{\mathrm{T}}$, where $\partial V_L(\zeta(t))$ denotes the Clarke generalized gradient of $V_L(\zeta)$. Since $V_L(\zeta)$ is continuously differentiable in ζ, then $\partial V_L(\zeta) = \{\nabla V_L(\zeta)\}$, where ∇ denotes the gradient operator.

Taking the generalized time-derivative of (4-13), then substituting in the mismatch between the ideal output-layer weight and the weight estimate in (4-4), the output-layer adaptation law in (4-9), the parameter estimate adaptation law in (4-10), and the closed-loop error system in (4-11) yields

$$\begin{aligned} \dot{\tilde{V}}_L(\zeta) \quad \subseteq \quad & e^{\mathrm{T}} \left(W^{*\mathrm{T}} \sigma\left(\Phi^*(x)\right) - \widehat{W}^{\mathrm{T}} K \left[\sigma\left(\widehat{\Phi}_i(x) \right) \right] \right) \\ & + e^{\mathrm{T}} \left(\varepsilon(x) - ke - k_s K[\mathrm{sgn}(e)] \right) - \mathrm{tr}\left(\widetilde{W}^{\mathrm{T}} K \left[\sigma\left(\widehat{\Phi}_i(x) \right) \right] e^{\mathrm{T}} \right). \end{aligned} \tag{4-14}$$

Using the trace operator property[3], the estimated mismatch for the ideal output-layer weight in (4-4), and adding and subtracting $e^{\mathrm{T}} W^{*\mathrm{T}} K \left[\sigma\left(\widehat{\Phi}_i(x) \right) \right]$ in (4-14) yields

$$\begin{aligned} \dot{\tilde{V}}_L(\zeta) \quad \subseteq \quad & -ke^{\mathrm{T}}e - k_s e^{\mathrm{T}} K[\mathrm{sgn}(e)] + e^{\mathrm{T}} \varepsilon(x) \\ & + e^{\mathrm{T}} W^{*\mathrm{T}} \left(\sigma\left(\Phi^*(x)\right) - K \left[\sigma\left(\widehat{\Phi}_i(x) \right) \right] \right). \end{aligned} \tag{4-15}$$

Hence, using the definition of $K[\mathrm{sgn}(e)]$ and Assumption 4.2, (4-15) can be upper bounded as

$$\dot{V}_L \overset{\text{a.e.}}{\leq} -k\|e\|^2 - \left(k_s - 2\overline{\sigma}\overline{W^*} - \overline{\varepsilon} \right) \|e\|. \tag{4-16}$$

By satisfying the gain condition described in (4-12), (4-16) can be further upper bounded as

$$\dot{V}_L \overset{\text{a.e.}}{\leq} -k\|e\|^2. \tag{4-17}$$

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

Using (4-13) and (4-17) implies $V_L \in \mathcal{L}_\infty$, which implies $z \in \mathcal{L}_\infty$. The definition of $z(t)$ implies $e, \tilde{\theta}, \widetilde{W} \in \mathcal{L}_\infty$. Using (4-4), (4-5) and (4-7) implies $x, \hat{\theta}, \widehat{W} \in \mathcal{L}_\infty$. Using Assumptions 4.2 and 4.3 implies $\sigma(\cdot), \varepsilon(x), \hat{g}^+(x), \hat{g}(x) \in \mathcal{L}_\infty$. Since $e, \hat{g}^+(x), \widehat{W} \in \mathcal{L}_\infty$, using (4-8) implies $u \in \mathcal{L}_\infty$. Since $\hat{g}(x), u, \hat{\theta} \in \mathcal{L}_\infty$, using (4-6) yields $Y(x,u) \in \mathcal{L}_\infty$. Furthermore, by the extension of the LaSalle-Yoshizawa theorem for non-smooth systems in [65] and [66], $k\|e\|^2 \to 0$, which implies $\|e(t)\| \to 0$ as $t \to \infty$ and $x(t) \in \Omega$ for all $t \geq t_0$. □

4.5 Simulation

To demonstrate the effectiveness of the developed method, a simulation is performed on a control-affine realization of a two-state Van der Pol oscillator. The dynamics used in the simulation are

$$f(x) = \begin{bmatrix} \mu\left(x_1 - \frac{1}{3}x_1^3 - x_2\right) \\ \frac{1}{\mu}x_1 \end{bmatrix}, \tag{4-18}$$

$$g(x) = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix}, \tag{4-19}$$

where $x = [x_1, x_2]^{\mathrm{T}}$ and $\mu = 10$. The desired trajectory is $x_d = [5\cos(t), 5\sin(t)]^{\mathrm{T}}$. The initial conditions of the system were $x(0) = [-5, 8]^{\mathrm{T}}$ and $\hat{\theta}(0) = [6, 6]^{\mathrm{T}}$. The user-defined parameters were selected as $k = 75$, $k_s = 0.05$, $\Gamma_W = 500 \cdot I_{13}$, and $\Gamma_\theta = \mathrm{diag}([0.1, 0.05])$.

The DNN used in this simulation was composed of 4 layers, each with 10, 5, 8, and 2 neurons, respectively. The DNN architecture is illustrated in Figure 4-2. Each layer is linear and the first, second, and third layers have tangent-sigmoid, logarithmic-sigmoid, and tangent-sigmoid activation functions, respectively. The learning rate (i.e., the learning gain parameter used to determine the step size in retraining the DNN weights at each iteration) was fixed as $\eta = 0.001$. The mean squared error (MSE) was used as the loss function for training. Each training iteration lasted until the MSE (i.e., the loss) was less than 10^{-3}. The Levenberg-Marquardt algorithm was used to train the weights of the DNN. For each DNN training iteration, 70% of the data was used for training, 15% was used for validation, and 15% was used for testing.

To pre-train the DNN, a 600 second simulation of a system with dynamics in (4-18) and $\mu = 10$ was performed. Training statistics for the offline training are shown in Figure 4-3. The real-time controller and the update laws in (4-8)-(4-10) are used to update their respective parametric estimates. Concurrent to the real-time controller execution, input-output data is collected to retrain the DNN. As shown in Figures 4-4-4-6, the training start time is denoted by the red dashed vertical line and the training completion time is denoted by the black dashed vertical line. The time (and corresponding amount of data) between retraining the inner-layer weights is a user-defined parameter of the simulation. After the prescribed time between retraining elapses, the inner-layer DNN weights begin updating via retraining. In this simulation, the time between retraining is 25 seconds. The first retraining starts at $t = 25$ seconds and ends at $t = 37.4$ seconds. The second retraining starts at $t = 62.4$ seconds and ends at $t = 68.3$ seconds. While the retraining is in process, the real-time controller and update laws continue uninterrupted as described in (4-8)-(4-10). Once the retraining is completed, the new inner-layer DNN weights are updated, overwriting the previous values.

As shown in Figures 4-4-4-6, the time for the MSE to be less than 10^{-3} took 12.4 seconds to complete. After the DNN has completed retraining, the controller implements the inner-layer weights at $t = 37.4$ seconds. After implementing the updated DNN weights, new data is collected for another 25 seconds. To further improve the DNN estimate, a second retraining is performed. During the second training iteration, data from the first 25 seconds and the second 25 seconds are both used. The second retraining took 5.9 seconds. The inner-layer weights from the second retraining are implemented at $t = 68.3$ seconds.

The tracking error performance in Figure 4-4 indicates that discretely retrained DNNs with an online adaptive output-layer weights are a viable method to perform trajectory tracking. The first iteration of the DNN (DNN1) is the offline generated DNN, DNN2 is the model after the first retraining, and DNN3 is the model after the second retraining. As shown by the root mean squared error (RMSE) in Table 4-1 (A), each subsequent DNN training iteration yielded improved performance, where $e \triangleq \begin{bmatrix} e_1 & e_2 \end{bmatrix}^{\mathrm{T}}$. The decrease in error after each retraining is expected since

a larger set of system data was used to train the DNN during each retraining. Figure 4-7 shows the phase plot of the system, and compares the performance of the tracking during the application of each DNN. DNN1 has the worst estimate of the system dynamics. DNN2 and DNN3 show significantly better tracking behavior, which is also reflected in Figure 4-4. Figure 4-6 presents the control input to the system for the duration of the simulation. DNN1 poorly approximates the dynamics near $x = [-3.5, 3.5]^{\mathrm{T}}$, and this error is further reflected in Figure 4-6 with the spikes in control input approximately at $t = 2$ seconds and $t = 8$ seconds.

4.5.1 Transfer Learning & Random Weights

To further demonstrate the flexibility of the developed real-time Lyapunov-based adjustment of the output-layer weights, two additional simulations were performed. In this section, transfer learning-based and randomly initialized DNN weights simulations were investigated. Transfer learning in this context is applying the learned DNN model of one system to another system. In the simulation, transfer learning is demonstrated by training a DNN model on a dataset of a system described by the dynamics in (4-18) with parameter $\mu = 1$, whereas the simulated system has parameter $\mu = 10$.

In the transfer learning-based approach, the DNN is pre-trained with 600 seconds of simulated data from a system with dynamics in (4-18), but parametrized with $\mu = 1$. Figures 4-8(a), 4-9(a), and Table 4-1 (B) show the tracking error, phase plot, and RMSE of the transfer learning approach over three iterations of DNN training, respectively. For situations where data cannot be collected *a priori*, initial inner-layer DNN weights can be selected by the user. In the third simulation, instead of pre-training the DNN, a simulation was performed with the initial DNN randomly selected weights. Figures 4-8(b), 4-9(b), and Table 4-1 (C) show the tracking error, phase plot, and RMSE of this approach over three iterations of DNN training, respectively.

The performance of the transfer learning-based approach and initial randomly selected DNN weights simulations are depicted in Figures 4-8 and 4-9. Iterations in the inner-layer weights are shown to improve performance. The first simulation, which was trained with 600 seconds of offline data using the dynamics in (4-18) with $\mu = 10$ has the best performance with

respect to the smallest RMSE within 25 second intervals compared to transfer learning and initial randomly selected DNN weights. Nevertheless, the proposed real-time Lyapunov-based adjustment of the output-layer weights accommodates for different methods to initialize the DNN inner-layer weights.

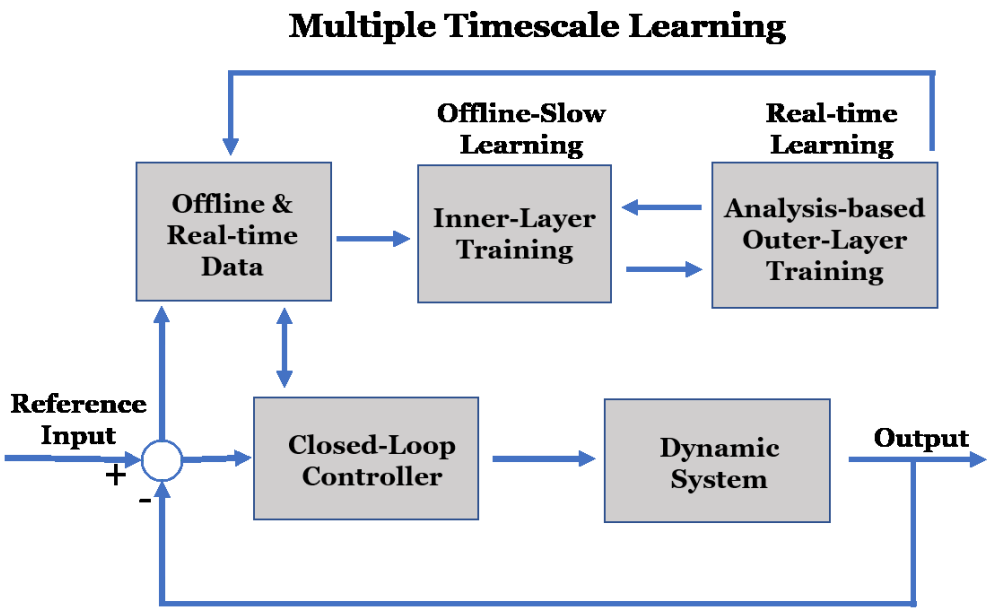


Figure 4-1. Multiple timescale learning architecture.

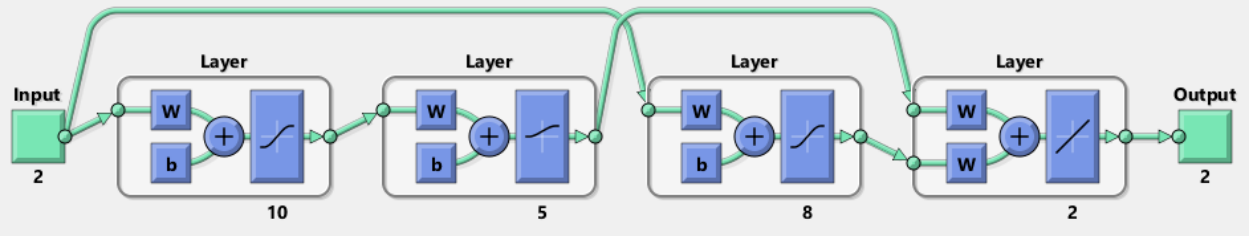


Figure 4-2. The DNN is composed of 4 layers, each with 10, 5, 8, and 2 neurons, respectively.

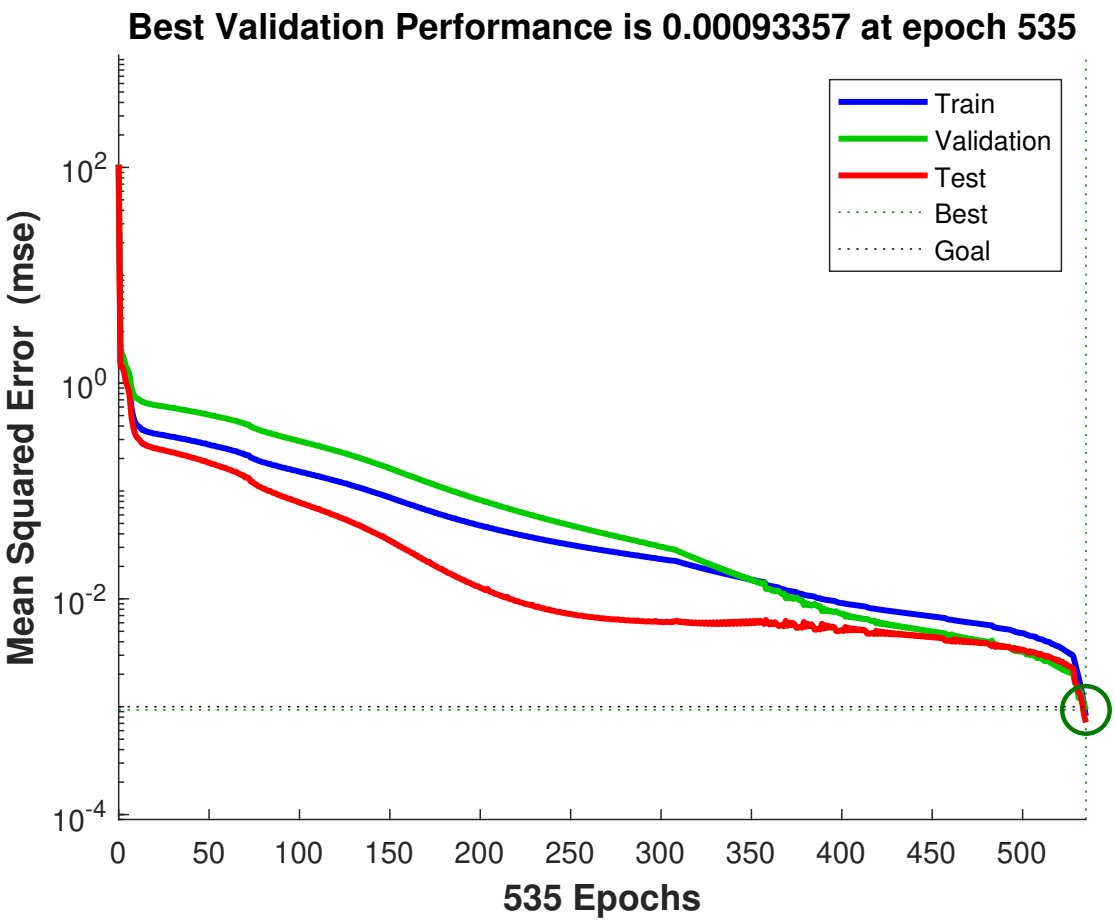


Figure 4-3. The MSE for the offline trained DNN with learning rate $\eta = 0.001$ for 535 epochs in logarithmic scale. The best validation performance for DNN1 is MSE $= 0.00093357$.

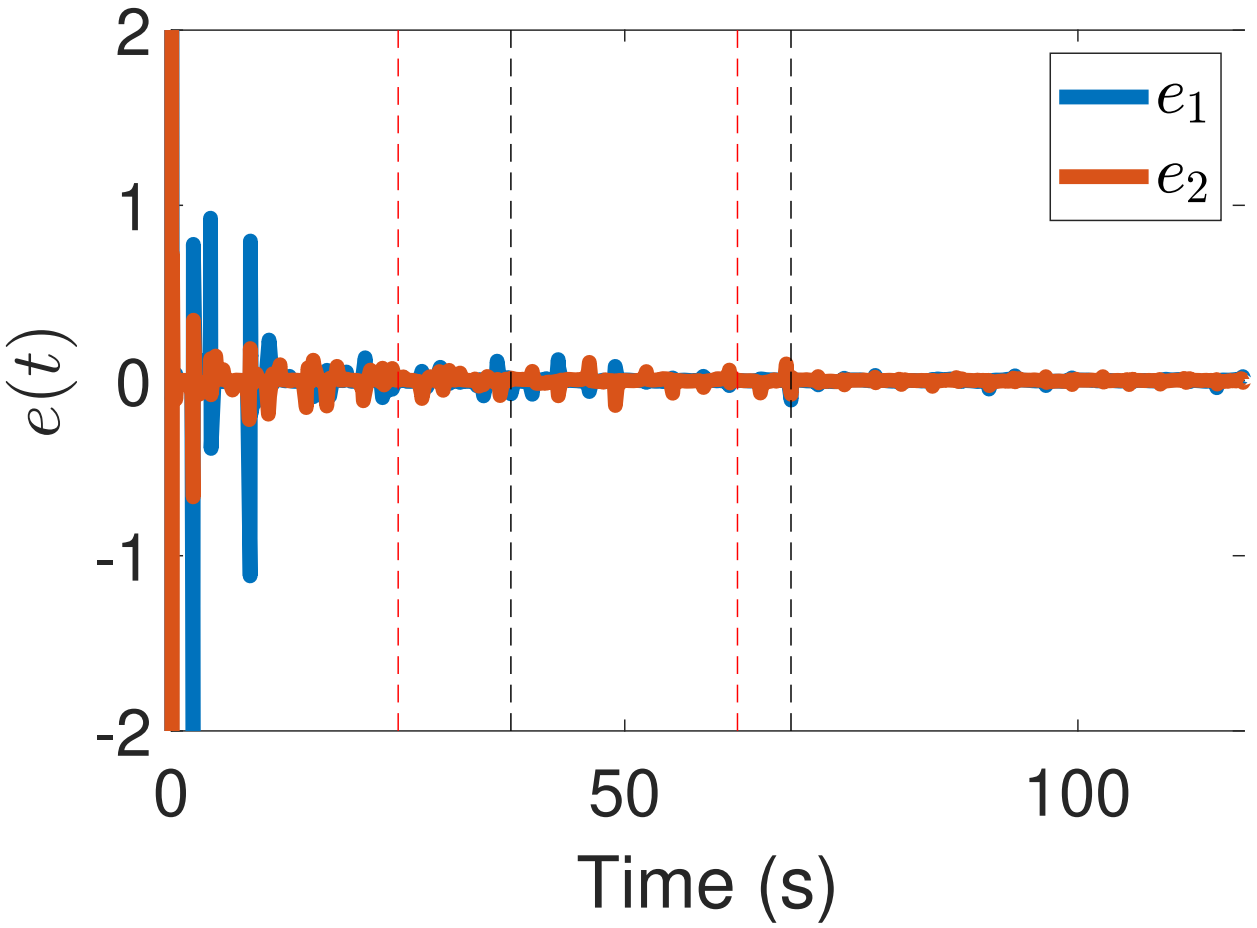


Figure 4-4. The tracking error over three iterations of DNN training, i.e., DNN1, DNN2, DNN3. At $t = 0$ seconds, the first iteration of the DNN (DNN1) is deployed. The red dashed line at $t = 25$ seconds represents the beginning of the first retraining (DNN2), and the black dashed line at $t = 37.4$ seconds represents the end of the first retraining. The red dashed line at $t = 62.4$ seconds represents the beginning of the second retraining (DNN3), and the black dashed line at $t = 68.3$ seconds represents the end of the second retraining.

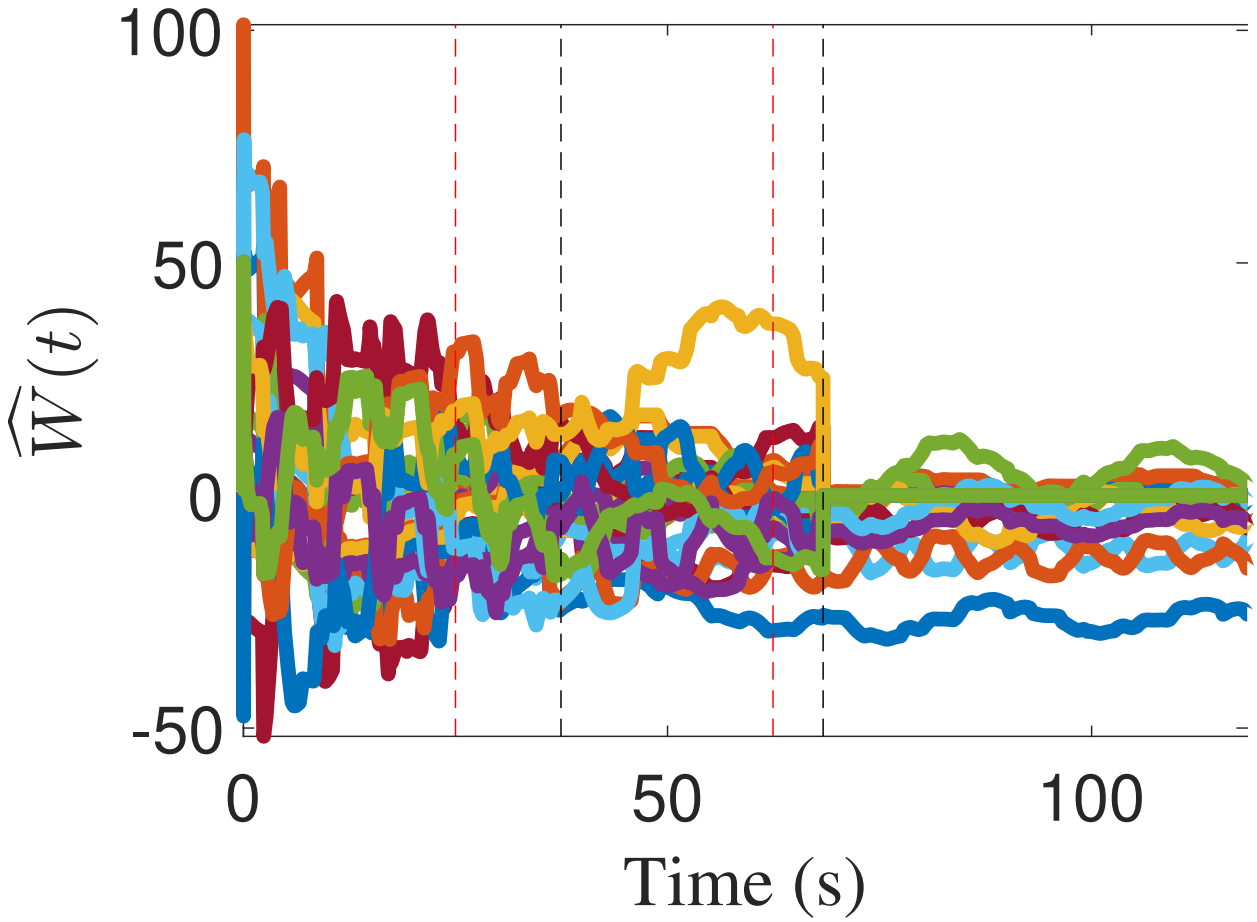


Figure 4-5. Weight estimates over three iterations of DNN training, i.e., DNN1, DNN2, DNN3.

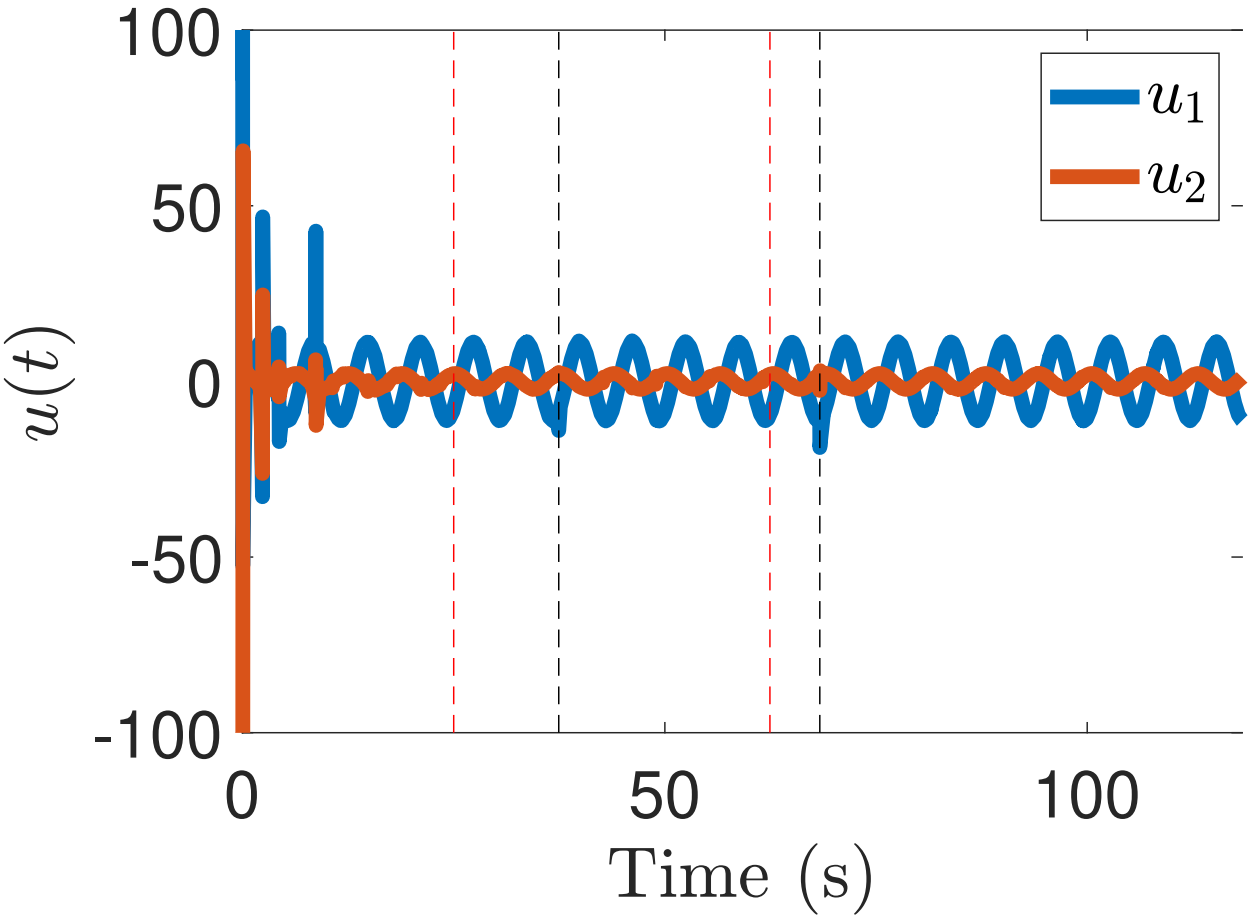


Figure 4-6. Applied control input over three iterations of DNN training, i.e., DNN1, DNN2, DNN3.

Table 4-1. Root Mean Squared Errors (RMSE)

		RMSE e_1	RMSE e_2	RMSE e
A	DNN1	0.18	0.18	0.25
	DNN2	0.01	0.02	0.02
	DNN3	0.01	0.00	0.01
B	DNN1	0.17	0.18	0.24
	DNN2	0.06	0.02	0.07
	DNN3	0.01	0.01	0.02
C	DNN1	0.21	0.22	0.29
	DNN2	0.02	0.00	0.02
	DNN3	0.01	0.01	0.01

RMSE of the tracking error for each simulation: (A) DNN pre-trained with dynamics in (4-18) parametrized with $\mu = 10$, (B) DNN pre-trained with dynamics in (4-18) parametrized with $\mu = 1$, (C) DNN initialized with randomly selected inner-layer weights.

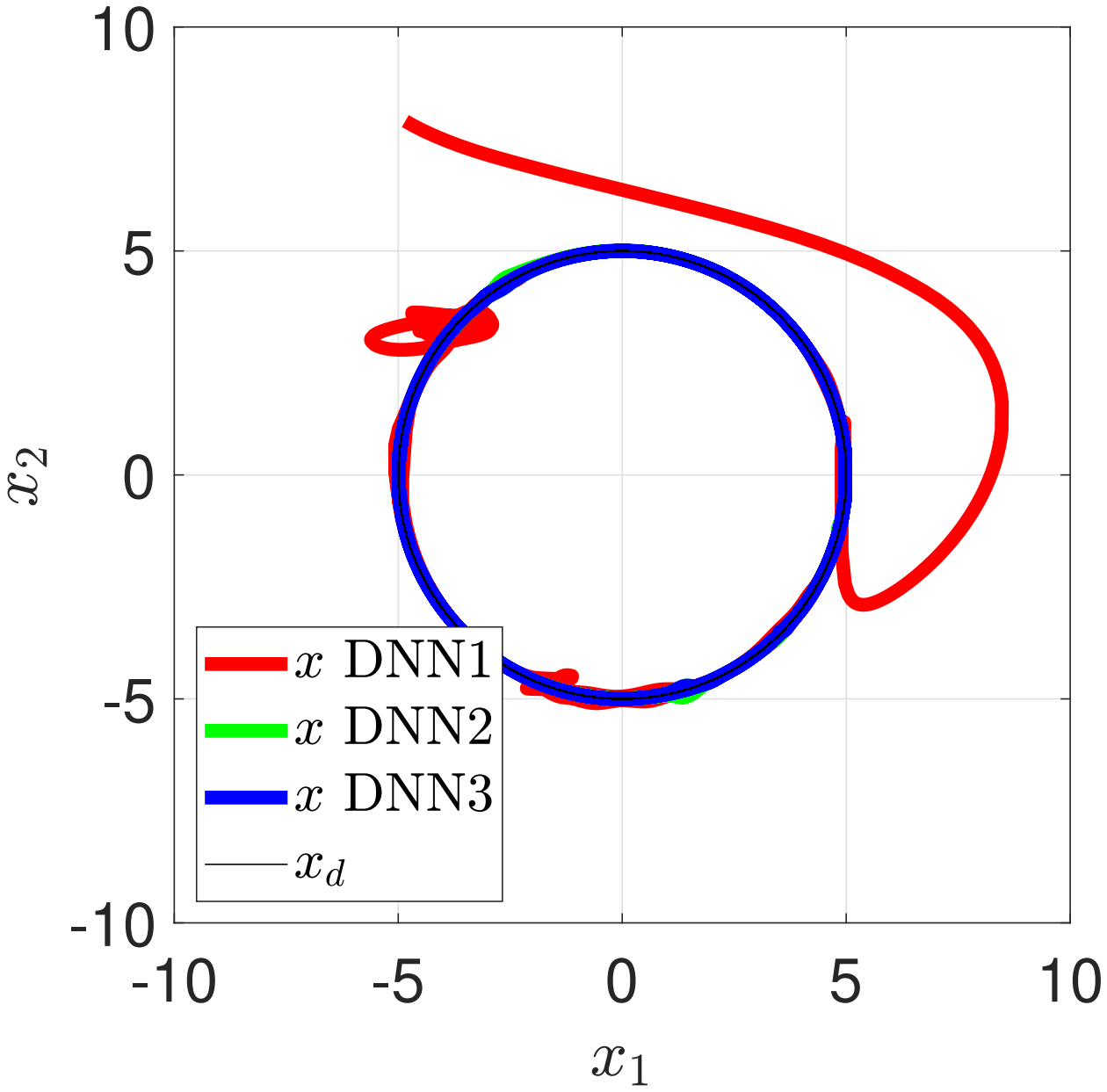
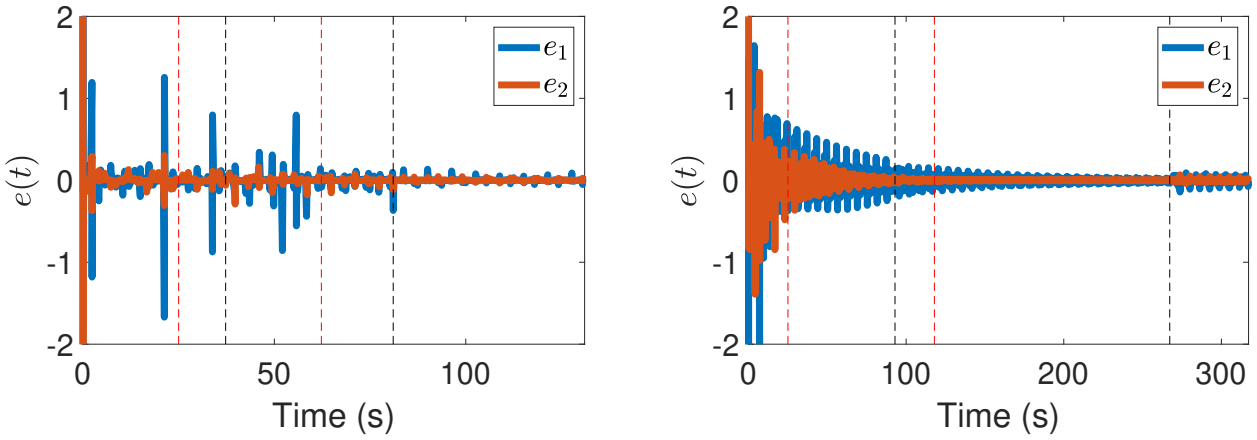
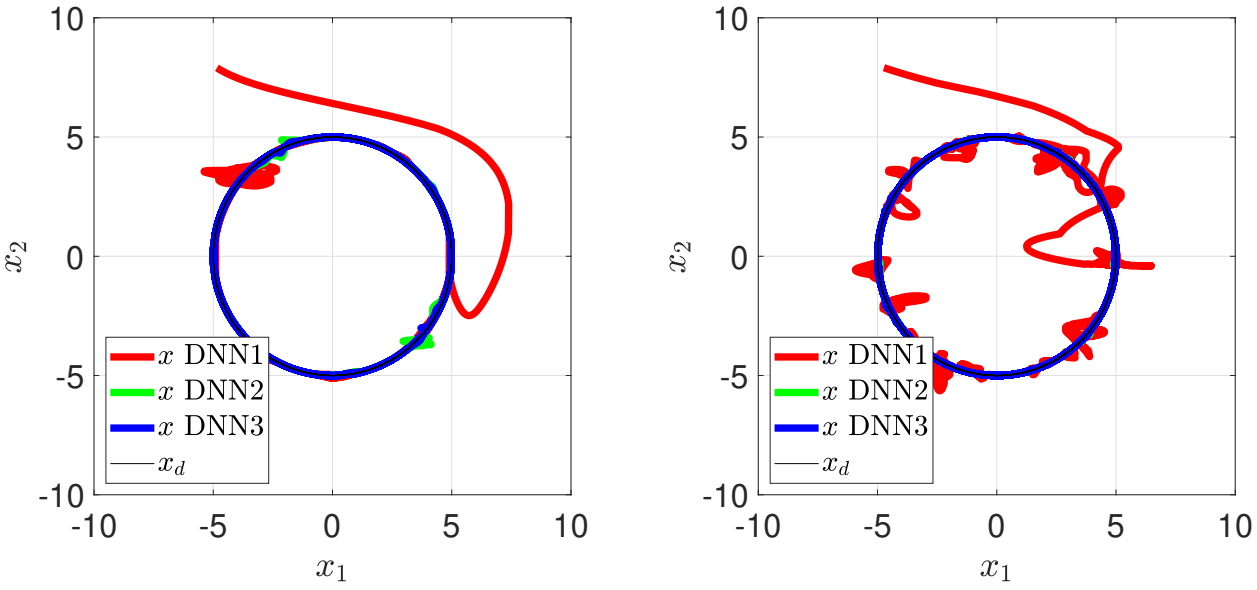


Figure 4-7. Phase plot of the dynamics in (4-18) over three iterations of DNN training, i.e., DNN1, DNN2, DNN3.



(a) (b)

Figure 4-8. Tracking error for (a) transfer learning, and (b) randomly selected initial inner-layer weights.



(a) (b)

Figure 4-9. The phase plot for (a) transfer learning, and (b) randomly selected initial inner-layer weights.

CHAPTER 5
CONCLUSIONS

Many autonomous applications experience intermittent state feedback due to practical reasons, e.g., the operating environment, task definition, system constraint. To compensate for the intermittency, a switched systems approach is leveraged to enable the state of a nonlinear dynamical system to switch between feedback available and feedback-denied environments. While state feedback is unavailable, state predictors or observers can be used to predict the state using estimated information. A stabilizing maximum dwell-time condition is developed using a Lyapunov-based analysis, which allow the autonomous agent to explore the feedback-denied environment for a pre-determined period of time before going to a feedback available region. While state feedback is available, the estimated state is reset to the true state to compensate for the state estimation error.

In Chapter 2, a relay-explorer control method is developed enabling a multi-agent system to explore an unknown environment using a switched systems approach. Specifically, a relay agent is switching between the feedback available and feedback-denied regions, providing state estimates to the explorer leader. The contributions in this chapter include developing a distributed formation controller using state estimates for the explorer follower agents to maintain a fixed formation with respect to the explorer leader agent, which significantly increases the exploring coverage compared to a single agent. The developed method also enables the relay and explorer agents to only use state estimates while operating in the feedback-denied environment by satisfying the designed parameters and the developed stabilizing dwell-time conditions. Future efforts could focus on extending the result to account for multiple relay servicing agents, where each relay agent is tasked with servicing multiple explorer agents. For example, the current result could combine with the preliminary results developed in [67] to incorporate more relay agents to increase the exploring coverage in the feedback-denied environment. Incorporating multiple relay agents could easily complicate the problem by introducing potential servicing responsibility conflicts. However, the robustness of the system can be improved provided coordination of the responsibilities of the relay agents can be carefully designed. Another future research direction is to consider incorporating multiple feedback regions, where the relay agent could select a specific

feedback region to get state update based on a designed cost function to either minimize travelling distance or control efforts.

In Chapter 3, a novel method that utilizes a switched systems approach to achieve unknown region exploration for nonholonomic systems under intermittent state feedback is presented. The approach relaxes the requirement of continuous state feedback and allows an agent to intermittently explore a feedback-denied environment. State estimates are used in the tracking controller to compensate for intermittence of state feedback. A controller design for a nonholonomic vehicle with an exogenous disturbance is developed to ensure the agent can track a desired trajectory in both position and orientation. The contributions in this chapter include developing a set of stabilizing dwell-time conditions via a Lyapunov-based analysis. A maximum dwell-time condition is developed to enable the nonholonomic agent exploring a feedback-denied region for a pre-determined time period before requiring state feedback information. A minimum-dwell time condition is developed to allow the nonholonomic agent to reduce the state estimation error within a desired value. Future efforts could focus on combining the current result with an existing localization method developed in [60] to use a monocular camera to localize the pose of the nonholonomic agent. Specifically, the poses of the nonholonomic agent, a set of images of landmarks, and the poses of landmarks are recorded and used as inputs to a DNN. Similar to [60], the DNN architecture consists of a visual encoder, an attention module, and a pose regressor. After the pre-training DNN phase, the outputs, i.e., pose of the nonholonomic agent, can be used as state feedback information for the control objective. With the proposed approach, the nonholonomic agent could use a monocular camera to obtain state feedback information by keeping a landmark within the FOV while implementing the pre-trained DNN. The nonholonomic agent will able to navigate the unknown environment using the developed method while acquiring the state information by using the onboard monocular camera. Another future research direction could be investigating a general method of optimal path planning while satisfying the maximum and minimum dwell-time conditions, so the agent can efficiently explore the unknown environment. Specifically, sampling-based algorithms (e.g., probabilistic roadmap,

rapidly-exploring random tree) can be used to design the optimal path for the agent to track, and a local planner using cubic Bezier curves can be used to compensate the nonholonomic constraint. However, it is not clear how to incorporate the maximum and minimum dwell-time conditions into the planning schemes. Therefore, some research efforts are required to achieve the proposed plan.

Uncertainties in dynamical systems are often experienced in practical autonomous applications, Chapter 4 developed a multiple timescale DNN-based adaptive control architecture for general nonlinear dynamical systems with unknown drift dynamics and uncertain control effectiveness matrix. The contributions in this chapter include a Lyapunov-based adaptive update law is developed to estimate the unknown output-layer weights of the DNN and the uncertain control effectiveness matrix in real-time. Simultaneous to real-time execution, data is collected and offline function approximation methods are used to update estimates of the inner-layer weights. Simulation examples showed promising results in improving the control performance while guaranteeing the system is stable with the Lyapunov-based stability analysis. Future efforts could focus on investigating dynamic NNs with feedback connections such as long short-term memory (LSTM) to not only process static data points, but also process the entire sequences of data. Using dynamic NNs to process and make predictions based on time series data while guaranteeing the system stability using a Lyapunov-based method, the results could be impactful for many practical applications. Furthermore, data-based learning methods developed in [50]-[52] could be combined with the developed approach to achieve system identification. However, the FE condition assumption made in the previous results might not be satisfied in the iterative training processes for the DNN update. Therefore, additional research efforts are needed to achieve the proposed objective.

REFERENCES

[1] G. Joshi and G. Chowdhary, "Deep Model Reference Adaptive Control," in *IEEE Conf. Decis. Control*. IEEE, 2019, pp. 4601–4608.

[2] G. Joshi, J. Virdi, and G. Chowdhary, "Design and Flight Evaluation of Deep Model Reference Adaptive Controller," in *AIAA Scitech 2020 Forum*, 2020, p. 1336.

[3] V. Gupta, A. F. Dana, J. P. Hespanha, R. M. Murray, and B. Hassibi, "Data Transmission Over Networks for Estimation and Control," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1807–1819, 2009.

[4] L. Shi, M. Epstein, and R. M. Murray, "Kalman Filtering Over a Packet-Dropping Network: A Probabilistic Perspective," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 594–604, 2010.

[5] A. Dana, V. Gupta, J. Hespanha, B. Hassibi, and R. Murray, "Estimation over Communication Networks: Performance Bounds and Achievability Results," in *Proc. Am. Control Conf.*, 9-13 2007, pp. 3450–3455.

[6] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, 2004.

[7] J. Cortes, "Discontinuous Dynamical Systems," *IEEE Control Sys.*, vol. 28, no. 3, pp. 36–73, 2008.

[8] C. Nowzari and J. Cortés, "Distributed Event-Triggered Coordination for Average Consensus on Weight-Balanced Digraphs," *Autom.*, vol. 68, pp. 237–244, 2016.

[9] A. Adaldo, F. Alderisio, D. Liuzza, G. Shi, D. V. Dimarogonas, M. di Bernardo, and K. H. Johansson, "Event-Triggered Pinning Control of Switching Networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 204–213, Jun. 2015.

[10] 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.

[11] 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.

[12] 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, 2015.

[13] 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, 2017.

[14] T. Huang, C. Li, W. Yu, and G. Chen, "Synchronization of Delayed Chaotic Systems with Parameter Mismatches by using Intermittent Linear State Feedback," *Nonlinearity*, vol. 22, no. 3, p. 569, 2009.

[15] D. Tolić and S. Hirche, *Networked Control Systems with Intermittent Feedback*. CRC Press, 2017.

[16] H.-Y. Chen, Z. I. Bell, R. Licitra, and W. E. Dixon, "A Switched Systems Approach to Vision-Based Tracking Control of Wheeled Mobile Robots," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 4902–4907.

[17] 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.

[18] 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.

[19] R. Sun, Z. Bell, F. Zegers, and W. E. Dixon, "A Switched Systems Approach to Unknown Environment Exploration with Intermittent State Feedback for Nonholonomic Systems," in *Proc. Am. Control Conf.*, 2020, pp. 5275–5280.

[20] F. Zegers, H.-Y. Chen, P. Deptula, and W. E. Dixon, "A Switched Systems Approach to Consensus of a Distributed Multi-Agent System with Intermittent Communication," in *Proc. Am. Control Conf.*, 2019, pp. 2372–2377.

[21] R. Sun, C. Harris, Z. Bell, and W. E. Dixon, "Relay-Explorer Approach for Multi-Agent Exploration of an Unknown Environment with Intermittent Communication," in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 5218–5223.

[22] R. T. M'closkey and R. M. Murray, "Exponential Stabilization of Driftless Nonlinear Control Systems using Homogeneous Feedback," *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 614–628, 1997.

[23] C. Samson, "Velocity and Torque Feedback Control of a Nonholonomic Cart," in *Adv. Robot Control*. Springer, 1991, pp. 125–151.

[24] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal, *Nonlinear Control of Wheeled Mobile Robots*, ser. Lecture Notes in Control and Inf. Sci. Springer-Verlag London Ltd, 2000, vol. 262.

[25] S. Hutchinson, G. Hager, and P. Corke, "A Tutorial on Visual Servo Control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[26] N. Gans, G. Hu, J. Shen, Y. Zhang, and W. E. Dixon, "Adaptive Visual Servo Control to Simultaneously Stabilize Image and Pose Error," *Mechatron.*, vol. 22, no. 4, pp. 410–422, 2012.

[27] N. R. Gans, G. Hu, and W. E. Dixon, "Keeping Multiple Objects in the Field of View of a Single PTZ Camera," in *Proc. Am. Control Conf.*, St. Louis, Missouri, Jun. 2009, pp. 5259–5264.

[28] G. Hu, N. Gans, and W. E. Dixon, "Quaternion-Based Visual Servo Control in the Presence of Camera Calibration Error," *Int. J. Robust Nonlinear Control*, vol. 20, no. 5, pp. 489–503, 2010. [Online]. Available: http://ncr.mae.ufl.edu/papers/RNC10.pdf

[29] G. Hu, N. Gans, N. Fitz-Coy, and W. E. Dixon, "Adaptive Homography-Based Visual Servo Tracking Control Via A Quaternion Formulation," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 1, pp. 128–135, 2010. [Online]. Available: http://ncr.mae.ufl.edu/papers/CST10.pdf

[30] G. Hu, W. Mackunis, N. Gans, W. E. Dixon, J. Chen, A. Behal, and D. Dawson, "Homography-Based Visual Servo Control with Imperfect Camera Calibration," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1318–1324, 2009. [Online]. Available: http://ncr.mae.ufl.edu/papers/tac09.pdf

[31] J. Chen, D. M. Dawson, W. E. Dixon, and V. Chitrakaran, "Navigation Function Based Visual Servo Control," *Autom.*, vol. 43, pp. 1165–1177, 2007. [Online]. Available: http://ncr.mae.ufl.edu/papers/auto07.pdf

[32] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, "Adaptive Homography-Based Visual Servo Tracking for Fixed and Camera-in-Hand Configurations," *IEEE Trans. Control Syst. Technol.*, vol. 13, pp. 814–825, 2005. [Online]. Available: http://ncr.mae.ufl.edu/papers/cst05.pdf

[33] 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.

[34] N. R. Gans and S. A. Hutchinson, "A Stable Vision-Based Control Scheme for Nonholonomic Vehicles to Keep a Landmark in the Field of View," in *Proc. IEEE Int. Conf. Robot. Autom.*, Roma, Italy, Apr. 2007, pp. 2196–2201.

[35] G. Lopez-Nicolas, N. R. Gans, S. Bhattacharya, C. Sagues, J. J. Guerrero, and S. Hutchinson, "Homography-Based Control Scheme for Mobile Robots With Nonholonomic and Field-of-View Constraints," *IEEE Trans. Syst. Man Cybern.*, vol. 40, no. 4, pp. 1115–1127, Aug. 2010.

[36] S. Mehta, G. Hu, N. Gans, and W. E. Dixon, *Robot Localization and Map Build.* InTech, 2010, ch. A Daisy-Chaining Visual Servoing Approach with Applications in Tracking, Localization, and Mapping, pp. 383–408. [Online]. Available: http://ncr.mae.ufl.edu/papers/sid-chap.pdf

[37] 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.

[38] 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.

[39] 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.

[40] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM Algorithms: A Survey from 2010 to 2016," *IPSJ Trans. on Comput. Vis. and Appl.*, vol. 9, no. 1, p. 16, 2017.

[41] M. Karrer, P. Schmuck, and M. Chli, "CVI-SLAM Collaborative Visual-Inertial SLAM," *IEEE Robot. and Autom. Lett.*, vol. 3, no. 4, pp. 2762–2769, Oct 2018.

[42] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[43] Q. Teng and L. Zhanga, "Data Driven Nonlinear Dynamical Systems Identification using Multi-Step CLDNN," *AIP Adv.*, 2019.

[44] F. L. Lewis, "Nonlinear Network Structures for Feedback Control," *Asian J. Control*, vol. 1, no. 4, pp. 205–228, 1999.

[45] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic Tracking for Uncertain Dynamic Systems via a Multilayer Neural Network Feedforward and RISE Feedback Control Structure," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, 2008.

[46] 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, 2010.

[47] W. Mackunis, F. Leve, P. Patre, N. Fitz-Coy, and W. E. Dixon, "Adaptive Neural Network-Based Satellite Attitude Control in the Presence of CMG Uncertainty," *Aerosp. Sci. Technol.*, vol. 54, pp. 218–228, 2016.

[48] G. V. Chowdhary and E. N. Johnson, "Theory and Flight-Test Validation of a Concurrent-Learning Adaptive Controller," *J. Guid. Control Dynam.*, vol. 34, no. 2, pp. 592–607, Mar. 2011.

[49] 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.

[50] 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.

[51] Z. Bell, J. Nezvadovitz, 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.

[52] R. Sun, C. Riano-Rios, R. Bevilacqua, N. G. Fitz-Coy, and W. E. Dixon, "Cubesat Adaptive Attitude Control with Uncertain Drag Coefficient and Atmospheric Density," *AIAA J. Guid. Control Dyn.*, vol. 44, no. 2, pp. 379–388, 2021.

[53] G. Joshi, J. Virdi, and G. Chowdhary, "Asynchronous Deep Model Reference Adaptive Control," in *Conf. on Robot Learn.*, 2021, pp. 984–1000.

[54] 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.

[55] 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.

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

[57] 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 Natl. Aerosp. and Electron. Conf. (NAECON)*. IEEE, 2019, pp. 606–613.

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

[59] W. E. Dixon, Z. P. Jiang, and D. M. Dawson, "Global Exponential Setpoint Control of Wheeled Mobile Robots: A Lyapunov Approach," *Autom.*, vol. 36, pp. 1741–1746, 2000. [Online]. Available: http://ncr.mae.ufl.edu/papers/auto00.pdf

[60] B. Wang, C. Chen, C. X. Lu, P. Zhao, N. Trigoni, and A. Markham, "AtLoc: Attention Guided Camera Localization," in *Proc. of the AAAI Conf. on Artif. Intell.*, vol. 34, no. 06, 2020, pp. 10 393–10 401.

[61] C. Samson, "Control of Chained Systems Application to Path Following and Time-Varying Point-Stabilization of Mobile Robots," *IEEE Trans. Autom. Control*, vol. 40, no. 1, pp. 64–77, 1995.

[62] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Philadelphia, PA: CRC Press, 1998.

[63] 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.

[64] D. Shevitz and B. Paden, "Lyapunov Stability Theory of Nonsmooth Systems," *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, 1994.

[65] 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.

[66] 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.

[67] R. Sun, S. Bharadwaj, Z. Xu, U. Topcu, and W. E. Dixon, "Reactive Synthesis for Relay-Explorer Consensus with Intermittent Communication."

BIOGRAPHICAL SKETCH

Runhan Sun received his bachelor's degree in mechanical engineering in 2015 from the University of California, Santa Barbara. He then worked as a mechanical engineer in Dimension Robotics LLC. in Beijing for a year. He received his master's degree in mechanical engineering in 2018 from the University of Florida. In the same year, he joined the Nonlinear Controls and Robotics group under the supervision of Dr. Warren E. Dixon. He received his Ph.D. in mechanical engineering from the University of Florida in 2022. His research focused on nonlinear and adaptive control, switched systems, deep neural network control, and image-based control systems.