Fall 2009

Formation control of mobile robots and unmanned aerial vehicles

Travis Alan Dierks

Department: Electrical and Computer Engineering

FORMATION CONTROL OF MOBILE ROBOTS AND UNMANNED AERIAL VEHICLES

by

TRAVIS DIERKS

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

2009

Approved by

Jagannathan Sarangapani, Advisor

Kelvin Erickson
Donald Wunsch
Cihan Dagli
S. N. Balakrishnan

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following six articles that have been submitted for publication as follows:

Paper 1, T. Dierks and S. Jagannathan, "Neural Network Control of Mobile Robot Formations using RISE Feedback," has been published in IEEE Trans. on Systems, Man, and Cybernetics—Part B, Vol. 39, no. 2, 2009.

Paper 2, T. Dierks and S. Jagannathan, "Neural Network Output Feedback Control of Robot Formations," will appear in IEEE Trans. on Systems, Man, and Cybernetics— Part B.

Paper 3, T. Dierks and S. Jagannathan, "Output Feedback Control of a Quadrotor UAV using Neural Networks," is provisionally accepted to IEEE Trans. on Neural Networks.

Paper 4, T. Dierks and S. Jagannathan, "Leader-Follower Formation Control of Multiple Quadrotor Unmanned Aerial Vehicles using Neural Networks," is under revision with Automatica.

Paper 5, T. Dierks and S. Jagannathan, "Optimal Control of Affine Nonlinear Discrete-time Systems with Unknown Internal Dynamics using Online Approximators," has been submitted to IEEE Trans. on Automatic Control.

Paper 6, T. Dierks and S. Jagannathan, "Optimal Control of Affine Nonlinear Continuous-time Systems using an Online Approximator," has been submitted to IEEE Trans. on Automatic Control.

ABSTRACT

In this dissertation, the nonlinear control of nonholonomic mobile robot formations and unmanned aerial vehicle (UAV) formations is undertaken and presented in six papers. In the first paper, an asymptotically stable combined kinematic/torque control law is developed for leader-follower based formation control of mobile robots using backstepping. A neural network (NN) is introduced along with robust integral of the sign of the error (RISE) feedback to approximate the dynamics of the follower as well as its leader using online weight tuning. Subsequently, in the second paper, a novel NN observer is designed to estimate the linear and angular velocities of both the follower and its leader robot and a NN output feedback control law is developed.

On the other hand, in the third paper, a NN-based output feedback control law is presented for the control of an underactuated quad rotor UAV, and a NN virtual control input scheme is proposed which allows all six degrees of freedom to be controlled using only four control inputs. The results of this paper are extended to include the control of quadrotor UAV formations, and a novel three-dimensional leader-follower framework is proposed in the fourth paper. Next, in the fifth paper, the discrete-time nonlinear optimal control is undertaken using two online approximators (OLA's) to solve the infinite horizon Hamilton-Jacobi-Bellman (HJB) equation forward-in-time to achieve nearly optimal regulation and tracking control. In contrast, paper six utilizes a single OLA to solve the infinite horizon HJB and Hamilton-Jacobi-Isaacs (HJI) equations forward-in-time for the near optimal regulation and tracking control of continuous affine nonlinear systems. The effectiveness of the optimal tracking controllers proposed in the fifth and sixth papers are then demonstrated using nonholonomic mobile robot formation control.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Jagannathan Sarangapani, for his guidance, support, and patience over the last several years. I also would like to thank Dr. Kelvin Erickson, Dr. Donald Wunsch, Dr. Cihan Dagli, and Dr. S. N. Balakrishnan for serving on my doctoral committee. I thank the Department of Education for providing financial support through the GAANN Fellowship program. In addition, I would like to thank the National Science Foundation for providing additional support.

I also thank my wife, Laura, my parents, Alan and Carrie, as well as the rest of my family for their love and support. Their encouragement and understanding over the course of my education has truly been a blessing. I would also like to thank God for the opportunities I have been presented with and the courage to pursue them.

TABLE OF CONTENTS

PAGE

PUBLICATION DISSERTATION OPTION....iii

ABSTRACT....iv

ACKNOWLEDGMENTS....v

LIST OF ILLUSTRATIONS....x

LIST OF TABLES....xiv

SECTION

1. INTRODUCTION.... 1

1.1 APPLICATIONS OF FORMATION CONTROL....3

1.2 OVERVIEW OF FORMATION CONTROL METHODOLOGIES....6

1.3 ORGANIZATION OF THE DISSERTATION....8

1.4 CONTRIBUTIONS OF THE DISSERTATION....12

1.5 REFERENCES....13

PAPER

1. NEURAL NETWORK CONTROL OF MOBILE ROBOT FORMATIONS USING RISE FEEDBACK....15

ABSTRACT....15

I. INTRODUCTION....16

II. LEADER-FOLLOWER FORMATION CONTROL....19

A. BACKSTEPPING CONTROLLER DESIGN....21

B. LEADER-FOLLOWER TRACKING CONTROL....22

C. DYNAMIC NN/RISE CONTROLLER DESIGN....25

D. LEADER CONTROL STRUCTURE....30

E. FORMATION STABILITY....33

III. LEADER-FOLLOWER OBSTACLE AVOIDANCE....34

IV. SIMULATION RESULTS....41

A. SCENARIO I: OBSTACLE FREE ENVIRONMENT....43

B. SCENARIO II: OBSTACLE RIDDEN ENVIRONMENT....46

V. CONCLUSIONS....47

REFERENCES ……48
APPENDIX ……50
2. NEURAL NETWORK OUTPUT FEEDBACK CONTROL OF ROBOT FORMATIONS ……...62
ABSTRACT ……....62
I. INTRODUCTION ……63
II. LEADER-FOLLOWER FORMATION CONTROL ……...66
A. BACKSTEPPING CONTROLLER DESIGN ……..68
B. LEADER-FOLLOWER TRACKING CONTROL ……..…..70
C. LEADER-FOLLOWER NN OBSERVER DESIGN ……72
D. DYNAMICAL NN TORQUE CONTROLLER ……....76
E. LEADER CONTROL STRUCTURE ……...……..81
F. FORMATION STABILITY ……..……...85
III. LEADER-FOLLOWER OBSTACLE AVOIDANCE ……..……..88
IV. SIMULATION RESULTS ……...91
A. SCENARIO I: OBSTACLE FREE ENVIRONMENT …….……93
B. SCENARIO II: OBSTACLE RIDDEN ENVIRONMENT……..96
V. CONCLUSIONS ……98
REFERENCES ……98
APPENDIX ……...……100
3. OUTPUT FEEDBACK CONTROL OF A QUADROTOR UAV USING NEURAL NETWORKS ……..111
ABSTRACT ……..111
I. INTRODUCTION ……...112
II. BACKGROUND ……..116
A. QUADROTOR UAV DYNAMICS ……..……116
B. CONSTRAINTS OF THE UNDERACTUATED SYSTEM ……..120
C. NEURAL NETWORKS ……...……121
III. NEURAL NETWORK OUTPUT FEEDBACK TRACKING CONTROL ……..…..122
A. NN OBSERVER DESIGN …….……..123
B. KINEMATIC CONTROL SYSTEM ……..…….127
C. NN VIRTUAL CONTROL DEVELOPMENT ……..133

D. NN Output Feedback Control Law ….137

E. Comments on Implementation and Practical Considerations ……145

IV. Simulation Results ……..…..147

V. Conclusions ……..156

VI. References ……...157

4. Leader-Follower Formation Control of Multiple Quadrotor Unmanned Aerial Vehicles using Neural Networks ……..160

Abstract ……..160

Nomenclature ……...……..…161

I. Introduction ……...163

II. Background ……..167

A. Quadrotor UAV Dynamics ……...……167

B. Neural Networks ……...……170

C. A Novel Three Dimensional Leader-Follower UAV Formation Control Framework ……..…171

III. Leader-Follower Formation Tracking Control ……..……174

A. Follower UAV Control Law ……...175

B. Formation Leader Control Law ……..191

C. Quadrotor UAV Formation Stability……...…..195

IV. Simulation Results ……....203

V. Conclusions ……..……..…...213

References ……..214

5. Optimal Control of Affine Nonlinear Discrete-time Systems with Unknown Internal Dynamics using Online Approximators ……..216

Abstract ……..216

I. Introduction ……...217

II. Background ……..220

III. Near Optimal Regulation of Nonlinear Systems ……..……222

A. Cost Function Approximation for Optimal Regulator Design ……..223

B. Estimation of the Optimal Feedback Control Signal ……229

C. Convergence Proof ……...…….233

IV. Near Optimal Tracking Control of Nonlinear Discrete-time Systems with Partially Unknown Dynamics ……..238

A. Cost Function Approximator Design for Tracking ……..……...242

B. Feedback Control Signal Design for Tracking ……....246

C. Nearly Optimal Control Input ……250

D. Convergence Proof ...……251

V. Simulation Results ……...…...256

VI. Conclusions ……266

References ……..……..266

6. Optimal Control of Affine Nonlinear Continuous-time Systems using an Online Approximator ……269

Abstract ……..269

I. Introduction ……...……270

II. Nonlinear Optimal Control in Continuous Time ……..……..274

A. Hamilton-Jacobi-Bellman Equation ……..……274

B. Hamilton-Jacobi-Isaacs Equation ……...277

III. Single Online Approximator-based Optimal Control Scheme ……..…..279

A. SOLA to Learn the HJB Function ……...……279

B. SOLA-based Scheme to Learn the HJI Function ……..…...292

IV. Continuous Time SOLA-based Scheme for Near Optimal Tracking ..….296

V. Simulation Results ……...……...300

A. Linear HJB Example ……..…..300

B. Nonlinear HJB Example ……302

C. Nonlinear HJI Example ……..305

D. Optimal Tracking Control of Nonholonomic Mobile Robot Formations ……308

VI. Conclusions ……..…….318

References ……..……..319

Section

2. Conclusions and Future Work..……..322

Vita ...……..328

LIST OF ILLUSTRATIONS

Figure Page

SECTION

INTRODUCTION

1.1 Leaf cutter ants, school of fish, and birds flying in formation……..2

1.2 NASA's twin Mars rovers, Spirit and Opportunity ……..4

1.3 Formation of fighter jets ……..….5

1.4 Missouri S&T autonomous trucks ……....7

1.5 Missouri S&T quadrotor UAV ……8

1.6 Dissertation outline ……...9

PAPER 1

1. Separation-bearing formation control ……20

2. General formation controller structure……34

3. Obstacle avoidance ……36

4. Formation structure ……43

5. Formation trajectories ……44

6. Formation errors……..45

7. Formation obstacle avoidance……47

PAPER 2

1. Separation-bearing formation control ……67

2. Formation controller structure ……87

3. Obstacle avoidance ……88

4. Formation structure ……92

5. Robot trajectories ……94

6. Observer estimation errors of F1 ……94

7. Formation errors……..……95

8. Trajectories in the presence of obstacles ……..……..97

9. Observer errors of F1 in the presence of obstacles ……97

PAPER 3

1. Quadrotor helicopter ……117

2. NN output feedback control structure……...141
3. UAV trajectory tracking ……...149
4. UAV tracking errors ……149
5. UAV observer estimation errors ……...150
6. UAV unmodeled dynamics, external disturbances and control signal ……152
7. UAV position error and control signals using PID control……...154
8. Control rate of change for the proposed NN controller and a conventional PID controller ……...……...156

PAPER 4

1. Quadrotor UAV ……168
2. UAV leader-follower formation control ……...172
3. Control structure for the follower and leader UAV ……..193
4. Formation control structure……...198
5. Desired formation topology ……..203
6. Quadrotor UAV formation trajectories……206
7. Leader position and orientation tracking errors……207
8. Leader translational and angular velocity tracking errors ……208
9. Position and orientation tracking errors for follower 1……...……..208
10. Velocity tracking errors for follower 1 ……209
11. Position and orientation tracking errors for follower 2 ……209
12. Velocity tracking errors for follower 2 ……210
13. Position and orientation tracking errors for follower 3……210
14. Velocity tracking errors for follower 3 ……211
15. Position and orientation tracking errors for follower 4……211
16. Velocity tracking errors for follower 4 ……212
17. Velocity tracking errors for follower 3 when the formation dynamics are ignored ……..213

PAPER 5

1. Near optimal regulator block diagram ……..233
2. Near optimal tracking control block diagram ……...251
3. Action error……...258
4. State trajectories……258

5. Adaptive critic weights ……...……...260
6. Action error ……...260
7. State trajectories for the nonlinear example……..260
8. Difference between optimal controls ……260
9. Robot trajectory ……264
10. OLA parameter estimates ……264
11. OLA feedback control and cost errors ……..265
12. OLA feedforward control term estimation error……...265

PAPER 6

1. Graphical representation of the convergence region ……...…….290
2. SOLA for HJI regulation design ……...……296
3. SOLA design for HJI optimal tracking ……..…...…….299
4. OLA weights for example 1……..301
5. System states for example 1……..302
6. OLA weights for example 2……..303
7. System states for example 2 ……...…..304
8. Divergence of the system states when the stabilizing OLA update is removed ($\Sigma(x,\hat{u}_1)=0$) for example 2……...……..305
9. OLA weights for example 3 ……..……...307
10. System states for example 3……..……307
11. Divergence of the system states when the stabilizing OLA update is removed ($\Sigma_2(x,\hat{u}_2,\hat{d})=0$) for example 3 ……..…….308
12. Leader-follower formation control ……...310
13. Desired formation ……..……310
14. OLA weights for the formation leader ……...……...312
15. OLA weights for follower 1 ……..…….……..312
16. OLA weights for follower 3 ……...……...313
17. Velocity tracking errors for the formation leader ……314
18. Velocity tracking errors for follower 1 ……314
19. Approximate Hamiltonian for the formation leader ……315
20. Formation trajectories ……...317

21. Costs for the formation leader……...317
22. Costs for follower 1 ……..318

LIST OF TABLES

Table Page

PAPER 1

I. Friction Coefficients……...……...42

II. Average Steady State Formation Errors……...46

PAPER 2

I. Friction Coefficients……...……...……....93

PAPER 3

I. Mean Squared Errors and Maximum Errors……151

PAPER 4

I. UAV Dynamic Parameters ……..205

PAPER 5

I. Cost Value Comparisons ……...……..265

SECTION

1. INTRODUCTION

In the past decade, social biological organisms such as ants, fish, and birds have inspired researchers to explore control objectives. For instance, the use of adaptive neural networks (NN's) in closed-loop feedback control systems has been motivated by biological processes such as the nervous system and its basic unit, the neuron.

Consider the social creatures shown in Fig. 1.1. The leaf cutter ants work together to harvest fresh plant matter to grow food to sustain and expand the colony. The fish swim in schools as a defense mechanism from predators and to aid in foraging for food, and the birds fly in formation to reduce the drag force that each bird experiences compared to if it was flying alone. While the objectives of each group of organisms are quite different, they share the same underlying theme. That is, by working together, the task or objective at hand can be completed more quickly and efficiently than if the task were undertaken alone. Recognizing these benefits, researchers have applied the lessons learned from nature to the control and coordination of multiple agents which include robots and unmanned aerial vehicles (UAVs), and the coordination of multiple agents has become known as robotic formation control where each robot or UAV in the group seeks to orient itself relative to its neighbor or a leader.

Just as the social organisms described above orient themselves relative to one another to complete there respective objectives more effectively, the concept of formation control is to arrange the robot or UAVs relative to each other so that the mission is successfully completed more quickly and efficiently. For example, the robots or UAVs

Fig 1.1 Leaf cutter ants[1], school of fish[2], and birds flying in formation[3].

may be equipped with sensors that have limited sensing capabilities. If a single agent were assigned to sweep a large area using its limited sensing, the task could take a very long time. However, by increasing the number of robots or UAVs and strategically arranging them, the formation of robots and UAVs can complete the task quicker and more efficiently than a single robot or UAV acting along can.

Thus, the benefits of controlling a team of robots or UAVs over controlling a single agent have stimulated the interests of many researchers, and the attention has shifted from the control of a single robot or UAV to controlling formations of robots or

[1] Photo courtesy of: http://dsc.discovery.com/news/2008/09/10/gallery/leaf-cutter-ants-324x205.jpg
[2] Photo courtesy of: http://photography.nationalgeographic.com/staticfiles/NGS/Shared/StaticFiles/Photography/Images/POD/f/fish-and-coral-tuamotu-513704-xl.jpg
[3] Photo courtesy of: http://www.wunderground.com/data/wximagenew/r/Ralfo/561.jpg

UAVs. Next, several applications of formation control of robots and UAV's are considered, and the benefits over controlling a single agent are discussed.

1.1 APPLICATIONS OF FORMATION CONTROL

A team of mobile robots or unmanned aerial vehicles (UAVs) working together is often more effective than a single agent acting alone in applications like surveillance, search and rescue, perimeter security, and exploration of unknown and/or hazardous environments to name a few. In addition to redundancy, a team of robots each with a variety of sensors offers the opportunity for increased sensor coverage when compared to a single mobile sensor or multiple stationary sensors. Therefore, mobile sensor networks are preferred over a single suite of sensors.

For example, in January 2004, the National Aeronautics and Space Administration (NASA) successfully landed two identical rovers on Mars known as Spirit and Opportunity, and shown in Fig. 1.2. For more than five years, the two robots have accumulated more than 15 miles in total odometry [2]. Had NASA only deployed a single rover, the total odometry could have been as low as 5 miles [2] (approximate total distance traveled by Spirit rover as of July 15, 2009). In contrast, by deploying a fleet of rovers to the surface of Mars, the unknown terrain could have been systematically divided and explored autonomously while providing scientists with an increased amount and wider variety of data from the Martian surface compared to the amount of data provided by just two rovers. In addition, increasing the number of robots provides redundancy and decreases the chances of complete mission failure.

In addition to exploring foreign planets, formation control can also be applied to satellite formation flying where multiple smaller satellites work together to perform the

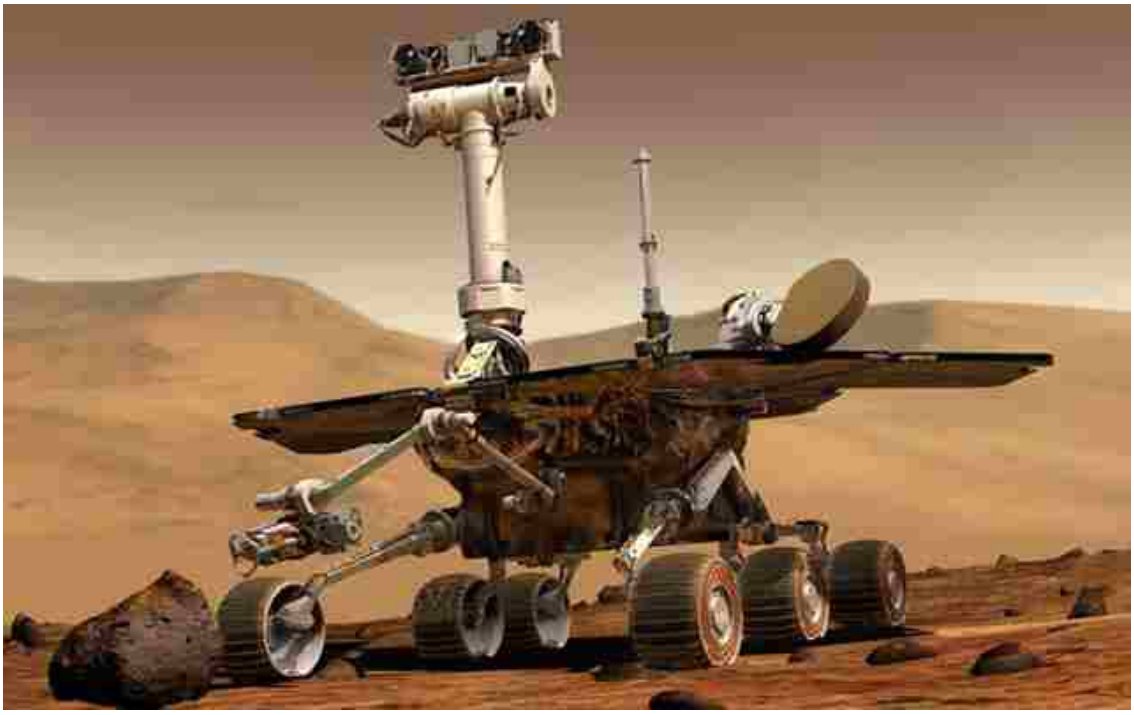


Fig. 1.2. NASA's twin Mars rovers, Spirit and Opportunity[4].

task normally accomplished by one larger and more expensive satellite [3]. Not only are the smaller satellites often cheaper and quicker to build, they provide an increase in the resolution that can be achieved by a single satellite, and they have the ability to view targets from multiple angles or at multiple times. These qualities make them ideal for meteorological, environmental, astronomy, and communications applications [3]. In addition, increasing the number of satellites adds redundancy and robustness for successfully completing the desired tasks.

A well known example of formation control is a squadron of fighter jets flying in formation as shown in Fig. 1.3 where the formation traditionally consists of jets flown by

[4] Photo courtesy of: http://www.jpl.nasa.gov/images/missions/Mer640.jpg

Fig. 1.3. Formation of fighter jets[5].

well-trained pilots. Now, recent advances in technology have paved the way for unmanned jets to fly in formation with both manned and unmanned aircraft [4] where objectives include flying in tight formations so that a reduction in the formation's induced drag is achieved [5]. By reducing the drag incurred on the formation, the team of UAV's reduces their fuel consumption, and thus, they can achieve longer flight durations [5]. However, as a result of a follower UAV flying in close proximity to its leader, the follower must not only consider its own dynamics, but also the dynamics of its leader. That is, the formation dynamics must be considered.

The examples above have illustrated three of the many possible applications of robotic formation control. In addition, these examples have brought to light several benefits of formation control over employing a single agent as well as several

[5] Photo courtesy of: http://www.baseops.net/militarypilot/at38_formation.jpg

considerations that should be taken into account in the design of formation control laws. That is, the designed formation control laws must ensure that the formation errors are small to ensure the success of the mission as well as the stability of the formation. In addition, the dynamic effects of the leader on the follower robot (formation dynamics) should be explicitly considered. Observing that autonomous robots and UAV's are often powered by batteries, the task of achieving and maintaining the desired formation should also be completed in an optimal manner to extend the duration of a mission and thus reduce the risk of mission failure due to depleted power. The optimal use of system resources becomes especially important in tasks such as the Mars rover and satellite formation examples described above where simply replacing batteries is not an option.

Next, an overview of current methodologies for robotic and UAV formation control is presented, and their shortcomings are exposed. Subsequently, the organization of this dissertation is presented.

1.2 Overview of Formation Control methodologies

For the formation control of wheeled mobile robots shown in Fig. 1.4, many researchers [1] have simplified their approaches by considering only the kinematic system of the robot thereby ignoring the robot and formation dynamics. As observed from robot arm control, the dynamics must be considered in practice to guarantee that the robots track a desired velocity while avoiding the use of large control gains which would become necessary to dominate the neglected dynamics in order to ensure an acceptable performance [6]. Similarly, experimental studies have illustrated the need for dynamical controllers for wheeled mobile robots with high inertia, high operating speeds, significant unmodeled dynamics, or high system noise [7].

Likewise, the control of quadrotor UAVs, similar to the UAV shown in Fig. 1.5, is often accomplished by making small angle approximations and considering simplified dynamics. However, experimental studies have shown that the above simplifications are valid only at very low speeds such as hovering while the aerodynamic effects can become significant even at moderate velocities causing instability of the UAV [8]. In addition, for the formation control of UAV's, cylindrical coordinates and contributions from wheeled mobile robot leader follower formation control [1] have been extended for aircrafts by assuming the dynamics are known [9]. However, it is desirable to solve the UAV formation control problem without requiring full knowledge of the system dynamics while in a coordinate system that is better suited for a three-dimensional (3D) formation, such as spherical coordinates, since the type of sensor measurements required to solve the 3D-formation control problem are often in a spherical coordinate system.

Fig 1.4. Missouri S&T autonomous trucks.

Fig. 1.5 Missouri S&T quadrotor UAV.

In addition, the stability of the wheeled mobile robot or UAV is often the sole consideration of many existing formation control schemes [1],[9]. However, as described above, optimal use of system resources if often required so as to extend the duration of the mission while improving the likelihood completing the task at hand. Thus, the control laws derived in this dissertation seek to address the shortcomings described above.

1.3 ORGANIZATION OF THE DISSERTATION

In this dissertation, the control of nonholonomic mobile robot formations and UAV formations is undertaken while relaxing the above common assumptions and simplifications. This dissertation is presented in six papers, and their relation to one another is illustrated in Fig. 1.6. The common theme of each paper is the formation control of wheeled mobile robots and UAV's. The first two papers deal with wheeled

mobile robots and address the asymptotic stability of the formation and output feedback controller designs, respectively, when the dynamics of the robots and formation are unknown. The third and forth papers consider the output feedback control of a single quadrotor UAV and state feedback control of formations of quadrotor UAVs, respectively, in the presence of unknown dynamics. The final two papers of the dissertation consider solving the Hamilton-Jacobi-Bellman (HJB) equation in both discrete and continuous time frameworks, respectively. Additionally, the contributions of the final paper are extended to solve the Hamilton-Jacobi-Isaacs (HJI) equation commonly used in H_∞ optimal control. The effectiveness of the optimal control laws derived in the final two papers is demonstrated using wheeled mobile robots.

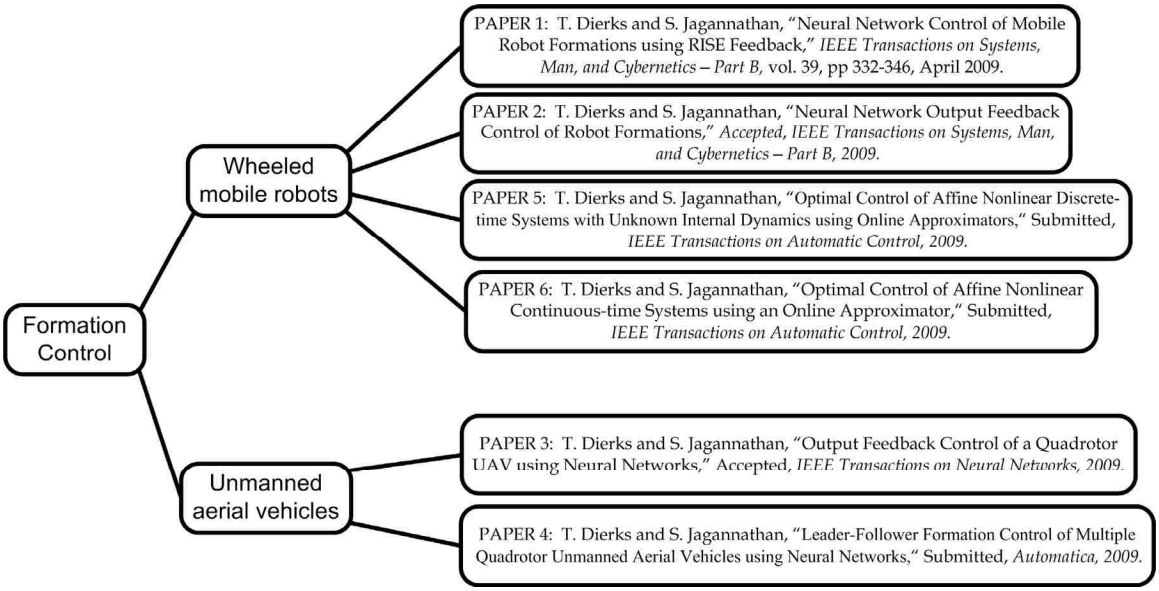


Fig 1.6. Dissertation outline.

In the first paper, an asymptotically stable combined kinematic/torque control law is developed for leader-follower based formation control of mobile robots using backstepping in order to accommodate the complete dynamics of the robots and the formation. A NN is introduced along with robust integral of the sign of the error (RISE) feedback to approximate the dynamics of the follower as well as its leader using online weight tuning, and Lyapunov theory guarantees that the tracking errors are asymptotically stable as opposed to uniformly ultimately bounded (UUB) stability which is typical with most NN controllers. In comparison to our previous work [10], the RISE method achieves asymptotic stability by using the integral of a high-gain term whereas the method in [10] attained asymptotic stability through a robust adaptive term. Subsequently, a NN output feedback control law is developed requiring minimal communication in the second paper. Further, a novel NN observer is designed to estimate the linear and angular velocities of both the follower robot and its leader.

In the third paper, a novel NN output feedback control law is presented for the control of an underactuated quad rotor UAV. Although a quadrotor UAV is underactuated, a novel NN virtual control input scheme is proposed which allows all six degrees of freedom of the UAV to be controlled using only four control inputs. Furthermore, a NN observer is introduced to estimate the translational and angular velocities of the UAV. In paper four, we extend the results of paper three to include the control of UAV formations, and a new leader-follower formation control framework is proposed for UAVs based on spherical coordinates where the desired trajectory of a follower UAV is specified using a desired- separation, angle of incidence, and bearing , s_d, α_d, β_d, respectively, relative to its leader. In the proposed formation control

formulation, the desired separation, angle of incidence and bearing angle will be utilized to define a desired trajectory of a follower UAV relative to its leader, so as to convert the formation control problem into a tracking control problem.

Our previous work [11] explored solving the HJB equation using offline training and NN's. For the approach in [11], an additional NN was utilized to relax the need of exact knowledge of the system dynamics. In contrast, direct dynamic programming techniques are utilized in paper five to solve the infinite horizon Hamilton Jacobi-Bellman (HJB) equation online and forward-in-time time for the optimal control of general affine nonlinear discrete-time systems. The proposed approach, referred normally as adaptive dynamic programming, uses online approximators (OLA's) to solve the infinite horizon optimal regulation and tracking control of affine nonlinear discrete-time systems in the presence of unknown internal dynamics and a known control coefficient matrix. Novel tuning laws for the OLA's are derived, and all parameters are tuned online. Lyapunov techniques are used to show that all signals are *UUB* and that the approximated control signals approach the optimal control inputs with small bounded error. The effectiveness of proposed nearly optimal tracking controller scheme is verified using a nonholonomic mobile robot. In addition, the online optimal control scheme is applied to the formation control of nonholonomic mobile robots in [12].

In the final paper, a novel single online approximator (SOLA)-based scheme is designed to solve the optimal regulation and tracking control problems for continuous nonlinear affine systems with known dynamics. The SOLA-based adaptive approach is designed to learn the infinite horizon continuous time HJB equation and the corresponding optimal control input that minimizes the HJB equation forward-in-time.

Subsequently, the SOLA architecture is extended to learn the HJI equation commonly used in H_∞ optimal control. Novel tuning algorithms are derived which not only ensures the optimal cost (HJB or HJI) function and control input are achieved, but also ensures the system states remain bounded during the online learning process. Lyapunov techniques are used to show that all signals are *UUB* and that the approximated control signals approach the optimal control inputs with small bounded error. In the absence of OLA reconstruction errors, an optimal control is demonstrated.

1.4 CONTRIBUTIONS OF THE DISSERTATION

This dissertation provides contributions to the field of robot and UAV formation control as well as to the control of general nonlinear systems. The control laws developed in this dissertation in the context of formation control explicitly compensate for the dynamics of the individual agents as well as the dynamics of the entire formation, and the stability of the formation is demonstrated in each case. Further, the contributions of paper 2 illustrate how the formation control objective can be achieved using limited communication and minimal sensor measurements by using output feedback. For the control of UAV's and UAV formations, the control laws derived in this dissertation are independent of a specific operating point and do not require any small angle approximations. Although a UAV underactuated, the control of all system states is achieved using a novel virtual controller structure. Additionally, the formation control laws derived in this work do not require complete knowledge of the system or formation dynamics as the NN's learn them all online.

In addition to robot formations, the NN/RISE feedback method developed in paper 1 allows the asymptotic stability of general nonlinear affine systems to be shown in the presence of uncertainties and disturbances that have time varying upper bounds. Asymptotic stability is a much stronger result than the boundedness results which typically arise in presence of bounded uncertainties and disturbances [13]. The contributions of papers 5 and 6 also pertain to general nonlinear affine systems in discrete- and continuous-time, respectively, and both provide novel online optimal control schemes to learn the HJB or HJI equations forward in time in contrast to optimal control methods which develop backwards in time [14]. Additionally, the schemes in papers 5 and 6 explicitly consider the approximation and OLA reconstruction errors in the stability proofs which is not typical of current approaches.

1.5 REFERENCES

[1] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *IEEE Trans. on Robotics and Automation*, vol. 18, pp. 813-825, October 2002.

[2] Mars Exploration Rover Mission (2009). Retrieved July 15, 2009, from http://marsrover.nasa.gov/mission/status.html.

[3] N. Short, *The Remote Sensing Tutorial*, NASA Reference Publication 1078, Library of Congress Catalog Card No. 81-600117, 2009.

[4] S. Waydo, J. Hauser, R. Bailey, E. Klavins, and R. M. Murray, "UAV as a reliable wingman: a flight demonstration," *IEEE Trans. on Control Systems Tech.* vol. 15, no. 4, pp. 680-688, July 2007.

[5] M. Pachter, J. J. D'Azzo, and A. W. Proud, "Tight formation flight control," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 246-254, 2001.

[6] T. J. Tarn, A. K. Bejczy, X. Yun, and Z. Li, "Effect of motor dynamics on nonlinear feedback robot arm control," *IEEE Trans. On Robotics and Automation,* vol. 7, pp. 114-122, February 1991.

[7] D. DeVon and T. Bretl, "Kinematic and dynamic control of a wheeled mobile robot," *Proc. IEEE Intl. Conference on Intelligent Robots and System,* pp. 4065-4070, November 2007.

[8] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: theory and experiment," *Proc. of the AIAA Guidance, Navigation and Control Conference and Exhibit*, South Carolina, pp. 1-20, August 2007.

[9] R. Fierro, C. Belta, J. P. Desai, and V. Kumar, "On controlling aircraft formations." *Proc. IEEE Conf. on Decision and Control,* pp. 1065-1079, 2001.

[10] T. Dierks and S. Jagannathan, "Asymptotic Adaptive Neural Network Tracking Control of Nonholonomic Mobile Robot Formations, *Journal of Intelligent and Robotic Systems,* vol. 56, no. 1, 2009.

[11] T. Dierks, B. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Networks,* vol. 22, pp. 851-860, 2009.

[12] T. Dierks, B. Brenner, and S. Jagannathan, "Discrete-time Optimal Control of Nonholonomic Mobile Robot Formations using Linearly Parameterized Neural Networks," *International Journal of Robotics and Automation, 2009 Special Issue*, submitted, 2009.

[13] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural network control of robot manipulators and nonlinear system,* Taylor and Francis, Philadelphia, PA, 1999.

[14] F. L. Lewis and V. L. Syrmos, *Optimal Control*, 2nd ed. Hoboken, NJ: Wiley, 1995.

PAPER 1

Neural Network Control of Mobile Robot Formations using RISE Feedback[1]

Travis Dierks and S. Jagannathan

***Abstract*—*In this paper, an asymptotically stable combined kinematic/torque control law is developed for leader-follower based formation control using backstepping in order to accommodate the complete dynamics of the robots and the formation, and a neural network (NN) is introduced along with robust integral of the sign of the error (RISE) feedback to approximate the dynamics of the follower as well as its leader using online weight tuning. It is shown using Lyapunov theory that the errors for the entire formation are asymptotically stable and the NN weights are bounded as opposed to uniformly ultimately bounded (UUB) stability which is typical with most NN controllers. Additionally, the stability of the formation in the presence of obstacles is examined using Lyapunov methods, and by treating other robots in the formation as obstacles, collisions within the formation do not occur. The asymptotic stability of the follower robots as well as the entire formation during an obstacle avoidance maneuver is demonstrated using Lyapunov methods, and numerical results are provided to verify the theoretical conjectures.*

Keywords: Neural network, formation control, Lyapunov method, kinematic/dynamic controller, RISE.

[1] Research Supported in part by GAANN Program through the Department of Education and Intelligent Systems Center. Authors are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology (formerly University of Missouri-Rolla), 1870 Miner Circle, Rolla, MO 65409. Contact author Email: tad5x4@mst.edu.

I. INTRODUCTION

For complex tasks like search and rescue operations, mapping unknown or hazardous environments, security and bomb sniffing, a team of robots working together offers many advantages over employing a single robot. Recognizing these benefits, robotic formation control has become the focus of many research efforts [1-18], and several different approaches to the problem have been proposed including behavior-based, generalized coordinates, virtual structures, and leader-follower, to name a few [1]. Separation-separation and separation-bearing [2-3] are two popular techniques in leader-follower formation control, and in this work, the latter will be considered where the followers stay at a specified separation and bearing from its designated leader.

Many formation control works [2-7] have proposed kinematic based control laws to keep the formation. Thus, perfect velocity tracking assumptions are required to ensure the desired formation is achieved as well as guarantee the stability of the formation. Therefore, numerous works [8-16] have proposed solution to formation control problem which include the robot dynamics. In [8], a neural network (NN) is introduced to learn the dynamics of the follower robots. The work in [9], [10], and [11] propose decentralized approaches based on virtual points, potential functions, and the abilities of the individual robots, respectively; however, in each case, only the inertial matrix of the robots is considered, and dynamics like the centripetal and coriolis matrix and the friction vector are ignored. In [12], a centralized control scheme is developed, and a PD controller is proposed to ensure velocity tracking; however, the derivative of the control velocity is neglected. Alternatively, the work in [13] proposes a dynamical control scheme for leader-follower based formation control which considers the dynamics of the

robots and guarantees collisions do not occur among them. However, this control scheme is derived using potential as well as bump functions which must be at least three times differentiable. In each of these works [8-13], the dynamics of the follower robots are considered whereas the effect of the dynamics of the leader on the follower (formation dynamics) is still ignored.

Our previous work [14] demonstrated that the dynamics of the lead robot are incorporated into the torque control inputs of the follower robots through the derivative of the follower's kinematic control velocity which was found to be a function of its leader's velocity. Consequently, in a formation of robots where a follower robot follows another robot directly in front of it, by considering its leader's dynamics, a robot inherently considers the dynamics of the robots in front of them. The dynamical extension in [14] provided a rigorous method of taking into account specific robot and formation dynamics; however, the dynamics of each robot were considered known. Therefore, in our previous work [15], a NN was introduced to learn the unknown dynamics of each robot as well as the dynamics of its respective leader, and the formation errors were shown to be *Uniformly Ultimately Bounded (UUB)* [20].

By contrast, the contribution of this work lies in a new asymptotically stable NN torque control law using a NN combined with the recently developed robust integral of sign of the error feedback method originating in [18] and referred to as RISE feedback in [19]. The asymptotic stability of the entire formation as well as the boundedness of the NN weights is shown using Lyapunov methods as opposed to UUB, a result common in the NN controls literature [15],[20]. The RISE method [19] is designed to reject bounded unmodeled disturbances, like NN functional reconstruction errors, to yield asymptotic

tracking. An approach to blend a multilayer NN with RISE feedback for a single rigid robot control is taken in [19] where the boundedness of the actual NN weights is shown separately using projection algorithm while the convergence of the tracking errors is then demonstrated by using constant controller gains. Selection of the predefined convex set in the projection algorithm to prevent the NN weights from diverging is a challenging task since the convex set must be carefully chosen to contain the ideal weights.

By contrast, in this work, a novel weight tuning is used in this work instead of the projection algorithm [19], and the constant bounds and gains in [19] are replaced here for formation control with time varying functions allowing bounds and gains to be determined with more certainty. Further, Lyapunov analysis is presented to show the asymptotic convergence of the tracking errors and boundedness of the NN weights simultaneously. The bounds and gains developed here are also applicable to single rigid robot control [19] besides formation control.

Finally, it is shown that the proposed formation controller achieves stability even in the presence of obstacles by integrating the RISE method into a simple, but effective obstacle avoidance scheme which allows each follower robot to navigate around obstacles while simultaneously tracking its leader. When an obstacle is encountered, the desired separation and bearing of the follower robot are modified so that the follower navigates around the obstacle. Similar to [13], collisions within the formation are avoided in this work too, but without the need of the additional assumption that higher order derivatives are available. Other works that have considered the formation in the presence of obstacles include [8] and [10] where potential functions were utilized. Additionally, the concept of potential trenches was applied in [16] whereas the dynamic

window approach was utilized [17]. Therefore, the contributions of this manuscript include: a) development of a novel formation control law by incorporating the dynamics of the leader, follower and formation; b) proof of asymptotic stability using Lyapunov stability even with using NN for approximating the leader and follower dynamics and their interactions; and c) simplified scheme to avoid collisions among the robots and with obstacles.

This paper is organized as follows. First, in Section II, the leader-follower formation control problem is introduced, and required background information is presented. Then, the NN/RISE feedback control law is developed for the follower robots as well as the formation leader, and the stability of the overall formation is presented along with a general formation controller structure. In Section III, a leader-follower obstacle avoidance scheme is developed, and Section IV presents numerical simulations. Section V provides some concluding remarks.

II. Leader-Follower Formation Control

Background information on leader-follower formation control is introduced next. Throughout the development, follower robots will be denoted with a subscript 'j' while the leader will be denoted by the subscript 'i'. The goal of separation-bearing formation control is to find a velocity control input such that

$$\lim_{t\to\infty}(L_{ijd} - L_{ij}) = 0 \text{ and } \lim_{t\to\infty}(\Psi_{ijd} - \Psi_{ij}) = 0 \tag{1}$$

where L_{ij} and Ψ_{ij} are obtained using local sensory information and denote the measured separation and bearing of the follower j with respect to leader i while L_{ijd} and Ψ_{ijd} represent desired distance and angles, respectively [2-3], as shown in Fig. 1.

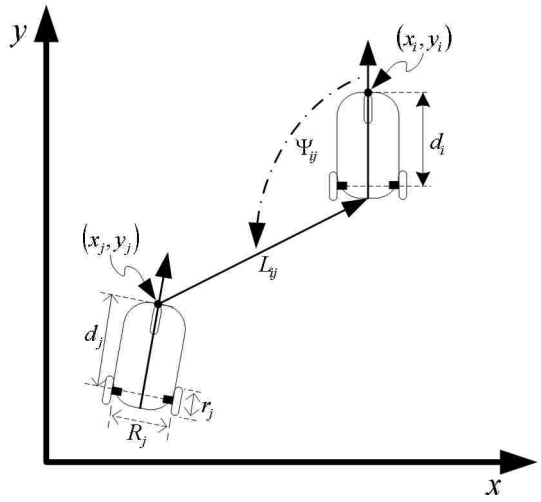


Fig. 1. Separation-bearing formation control.

The kinematic equations for the front of the j^{th} follower robot, (x_j, y_j), can be written as

$$\dot{q}_j = \begin{bmatrix} \dot{x}_j \\ \dot{y}_j \\ \dot{\theta}_j \end{bmatrix} = \begin{bmatrix} \cos\theta_j & -d_j \sin\theta_j \\ \sin\theta_j & d_j \cos\theta_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_j \\ \omega_j \end{bmatrix} = S_j(q_j)\bar{v}_j \tag{2}$$

where d_j is the distance from the rear axle to the to front of the robot, $q_j = [x_j \;\; y_j \;\; \theta_j]^T$ denotes the actual Cartesian position for the front of the robot and orientation, respectively, v_j, and ω_j represent linear and angular velocities, respectively, and $\bar{v}_j = [v_j \;\; \omega_j]^T$. Many robotic systems can be characterized as a system having an *n*-dimensional configuration space $\mathcal{C}$ with generalized coordinates $(q_1, \ldots q_n)$ subject to ℓ constraints [23]. Applying the transformation [23], the dynamics of the mobile robots are given by

$$\bar{M}_j \dot{\bar{v}}_j + \bar{V}_{mj}(q_j, \dot{q}_j)\bar{v}_j + \bar{F}_j(\bar{v}_j) + \bar{\tau}_{d_j} = \bar{\tau}_j \tag{3}$$

where $\overline{M}_j \in \Re^{\rho x \rho}$ is a constant positive definite inertia matrix, $\overline{V}_{mj} \in \Re^{\rho x \rho}$ is the bounded centripetal and coriolis matrix, $\overline{F}_j \in \Re^{\rho}$ is the friction vector, $\overline{\tau}_{dj} \in \Re^{\rho}$ represents unknown bounded disturbances such that $\left\|\overline{\tau}_{dj}\right\| \leq d_M$ and $\left\|\dot{\overline{\tau}}_d \;\; \ddot{\overline{\tau}}_d\right\| \leq d'_M$ for known constants d_M and d'_M, $\overline{B}_j \in \Re^{\rho x \rho}$ is a constant, nonsingular input transformation matrix, $\overline{\tau}_j = \overline{B}_j \tau_j \in \Re^{\rho}$ is the input vector, and $\tau_j \in \Re^{\rho}$ is the control torque vector. For complete details on (3) and the parameters that comprise it, see [23]. It should be noted that for the nonholonomic system of (2) and (3) with n generalized coordinates q, ℓ independent constraints, and ρ actuators, the number of actuators is equal to $n - \ell$, and for this work $n = 3, \ell = 1, \rho = 2$. We will also apply the assumption from [23] that the linear and angular velocities are bounded for all time, *t*.

A. Backstepping Controller Design

The complete description of the behavior of a mobile robot is given by (2) and (3). The NN/RISE controller is introduced so that the specific torque $\overline{\tau}_j(t)$ may be calculated in order that (2) and (3) exhibit the desired behavior for a given control velocity $v_{jc}(t)$ without knowing the complete dynamics of the formation.

In this work, a two-layer NN consisting of one layer of randomly assigned constant weights $V \in \Re^{axL}$ in the input layer and one layer of tunable weights $W \in \Re^{Lxb}$ in the output layer, with a inputs, b outputs, and L hidden neurons are considered. The *universal approximation property* for NN [20] states that for any smooth function $f(x)$, there exists a NN such that $f(x) = W^T \sigma(V^T x) + \varepsilon$ for some ideal weights W, V, where ε is the NN functional approximation error, and $\sigma(\cdot) : \Re^a \rightarrow \Re^L$ is the activation function

in the hidden layers. It has been shown that by randomly selecting the input layer weights V, the activation function $\sigma(\bar{x}) = \sigma(V^T x)$ forms a stochastic basis, and thus the approximation property holds for all inputs, $x \in \Re^a$, in the compact set S [20]. Also, the functional approximation error is bounded such that $\|\varepsilon\| < \varepsilon_N$ where ε_N is a known bound and dependent on S [20]. The sigmoid activation function is considered here. For complete details of the NN and its properties, see [20].

***Remark 1**:* Throughout this paper, $\|\cdot\|$ and $\|\cdot\|_F$ will be used interchangeably as the Frobenius vector and matrix norms, respectively [20].

B. Leader-Follower Tracking Control

To complete the separation-bearing formation control objective (1), contributions from single robot control frameworks such as [23] are extended to leader-follower formation control. Consider the tracking controller error system from [23] for a single robot as

$$e_j = T_{ej}(q_{jr} - q_j) = \begin{bmatrix} e_{j1} \\ e_{j2} \\ e_{j3} \end{bmatrix} = \begin{bmatrix} \cos\theta_j & \sin\theta_j & 0 \\ -\sin\theta_j & \cos\theta_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{jr} - x_j \\ y_{jr} - y_j \\ \theta_{jr} - \theta_j \end{bmatrix} \tag{4}$$
$$\dot{x}_{jr} = v_{jr}\cos\theta_{jr},\ \dot{y}_{jr} = v_{jr}\sin\theta_{jr},\ \dot{\theta}_{jr} = \omega_{jr},\ \dot{q}_{jr} = \begin{bmatrix} \dot{x}_{jr} & \dot{y}_{jr} & \dot{\theta}_{jr} \end{bmatrix}^T$$

where x_{jr}, y_{jr}, θ_{jr}, and $\bar{v}_{jr} = [v_{jr} \quad \omega_{jr}]^T$ are the Cartesian position in the x and y direction, orientation and the linear and angular velocities, respectively, of a virtual reference robot for robot j [23]. In a single robot control, a steering control input $v_{jc}(t)$ is designed to solve three basic problems: path following, point stabilization, and trajectory following such that $\lim_{t\to\infty}(q_{jr} - q_j) = 0$ and $\lim_{t\to\infty}(\bar{v}_{jr} - \bar{v}_j) = 0$ [23]. If the mobile robot

controller can successfully track a class of smooth velocity control inputs, then all three problems can be solved with the same controller [23].

To extend the contributions from single robot control frameworks such as (4) to leader-follower formation control, we begin by replacing the virtual reference cart with a physical mobile robot acting as the leader i for follower j subject to kinematics and dynamics that are defined similarly to (2) and (3), respectively. Then, define a reference position at a desired separation L_{ijd} and a desired bearing Ψ_{ijd} for follower j with respect to the rear of leader i as

$$x_{jr} = x_i - d_i \cos\theta_i + L_{ijd} \cos(\Psi_{ijd} + \theta_i), \qquad y_{jr} = y_i - d_i \sin\theta_i + L_{ijd} \sin(\Psi_{ijd} + \theta_i) \tag{5}$$

as well as a reference orientation, θ_{jr} that will be defined in the proceeding discussion. Next, define the actual position and orientation of follower j as

$$x_j = x_i - d_i \cos\theta_i + L_{ij} \cos(\Psi_{ij} + \theta_i), \qquad y_j = y_i - d_i \sin\theta_i + L_{ij} \sin(\Psi_{ij} + \theta_i), \qquad \theta_j = \theta_j \tag{6}$$

where L_{ij} and Ψ_{ij} are the actual separation and bearing of follower j measured relative to the rear of the leader i. Substitution of (5) and (6) into the error system (4), and applying basic trigonometric identities, the kinematic error for leader-follower formation control is obtained as

$$e_j = \begin{bmatrix} e_{j1} \\ e_{j2} \\ e_{j3} \end{bmatrix} = \begin{bmatrix} L_{ijd} \cos(\Psi_{ijd} + \theta_{ij}) - L_{ij} \cos(\Psi_{ij} + \theta_{ij}) \\ L_{ijd} \sin(\Psi_{ijd} + \theta_{ij}) - L_{ij} \sin(\Psi_{ij} + \theta_{ij}) \\ \theta_{jr} - \theta_j \end{bmatrix} \tag{7}$$

where $\theta_{ij} = \theta_i - \theta_j$ and θ_{jr} is the reference orientation. Due to the nonholonomic constraint as well as the separation-bearing formation control objective, the orientations of each robot in the formation will not be equal while the formation is turning, and thus, the reference orientation of each robot cannot be chosen such that $\theta_{jr} = \theta_i$. However,

choosing the reference orientation relative to the leader satisfying the differential equation

$$\dot{\theta}_{jr} = \frac{1}{d_j}\left(\omega_i L_{ijd}\cos(\psi_{ijd} + \theta_{ij}) + v_i \sin(\theta_{ijr}) + k_{j2} e_{j2}\right), \tag{8}$$

the asymptotic stability of all three error states can be shown, where $\theta_{ijr} = \theta_i - \theta_{jr} \in [-\pi, \pi]$ and k_{j2} is a positive design constant. Further, it can be shown that the reference orientation of the follower will become equal the orientation of the leader ($\theta_i - \theta_{jr} = 0$) after formation errors have converged to zero and when $v_i > 0$ and $\omega_i = 0$ which is a desirable attribute. The transformed error system (7) now acts as a formation tracking controller which not only seeks to remain at a fixed desired distance L_{ijd} with a desired angle Ψ_{ijd} relative to the leader robot i, but also will achieve a relative orientation with respect to the leader. By taking the desired separation and bearing, L_{ijd} and Ψ_{ijd}, as constants similar to other works, and observing the derivatives of the separation and bearing, $\dot{L}_{ij}$ and $\dot{\Psi}_{ij}$ defined in [2], the error dynamics of (7) are found to be

$$\begin{bmatrix} \dot{e}_{j1} \\ \dot{e}_{j2} \\ \dot{e}_{j3} \end{bmatrix} = \begin{bmatrix} -v_j + v_i \cos\theta_{ij} + \omega_j e_{j2} - \omega_i L_{ijd} \sin(\Psi_{ijd} + \theta_{ij}) \\ -\omega_j e_{j1} + v_i \sin\theta_{ij} - d_j \omega_j + \omega_i L_{ijd} \cos(\Psi_{ijd} + \theta_{ij}) \\ \frac{1}{d_j}\left(\omega_i L_{ijd}\cos(\Psi_{ijd} + \theta_{ij}) + v_i \sin(\theta_{ijr}) + k_{j2} e_{j2}\right) - \omega_j \end{bmatrix}. \tag{9}$$

To stabilize the kinematic system, we propose the following velocity control inputs which are derived using Lyapunov methods for follower robot j to achieve the desired position and orientation with respect to leader i as

$$v_{jc} = \begin{bmatrix} v_{jc1} \\ v_{jc2} \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_{ij} + k_{j1} e_{j1} - \omega_i L_{ijd} \sin(\Psi_{ijd} + \theta_{ij}) \\ \frac{1}{d_j}\left(\omega_i L_{ijd}\cos(\Psi_{ijd} + \theta_{ij}) + v_i \sin(\theta_{ijr}) + k_{j2} e_{j2} + k_{j3} e_{j3}\right) \end{bmatrix} \tag{10}$$

where $K_j = [k_{j1} \; k_{j2} \; k_{j3}]^T$ is a vector of positive design constants. Next, define the velocity tracking error as

$$e_{jc} = [e_{j4} \quad e_{j5}]^T = v_{jc} - \bar{v}_j = [v_{jc1} \quad v_{jc2}]^T - [v_j \quad \omega_j]^T \tag{11}$$

Observing $\bar{v}_j = v_{jc} - e_{jc}$, substituting the control velocity (10) into the error dynamics of (9) and applying basic trigonometric identities reveals that

$$\begin{bmatrix} \dot{e}_{j1} \\ \dot{e}_{j2} \\ \dot{e}_{j3} \end{bmatrix} = \begin{bmatrix} -k_{j1}e_{j1} + \omega_j e_{j2} + e_{j4} \\ 2v_i \sin\left(\frac{e_{j3}}{2}\right)\cos\left(\theta_i - \frac{\theta_{jr} + \theta_j}{2}\right) - k_{j2}e_{j2} - k_{j3}e_{j3} - \omega_j e_{j1} + d_j e_{j5} \\ -\frac{k_{j3}}{d_j}e_{j3} + e_{j5} \end{bmatrix}. \tag{12}$$

Examining the closed loop error dynamics (12), it is clear that the stability of the kinematic system is dependent on the velocity tracking error. Additionally, the origin $e_j = 0$ and $e_{jc} = 0$ consisting of the position, orientation and velocity tracking errors for follower *j,* is an equilibrium point of the closed loop kinematic error dynamics (12).

C. Dynamical NN/RISE Controller Design

In the previous section, it was shown that the stability of the kinematic error system depends on the velocity tracking error. Therefore, the dynamics of the mobile robot are now considered, and a velocity tracking loop is designed to ensure $\bar{v}_j \to v_{jc}$ asymptotically.

To begin the development, define the velocity filtered tracking errors as

$$r_j = \dot{e}_{jc} + \alpha_j(t) e_{jc} \tag{13}$$

where $\alpha_j(t)$ is a time varying real function greater than zero defined as $\alpha_j(t) = \alpha_{j0} + \alpha_{j1}(t)$ where α_{j0} is a constant and $\alpha_{j1}(t)$ is a time varying term. Multiplying both sides of (13)

by $\overline{M}_j$, adding and subtracting $\overline{V}_{m_j} v_{jc}$ and $\overline{F}_j(v_{jc})$, and substituting the robot dynamics (3) allows (13) to be rewritten as

$$\overline{M}_j r_j = f_{d_j} + T_j + \overline{\tau}_{d_j} - \overline{\tau}_j \tag{14}$$

where

$$f_{d_j} = \overline{M}_j \dot{v}_{jc} + \overline{V}_{m_j} v_{jc} + \overline{F}_j(v_{jc}), \quad T_j = e_{jc}(\alpha_j(t)\overline{M}_j - \overline{V}_{m_j}) + \overline{F}_j(\overline{v}_j) - \overline{F}_j(v_{jc}) \tag{15}$$

Differentiating (14) then yields the filtered tracking error dynamics

$$\overline{M}_j \dot{r}_j = -\dot{\overline{M}}_j r_j + \dot{f}_{d_j} + \dot{T}_j + \dot{\overline{\tau}}_{d_j} - \dot{\overline{\tau}}_j. \tag{16}$$

Using the *universal approximation property* for NN's [20], define $\dot{f}_{d_j} = W_j^T \sigma(V_j^T x_{dj}) + \varepsilon_j$ where W_j^T, V_j^T are bounded constant ideal weights such that $\|W_j\|_F \le W_M$ for a known constant W_M, ε_j is the bounded NN reconstruction error such that $\|\varepsilon_j\| \le \varepsilon_{jM}, \|\dot{\varepsilon}_j\| \le \varepsilon'_{jM}$ for known constants ε_{jM} and ε'_{jM}, and x_{dj}=[1 v_{jc} $\dot{v}_{jc}$ $\ddot{v}_{jc}$ θ_j]^T. Examining the definition of the NN input, x_{dj}, reveals $\dot{v}_{jc}$ and $\ddot{v}_{jc}$ are necessary; however, recalling v_{jc} in (10) is a function of the leader's velocity reveals $\dot{v}_{jc} = f_j(\dot{v}_i, \dot{\omega}_i, v_i, \omega_i, e_j, \dot{e}_j)$ where $f_j(\bullet)$ is the function describing $\dot{v}_{jc}$. The leader *i*'s dynamics written in the form of (3) can be rewritten as $\dot{\overline{v}}_i = \overline{M}_i^{-1}(\overline{\tau}_i - \overline{F}_i(\overline{v}_i) - \overline{V}_{mi}\overline{v}_i)$, and substituting $\dot{\overline{v}}_i$ and (9) into $f_j(\bullet)$ results in the kinematic error dynamics of follower *j* and the dynamics of leader *i* to become apart of $\dot{v}_{jc}$ as

$$\dot{v}_{jc} = f_j(\overline{v}_i, \theta_i, \tau_i, \overline{v}_j, \theta_j, e_j). \tag{17}$$

It is not difficult to observe that $\ddot{\overline{v}}_i$, v_{jc}, and $\dot{v}_{jc}$ are also smooth functions since the leader and follower robots' dynamics are sufficiently smooth. As a

consequence, $\ddot{v}_{jc}$ can be approximated with relatively small error by the standard second order backwards difference equation for a small sample period Δt as

$$\ddot{\hat{v}}_{jc} = v_{jc}(t) - 2v_{jc}(t-\Delta t) + v_{jc}(t-2\Delta t) \tag{18}$$

Using (18) and forming $\dot{v}_{jc}$ under the assumption that $\dot{v}_i = 0$ as well as including the terms $\bar{v}_i, \theta_i,$ and τ_i of the function defined in (17), the estimated input to the NN $\hat{x}_{dj}$ takes the form of $\hat{x}_{dj} = [1 \; v_{jc}^T \; \dot{v}_{jc}^T\big|_{\dot{v}_i=0} \; \ddot{\hat{v}}_{jc}^T \; \theta_j \; \bar{v}_i \; \tau_i^T \; \theta_i \; e_j^T]^T$ so that the dynamics of the leader *i* can be estimated by the NN, and the terms of $\dot{v}_{jc}$ omitted by assuming $\dot{v}_i = 0$ can be accounted for.

Remark 3: In the formation of estimated NN input $\hat{x}_{dj}$, the terms $\bar{v}_i$, τ_i^T, θ_i are considered available via a wireless communication link which is a standard assumption; see [13].

The NN approximation of $\dot{f}_{d_j}$ is now defined as

$$\dot{\hat{f}}_{d_j} = \hat{W}_j^T \sigma(V_j^T \hat{x}_{dj}) \tag{19}$$

where $\hat{W}_j^T$ is the NN estimate of the ideal weight matrix W_j^T, and the control torque is now defined similarly [19] to be

$$\bar{\tau}_j = \hat{f}_{d_j} + \mu_j \tag{20}$$

where μ_j is the RISE feedback term defined similarly to [18] and [19] as

$$\mu_j = (k_{js}+1)e_{jc}(t) - (k_{js}+1)e_{jc}(0) + \int_0^t [(k_{js}+1)\alpha_j(s)e_{jc}(s) + (\beta_{j1}(s)+\beta_{j2})\text{sgn}(e_{jc}(s))]ds \tag{21}$$

such that $\dot{\mu}_j = (k_{js}+1)r_j + (\beta_{j1}(t)+\beta_{j2})\,\text{sgn}(e_{jc})$ with $\beta_{j1}(t)$ a positive, time varying gain real function, k_{js} and β_{j2} positive real constants, and sgn(•) the signum function.

***Remark 4**:* The projection algorithm is not used in this work to tune the NN weights as in [19], and as a result, the constant gains of [19] become time varying. Here $\beta_{j1}(t)$ and $\alpha_j(t)$ are time varying functions to facilitate in defining the upper bounds necessary for the RISE aspects of the NN/RISE controller which will be discussed in the proceeding development and in the Appendix in comparison to [18-19]. Further, the constant term β_{j2} is not same as constant term β_2 from [18] and [19] and is included here to aid in the forthcoming stability analysis.

Next, substituting the derivative of (20), as well as adding and subtracting e_{jc} and $W_j^T\hat{\sigma}(V_j^T\hat{x}_{dj})$ into (16) yields

$$\overline{M}_j\dot{r}_j = -\frac{1}{2}\dot{\overline{M}}_j r_j + \widetilde{N}_j + N_{Bj1} + N_{Bj2} - e_{jc} - (k_{js}+1)r_j - (\beta_{j1}(t)+\beta_{j2})\,\text{sgn}(e_{jc}) \tag{22}$$

where

$$\widetilde{N}_j = -\frac{1}{2}\dot{\overline{M}}_j r_j + \dot{T}_j + e_{jc} \tag{23}$$

$$N_{Bj1} = \varepsilon_j + \dot{\tilde{\tau}}_{d_j} + W_j^T\widetilde{\sigma}_j, \qquad N_{Bj2} = \widetilde{W}_j^T\sigma(V_j^T\hat{x}_{dj}) = \widetilde{W}_j^T\hat{\sigma}_j \tag{24}$$

and $\widetilde{W}_j = W_j - \hat{W}_j$, $\widetilde{\sigma}_j = \sigma(V_j^T x_{dj}) - \sigma(V_j^T\hat{x}_{dj})$. An upper bound for $\widetilde{N}_j$ can be obtained using the Mean Value Theorem as [18] and [19]

$$\left\|\widetilde{N}_j\right\| \le \rho(\left\|z_j\right\|)\left\|z_j\right\| \tag{25}$$

where $z_j = [e_{jc}^T \quad r_j^T]^T$ and $\rho(\left\|z_j\right\|)$ is a positive, globally invertible, non-decreasing function.

Lemma 1: The expressions in (24) and their derivatives are upper bounded according to:

$$\left\| N_{Bj1} \right\| \le \varepsilon_N + d'_M + 2W_M \sqrt{N_h} \equiv \varsigma_{j1} \tag{26}$$

$$\left\| \dot{N}_{Bj1} \right\| \le \varepsilon'_N + d'_M + \left(\sqrt{N_h} + N_h \right)\left(C_1 + C_2 \left\| e_j \right\| \right) \equiv \varsigma'_{j1}(t) \tag{27}$$

$$\left\| N_{Bj2} \right\| \le \left(W_M + \left\| \hat{W}_j \right\|_F \right)\sqrt{N_h} \equiv \varsigma_{j2}(t) \tag{28}$$

$$\left\| \dot{N}_{Bj2} \right\| \le C_3 \left\| e_{jc} \right\| + \left(\sqrt{N_h} + N_h \right)\left(W_M + \left\| \hat{W}_j \right\|_F \right) c_2(t) \equiv \varsigma'_{j2}(t) \tag{29}$$

where C_1, C_2, C_3 are known positive constants and $c_2(t)$ is a positive time-varying function based on $\left\| \dot{\hat{x}}_{dj} \right\|$.

Proof: See Appendix.

To aid in the forthcoming stability analysis and to facilitate time varying gains, we define an auxiliary function as

$$L_j = r_j^T \left(N_{Bj1} + N_{Bj2} - \beta_{j1} \operatorname{sgn}(e_{jc}) \right) - e_{jc}^T N_{Bj2} - \dot{\beta}_{j1} \left\| e_{jc} \right\| - \dot{e}_{jc}^T \beta_{j2} \operatorname{sgn}(e_{jc}) - \alpha_{jo} \beta_{j2} \left\| e_{jc} \right\|.$$

Lemma 2: Given the auxiliary function L_j, let $\beta_{j1}(t)$ and β_{j2} be chosen according to

$$\beta_{j1}(t) \ge K_{j\beta} + K_{jW} \left\| \hat{W}_j \right\|_F + K_{je} \left\| e_j \right\| + K_{jWe} \left\| e_j \right\| \left\| \hat{W}_j \right\|_F + K_{jec} \left\| e_{jc} \right\|, \qquad \beta_{j2} > 0 \tag{30}$$

with $K_{j\beta}$, K_{jW}, K_{je}, K_{jWe}, K_{jec} known positive constants, then

$$\int_0^t L_j(s) ds \le \gamma_j$$

where $\gamma_j = \left\| e_{jc}(0) \right\| \left(\beta_{j1}(0) + \beta_{j2} \right) - e_{jc}^T(0) N_{Bj3}(0) \ge 0$ with $N_{Bj3} = N_{Bj1} + N_{Bj2}$.

Proof: See Appendix.

Before proceeding, it is important to note that $r_j = 0,\ e_{jc} = 0,$ and $\tilde{W}_j^T = 0$ are equilibrium points of (22) in the absence of disturbances and NN functional reconstruction error ($N_{Bj1} = 0$). Proof of this claim is straight forward through examination of (13) and (22).

Theorem 1*: (Follower Dynamic Control)* Given the nonholonomic robot system consisting of (2) and (3) along with the leader follower criterion of (1), let a smooth velocity control input $v_{jc}(t)$ for follower *j* be given by (10), and the torque control for follower *j* given by (20) be applied to (3). Let the NN weight tuning law be given as

$$\dot{\hat{W}}_j = F_j \hat{\sigma}_j e_{jc}^T \tag{31}$$

where $F_j = F_j^T > 0$ is a design parameter. Then there exists a vector of positive constants $K_j = [k_{j1}\ \ k_{j2}\ \ k_{j3}]^T$, positive constants $k_{js}, \beta_{j2}, \alpha_{j0}$, and positive time varying functions $\beta_{j1}(t), \alpha_j(t)$, such that the position, orientation, and velocity tracking errors e_j and e_{jc} are asymptotically stable, and the neural network weight estimate errors $\tilde{W}_j$ are bounded for follower j provided that $\beta_{j1}(t)$ and β_{j2} are selected as in (30).

Proof: See Appendix.

D. Leader Control Structure

In every formation, there is a formation leader *i* whose kinematics and dynamics are defined similarly to (2) and (3), respectively. From [23], the leader tracks a virtual reference robot, and the tracking error for the leader and its derivative are found to be

$$e_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_i \\ y_r - y_i \\ \theta_r - \theta_i \end{bmatrix} \text{ and } \dot{e}_i = \begin{bmatrix} \dot{e}_{i1} \\ \dot{e}_{i2} \\ \dot{e}_{i3} \end{bmatrix} = \begin{bmatrix} -v_i + v_{ir}\cos e_{i3} + \omega_i e_{i2} \\ -\omega_i e_{i1} + v_{ir}\sin e_{i3} \\ \omega_{ir} - \omega_i \end{bmatrix} \tag{32}$$

where x_{ir}, y_{ir}, θ_{ir} v_{ir} and ω_{ir} are the states of a virtual reference robot for leader i defined as in (4). In this work, the virtual leader's velocity $\overline{v}_{ir}$ is defined by a time varying function that is twice differentiable. The leader's control velocity $v_{ic}(t)$ is then defined similarly to [23] as

$$v_{ic}\begin{bmatrix} v_{ic1} \\ v_{ic2} \end{bmatrix} = \begin{bmatrix} v_{ir}\cos e_{i3} + k_{i1}e_{i1} \\ \omega_{ir} + k_{i2}v_{ir}e_{i2} + k_{i3}k_{i2}\sin e_{i3} \end{bmatrix} \tag{33}$$

where $K_i = [k_{i1} \;\; k_{i2} \;\; k_{i3}]^T$ is a vector of positive constants, and the third term of v_{ic2} in (33) has been altered from [23] to facilitate in the stability analysis to come. To construct the dynamical NN/RISE controller for the leader i, define the velocity tracking and filtered tracking errors as

$$e_{ic} = v_{ic} - \overline{v}_i, \qquad r_i = \dot{e}_{ic} + \alpha_i(t)e_{ic} \tag{34}$$

Using similar steps and justifications used to form (14) for follower j, construct the error system for leader i to be $\overline{M}_i r_i = f_{d_i} + T_i + \overline{\tau}_{d_i} - \overline{\tau}_i$ where f_{d_i} and T_i are defined similarly to (15). The control torque, $\overline{\tau}_i$, for leader i can be defined similarly to follower j's as

$$\overline{\tau}_i = \hat{f}_{d_i} + \mu_i \tag{35}$$

where $\hat{f}_{d_i}$ is the estimate of f_{d_i}, μ_i is the RISE feedback term defined similarly the follower's in (19)-(21). The NN input vector for leader i is defined as $\hat{x}_{di} = [1 \;\; v_{ic}^T \;\; \dot{v}_{ic}^T \;\; \ddot{v}_{ic}^T\big|_{\dot{v}_i=0} \;\; \theta_i \;\; v_i \;\; \omega_i \;\; \tau_i^T]^T$ where the term $\dot{v}_{ic}^T$ is available while the term $\ddot{v}_{ic}^T$ is not due to its dependence on $\dot{\overline{v}}_i$ which is not known. As a result $\dot{v}_{ic}^T$ is calculated

assuming $\dot{\bar{v}}_i = 0$, and including the terms $\bar{v}_i, \theta_i, \tau_i^T$ in $\hat{x}_{di}$ so that the unknown dynamics can be accounted for by the NN similarly to the treatment of (17).

Using the same steps and justifications used to form (22), the closed loop error system for the for the lead robot i can be formed as

$$\overline{M}_i \dot{r}_i = -\frac{1}{2}\dot{\overline{M}}_i r_i + \widetilde{N}_i + N_{Bi1} + N_{Bi2} - e_{ic} - (k_{is}+1) r_i - \left(\beta_{i1}(t) + \beta_{i2}\right)\text{sgn}(e_{ic}) \tag{36}$$

where k_{is} is a positive control gain parameter, and $\widetilde{N}_i$, N_{Bi1} and N_{Bi2} are defined similarly to (23), and (24), respectively, and are bounded similarly to the bounds defined in (25)-(29). Further, $r_i = 0,\ e_{ic} = 0,$ and $\widetilde{W}_i^T = 0$ are equilibrium points of (36) in the absence of disturbances and NN functional reconstruction error ($N_{Bi1} = 0$).

Theorem 2*: (Leader Stability)* Let the smooth velocity control input for leader *i* be given by (33) and let the toque control input defined by (35) be applied to the leader robot *i,* defined similarly to (3). Let the NN tuning law for leader *i* be defined similarly to (31). Then there exists a vector of positive constants $K_i = [k_{i1}\ k_{i2}\ k_{i3}]^T$, positive constants $k_{is}, \beta_{i2},\ \alpha_{i0}$, and positive time varying functions $\beta_{i1}(t), \alpha_i(t)$, such that the position, orientation, and velocity tracking errors e_i and e_{ic} are asymptotically stable, and the NN weight estimate errors $\widetilde{W}_i$ are bounded for follower *j* provided that $\beta_{i1}(t)$ and β_{i2} are selected similarly to (30).

Proof : See Appendix.

Next, the stability of the entire formation is demonstrated in the following theorem.

E. Formation Stability

Theorem 3: *(Formation Stability)* Consider a formation of *N+1* robots consisting a leader *i* and *N* followers, and let the hypotheses of *Theorems 1* and *2* hold. Then the formation error $e_{ij} = [e_i^T \quad e_{ic}^T \quad e_j^T \quad e_{jc}^T]^T$ where $e_{ij} \in \Re^{(n+\rho)(1+N)}$ represents the augmented position, orientation and velocity tracking error systems for the leader *i* and *N* followers, respectively, is asymptotically stable, and the NN weight estimation errors $\widetilde{W}_i^T, \widetilde{W}_j^T$, $j = 1,2,...N$ for the leader *i* and *N* followers, respectively, are bounded.

Proof: See Appendix.

Remark 5: The stability of the entire formation for the case when follower *j* becomes a leader to follower *j+1* follows directly from *Theorem 1* and selecting a Lyapunov candidate to be the sum of the Lyapunov candidates for follower *j* and follower *j+1*, respectively. In this case, follower *j* becomes the reference for follower *j+1,* and thus the dynamics of follower *j* must be considered by follower *j+1*. Since the dynamics of follower *j* incorporates the dynamics of leader *i,* follower *j+1* inherently brings in the dynamics of leader *i* by considering the dynamics of follower *j*.

A general formation controller structure is shown in Fig. 2 which includes the controller structures for the leader *i* and multiple followers. Additionally, communication between the robots is indicated. In the figure, leader *i* communicates its velocity, orientation, and control torque to follower *j*, and follower *j* communicates its velocity, orientation, and control torque to follower *j+2,* but it is not necessary for follower *j* to relay the states of leader *i* to follower *j+2*. Also note that in a formation of robots, each robot may have more than one follower.

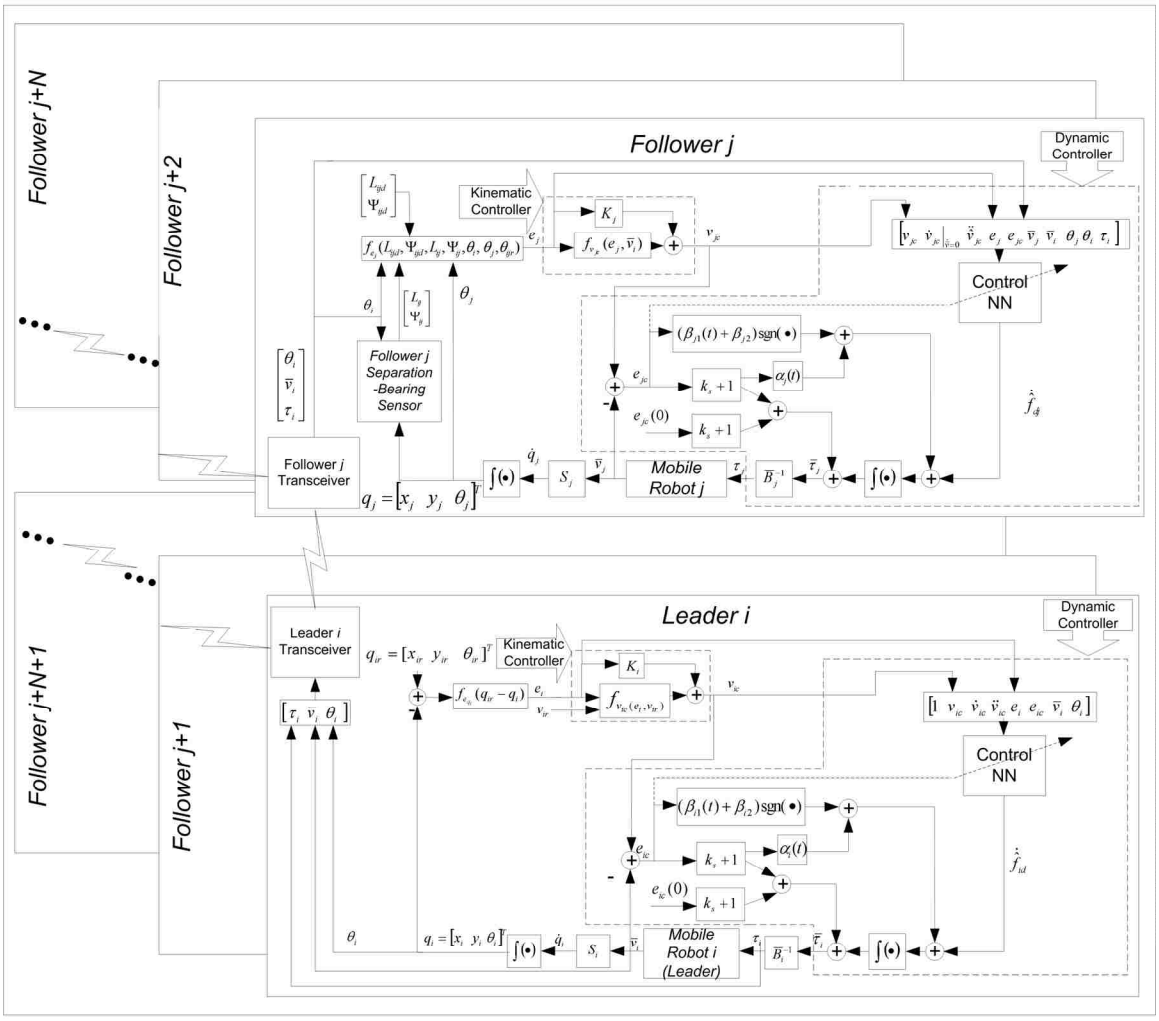


Fig. 2. General formation controller structure.

III. Leader-Follower Obstacle Avoidance

Next, a simple but effective obstacle avoidance scheme is proposed that will allow follower *j* to track its leader while simultaneously avoiding obstacles. To accomplish this, the desired separation and bearing are no longer considered to be constants but are considered to be time varying, and through the incorporation of RISE feedback, each follower in the formation asymptotically tracks the new reference position while avoiding obstacles. In this section, the time varying desired separation and bearing

will be denoted as $L_{ijd}(t)$ and $\Psi_{ijd}(t)$ while the constant desired separation and bearing will be written as L_{ijd} and Ψ_{ijd}. Furthermore, the distance from the center of follower j to an obstacle, s_j, and relative angle of the obstacle, θ_{js}, are considered measurable while the velocity vector, $\bar{v}_o = [v_o \ \omega_o]^T$, and orientation, θ_o, of the obstacle are unavailable. It is standard to assume that the formation leader i utilizes a path planning scheme such that by tracking the virtual reference cart described in [23], the lead robot i navigates around any encountered obstacles.

To begin, consider the configuration shown in Fig. 3 where it is desirable that the follower robot j maintains a safe distance, s_d, from the closest obstacle. When the nearest edge of an obstacle is detected at an angle θ_{js} and distance s_j relative to center of follower j such that $s_j < s_d$, the desired separation and bearing, $L_{ijd}(t)$ and $\Psi_{ijd}(t)$, are modified to ensure the follower is steered away from obstacle by

$$L_{ijd}(t) = L_{ijd} - \frac{1}{2}K_L\left(\frac{1}{s_j} - \frac{1}{s_d}\right)^2 \operatorname{sgn}(\theta_{js}\Psi_{ijd}), \quad \Psi_{ijd}(t) = \Psi_{ijd} + \frac{1}{2}K_\Psi\left(\frac{1}{s_j} - \frac{1}{s_d}\right)^2 \xi_j \qquad (37)$$

where $\xi_j = \operatorname{sgn}(\Psi_{ijd})\operatorname{sgn}(\theta_{js}\Psi_{ijd})$, with *sgn* is the signum function and K_L and K_ψ are positive design constants. Examining (37), one can see that the shifts introduced to the desired separation and bearing are similar to repulsive potential functions commonly used in robotic path planning [22]. Here we use the potential like function to push the desired set point of the follower robot j away from the encountered obstacle thus steering the robot around the obstruction. Incorporation of $\operatorname{sgn}(\theta_{js}\Psi_{ijd})$ allows obstacles to be avoided on the left or the right, depending on where the follower is located in the

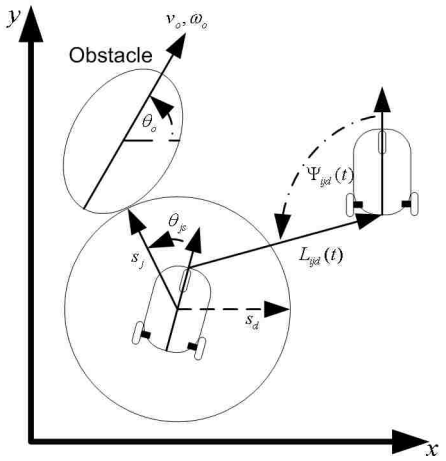


Fig. 3. Obstacle avoidance.

formation and where the obstacle is located relative to the follower. This term also allows collisions to be avoided within the formation by considered neighboring robots as obstacles.

With the introduction of obstacle avoidance schemes, the orientation of the follower *j* will vary from its reference orientation as a result of avoiding an obstacle that was in the path of the follower *j* but not its leader. Therefore, while avoiding an obstacle, it is logical for follower *j* to track a reference point, but no specific orientation with respect to its leader. Thus, consider the formation tracking control error system presented in (7), but rewritten to include only the normal and tangential position error components as

$$e_{jo} = \begin{bmatrix} e_{jo1} \\ e_{jo2} \end{bmatrix} = \begin{bmatrix} L_{ijd}(t)\cos(\Psi_{ijd}(t)+\theta_{ij}) - L_{ij}\cos(\Psi_{ij}+\theta_{ij}) \\ L_{ijd}(t)\sin(\Psi_{ijd}(t)+\theta_{ij}) - L_{ij}\sin(\Psi_{ij}+\theta_{ij}) \end{bmatrix}. \tag{38}$$

The dynamics of (38) can be found in a similar manner as that of (9), and written as

$$\begin{bmatrix} \dot{e}_{jo1} \\ \dot{e}_{jo2} \end{bmatrix} = \begin{bmatrix} \dot{\bar{e}}_{jo1} - v_j + v_i\cos\theta_{ij} + \omega_j e_{jo2} - \omega_i L_{ijd}(t)\sin(\Psi_{ijd}(t)+\theta_{ij}) \\ \dot{\bar{e}}_{jo2} - \omega_j e_{jo1} + v_i\sin\theta_{ij} - d_j\omega_j + \omega_i L_{ijd}(t)\cos(\Psi_{ijd}(t)+\theta_{ij}) \end{bmatrix} \tag{39}$$

where

$$\begin{aligned}\dot{\bar{e}}_{jo1} &= \dot{L}_{ijd}(t)\cos(\Psi_{ijd}(t)+\theta_{ij}) - \dot{\Psi}_{ijd}(t)L_{ijd}(t)\sin(\Psi_{ijd}(t)+\theta_{ij}) \\ \dot{\bar{e}}_{jo2} &= \dot{L}_{ijd}(t)\sin(\Psi_{ijd}(t)+\theta_{ij}) + \dot{\Psi}_{ijd}(t)L_{ijd}(t)\cos(\Psi_{ijd}(t)+\theta_{ij})\end{aligned}. \tag{40}$$

The dynamics of the desired separation and bearing, $L_{ijd}(t)$ and $\Psi_{ijd}(t)$ in (37), respectively, are necessary in the calculation of (40), and therefore, the derivative $\dot{s}_j$ is also required. The measured distance s_j can be written in terms of the x and y components of s_j as $s_j^2 = s_{jx}^2 + s_{jy}^2$ where $s_{jx} = x_j - x_o$ and $s_{jy} = y_j - y_o$ and x_o and y_o are the coordinates of the obstacle. Note that the obstacle is not necessarily stationary, and therefore assume that the obstacle can be described using the kinematic model as $\dot{x}_o = v_o \cos\theta_o$ and $\dot{y}_o = v_o \sin\theta_o$. Using this information along with (2), it is evident that the derivative of s_j is a function of the velocity $\bar{v}_j$ and orientation θ_j of follower j as well as the velocity $\bar{v}_o$ and orientation, θ_o, of the encountered obstacle. Since the velocity $\bar{v}_o$ and orientation θ_o of the obstacle are not available to follower *j*, $\dot{s}_j$ must be estimated, and as a result, $\dot{L}_{ijd}(t)$ and $\dot{\Psi}_{ijd}(t)$ must also be estimated. Assuming that s_j is a smooth function, define the aforementioned estimates to be

$$\hat{\dot{L}}_{ijd}(t) = \text{sgn}(\theta_{js}\Psi_{ijd})K_L\left(\frac{1}{s_j} - \frac{1}{s_d}\right)\frac{1}{s_j^2}\dot{\hat{s}}_j, \quad \hat{\dot{\Psi}}_{ijd}(t) = -\xi_j K_\psi\left(\frac{1}{s_j} - \frac{1}{s_d}\right)\frac{1}{s_j^2}\dot{\hat{s}}_j, \tag{41}$$

and $\dot{\hat{s}}_j = s_j(t) - s_j(t-\Delta t)$ is the estimate of $\dot{s}_j$ for an arbitrarily small time interval, Δt.

In order to show that the obstacle avoidance method is asymptotically stable in the presence of uncertainties, the RISE method described in the previous section will

again be utilized. To use the RISE method, we begin by defining a filtered tracking error as

$$\vartheta_j = \dot{e}_{jo} + \kappa e_{jo} \tag{42}$$

where κ is a positive, real design constant. Utilizing the error dynamics (39) and (40), the filtered tracking error (42) can be rewritten as

$$\vartheta_j = J_j + H_j - E_j v_j + \kappa e_{jo} \tag{43}$$

where

$$J_j = \left(\frac{1}{s_j} - \frac{1}{s_d}\right)\frac{1}{s_j^2}\dot{s}_j \begin{bmatrix} \operatorname{sgn}(\theta_{js}\Psi_{ijd})K_L \cos(\Psi_{ijd}(t)+\theta_{ij}) + \xi_j K_\Psi L_{ijd}(t)\sin(\Psi_{ijd}(t)+\theta_{ij}) \\ \operatorname{sgn}(\theta_{js}\Psi_{ijd})K_L \sin(\Psi_{ijd}(t)+\theta_{ij}) - \xi_j K_\Psi L_{ijd}(t)\cos(\Psi_{ijd}(t)+\theta_{ij}) \end{bmatrix} \tag{44}$$

$$H_j = \begin{bmatrix} v_i \cos(\theta_{ij}) - \omega_i L_{ijd}(t)\sin(\Psi_{ijd}(t)+\theta_{ij}) + e_{jo2}\omega_j \\ v_i \sin(\theta_{ij}) + \omega_i L_{ijd}(t)\cos(\Psi_{ijd}(t)+\theta_{ij}) - e_{jo1}\omega_j \end{bmatrix}, \quad E_j = \begin{bmatrix} 1 & 0 \\ 0 & d_j \end{bmatrix} \tag{45}$$

and $\dot{s}_j$ is the real dynamics of s_j. To stabilize the filtered tracking error dynamics in the presence of an obstacle, the following velocity control input for follower robot j is proposed

$$v_{jco} = E_j^{-1}\left(H_j + \hat{J}_j + (G+\kappa)e_{jo} + \int_0^t \left(G\kappa e_{jo} + \beta_{jo}\operatorname{sgn}(e_{jo})\right)ds\right) \in \Re^2 \tag{46}$$

where $\hat{J}_j$ is the estimate of J_j as a result of using $\hat{\dot{s}}_j$, and G, β_{jo} are positive, real design constants. For analysis purposes, we will assume $J_j = \hat{J}_j + \zeta_j$ where ζ_j is the error in estimation. Furthermore, we assume that the estimation error and its derivative are bounded by a positive real values ζ_M and ζ'_M, respectively, such that $\|\zeta_j\| \le \zeta_M$ and $\|\zeta'_j\| \le \zeta'_M$ for all time.

Defining the velocity tracking error e_{jco} identically to (11), substituting $\bar{v}_j = v_{jco} - e_{jco}$ into (43) and taking the derivative, the close loop kinematic filtered error dynamics can be written as

$$\dot{\vartheta}_j = -G\vartheta_j + \zeta_j - \beta_{jo}\,\mathrm{sgn}(e_{jo}) + E_j e_{jco}, \tag{47}$$

and when there is zero estimation error, ζ_j=0, the origin $\vartheta_j = 0$, $e_{jo} = 0$ and $e_{jco} = 0$ is an equilibrium point of $\dot{\vartheta}_j$. To aid in the stability analysis of the follower robot in the presence of obstacles, an auxiliary function is defined as $R_j(t) = \vartheta_j^T(\zeta_j - \beta_{jo}\,\mathrm{sgn}(e_{jo}))$.

Lemma 3: Given the auxiliary function $R_j(t)$, then

$$\int_0^t R_j(s)ds \le \beta_{jo}\left\|e_{jo}(0)\right\| - e_{jo}^T(0)\zeta_j(0)$$

provided β_{jo} is selected as

$$\beta_{jo} \ge \zeta_M + \frac{1}{\kappa}\zeta'_M \tag{48}$$

Proof: See Appendix.

Theorem 4*: (Follower Obstacle Avoidance)* Let the hypothesis of *Theorem 1* hold with (10) replaced by (46). Then, there exists positive constants G, β_{jo} K_L and K_ψ such that position and velocity tracking errors for the follower are asymptotically stable in the presence of obstacles provided β_{jo} is selected as (48).

Proof: See Appendix.

Remark 6: Since leader robot *i* does not track a physical robot, any existing asymptotically stable obstacle avoidance method can be utilized by the leader to ensure the stability of the entire formation in the presence of obstacles. The path planning

algorithm for the leader *i* is beyond the scope of this paper and therefore is not included here.

Remark 7: The stability of a formation of *N+1* robots consisting of a leader *i* and *N* followers in the presence of obstacles follows directly by combining the results of *Theorem 2* and *Theorem 4* for $j = 1, 2, 3, ...N$, respectively. Further, the stability of a formation in the presence of obstacles for the case when follower *j* becomes a leader to follower *j+1* follows directly from *Theorem 4* and combining the Lyapunov candidates for follower *j* and follower *j+1* into a single Lyapunov function.

Remark 8: The proposed obstacle avoidance scheme is observed to have potential limitations. Since the scheme only considers the closest obstruction, it is possible that in a highly cluttered environment there may be more than one obstacle within the robot's safety zone; one of which could potentially be another robot in the formation. In this case, the follower may exhibit an oscillatory behavior between multiple obstructions located within the safety zone which is not ideal; however, the goal of the obstacle avoidance scheme is still achieved in that collisions are avoided. In the event that two or more obstacles are located at the same distance from follower *j*, the obstacle which poses the greatest immediate threat of collision is considered. Future efforts will work to remove these limitations and the obstacle avoidance is not the focus of this effort.

Remark 9: The control velocity (46) can be applied for *any* obstacle avoidance scheme in which the desired separation and bearing are modified to steer the robot around the obstruction. The only required modified to the control velocity (46) is with respect to the vector J_j in (44) which contains the dynamics of $L_{ijd}(t)$ and $\Psi_{ijd}(t)$, respectively.

Remark 10: By the design of the obstacle avoidance scheme, the follower robot continues to track its leader while it navigates around an obstacle through the use of the time varying desired separation and bearing. As the robot navigates around the obstruction and the obstruction leaves the robot's safety zone, the time varying desired separation and bearing naturally return to the constant desired values. Thus, the robot itself returns to its location in the formation.

IV. Simulation Results

A formation of identical nonholonomic mobile robots is considered where the leader's trajectory is the desired formation trajectory and simulations are carried out in MATLAB under two scenarios: with and without obstacles. In the first scenario, the NN controller which renders *Uniformly Ultimately Bounded* (UUB) in [15] is considered, and then the NN/RISE controller which has been shown to be *asymptotically stable* (AS) in this paper is tested. The torque controller developed in [15] is similar the torque control of (20), but without the extra RISE terms added in (21) and takes the form of $\bar{\tau}_j = \hat{W}_j^T \sigma(\bar{x}_j) + (k_{js}+1)e_{jc} = \hat{f}_j + (k_{js}+1)e_{jc}$ where $\hat{f}_j$ is the NN estimate of an unknown function.

An additional difference between the torque control of this work and that of [15] is the fact that the NN estimates the derivative of an unknown function in this work. In both cases, unmodeled dynamics are introduced in the form of friction as $\overline{F}_j = [\mu_{j1} sign(v_j) + \mu_{j2} v_j, \quad \mu_{j3} sign(\omega_j) + \mu_{j4}\omega_j]^T$ where μ_{ji} are the coefficients of friction and summarized in Table I. Additionally, disturbance and sensor noise terms are added to the robot dynamics and state measurements, respectively. Disturbances are added to

TABLE I: Friction Coefficients

	L	F1	F2	F3	F4
μ_1	0.5	0.05	0.01	0.015	0.025
μ_2	0.75	0.75	0.65	0.15	0.50
μ_3	0.25	0.025	0.025	0.05	0.015
μ_4	0.03	0.30	0.20	0.25	0.03

the robot dynamics and are generated from a normal distribution with mean zero, variance one and standard deviation one. The magnitude of the disturbances is taken as two.

Sensor noise is also generated from an identical normal distribution with magnitudes of $\eta_v = 0.25, \eta_L = 0.1, \eta_\Psi = 0.05$ where η_v, η_L, η_Ψ for the velocity, separation, and bearing measurements, respectively. In the second scenario, obstacles are added in the path of the follower robots and the obstacle avoidance scheme of *Theorem 4* is demonstrated, and both a static and dynamic obstacle environment is considered.

In the simulations, followers 1 and 2 track the leader while followers 3 and 4 track followers 1 and 2, respectively, as depicted in Fig. 4. The following parameters are considered for the leader and its followers: $m = 5kg$, $I = 3kg^2$, $R = .175m$, $r = 0.08m$, and $d = 0.4m$. The control gains for the leader were selected as $k_{i1} = 10$, $k_{i2} = 5$, $k_{i3} = 4$, $K_{is} = 35$, and for each follower, gains were selected as $k_{j1} = 5$, $k_{j2} = 5$, $k_{j3} = 16.5$ and $K_{js} = 35$, respectively. Five hidden layer neurons are considered in the NN for the leader and each follower such that $N_h = 5$, and the NN parameters for both the leader and each follower were selected as, $F_j = F_i = 10$. In addition, the RISE terms are selected according to (30) with $K_W = 8$, $K_e = 15$, $K_{We} = 8$, $K_\beta = 15$, $K_{ec} = 10$,

and the filtered tracking error gain $\alpha(t)$ is selected as $\alpha(t)=5+\frac{1}{\beta_2}\left(K_W\sqrt{N_h}F\|e_c\|+K_{ec}(10+2.5\|e\|)+K_e(8+2.5\|e\|)+K_{We}\left((8+2.5\|e\|)\|\hat{W}\|_F+F\sqrt{N_h}\|e_c\|\|e\|\right)\right)$ with $\beta_2 = 20$.

Remark 11: In the proceeding analysis, $L, F1, F2, F3,$ and $F4$ will be used to denote the *leader, follower 1*, *follower 2, follower 3,* and *follower 4,* respectively.

A. Scenario I: Obstacle Free Environment

In this scenario, the leader follows a virtual robot traveling at a constant linear velocity of $v_{ir} = 5\ m/s$ with a time varying reference angular velocity, and the NN controller of our previous work and the NN/RISE controller are tested. The formation is selected to be a wedge shape as in Fig. 4 where each follower is to track its leader at a desired separation of $L_{ijd} = 2$ meters with a bearing of $\Psi_{ijd} = \pm 120°$ depending on the follower's location, and for illustrative purposes, a fifth follower has been added to track follower 2.

Fig. 5 displays the formation trajectories for both controllers as the formation performs a sharp turn while navigating around a barrier. Examining the trajectories reveals that both controllers successfully perform the maneuver; however upon closer

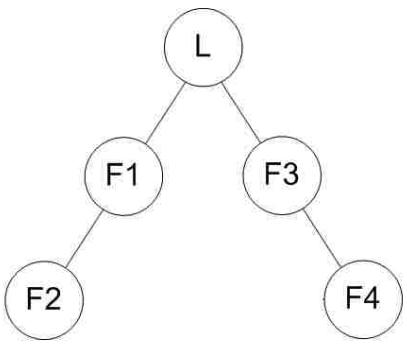


Fig. 4. Formation structure.

examination, formation errors are seen propagating throughout the formation for the case when the NN controller is used. The evidence of the error propagation is best seen in the trajectories of the robots on the inside of the turn which have been enlarged to facilitate viewing. Examining the trajectories in the bottom right corner of Fig. 5, small errors can been seen in the path of follower 2 while larger errors are seen in the path of follower 5 for the case when the NN controller is applied. On the other hand, evidence of this error propagation is not present in the paths of either robot when the NN/RISE controller is applied. Thus, the theoretical conjectures of *Theorem 1* are verified in that the formation achieves asymptotic tracking in the presence of bounded disturbances.

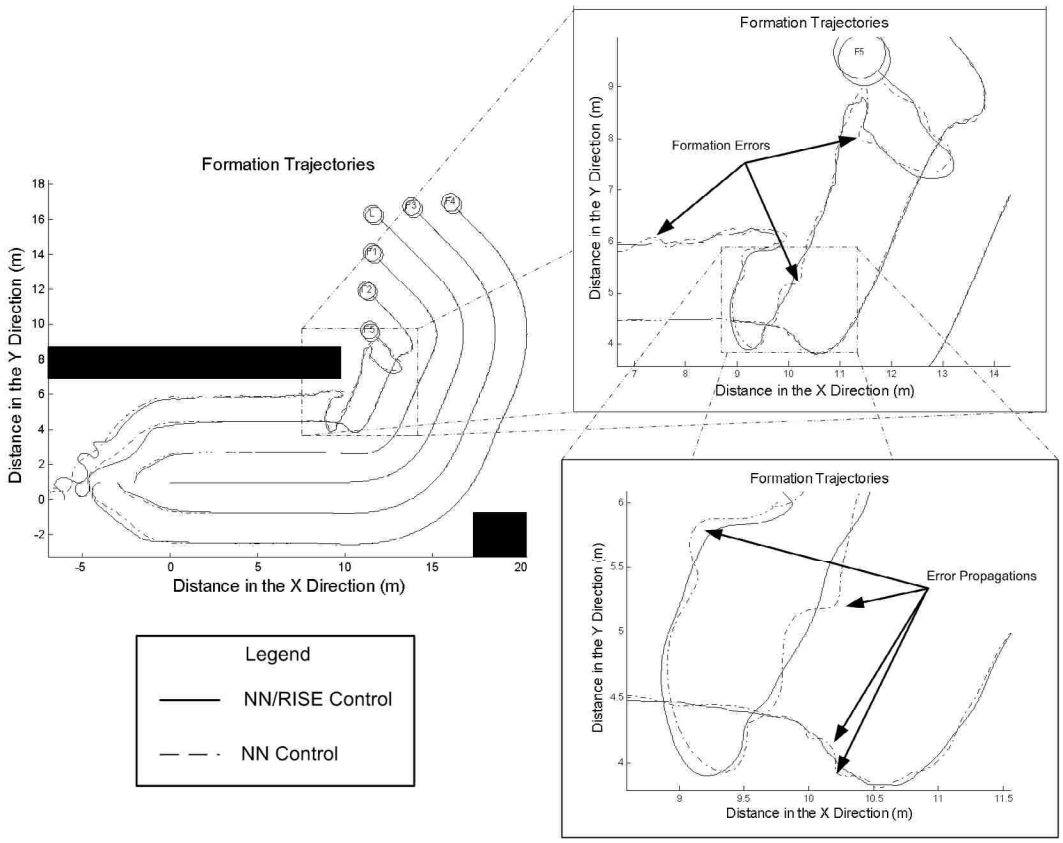


Fig. 5: Formation trajectories.

Fig. 6 displays the steady-state formation errors of each follower in the formation. The improved performance of the NN/RISE controller over the NN controller is again observed, especially in the formation errors for follower 1, 2, and 3, respectively, and the strength of AS over UUB is revealed. The average formation errors for each follower are shown in Table II where it is observed that the average error was reduced for each follower when the NN/RISE controller was utilized. In some cases, as with follower 1, errors were reduced by 50%, while marginal error reduction was observed for follower 5. Reducing the formation errors for the robots near the front of the formation helps prevent formation errors from propagating through the formation, which was observed for the case with the NN controller was applied.

Remark 12: The reference position of each robot in the formation is defined with respect to its respective leader, not the leader of the entire formation. As a result, the movement of each robot propagates to its followers, a phenomenon observed in Fig. 5

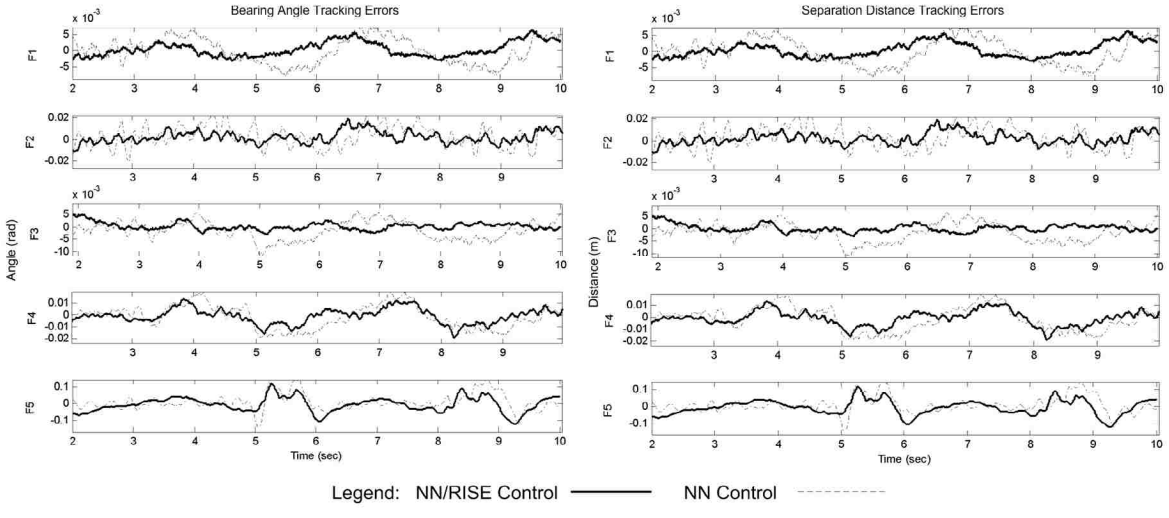


Fig. 6. Formation errors.

TABLE II. Average Steady State Formation Errors.

Average Errors	NN/RISE Controller					NN Controller				
	F1	F2	F3	F4	F5	F1	F2	F3	F4	F5
Separation error (m)	0.0048	0.0117	0.0029	0.0066	0.0442	0.0094	0.0179	0.0048	0.077	0.0462
Bearing error (rad)	0.0030	0.0069	0.0037	0.0049	0.0449	0.0036	0.0089	0.0041	0.0093	0.0467

with followers 2 and 5 for the case when the NN control was applied. Additionally, it was observed in Table II that formation errors for follower 5 were marginally reduced when the NN/RISE controller was applied; however, although the reduction in the error was small, the improved performance in the NN/RISE controller over the standard NN controller is still significant since the oscillatory movements observed for the NN controller in Fig. 5 are not observed for the case when the NN/RISE control was applied.

B. Scenario II: Obstacle Ridden Environment

Now, consider stationary and moving obstacles for the wedge formation along with the controller gains outlined above along with $K_L = .9$, $K_\psi = 1.5$, $\beta_o = 0.5$, and $\kappa = 2$. The robots are initialized so that they must avoid one another while attempting to reach their desired location in the formation.

Fig. 7 depicts the formation trajectories in the presence of both stationary and moving obstacles, and examining this figure, it is evident that the robots are able avoid collisions with their neighbors and maneuver around the encountered obstacles while simultaneously tracking their leaders. Because the followers on the outside of the formation track the robots in the inner formation, the movements of the robots in the interior of the formation propagate to followers on the exterior of the formation. Thus,

when a robot on the interior of the formation performs an obstacle avoidance maneuver, their movements are mimicked by their followers, which is evident in Fig. 7. As previously identified, the obstacle avoidance scheme poses potential short comings in heavily cluttered environment. However, as illustrated in Fig. 7, the obstacle avoidance scheme can be effective in undemanding environments as well as ensure collisions between robots in the formation do not occur.

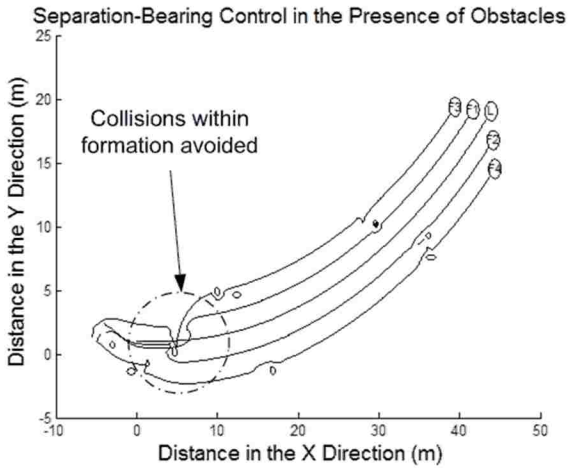


Fig. 7. Formation obstacle avoidance.

V. CONCLUSIONS

In the absence of obstacles, an asymptotically stable NN tracking controller for leader-follower based formation control was presented that considers the dynamics of the leader and the followers using backstepping with RISE feedback. The feedback control scheme is valid even when the dynamics of the followers and their leader are unknown since the NN learns them all online. Numerical results were presented and the asymptotic stability of the system was verified. Simulation results verify the theoretical

conjecture and reveal the strength of asymptotic stability over the common result of most NN literature, UUB. The asymptotic stability of the formation in the presence of obstacles was also demonstrated by applying the RISE method to a leader-follower obstacle avoidance scheme. The control was shown to be effective in both a static and dynamic obstacle environment, and numerical results were presented. Further, by treating robots in the formation as obstacles, collisions within the formation were guaranteed not to occur. The stability of the system was verified, and the simulation results verified the theoretical conjecture.

Future efforts will address a more comprehensive obstacle avoidance scheme for leader-follower formation control. This work will focus on alleviating the previously observed limitations of the current obstacle avoidance scheme so that multiple objects and more complex environments can be navigated while completing the leader-follower formation control objective.

REFERENCES

[1] Y. Q. Chen and Z. Wang, "Formation control: a review and a new consideration," *Proc. IEEE International* Conference *on Intelligent Robots and Systems,* pp. 3181-3186, August 2005.

[2] G. L. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis, "Vision-based localization of leader-follower formations," *Proc. IEEE European Control Conference on Decision and Control*, pp 635-640, December 2005.

[3] Jaydev P. Desai, Jim Ostrowski, and Vijay Kumar, "Controlling formations of multiple mobile robots," *Proc. IEEE International Conference on Robotics and Automation*, pp. 2864-2869, May 1998.

[4] J. Shao, G. Xie, J. Yu, and L. Wang, "A tracking controller for motion coordination of multiple mobile robots," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp 783-788, August 2005.

[5] H. Hsu and A. Liu, "Multi-agent based formation control using a simple representation," *Proc. IEEE International Conference on Networking, Sensing & Control*, pp 276-282, March 2004.

[6] X. Li, J. Xiao, and Z. Cai, "Backstepping based multiple mobile robots formation control," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp 887-892, August 2005.

[7] S. Mastellone, D. Stipanović, C. Graunke, K. Intlekofer, and M. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: theory and experiments," The *International Journal of Robotics Research,* vol. 27, pp 107-126, January 2008.

[8] Y. Li and X. Chen, "Dynamic control of multi-robot formation," P*roc. IEEE International Conference on Mechatronics*, pp 352-357, July 2005.

[9] M. Breivik, M. Subbotin, and T. Fossen, "Guided formation control for wheeled mobile robots," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1-7, December 2006.

[10] Y. Liang and H. Lee, "Decentralized formation control and obstacle avoidance for multiple robots with nonholonomic constraints," *Proc. IEEE American Control Conference*, pp. 5596-5601, June 2006.

[11] H. Takahashi, H. Nishi, and K. Ohnishi, "Autonomous decentralized control for formation of multiple mobile robots considering ability of robot," *IEEE Trans. on Industrial Electronics*, vol. 51, pp. 1272-1279, December 2004.

[12] C. De La Cruz and R. Carelli, "Dynamic modeling and centralized formation control of mobile robots," *Proc. IEEE Conference on Industrial Electronics,* pp. 3880-3885, November 2006.

[13] K. D. Do, "Formation tracking control of unicycle-type mobile robots with limited sensing ranges," *IEEE Trans. On Control Systems Technology, vol. 16, pp. 527-538, May 2008.*

[14] T. Dierks and S. Jagannathan, "Control of nonholonomic mobile robot formations: backstepping kinematics into dynamics," *Proc. IEEE International Conference on Control Applications*, pp.94-99, Oct. 2007.

[15] T. Dierks, "Nonlinear control of nonholonomic mobile robot formations," MS Thesis, at the University of Missouri-Rolla, 2007. Available via the internet: (URL: http://scholarsmine.mst.edu)

[16] S. S. Ge and C. H. Fua, "Queues and artificial potential trenches for multirobot formations," *IEEE Trans. on Robotics,* vol.21, pp. 646-656, August 2005.

[17] P. Ogren and N.E. Leonard, "Obstacle avoidance in formation," *Proc. IEEE Conference on Robotics and Automation,* pp. 2492-2497 14-19 September 2003.

[18] B. Xian, M.S. de Queiroz, and D. M. Dawson, "A continuous control mechanism for uncertain nonlinear systems in optimal control," *Stabilization and Nonsmooth Analysis,* pp. 251-262, Heidelberg, Germany: Springer-Verlag, 2004.

[19] P. M. Patre, W. E. Dixon, K. Kaiser, and W. MacKunis, "Asymptotic tracking for uncertain dynamic systems via a multilayer NN feedforward and RISE feedback control structure," *IEEE American Control Conference*, pp. 5989-5994, July 2007.

[20] F.L. Lewis, S. Jagannathan, and A. Yesilderek, "*Neural network control of robot manipulators and nonlinear systems*," Taylor and Francis, London, UK, 1999.

[21] H. K. Khalil, *Nonlinear Systems*, 3rd edition, Prentice-Hall, NJ, 2002.

[22] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Intl. Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[23] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Transactions on Neural Networks*, vol. 8, pp. 589-600, July 1998.

APPENDIX

Remark A.1*:* To begin, certain bounds must be established, and for generality, the subscripts *i* and *j* will be not be used here. First, bounds on NN quantities will be frequently used as

$$\|W\|_F \le W_M, \quad \|\sigma\| \le \sqrt{N_h}, \quad \|\sigma(1-\sigma)\| \le \sqrt{N_h} + N_h, \quad \left\|\dot{\hat{W}}\right\|_F \le F_M \sqrt{N_h} \|e_c\| \tag{A.1}$$

where $\|e_c\|$ refers to the velocity tracking error, N_h is the constant number of hidden layer neurons, W_M is the upper bound of the ideal NN weights W, and $F_M = \|F\|_F$ is a constant. Next, bounds relating the physical robotic system are written as

$$|\theta| \le q_M, \quad \|[v \;\; \omega \;\; \dot{v} \;\; \dot{\omega} \;\; \ddot{v} \;\; \ddot{\omega}]\| \le V_M, \quad \left\|[\tau^T \;\; \dot{\tau}^T]\right\| \le \mathrm{T}_M \tag{A.2}$$

where q_M, V_M, T_M are known constants relating to the physical capabilities of the mobile robot. Additionally, bounds on the velocity control (10) and its derivatives can be established as

$$\left\| [v_c^T \quad \dot{v}_c^T \quad \ddot{v}_c^T \quad \dddot{v}_c^T] \right\| \leq C_4 + C_5 \|e\| \tag{A.3}$$

where $\|e\| = \|[e_1 \ e_2 \ e_3]\|$ refers to the position and orientation tracking errors with $C_i, i = 4,5$ computable constants dependant on (A.2) and the selection of the velocity control gains in (10). Since the backwards difference equation (18) is utilized to estimate the higher order derivatives of the control velocity (10), the following bound must also be established

$$\left\| [v_c^T \quad \dot{v}_c^T \quad \ddot{\hat{v}}_c^T \quad \dddot{\hat{v}}_c^T] \right\| \leq C_6 + C_7 \|e\| \tag{A.4}$$

with $C_i, i = 6,7$ computable constants. Now, the bounds on the derivative of the ideal NN input x_d as well as the derivative of the estimated NN input $\hat{x}_d$ are found to be

$$\|\dot{x}_d\| \leq C_8 + C_9 \|e\| \equiv c_1(t), \qquad \|\dot{\hat{x}}_d\| \leq C_{10} + C_{11} \|e\| \equiv c_2(t) \tag{A.5}$$

with $C_i, i = 8,9,10,11$ computable constants. Proof of (A.5) is straight forward using (A.2), (A.2), (A.3) and (A.4) along with using similar steps described in [20].

Lemma 1: Upper bounds for N_{B1} and N_{B2} in (24) as well as their derivatives can be defined as in (26), (27), (28), and (29).

Proof: Recalling $\|\varepsilon\| \leq \varepsilon_N$, $\left\| [\dot{\bar{\tau}}_d^T \ \ddot{\bar{\tau}}_d^T] \right\| \leq d'_M$ as well as observing (A.1) reveals (26). Next, differentiating N_{B1} reveals $\dot{N}_{B1} = \dot{\varepsilon} + \ddot{\bar{\tau}}_d + W^T \dot{\tilde{\sigma}}$. Then, recalling $\|\dot{\varepsilon}\| \leq \varepsilon'_N$ and again applying the bounds in (A.1) reveals $\|\dot{N}_{B1}\| \leq \varepsilon'_N + d'_M + W_M \|\dot{\tilde{\sigma}}\|$, and the bound in

(27) follows by observing $\tilde{\sigma} = \sigma - \hat{\sigma}$ and applying the chain rule for derivatives written as

$$\left\|\dot{\tilde{\sigma}}\right\| \le \left(\sqrt{N_h} + N_h\right)\left(\left\|\dot{x}_d\right\| + \left\|\dot{\hat{x}}_d\right\|\right) = \left(\sqrt{N_h} + N_h\right)\left(c_1(t) + c_2(t)\right) = \left(\sqrt{N_h} + N_h\right)\left(C_1 + C_2\|e\|\right)$$

with $C_1 = C_8 + C_{10}$ and $C_2 = C_9 + C_{11}$.

Now, considering N_{B2}, recalling $\tilde{W} = W - \hat{W}$, and applying (A.1) reveals the bound in (28).

Finally, differentiating N_{B2} reveals $\dot{N}_{B2} = \dot{\tilde{W}}^T \hat{\sigma} + \tilde{W}^T \dot{\hat{\sigma}}$, and observing $\dot{\tilde{W}} = -\dot{\hat{W}}$, utilizing the NN weight update law (31), and applying (A.1) $\dot{N}_{B2}$ is bounded as shown in (29) with $C_3 = F_M \sqrt{N_h}$.

Lemma 2: Given the auxiliary function

$$L = r^T\left(N_{B1} + N_{B2} - \beta_1 \operatorname{sgn}(e_c)\right) - e_c^T N_{B2} - \dot{\beta}_1 \|e_c\| - \dot{e}_c^T \beta_2 \operatorname{sgn}(e_c) - \alpha_0 \beta_2 \|e_c\|, \tag{A.6}$$

let $\beta_1(t)$ and β_2 be chosen according to (30), then

$$\int_0^t L(s)ds \le \gamma \tag{A.7}$$

where $\gamma = \|e_c(0)\|\left(\beta_1(0) + \beta_2\right) - e_c^T(0) N_{B3}(0) \ge 0$ with $N_{B3} = N_{B1} + N_{B2}$.

Proof: Integrating both sides of (A.6), substituting (13) and defining $N_{B3} = N_{B1} + N_{B2}$ yields

$$\begin{aligned}\int_0^t L\, ds = {} & \int_0^t \dot{e}_c^T\left(N_{B3} - \left(\beta_1 + \beta_2\right)\operatorname{sgn}(e_c)\right)ds + \int_0^t \alpha(t) e_c^T\left(N_{B1} + N_{B2}\left(1 - \frac{1}{\alpha(t)}\right) - \beta_1 \operatorname{sgn}(e_c)\right)ds \\ & - \int_0^t \dot{\beta}_1 \|e_c\| ds - \int_0^t \alpha_0 \beta_2 \|e_c\| ds\end{aligned} \tag{A.8}$$

Using integration by parts, the first term can be written as

$$\int_0^t \dot{e}_c^T \left(N_{B3} - (\beta_1 + \beta_2)\text{sgn}(e_c)\right) ds = e_c^T \left(N_{B3} - (\beta_1 + \beta_2)\text{sgn}(e_c)\right)\Big|_0^t - \int_0^t e_c^T \left(\dot{N}_{B3} - \dot{\beta}_1 \text{sgn}(e_c)\right) ds, \quad \text{(A.9)}$$

and substituting (A.9) into (A.8) reveals

$$\int_0^t L\, ds \le \int_0^t \alpha(t) \left\| e_c \right\| \left(\left\| N_{B1} \right\| + \left\| N_{B2} \right\| \left(1 - \frac{1}{\alpha(t)} \right) + \frac{\left\| \dot{N}_{B3} \right\|}{\alpha(t)} - \beta_1 - \frac{\alpha_0}{\alpha(t)} \beta_2 \right) ds + e_c^T \left(N_{B3} - (\beta_1 + \beta_2)\text{sgn}(e_c)\right)\Big|_0^t. \quad \text{(A.10)}$$

Recalling $\alpha(t) = \alpha_0 + \alpha_1(t)$ substituting the bounds (26), (27), (28), and (29) into (A.10) and rearranging allows the terms to be written as

$$\int_0^t L\, ds \le \int_0^t \alpha(t) \left\| e_c \right\| \left(\varsigma_1 + \varsigma_2(t) \left(1 - \frac{1}{\alpha(t)} \right) + \frac{\varsigma_1'(t) + \varsigma_2'(t)}{\alpha(t)} - \beta_1 - \frac{\alpha_0}{\alpha(t)} \beta_2 \right) ds + e_c^T N_{B3}\Big|_0^t - \left\| e_c \right\| (\beta_1 + \beta_2)\Big|_0^t. \quad \text{(A.11)}$$

Next, observing $\alpha(t) \ge \alpha_0$, $\frac{1}{\alpha(t)} \le \frac{1}{\alpha_0}$, and $0 \le 1 - \frac{1}{\alpha(t)} < 1$ for $\alpha_0 \ge 1$, (A.11) can be rewritten to reveal

$$\begin{aligned} \int_0^t L\, ds \le{}& \left\| e_c \right\| (\varsigma_1 + \varsigma_2(t)) - \left\| e_c \right\| (\beta_1(t) + \beta_2) + \left\| e_c(0) \right\| (\beta_1(0) + \beta_2) - e_c^T(0) N_{B3}(0) \\ &+ \int_0^t \alpha(t) \left\| e_c \right\| \left(\varsigma_1 + \varsigma_2(t) + \frac{\varsigma_1'(t) + \varsigma_2'(t)}{\alpha_0} - \beta_1 - \frac{\alpha_0}{\alpha(t)} \beta_2 \right) ds \end{aligned} \quad \text{(A.12)}$$

Examining the first term on the right side of (A.12), it can be concluded that $\left\| e_c \right\| (\varsigma_1 + \varsigma_2(t) - \beta_1(t) - \beta_2) \le 0$ if

$$\beta_1(t) + \beta_2 \ge \varsigma_1 + \varsigma_2(t). \quad \text{(A.13)}$$

If the inequality of (A.13) is satisfied, then the constant term $\left\| e_c(0) \right\| (\beta_1(0) + \beta_2) - e_c^T(0) N_{B3}(0)$ is guaranteed to be greater than zero. Next, the last term in (A.12) is less than zero provided

$$\beta_1(t) + \frac{\alpha_0}{\alpha(t)} \beta_2 \ge \varsigma_1 + \varsigma_2(t) + \frac{\varsigma_1'(t) + \varsigma_2'(t)}{\alpha_0}. \quad \text{(A.14)}$$

Finally, selecting $\beta_1(t) \geq \varsigma_1 + \varsigma_2(t) + \left(\varsigma_1'(t) + \varsigma_2'(t)\right)/\alpha_0$ and $\beta_2 > 0$ and, the inequalities of (A.13) and (A.14) both hold. Through expansion of the bounds in (26), (27), (28), and (29), the gain terms defined in (30) are revealed to be

$$K_W = \sqrt{N_h} + C_{10}\left(\sqrt{N_h} + N_h\right)/\alpha_0, \qquad K_e = \left(\sqrt{N_h} + N_h\right)\left(C_2 + W_M C_{11}\right)/\alpha_0, \qquad K_{ec} = C_3/\alpha_0$$
$$K_{We} = \left(\sqrt{N_h} + N_h\right)C_{11}/\alpha_0, \qquad K_\beta = \varsigma_1 + W_M\sqrt{N_h} + \left(\varepsilon_N' + d_M' + \left(\sqrt{N_h} + N_h\right)\left(C_1 + W_M C_{10}\right)\right)/\alpha_0$$

Remark A.2: In the proof of the following theorems, the subscripts *i* and *j* will be reinstated.

Proof of Theorem 1: *(Follower Dynamic Control)* Consider the following positive definite Lyapunov candidate

$$V_j' = \alpha_{jo} V_j + \Lambda_j V_{jNN} \tag{A.15}$$

where $\Lambda_j = k_{j2} + d_j(k_{j2} + k_{j3}) > 0$, $V_j = \frac{k_{j2}}{2}(e_{j1}^2 + e_{j2}^2) + \frac{d_j k_{j3}}{2} e_{j3}^2$ and

$$V_{jNN} = \frac{1}{2} e_{jc}^T e_{jc} + \frac{1}{2} r_j^T \overline{M}_j r_j + P_j + Q_j \tag{A.16}$$

$$P_j = \left\| e_{jc}(0) \right\| \left(\beta_{j1}(0) + \beta_{j2}\right) - e_{jc}^T(0) N_{Bj3}(0) - \int_0^t L_j(s) ds \tag{A.17}$$

$$Q_j = \frac{1}{2} tr\{\widetilde{W}_j^T F_j^{-1} \widetilde{W}_j\} \tag{A.18}$$

and $L_j(t)$ is defined in (A.6). By Lemma 2, it can be concluded that $P_j \geq 0$. Before proceeding, it is important to observe the existence of the functions $U_1(y_j)$ and $U_2(y_j)$ such that

$$U_1(y_j) \leq V_j' \leq U_2(y_j) \tag{A.19}$$

where $y_j = [e_j^T \quad e_{jc}^T \quad r_j^T \quad \sqrt{P_j} \quad \sqrt{Q_j}] \in \Re^{n+2(r+1)}$, $U_1(y_j)$ and $U_2(y_j)$ are defined by $U_1(y_j) = \lambda_{j1}\|y_j\|^2$ and $U_2(y_j) = \lambda_{j2}\|y_j\|^2$, respectively, with $\lambda_{j1} = \frac{1}{2}\min\{\Lambda_j, \Lambda_j \overline{m}_1, \alpha_{jo} k_{j2}, \alpha_{jo} d_j k_{j3}\}$, $\lambda_{j2} = \max\{\Lambda_j, \frac{\Lambda_j}{2}\overline{m}_2, \alpha_{jo} k_{j2}, \alpha_{jo} d_j k_{j3}\}$ and $\overline{m}_1, \overline{m}_2$ are known positive constants satisfying

$$\overline{m}_1\|y_j\|^2 \le y_j^T \overline{M}_j y_j \le \overline{m}_2\|y_j\|^2.$$

Differentiating V_j, and substitution of the kinematic error dynamics (12)

$$\dot{V}_j = -k_{j2}k_{j1}e_{j1}^2 - k_{j2}^2 e_{j2}^2 - k_{j3}^2 e_{j3}^2 + 2k_{j2}e_{j2}v_i \sin\left(\frac{e_{j3}}{2}\right)\cos\left(\theta_i - \frac{\theta_{jr} + \theta_j}{2}\right) - k_{j3}k_{j2}e_{j2}e_{j3} + k_{j2}e_{j1}e_{j4} + d_j\left(k_{j2}e_{j2} + k_{j3}e_{j3}\right)e_{j5}.$$

Noting that $\left|\sin(e_{j3}/2)\right| \le \left|e_{j3}\right|$ for all $e_{j3} \in [-\pi, \pi]$, $\dot{V}_j$ takes the form of

$$\dot{V}_j \le -k_{j2}k_{j1}e_{j1}^2 - k_{j2}^2 e_{j2}^2 - k_{j3}^2 e_{j3}^2 + k_{j2}\left|e_{j2}e_{j3}\right|\left(k_{j3} + 2v_{i\max}\right) + k_{j2}e_{j1}e_{j4} + d_j\left(k_{j2}e_{j2} + k_{j3}e_{j3}\right)e_{j5} \quad \text{(A.20)}$$

In the next step, it is desired to select k_{j3} such that $\left(k_{j3} + 2v_{i\max}\right) < 2k_{j3}$, and for any $\varepsilon_{vi} > 0$, selecting $k_{j3} = 2v_{i\max} + \varepsilon_{vi}$ ensures this inequality holds. Specifically, we select $\varepsilon_{vi} = 2\varepsilon_{k3}k_{j3}$ where $\varepsilon_{k3} \in (0, 1/2)$ so that $k_{j3} = 2v_{i\max}/(1 - 2\varepsilon_{k3})$, and selecting k_{j3} in this way allows $\dot{V}_j$ to be written as

$$\dot{V}_j \le -k_{j2}k_{j1}e_{j1}^2 - \varepsilon_{k3}k_{j2}^2 e_{j2}^2 - \varepsilon_{k3}k_{j3}^2 e_{j3}^2 - (1 - \varepsilon_{k3})\left(k_{j3}\left|e_{j3}\right| - k_{j2}\left|e_{j2}\right|\right)^2 + k_{j2}e_{j1}e_{j4} + d_j\left(k_{j2}e_{j2} + k_{j3}e_{j3}\right)e_{j5}. \quad \text{(A.21)}$$

Next, differentiating (A.16), noting $\dot{P}_j = -L_j$, utilizing the definition of the filtered tracking error (13), and substituting the filter tracking error dynamics and the derivatives of (A.17) and (A.18) reveals

$$\dot{V}_{jNN} = -\alpha_j(t)e_{jc}^T e_{jc} - (k_{js}+1)r_j^T r_j + r_j^T \widetilde{N}_j + r_j^T\left(N_{Bj1} + N_{Bj2}\right) - r_j^T \beta_{j1}(t)\,\mathrm{sgn}(e_{jc}) - r_j^T \beta_{j2}\,\mathrm{sgn}(e_{jc})$$
$$- L_j + tr\{\widetilde{W}_j^T F^{-1}\dot{\widehat{W}}_j\}$$

Then, substitution of the NN weight tuning law (31) and $L_j(t)$ in (A.6) reveals

$$\dot{V}_{jNN} \le -\alpha_j(t)\left\|e_{jc}\right\|^2 - (k_{js}+1)\left\|r_j\right\|^2 + \left\|r_j\right\|\left\|\widetilde{N}_j\right\| + \left(\left\|\dot{\beta}_{j1}\right\| - \alpha_j(t)\beta_{j2}\right)\left\|e_{jc}\right\| + \alpha_{j0}\beta_{j2}\left\|e_{jc}\right\| \qquad \text{(A.22)}$$

Recalling $\alpha_j(t) = \alpha_{j0} + \alpha_{j1}(t)$ and selecting $\alpha_{j1}(t) \ge \left\|\dot{\beta}_{j1}(t)\right\| / \beta_{j2}$, allows (A.22) to be rewritten as

$$\dot{V}_{jNN} \le -\alpha_{j0}\left\|e_{jc}\right\|^2 - (k_{js}+1)\left\|r_j\right\|^2 + \left\|r_j\right\|\left\|\widetilde{N}_j\right\| \qquad \text{(A.23)}$$

Next, combining (A.21) and (A.23) and completing the squares with respect to e_{j1}, e_{j2}, and e_{j3} yields

$$\dot{V}_j' \le -\alpha_{j0}\lambda_{j3}\left\|e_j\right\|^2 - \alpha_{j0}\lambda_{j4}\left\|e_{jc}\right\|^2 - \Lambda_j(k_{js}+1)\left\|r_j\right\|^2 + \Lambda_j\left\|r_j\right\|\left\|\widetilde{N}_j\right\|$$

where $\lambda_{j3} = \min(k_{j2}\left(k_{j1} - 1/2\right),\ \varepsilon_{k3}k_{j2}\left(k_{j2} - d_j/2\varepsilon_{k3}\right),\ \varepsilon_{k3}k_{j3}\left(k_{j3} - d_j/2\varepsilon_{k3}\right)) > 0$ and provided $k_{j1} > 1/2,\ k_{j2} > d_j/(2\varepsilon_{k3}),\ \text{and}\ k_{j3} > d_j/(2\varepsilon_{k3}),$ and $\lambda_{j4} = \min\left(k_{j2}/2 + d_j(k_{j2}+k_{j3}),\ k_{j2} + d_j(k_{j2}+k_{j3})/2\right) > 0.$ Recalling $k_{j3} = 2v_{i\max}/(1-2\varepsilon_{k3})$, the third inequality can be rewritten as $\dfrac{2v_{i\max}}{1-2\varepsilon_{k3}} > \dfrac{d_j}{2\varepsilon_{k3}}$ or $\varepsilon_{k3} > d_j/\left(4v_{i\max} + 2d_j\right)$, and it is worth noting $\varepsilon_{k3} \in (0,1/2)$ as required since $v_{i\max} > 0$. Next, completing the square with respect to $\left\|r_j\right\|$ and recalling the bound defined in (25), $\dot{V}_j'$ becomes

$$\dot{V}_j' \le -\alpha_{j0}\lambda_{j3}\left\|e_j\right\|^2 - \lambda_{j5}\left\|z_j\right\|^2 - k_{js}\Lambda_j\left(\left\|r_j\right\| - \frac{\rho\left(\left\|z_j\right\|\right)\left\|z_j\right\|}{2k_{js}}\right)^2 + \frac{\Lambda_j\rho\left(\left\|z_j\right\|\right)^2\left\|z_j\right\|^2}{4k_{js}} \qquad \text{(A.24)}$$

where $\lambda_{j5} = \min(\alpha_{j0}\lambda_{j4}, \Lambda_j)$ and greater than zero provided $\alpha_{j0} > 0$. The third term in (A.24) is always less than or equal to zero, so consider the first, second and fourth terms in the following inequality

$$\dot{V}_j' \le -\alpha_{j0}\lambda_{j3}\left\|e_j\right\|^2 - \left(\lambda_{j5} - \frac{\Lambda_j \rho\left(\left\|z_j\right\|\right)^2}{4k_{js}}\right)\left\|z_j\right\|^2 \le -U(y_j) \tag{A.25}$$

where $U(y_j) = c\left\|\begin{bmatrix} e_j^T & z_j^T \end{bmatrix}\right\|^2$ is a continuous positive-semi-definite function for some real positive constant c defined on the domain D such that

$$\dot{V}_j' \le -U(y_j) \text{ for } D = \{y_j \in \Re^{n+2(r+1)} \mid \left\|y_j\right\| \le \rho^{-1}(\sqrt{4\lambda_{j5}k_{js}/\Lambda_j})\}. \tag{A.26}$$

The inequalities in (A.19) and (A.26) can be used to show that $V_j' < \infty$ and bounded in D, and therefore e_j, e_{jc}, r_j, P_j and Q_j are also bounded in D. Continuing this way by observing the boundedness of e_j, e_{jc} and r_j in D, standard linear analysis methods can be used to prove that all of the quantities in (7), (9), (10), (11), (13), (14), (20), and (22) are also bounded in D. Therefore, using the definitions for $U(y_j)$ and $z_j(t)$ it can be concluded that $U(y_j)$ is uniformly continuous. For complete details of the steps to draw this conclusion, see [19].

Let $S \subset D$ denote a region of attraction such that

$$S = \left\{y_j(t) \subset D \mid U_2(y_j(t)) < \lambda_{j2}(\rho^{-1}(\sqrt{4\lambda_{j5}k_{js}/\Lambda_j}))^2\right\}. \tag{A.27}$$

Applying Theorem 8.4 of [21], it can be concluded $c\left\|\begin{bmatrix} e_j^T & z_j^T \end{bmatrix}\right\|^2 \to 0$ as $t \to \infty$ $\forall$ $y_j(0) \in S$. Thus, $\left\|e_j\right\| \to 0$ as $t \to \infty$, and from the definition of $z_j(t)$, it is clear that

$\|e_{jc}\| \to 0$ as $t \to \infty$ for all $y_j(0) \in S$ thus illustrating the asymptotic stability of the tracking errors and the boundedness of the neural network weight estimates.

Remark A.3: The region of attraction (A.27) can be made arbitrarily large to include a larger set of initial conditions by increasing the gain k_{js}. Also, the boundedness $\hat{W}_j$ does not guarantee that the estimates converge to the ideal W unless certain signals are persistently excited [20].

Proof of Theorem 2: *(Leader Stability)* Consider the Lyapunov candidate

$$V_i' = \alpha_{i0} V_i + \Lambda_i V_{iNN} \tag{A.28}$$

where $\Lambda_i = 1 + 1/k_{i2}$, $V_i = \frac{1}{2}(e_{i1}^2 + e_{i2}^2) + \frac{1 - \cos e_{i3}}{k_{i2}}$ and

$$V_{iNN} = \frac{1}{2} e_{ic}^T e_{ic} + \frac{1}{2} r_i^T \overline{M}_i r_i + P_i + Q_i . \tag{A.29}$$

where P_i and Q_i are defined similarly to (A.17) and (A.18), respectively.

First, taking the derivative of V_i and substitution of the error dynamics (32), control velocity (33) and velocity tracking error (34) reveals the following after simplification

$$\dot{V}_i = -k_{i1} e_{i1}^2 - k_{i3} \sin^2 e_{i3} + e_{i1} e_{ic4} + \frac{\sin e_{i3}}{k_{i2}} e_{ic5} \tag{A.30}$$

Then, examining (A.29), one can see that it is defined similarly to the Lyapunov function (A.16) defined for follower j. Exploiting these similarities and applying steps and justifications similar to the ones used to derive (A.22)-(A.27), it is straight forward to show that there exists a domain D_i and region of attraction S_i such that $\dot{V}_{iNN} \leq -U(y_i)$ and thus $\dot{V}_{iNN}$ is uniformly continuous provided $k_{i1} > 1/2$ and $k_{i3} > 1/(2k_{i2})$. Therefore, again

applying Theorem 8.4 of [21], it can be concluded $c_i \left\| \begin{bmatrix} e_{i1} & e_{i3} & z_i^T \end{bmatrix} \right\|^2 = c_i \left\| \begin{bmatrix} e_{i1} & e_{i3} & e_{ic}^T & r_i^T \end{bmatrix} \right\|^2 \to 0$ as $t \to \infty \quad \forall \quad y_i(0) \in S_i$ where c_i is a positive real constant. Thus $\left\| \begin{bmatrix} e_{i1} & e_{i3} \end{bmatrix} \right\| \to 0$, and from the definition of $z_i(t)$, it is clear that $\|e_{ic}\| \to 0$ as $t \to \infty$ for all $y_i(0) \in S_i$ and thus $\dot{V}_{iNN} \to 0$ as $t \to \infty$.

Using the knowledge $\left\| \begin{bmatrix} e_{i1} & e_{i3} \end{bmatrix} \right\| \to 0$ and examining (31) and the definition of e_{ic}, it is then straight forward to verify that $e_{i2} \to 0$ as $t \to \infty$. Thus, the asymptotic stability of the position and velocity tracking errors and the boundedness of the NN weight estimates for leader i follows.

Proof of Theorem 3: *(Formation Stability)* Consider the following Lyapunov candidate

$$V_{ij} = \sum_1^N V_j' + V_i' \tag{A.31}$$

where V_j' is defined by (A.15), V_i' is defined in (A.28). Taking the derivative of (A.31) yields $\dot{V}_{ij} = \sum_1^N \dot{V}_j' + \dot{V}_i'$, and using the results of *Theorems 1* and *2*, there exists a region of attraction S_{ij} defined similarly to (A.27) such that the positions, orientation, and velocity tracking errors for the entire formation are asymptotically stable and the NN weights remain bounded.

Lemma 3: If β_{jo} is chosen according to (45) so that $\beta_{jo} \geq \zeta_M + \frac{1}{\kappa}\zeta_M'$, then

$$\int_0^t R_j(s)ds \leq \beta_{jo} \left\| e_{jo}(0) \right\| - e_{jo}^T(0)\zeta_j(0). \tag{A.32}$$

Proof: Define $R_j(t) = \vartheta_j^T(\zeta_j - \beta_{jo}\,\mathrm{sgn}(e_{jo}))$. Integrating both sides and using (38) yields

$$\int_0^t R_j(s)ds = \int_0^t \kappa e_{jo}(\zeta_j - \beta_{jo}\,\mathrm{sgn}(e_{jo}))ds + \int_0^t \dot{e}_{jo}(\zeta_j - \beta_{jo}\,\mathrm{sgn}(e_{jo}))ds\,. \tag{A.33}$$

Then, applying integration by parts to the second term on the right side of (A.33) reveals

$$\int_0^t R_j(s)ds \le e_{jo}^T(t)\zeta_j(t) - \beta_{jo}\|e_{jo}(t)\| + \beta_{jo}\|e_{jo}(0)\| - e_{jo}^T(0)\zeta_j(0) + \int_0^t \kappa\|e_{jo}\|(\|\zeta_j\| + \frac{1}{\kappa}\|\zeta_j'\| - \beta_{jo})ds. \tag{A.34}$$

Recalling $\|\zeta_j\| \le \zeta_M$ and $\|\zeta_j'\| \le \zeta_M'$ and selecting β_{jo} according to (45), the inequality of (A.32) follows.

Proof of Theorem 4: *(Follower Obstacle Avoidance)* Consider the Lyapunov candidate $V_{jo}' = \alpha_{j0}V_{jo} + \Lambda_{jo}V_{jNN}$ where $\Lambda_{jo} = 1 + d_j$, $V_{jo} = \frac{1}{2}e_{jo}^T e_{jo} + \frac{1}{2}\vartheta_j^T\vartheta_j + \Gamma_j$, V_{jNN} as defined in (A.16) with e_{jc} and r_j replaced by e_{jco} and r_{jo}, respectively, and $\Gamma_j = \beta_{j3}\|e_{jo}(0)\| - e_{jo}^T(0)\zeta_j(0) - \int_0^t R_j(s)ds$. By *Lemma 3*, it can be concluded that $\Gamma_j \ge 0$.

Taking the time derivative of V_{jo} and utilizing (42) and (47) yields

$$\dot{V}_{jo} = e_{jo}^T\vartheta_j - \kappa e_{jo}^T e_{jo} - G\vartheta_j^T\vartheta_j + \vartheta_j^T E_j e_{jco} \tag{A.35}$$

Noting that $e_j^T\vartheta_j \le \frac{1}{2}(e_j^T e_j + \vartheta_j^T\vartheta_j) \le e_j^T e_j + \vartheta_j^T\vartheta_j$ and using the definition of E_j in (45), (A.35) can be rewritten as

$$\dot{V}_{jo} \le -(\kappa - 1)e_{jo}^T e_{jo} - (G-1)\vartheta_j^T\vartheta_j + \vartheta_{j1}e_{jco4} + d_j\vartheta_{j2}e_{jco5} \tag{A.36}$$

Then, differentiating V_{jNN} and applying steps and justifications similar to the ones used to derive (A.22)-(A.27) except completing the squares with respect to ϑ_j instead of e_j, it is

straight forward to show the asymptotic stability of the position and velocity tracking errors and the boundedness of the NN weight estimates provided $\kappa > 1 \text{ and } G > 1 + (1/2)\max(1,\ d_j)$.

2. Neural Network Output Feedback Control of Robot Formations[1]

Travis Dierks and S. Jagannathan

***Abstract*—*In this paper, a combined kinematic/torque output feedback control law is developed for leader-follower based formation control using backstepping in order to accommodate the dynamics of the robots and the formation in contrast with kinematic-based formation controllers. A neural network (NN) is introduced to approximate the dynamics of the follower as well as its leader using online weight tuning. Further, a novel NN observer is designed to estimate the linear and angular velocities of both the follower robot and its leader. It is shown using Lyapunov theory that the errors for the entire formation are uniformly ultimately bounded while relaxing the separation principle. Additionally, the stability of the formation in the presence of obstacles is examined using Lyapunov methods, and by treating other robots in the formation as obstacles, collisions within the formation are prevented. Numerical results are provided to verify the theoretical conjectures.*

Keywords: Formation Control, Output Feedback, Backstepping Control, Lyapunov Stability, Obstacle Avoidance

[1] Research Supported in part by GAANN Program through the Department of Education, NSF #0621924 and Intelligent Systems Center. Authors are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology (formerly University of Missouri-Rolla), 1870 Miner Circle, Rolla, MO 65409. Contact author Email: tad5x4@mst.edu.

I. INTRODUCTION

There are several methodologies [1] to robotic formation control such as behavior-based, generalized coordinates, virtual structures, and, perhaps the most popular and intuitive approach, leader-follower, to name a few. Separation-separation and separation-bearing [2-3] are two popular techniques in leader-follower formation control, and the latter will be considered in this work where the followers stay at a specified separation and bearing from its designated leader.

A characteristic that is common in many formation control schemes [2-6] is the design of a kinematic controller to keep the formation which requires a perfect velocity tracking assumption. Thus, where only velocity commands are treated [2-6], the stability of the formation is entirely dependent on the assumption that the robot perfectly tracks the designed control velocity. In practice, the individual robot and formation dynamics must be considered to ensure that not only the robots track a desired velocity but also the formation errors go to zero.

As observed from robot arm control [16], the dynamics must be considered in practice to guarantee that the robots track a desired velocity while avoiding the use of large control gains which would become necessary to dominate the neglected dynamics in order to ensure an acceptable performance. Similarly, the work in [17] illustrates the need for dynamical controllers for wheeled mobile robots with high inertia, high operating speeds, significant unmodeled dynamics, or high system noise. Therefore, in [7], a neural network (NN) is introduced to learn the dynamics of the follower robots to achieve formation stability using state feedback. Similarly, the work in [8] proposes a decentralized state feedback formation controller based on virtual points for the robots to

track in the formation; however, only the inertial matrix of the robots is considered. In [9], a leader-follower based state feedback formation control scheme is recently introduced using potential as well as bump functions which must be at least three times differentiable and by considering the dynamics of the robots while guaranteeing that collisions do not occur among them. On the other hand, in [10], the robot dynamics are considered using linear parameterization and input-output feedback linearization, and a centralized state feedback formation controller is developed. However, only a basic PD controller is utilized to ensure velocity tracking, and the derivatives of the control velocities are neglected. In each of these works [7-10], the follower dynamics are considered alone whereas the effects of the leader's dynamics on the followers (formation dynamics) are still ignored.

Consequently, in our previous work [11], it was shown that the dynamics of the leader become an important part of its follower robots. In addition, in a string formation of robots where a robot follows another robot directly in front of it, by considering its leader's dynamics, a robot inherently considers the dynamics of the robots in front of them. The dynamical extension in [11] provides a rigorous method of taking into account the specific robot and formation dynamics to convert a steering system command into control inputs via the backstepping approach, and a state feedback controller by assuming that the leader communicates all of its states to its followers is developed using a NN combined with a robustifying feedback term.

By contrast, in this paper, we develop an NN output feedback controller for leader-follower based formation control. The universal approximation property of NN is utilized to learn the complete dynamics of the follower robots and the formation using

online weight tuning. Then, a NN observer is introduced to estimate the linear and angular velocity of the follower as well as its leader so that a specific torque command for the follower robots can be calculated using local sensor measurements with minimal communication between the leader and its followers as opposed to communicating leader's orientation, linear and angular velocities and their control torque [11]. In this work, only the orientation of the leader is assumed available while the separation principle is relaxed. Finally, it is shown that the proposed output feedback controller achieves stability even in the presence of obstacles. Similar to [9], collisions within the formation are avoided in this work too, but without the need of the additional assumption that higher order derivatives are available.

This paper is organized as follows. In Section II, the leader-follower formation control problem and required background information is introduced. Then, a NN output feedback control law is developed for the follower robots by designing a NN observer followed by the design of a NN torque control input, and the stability of the combined systems is examined. Next, a NN output feedback control law and its stability are presented for the leader robot. Finally, the stability of the overall formation is presented, and a general formation controller structure is given which shows the controllers for the leader and followers as well as the interactions between them. In Section III, the NN output feedback control law for the followers is integrated with the leader-follower obstacle avoidance scheme of our previous work [11], and the stability of the modified obstacle avoidance scheme is presented. Section IV presents numerical simulations, and Section V provides some concluding remarks.

II. LEADER-FOLLOWER FORMATION CONTROL

Background information on leader-follower formation control is introduced next. Throughout the development, follower robots will be denoted with a subscript '*j*' while the formation leader will be denoted by the subscript '*i*'. The goal of separation-bearing formation control is to find a velocity control input such that

$$\lim_{t\to\infty}(L_{ijd} - L_{ij}) = 0 \text{ and } \lim_{t\to\infty}(\Psi_{ijd} - \Psi_{ij}) = 0 \tag{1}$$

where L_{ij} and Ψ_{ij} are the measured separation and bearing of the follower *j* with respect to leader *i,* and L_{ijd} and Ψ_{ijd} represent desired distance and angles [2-3], respectively, as shown in Fig. 1. Note that limited sensing capabilities restrict the types of achievable formation topologies. Therefore, care must be taken during the selection of the desired separation and bearing, L_{ijd} and Ψ_{ijd}, respectively, to ensure follower *j* can detect its leader.

The kinematic equations for the front of the j^{th} follower robot can be written as

$$\dot{q}_j = \begin{bmatrix} \dot{x}_j \\ \dot{y}_j \\ \dot{\theta}_j \end{bmatrix} = \begin{bmatrix} \cos\theta_j & -d_j \sin\theta_j \\ \sin\theta_j & d_j \cos\theta_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_j \\ \omega_j \end{bmatrix} = S_j(q_j)\bar{v}_j \tag{2}$$

where d_j is the distance from the rear axle to the to front of the robot, $q_j = [x_j \;\; y_j \;\; \theta_j]^T$ denotes the actual Cartesian position and orientation of the physical robot, v_j, and ω_j represent linear and angular velocities, respectively, and $\bar{v}_j = [v_j \; \omega_j]^T$. Many robotic systems can be characterized as a system having an *n*-dimensional configuration space $\mathcal{C}$ with generalized coordinates $(q_1,...q_n)$ subject to ℓ constraints [12]. Applying the transformation [12], the dynamics of the mobile robots are given by

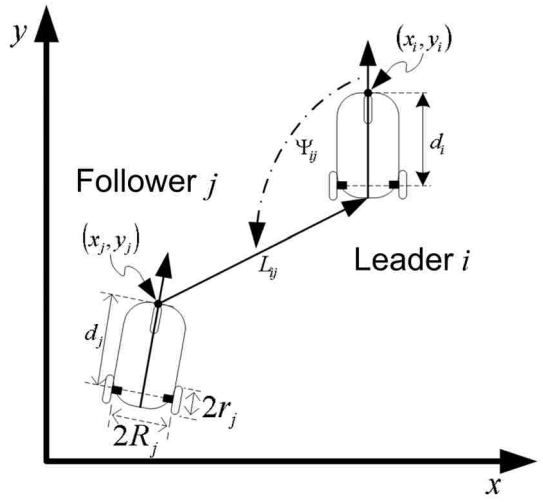


Fig. 1. Separation-bearing formation control.

$$\overline{M}_j \dot{\bar{v}}_j + \overline{V}_{mj}(q_j, \dot{q}_j)\bar{v}_j + \overline{F}_j(\bar{v}_j) + \overline{\tau}_{d_j} = \bar{\tau}_j \tag{3}$$

where $\overline{M}_j \in \Re^{\rho x \rho}$ is a constant positive definite inertia matrix, $\overline{V}_{mj} \in \Re^{\rho x \rho}$ is the bounded centripetal and coriolis matrix, $\overline{F}_j \in \Re^{\rho}$ is the friction vector, $\bar{\tau}_{dj} \in \Re^{\rho}$ represents unknown bounded disturbances such that $\left\| \bar{\tau}_{dj} \right\| \le d_M$ for a known constant, d_M, $\overline{B}_j \in \Re^{\rho x \rho}$ is a constant, nonsingular input transformation matrix, $\bar{\tau}_j = \overline{B}_j \tau_j \in \Re^{\rho}$ is the input vector, and $\tau_j \in \Re^{\rho}$ is the control torque vector. For complete details on (3) and the parameters that comprise it, see [12]. For this work $n = 3$, $\ell = 1$, $\rho = 2$, and the inertial and input transformation matrices are considered to be known while centripetal, friction, and coriolis forces are considered unknown. We will also apply the assumption from [12] that the linear and angular velocities of each robot are bounded for all time, *t*. Robotic systems satisfy the following properties [12]:

1. *Boundedness:* $\overline{M}_j$, the norm of $\overline{V}_{mj}$, and $\bar{\tau}_{dj}$ are all bounded.

2. *Skew Symmetric:* The matrix $\dot{\overline{M}}_j - 2\overline{V}_{mj}$ is skew-symmetric.

A. Backstepping Controller Design

The complete description of the behavior of a mobile robot is given by (2) and (3). The NN output feedback controller is introduced so that the specific torque $\tau_j(t)$ may be calculated so that the alternative control velocity $v_{jc}(t)$ derived in [11] can be tracked without knowing the complete dynamics of the formation while minimizing communication requirements and relaxing the availability of state variables. In this work, each robot is not aware of its velocity or the velocity of its leader. In addition, each robot only has knowledge of its constant inertial and input transformation matrices and no knowledge of its leader's dynamics. Thus, each robot has many challenging uncertainties that must be overcome in order to complete its control objective. The NN in the observer and controller will overcome these problems.

In this work, a two-layer NN consisting of one layer of randomly assigned constant weights $V \in \Re^{axL}$ in the input layer and one layer of tunable weights $W \in \Re^{Lxb}$ in the output layer, with a inputs, b outputs, and L hidden neurons are considered. The *universal approximation property* for NN [13] states that for any smooth function $f(z)$, there exists a NN such that $f(z) = W^T\sigma(V^T z) + \varepsilon$ for some ideal weights W, V, where ε is the NN functional approximation error, and $\sigma(\cdot): \Re^a \to \Re^L$ is the activation function in the hidden layers. It has been shown that by randomly selecting the input layer weights V, the activation function $\sigma(\bar{z}) = \sigma(V^T z)$ forms a stochastic basis, and thus the approximation property holds for all inputs, $z \in \Re^a$, in the compact set S [13]. The

sigmoid activation function is considered here. For complete details of the NN and its properties, see [13].

***Remark 1**:* Throughout this paper, $\|\cdot\|$ and $\|\cdot\|_F$ will be used as the vector and Frobenius matrix norms, respectively [13].

Before we proceed, the following definition and assumptions which are standard in leader-follower formation control [4],[6],[9] and NN literature [13] will be revisited.

Definition 1: An equilibrium point x_e is said to be *uniformly ultimately bounded (UUB)* if there exists a compact set $S \subset \Re^n$ so that for all initial states $x_0 \in S$ there exists a bound B and a time $T(B, x_o)$ such that $\|x(t) - x_e\| \leq B$ for all $t \geq t_0 + T$ [13].

Assumption 1. The separation L_{ij} and bearing Ψ_{ij} [4], [6], and the position and orientation [12] of all the robots are measured whereas velocity measurements are not available.

Assumption 2. Leader i communicates its orientation θ_i to its followers [9].

Assumption 3. On any compact subset of $\Re^n$, the target NN weights W_j and reconstruction errors ε_j are bounded by known positive values for all followers $j = 1,2,...N$ such that $\|W_j\|_F \leq W_M$ and $\|\varepsilon_j\| < \varepsilon_N$, respectively, and all disturbances are bounded such that $\|\bar{\tau}_{dj}\| \leq d_M$ [13].

Remark 2: Ideally, we would like to solve the leader-follower formation control problem using minimal communication. If the follower robots can measure or estimate the orientation of their respective leader, then the proposed output feedback scheme could be implement in a decentralized manner. However, in this work we assume that follower

robots cannot directly detect the orientation of their respective leader, and thus, the orientation of the leader must be communicated to its follower. In addition, this work is developed under the assumption that the velocity vector of each robot is not measureable. As a result, the leader cannot communicate its velocity vector to its followers unless it uses the velocity estimate generated by its observer. Therefore, the follower robots' control laws would be reliant on the accuracy of the leader's observer and susceptible to the leader's observer estimation errors. Since each robot estimates its leader's velocity vector online locally, the risk of observer estimation errors propagating throughout a formation is removed.

B. Leader-Follower Tracking Control

In [11], single robot control frameworks such as [12] were extended to leader-follower formation control subject to the kinematics and dynamics defined by (2) and (3), respectively. Then, a reference position at a desired separation L_{ijd} and bearing Ψ_{ijd} for follower j with respect to the rear of leader i was defined, and the kinematic error system was found to be [11]

$$e_j = \begin{bmatrix} e_{j1} \\ e_{j2} \\ e_{j3} \end{bmatrix} = \begin{bmatrix} L_{ijd}\cos(\Psi_{ijd} + \theta_{ij}) - L_{ij}\cos(\Psi_{ij} + \theta_{ij}) \\ L_{ijd}\sin(\Psi_{ijd} + \theta_{ij}) - L_{ij}\sin(\Psi_{ij} + \theta_{ij}) \\ \theta_{jr} - \theta_j \end{bmatrix} \tag{4}$$

where $\theta_{ij} = \theta_i - \theta_j$ and θ_{jr} is the reference orientation. The reference orientation for follower j is defined relative to the leader satisfying the differential equation as

$$\dot{\theta}_{jr} = \left(\omega_i L_{ijd}\cos(\Psi_{ijd} + \theta_{ij}) + v_i \sin(\theta_{ijr}) + k_{j2} e_{j2}\right)/d_j \,, \tag{5}$$

where $\theta_{ijr} = \theta_i - \theta_{jr} \in [-\pi, \pi]$ and k_{j2} is a positive design constant. It is noted in our previous work [11] that due to the nonholonomic constraint [12] as well as the

separation-bearing formation control objective, the orientations of each robot in the formation will not be equal while the formation is turning, and thus, the reference orientation of each robot cannot be chosen such that $\theta_{jr} = \theta_i$. However, defining θ_{jr} by (5), allows the stability of all three error states to be shown. It can be shown that the reference orientation, θ_{jr}, converges to the orientation of leader i when $\omega_i = 0$ (traveling in a straight path) and formation errors have converged to zero.

The transformed error system (4) now acts as a formation tracking controller which not only seeks to remain at a fixed desired distance L_{ijd} with a desired angle Ψ_{ijd} relative to the leader robot i, but also will achieve a relative orientation with respect to the leader. Further, the orientation of the follower will become the orientation of the leader when $\omega_i = 0$. Finally, the error dynamics of (4) are found to be [11]

$$\begin{bmatrix} \dot{e}_{j1} \\ \dot{e}_{j2} \\ \dot{e}_{j3} \end{bmatrix} = \begin{bmatrix} -v_j + v_i \cos\theta_{ij} + \omega_j e_{j2} - \omega_i L_{ijd} \sin(\Psi_{ijd} + \theta_{ij}) \\ -\omega_j e_{j1} + v_i \sin\theta_{ij} - d_j \omega_j + \omega_i L_{ijd} \cos(\Psi_{ijd} + \theta_{ij}) \\ (\omega_i L_{ijd} \cos(\Psi_{ijd} + \theta_{ij}) + v_i \sin(\theta_{ijr}) + k_{j2} e_{j2}) / d_j - \omega_j \end{bmatrix}. \tag{6}$$

To stabilize the kinematic system, the following velocity control inputs for follower robot *j* were derived using Lyapunov theory [11] to achieve the desired position and orientation with respect to leader i as

$$v_{jc} = \begin{bmatrix} v_{jc1} \\ v_{jc2} \end{bmatrix} = \begin{bmatrix} v_i \cos\theta_{ij} + k_{j1} e_{j1} - \omega_i L_{ijd} \sin(\Psi_{ijd} + \theta_{ij}) \\ \left(\omega_i L_{ijd} \cos(\Psi_{ijd} + \theta_{ij}) + v_i \sin(\theta_{ijr}) + k_{j2} e_{j2} + k_{j3} e_{j3}\right) / d_j \end{bmatrix} \tag{7}$$

where $K_j = [k_{j1} \;\; k_{j2} \;\; k_{j3}]^T$ is a vector of positive design constants. Examining (7), one can see that the linear and angular velocities of leader *i* must be available in order to calculate v_{jc}. However, given only the orientation of the leader (*Assumption 2)*, each follower must estimate the velocity vector of its respective leader online.

C. Leader-Follower NN Observer Design

In order to estimate the linear and angular velocities of the leader, the follower must be able to measure either its x_j and y_j coordinates or its own linear velocity v_j. If neither the position nor the velocity information of the follower is available, only the relative linear velocity between the leader and its follower $v_i - v_j$, can be recovered. In this work, the velocity vector of the follower, $\bar{v}_j$ is considered not measurable, therefore, the position measurements will be used. If the linear velocity of the follower was directly measurable, the observer could easily be modified to recover the linear and angular velocities of the leader.

To begin the development of the NN observer, we define the auxiliary system states as $X_{j1} = [-L_{ij}\cos\Psi_{ij} \quad \theta_i \quad x_j \quad y_j]^T \in \Re^{4x1}$ and $X_{j2} = [v_i \quad \omega_i \quad v_j \quad \omega_j]^T \in \Re^{4x1}$. Note that X_{j1} is available under *Assumptions 1-2*, while X_{j2} is not. The dynamics of L_{ij} and Ψ_{ij} can be written as [2]

$$\begin{aligned} \dot{L}_{ij} &= v_j \cos(\Psi_{ij}+\theta_{ij}) - v_i \cos\Psi_{ij} + d_j \omega_j \sin(\Psi_{ij}+\theta_{ij}) \\ \dot{\Psi}_{ij} &= (v_i \sin\Psi_{ij} - v_j \sin(\Psi_{ij}+\theta_{ij}) + d_j \omega_j \cos(\Psi_{ij}+\theta_{ij}) - L_{ij}\omega_i)/L_{ij} \end{aligned} \tag{8}$$

Differentiating (8), the dynamics of the leader can be written in terms of the formation dynamics and the dynamics of the follower as

$$\dot{\bar{v}}_i = \begin{bmatrix} \cos(\Psi_{ij}) & 0 \\ \dfrac{-\sin(\Psi_{ij})}{L_{ij}} & 1 \end{bmatrix}^{-1} \left(\begin{bmatrix} \cos(\Psi_{ij}+\theta_{ij}) & d_j \sin(\Psi_{ij}+\theta_{ij}) \\ -\dfrac{\sin(\Psi_{ij}+\theta_{ij})}{L_{ij}} & \dfrac{d_j \cos(\Psi_{ij}+\theta_{ij})}{L_{ij}} \end{bmatrix} \dot{\bar{v}}_j - \begin{bmatrix} \ddot{L}_{ij} - L_{ij}(\dot{\Psi}_{ij}+\omega_i)(\dot{\Psi}_{ij}+\dot{\theta}_{ij}) + v_j \sin(\Psi_{ij})\dot{\theta}_{ij} \\ \ddot{\Psi}_{ij} + \dfrac{\dot{L}_{ij}}{L_{ij}}(2\dot{\Psi}_{ij} - 2\omega_i + \omega_j) + \dfrac{v_i}{L_{ij}}\cos(\Psi_{ij})\dot{\theta}_{ij} \end{bmatrix} \right) \tag{9}$$

Using (2), (3), (8), and (9), the dynamics of the auxiliary error system can now be written as

$$\begin{aligned} \dot{X}_{j1} &= A_j(X_{j1})X_{j2} + \zeta_{jo1} \\ \dot{X}_{j2} &= f_{jo}(X_{j1}, X_{j2}) + g_{jo}u_{jo} + \zeta_{jo2} \end{aligned} \tag{10}$$

where $A_j(X_{j1})$ is an invertible, time varying, nonlinear matrix formulated from (2) and (8) and comprised of measurable terms as

$$A_j(X_{j1}) = A_j = \begin{bmatrix} 1 & -L_{ij}\sin(\Psi_{ij}) & -\cos(\theta_{ij}) & -d_j\sin(\theta_{ij}) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta_j & -d_j\sin\theta_j \\ 0 & 0 & \sin\theta_j & d_j\cos\theta_j \end{bmatrix},$$

where $\zeta_{jok} \in \Re^{4x1}$, $k = 1,2$ represents unknown but bounded measurement errors and disturbances such that $\left\|\zeta_{jok}\right\| \le \zeta_{joB}$, $k = 1,2$, $f_{jo}(X_{j1}, X_{j2}) \in \Re^{4x1} = f_{jo} = [f_{jo1}^T \; f_{jo2}^T]^T$ is the vector of robot dynamics formed from (9) and (3) as $f_{jo1} = \dot{\bar{v}}_i \in \Re^{2x1}$ and $f_{jo2} = \overline{M}_j^{-1}\left(\overline{V}_{mj}\bar{v}_j + \overline{F}_j(\bar{v}_j)\right) \in \Re^{2x1}$, respectively, $g_{jo} \in \Re^{4x2}$ is a known constant matrix defined as $g_{jo} = D\overline{M}_j^{-1}$ where $D \in \Re^{4x2}$ is the constant matrix $D = [0\ 0\ 1\ 0; 0\ 0\ 0\ 1]^T$, and $u_{jo} = \bar{\tau}_j \in \Re^{2x1}$ is the control torque for follower j.

Examining the definition of X_{j1}, it is observed that $L_{ij}\cos\Psi_{ij} = 0$ when $\Psi_{ij} = \pm\pi/2$. Further, the dynamics (9) contain a singularity when $\Psi_{ij} = \pm\pi/2$. The singularity problem in (9) is avoided in (10) by using a NN to estimate $\dot{\bar{v}}_i$ and is described in detail in the subsequent development; however, in order to avoid the first term in X_{j1} from becoming zero, formations defined by $\Psi_{ijd} = \pm\pi/2$ are not allowed.

Before proceeding, some useful properties of A_j should be highlighted. First, in the presence of limited sensing capabilities, there exists a maximum measurable

separation distance $L_{ij\max}$; therefore, both A_j and its inverse are bounded by computable constants A_{jM} and A_{jM}^I, respectively. Also, the terms $c_1 I \pm A_j^{-1}\dot{A}_j/2$ and $c_2 I \pm A_j \dot{A}_j^{-1}/2$ are shown to be positive definite using Sylvester's criterion for any choice of real positive constants, c_1 and c_2.

Next, we define a change of variables as $z_{j1} \equiv X_{j1}$ and $z_{j2} \equiv A_j X_{j2}$ and the new system dynamics are given by

$$\begin{aligned} \dot{z}_{j1} &= z_{j2} + \zeta_{jo1} \\ \dot{z}_{j2} &= A_j\left(\bar{f}_{jo} + g_{jo}u_{jo} + \zeta_{jo2}\right) \end{aligned} \tag{11}$$

where $\bar{f}_{jo} = f_{jo} + A_j^{-1}\dot{A}_j A_j^{-1} z_{j2}$. According to the *universal approximation property* of NN, there exist target weights W_{jo}, V_{jo} and a NN approximation error ε_{jo} such that $\bar{f}_{jo} = W_{jo}^T \sigma(V_{jo}^T z_{jo}) + \varepsilon_{jo}$ where z_{jo} is the NN input. Then, the NN estimate $\hat{\bar{f}}_{jo} = \hat{W}_{jo}^T \sigma(V_{jo}^T \hat{z}_{jo})$ can be defined where $\hat{W}_{jo}$ is the NN estimate of W_{jo} and $\hat{z}_{jo}$ is the NN input defined using the estimated states of the observer which will be identified later.

Remark 3: In [14] and [15], observers were proposed utilizing adaptive fuzzy logic and a NN, respectively, and by defining a change of variable. In both these approaches, A_j is the identity matrix, while in this work, A_j is an invertible, time varying, nonlinear matrix.

The NN observer is now defined as

$$\begin{aligned} \dot{\hat{z}}_{j1} &= \hat{z}_{j2} + K_{jo1}\tilde{z}_{j1} \\ \dot{\hat{z}}_{j2} &= A_j\left(\hat{\bar{f}}_{jo} + g_{jo}u_{jo} + \gamma_j\right) + K_{jo2}\tilde{z}_{j1} \end{aligned} \tag{12}$$

where K_{jo1} and K_{jo2} are positive design constants, γ_j is a robustifying term, and $\tilde{z}_{j1} = z_{j1} - \hat{z}_{j1}$ is the error between the measured and observed states. The observer estimation error dynamics are found by subtracting (12) from (11) and adding and subtracting $A_j W_{jo}^T \sigma(V_{jo}^T \hat{z}_{jo})$ to $\dot{\tilde{z}}_{j2}$ to yield

$$\begin{aligned} \dot{\tilde{z}}_{j1} &= \tilde{z}_{j2} - K_{jo1}\tilde{z}_{j1} + \zeta_{jo1} \\ \dot{\tilde{z}}_{j2} &= A_j\left(\tilde{W}_{jo}^T \hat{\sigma}_{jo} - \gamma_j + \delta_{jo}\right) - K_{jo2}\tilde{z}_{j1} \end{aligned} \tag{13}$$

where $\tilde{W}_{jo} = W_{jo} - \hat{W}_{jo}$ is the weight estimation error, $\hat{\sigma}_{jo} = \sigma(V_{jo}^T \hat{z}_{jo})$ is the hidden-layer activation function vector, and $\delta_{jo} = \zeta_{jo2} + \varepsilon_{jo} + W_{jo}^T\left(\sigma(V_{jo}^T z_{jo}) - \sigma(V_{jo}^T \hat{z}_{jo})\right)$. Utilizing *Assumption 3* along with properties of the sigmoid activation function [13], it is straight forward to show $\left\|\delta_{jo}\right\| \le \delta_{jB}$ for all time t where $\delta_{jB} > 0$ is a computable constant.

The robot velocity estimates $\hat{X}_{j2}$ are then defined as

$$\hat{X}_{j2} = A_j^{-1}\left(\hat{z}_{j2} + K_{jo3}\tilde{z}_{j1}\right)$$

where K_{jo3} is a positive design constant. The estimation errors for the auxiliary system (10) are defined as $\tilde{X}_{j1} = \tilde{z}_{j1}$ and $A_j\tilde{X}_{j2} = \tilde{z}_{j2} - K_{jo3}\tilde{z}_{j1}$. Differentiating $\tilde{X}_{j1}$ and $A_j\tilde{X}_{j2}$ yields the error dynamics of the auxiliary system as

$$\begin{aligned} \dot{\tilde{X}}_{j1} &= -(K_{jo1} - K_{jo3})\tilde{X}_{j1} + A_j\tilde{X}_{j2} + \zeta_{jo1} \\ \dot{\tilde{X}}_{j2} &= -\left(K_{jo3}I + A_j^{-1}\dot{A}_j\right)\tilde{X}_{j2} + \tilde{W}_{jo}^T\hat{\sigma}_{jo} - \gamma_j - K_{jo2}A_j^{-1}\tilde{X}_{j1} - A_j^{-1}K_{jo3}\left(K_{jo3} - K_{jo1}\right)\tilde{X}_{j1} + \delta'_{jo} \end{aligned} \tag{14}$$

where $\delta'_{jo} = A_j^{-1}K_{jo3}\zeta_{jo1} + \delta_{jo}$ and $\left\|\delta'_{jo}\right\| \le \delta'_{jB}$ where δ'_{jB} is another computable constant. Finally, the observer NN input can be defined as $\hat{z}_{jo} = [1\ X_{j1}^T\ \ \hat{X}_{j2}^T\ \ \Psi_{ij}\ \ L_{ij}\ \ \dot{\hat{\Psi}}_{ij}\ \ \dot{\hat{L}}_{ij}\ \ \ddot{\hat{\Psi}}_{ij}\ \ \ddot{\hat{L}}_{ij}]$,

where $\dot{\hat{\Psi}}_{ij}, \dot{\hat{L}}_{ij}, \ddot{\hat{\Psi}}_{ij},$ and $\ddot{\hat{L}}_{ij}$ are the estimates and derivative estimates of (8), respectively, which are estimated using the observer velocity estimates.

Theorem 1: (Follower Robot NN Observer) Given the auxiliary system (10), and NN observer (12) for follower j, and let *Assumptions 1-3* hold. Define the robustifying signal

$$\gamma_j = -A_j^{-1} K_{jo3} (K_{jo3} - K_{jo1}) \tilde{X}_{j1} \tag{15}$$

with the NN update law for the observer given by

$$\dot{\hat{W}}_{jo} = F_o \hat{\sigma}_{jo} \left\| A_j^{-1} \right\|_F \tilde{X}_{j1}^T - F_o \left(\kappa_{o1} \left\| \tilde{X}_{j1} \right\| \left\| A_j^{-1} \right\|_F + \kappa_{o2} \left\| A_j \right\|_F \right) \hat{W}_{jo} \tag{16}$$

where $F_o = F_o^T > 0$ and $\kappa_{o1}, \kappa_{o2} > 0$ are design parameters. Then there exists positive design parameters $K_{jo1}, K_{jo2},$ and K_{jo3} such that the observer estimation errors $\tilde{X}_{j1}, \tilde{X}_{j2}$ and the NN observer weight estimation errors, $\tilde{W}_{jo}$, are *UUB*.

Please see the Appendix for proof of *Theorem 1.*

D. Dynamical NN Torque Controller

In this work, the velocity vectors of each mobile robot are not available. Therefore, the control laws must be defined using the velocity estimates of the NN observer derived in the previous section. To begin, the control velocity (7) is estimated using the observed velocities of the leader and written as

$$\hat{v}_{jc} = \begin{bmatrix} \hat{v}_{jc1} \\ \hat{v}_{jc2} \end{bmatrix} = \begin{bmatrix} (\hat{v}_i + K_{jo4} \tilde{X}_{j11}) \cos\theta_{ij} + k_{j1} e_{j1} - (\hat{\omega}_i + K_{jo4} \tilde{X}_{j12}) L_{ijd} \sin(\Psi_{ijd} + \theta_{ij}) \\ \left((\hat{\omega}_i + K_{jo4} \tilde{X}_{j12}) L_{ijd} \cos(\Psi_{ijd} + \theta_{ij}) + (\hat{v}_i + K_{jo4} \tilde{X}_{j11}) \sin(\theta_{ijr}) + k_{j2} e_{j2} + k_{j3} e_{j3} \right) / d_j \end{bmatrix} \tag{17}$$

where K_{jo4} is a positive design constant, and $\widetilde{X}_{j11}$ and $\widetilde{X}_{j12}$ are the first and second elements of the observer error vector $\widetilde{X}_{j1}$, respectively. Next, we define the actual and estimated velocity tracking errors, e_{jc} and $\hat{e}_{jc}$, respectively, as

$$e_{jc} = [e_{j4} \;\; e_{j5}]^T = [v_{jc1} \;\; v_{jc2}]^T - [v_j \;\; \omega_j]^T = v_{jc} - \bar{v}_j \tag{18}$$

and

$$\hat{e}_{jc} = [\hat{e}_{j4} \;\; \hat{e}_{j5}]^T = [\hat{v}_{jc1} \;\; \hat{v}_{jc2}]^T - [\hat{v}_j \;\; \hat{\omega}_j]^T = \hat{v}_{jc} - \hat{\bar{v}}_j . \tag{19}$$

Defining $\widetilde{\bar{v}}_j = \bar{v}_j - \hat{\bar{v}}_j$, observing $\bar{v}_j = \hat{v}_{jc} - \hat{e}_{jc} + \widetilde{\bar{v}}_j$, and substituting this along with the control velocity (17) into (6), reveals the closed loop kinematic error dynamics as

$$\begin{bmatrix} \dot{e}_{j1} \\ \dot{e}_{j2} \\ \dot{e}_{j3} \end{bmatrix} = \begin{bmatrix} \hat{e}_{j4} - k_{j1} e_{j1} + \omega_j e_{j2} \\ d_j \hat{e}_{j5} + 2 v_i \sin\left(\frac{e_{j3}}{2}\right) \cos\left(\theta_i - \frac{\theta_{jr} + \theta_j}{2}\right) - k_{j2} e_{j2} - k_{j3} e_{j3} - \omega_j e_{j1} \\ \hat{e}_{j5} - k_{j3} e_{j3} / d_j \end{bmatrix} + \widetilde{\widetilde{X}}_j \tag{20}$$

where

$$\widetilde{\widetilde{X}}_j = \begin{bmatrix} \cos\theta_{ij} & -L_{ijd}\sin(\Psi_{ijd} + \theta_{ij}) & -1 & 0 \\ \sin\theta_{ijr} & L_{ijd}\cos(\Psi_{ijd} + \theta_{ij}) & 0 & -d_j \\ 0 & 0 & 0 & -1 \end{bmatrix} \widetilde{X}_{j2} - K_{jo4} \begin{bmatrix} \cos\theta_{ij} & -L_{ijd}\sin(\Psi_{ijd} + \theta_{ij}) \\ \sin\theta_{ijr} & L_{ijd}\cos(\Psi_{ijd} + \theta_{ij}) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \widetilde{X}_{j11} \\ \widetilde{X}_{j12} \end{bmatrix}$$

and $\tilde{X}_{j2}$ is the observer estimation error vector of the velocities of the robots.

Remark 4: Recall that the reference orientation dynamics (5) are defined in terms of the linear and angular velocities of the leader; therefore, the reference orientation must be rewritten in terms of the observer velocity estimates. Thus, the observer estimation errors are not present in the dynamics of the third error state, $\dot{e}_{j3}$. Note that the stability of the kinematic error system now depends on the estimated velocity tracking errors, demonstrating the necessity for a dynamical velocity tracking control loop.

Moving on and observing $\bar{v}_j = \hat{\bar{v}}_j + \tilde{\bar{v}}_j$, and $v_{jc} = \hat{v}_{jc} + \tilde{X}'_{j2} - \tilde{X}'_{j1}$ where

$$\tilde{X}'_{j2} = \begin{bmatrix} \cos\theta_{ij} & -L_{ijd}\sin(\Psi_{ijd}+\theta_{ij}) & 0 & 0 \\ \sin\theta_{ijr} & L_{ijd}\cos(\Psi_{ijd}+\theta_{ij}) & 0 & 0 \end{bmatrix}\tilde{X}_{j2} \text{ and } \tilde{X}'_{j1} = K_{jo4}\begin{bmatrix} \cos\theta_{ij} & -L_{ijd}\sin(\Psi_{ijd}+\theta_{ij}) \\ \sin\theta_{ijr} & L_{ijd}\cos(\Psi_{ijd}+\theta_{ij}) \end{bmatrix}\begin{bmatrix} \tilde{X}_{j11} \\ \tilde{X}_{j12} \end{bmatrix},$$

e_{jc} and $\hat{e}_{jc}$ can be related by

$$\hat{e}_{jc} = e_{jc} - \tilde{X}'_{j2} + \tilde{X}'_{j1} + \tilde{\bar{v}}_j \tag{21}$$

To form the error dynamics of (19), we first find the error dynamics of (18). To begin, add and subtract $\bar{M}_j \dot{v}_{jc}$ and $\bar{V}_{mj} v_{jc}$ to (3), and substitute (18) and its derivative into (3) to reveal the actual velocity tracking error dynamics to be

$$\bar{M}_j \dot{e}_{jc} = -\bar{V}_{mj} e_{jc} + \bar{M}_j \dot{v}_{jc} + \bar{V}_{mj} v_{jc} + \bar{F}_j(\bar{v}_j) - \bar{\tau}_j + \bar{\tau}_{dj}. \tag{22}$$

Then, multiplying both sides of (21) by $\bar{M}_j$ and taking its first derivative with respect to time as well as substituting (22) into the derivative of (21) reveals the dynamics of the estimated velocity tracking error to be

$$\bar{M}_j \dot{\hat{e}}_{jc} = -\bar{V}_{mj} \hat{e}_{jc} + f_j(z_j) - \bar{\tau}_j + \bar{\tau}_{dj} \tag{23}$$

where $f_j(z_j) = \bar{M}_j \dot{v}_{jc} + \bar{V}_{mj} v_{jc} + \bar{F}_j(\bar{v}_j) - \bar{M}_j(\dot{\tilde{X}}'_{j2} - \dot{\tilde{X}}'_{j1} - \dot{\tilde{\bar{v}}}_j) + \bar{V}_{mj}(\tilde{X}'_{j2} - \tilde{X}'_{j1} - \tilde{\bar{v}}_j)$ and $z_j = [v_{jc}^T, \dot{v}_{jc}^T, \bar{v}_j^T, \tilde{X}'^T_{j2}, \tilde{X}'^T_{j1}, \tilde{\bar{v}}_j^T, \dot{\tilde{X}}'^T_{j2}, \dot{\tilde{X}}'^T_{j1}, \dot{\tilde{\bar{v}}}_j^T]^T$. The nonlinear function $f_j(z_j)$ brings in the dynamics of leader *i* through $\dot{v}_{jc}$ as given by $\dot{v}_{jc} = f_{\dot{v}cj}(\dot{\bar{v}}_i, \bar{v}_i, e_j, \dot{e}_j)$ where $f_{\dot{v}cj}(\bullet)$ is the nonlinear function that relates $\dot{v}_{jc}$.

According the NN *universal approximation property*, there exists constant, bounded, ideal weights W_j, V_j, such that $\|W_j\|_F \le W_M$ and $f_j(z_j) = W_j^T \sigma(V_j^T z_j) + \varepsilon_j$ where ε_j is the bounded NN approximation error such

that $\varepsilon_j \le \varepsilon_M$ [13]. Since the complete dynamics (3) are not known, the *universal approximation property* will be invoked to estimate the function $f_j(z_j)$ and thus estimate the dynamics of follower j by $\hat{f}_j(\hat{z}_j) = \hat{W}_j^T \sigma(V_j^T \hat{z}_j) = \hat{W}_j^T \hat{\sigma}_j$ where $\hat{f}_j(\hat{z}_j)$ and $\hat{W}_j$ are the NN approximations of $f_j(z_j)$ and W_j, respectively. In order to accommodate the dynamics of the follower and the formation, the estimated NN input, $\hat{z}_j$, is defined as $\hat{z}_j = [1 \;\; \hat{v}_{jc}^T \;\; \dot{\hat{v}}_{jc}^T \;\; \hat{\bar{v}}_j^T \;\; \theta_j \;\; \hat{\bar{v}}_i^T \;\; \theta_i \;\; \hat{e}_{jc}^T \;\; e_j^T]^T$ where $\dot{\hat{v}}_{jc}^T$ is the estimate of $\dot{v}_{jc}^T$ calculated using the observer velocity estimates and their dynamics. Then, the torque control input for follower j can be written as

$$\bar{\tau}_j = K_{j4} \hat{e}_{jc} + \hat{f}_j(\hat{z}_j) + u_j \tag{24}$$

where u_j is a robustifying term defined as

$$u_j = \alpha_j \begin{bmatrix} k_{j2} e_{j1} \\ d_j (k_{j2} e_{j2} + k_{j3} e_{j3}) \end{bmatrix} \tag{25}$$

and α_j is a positive design constant.

Substitution of (24) into the error dynamics (23) as well as adding and subtracting $W_j^T \hat{\sigma}_j$ reveals

$$\bar{M}_j \dot{\hat{e}}_{jc} = -\left(K_{j4} + \bar{V}_{mj}\right) \hat{e}_{jc} - u_j + \tilde{W}_j^T \hat{\sigma}_j + \zeta_j \tag{26}$$

where $\tilde{W}_j = W_j - \hat{W}_j$, $\zeta_j = W_j^T \tilde{\sigma}_j + \bar{\tau}_{dj} + \varepsilon_j$ and $\tilde{\sigma}_j = \sigma_j - \hat{\sigma}_j$. It is noted that $\left\| \zeta_j \right\| \le \zeta_{jM}$ for a computable constant $\zeta_{jM} = 2W_M \sqrt{N_c} + d_M + \varepsilon_M$ where N_c is the number of hidden layer neurons in the control NN and the relation $\left\| \tilde{\sigma}_j \right\| \le 2\sqrt{N_c}$ [15] was utilized. Note in the absence of disturbances, observer estimation errors, and NN

approximation errors, the origin $\hat{\bar{e}}_j = [e_j^T \ \hat{e}_{jc}^T]^T = 0$ is an equilibrium point of (20) and (26).

Remark 5: Note that the robust term u_j (25) is not necessary to prove the stability of the error system for the follower robot and the entire formation. However, investigating the control torque (24) one can see u_j is the only term that is well known and measurable. As a result, the reliable signal u_j is included in the control input (24).

Next, the stability of the combined NN observer and output feedback controller is established in *Theorem 2* using Lyapunov analysis methods without the need of the separation principle where it will be shown that the observer estimation errors, position errors, and estimated velocity tracking errors are all *UUB*.

Theorem 2: (Follower Output Feedback Control): Given the nonholonomic robot system consisting of (2) and (3), the leader follower criterion of (1) as well as the auxiliary system for follower *j* given by (10) and the NN observer defined by (12), let *Assumptions 1-3* hold. Let a smooth velocity control input, $\hat{v}_{jc}$, and torque control, $\bar{\tau}_j$, for the follower *j* be given by (17) and (24), respectively, along with the robustifying term be given by (25). Consider the NN observer update law (16) and dynamic NN controller update law as

$$\dot{\hat{W}}_j = F\sigma_j \hat{e}_{jc}^T - F\left(\kappa_1 \left\|\hat{\bar{e}}_j\right\| + \kappa_2\right)\hat{W}_j \tag{27}$$

where $F = F^T > 0$ and $\kappa_1 > 0, \kappa_2 > 0$ are small design parameters. Then, there exist positive constants $K_{jo1}, K_{jo2}, K_{jo3}$ and K_{jo4}, a vector of positive constants, $K_j = [k_{j1} \quad k_{j2} \quad k_{j3}]^T$, k_{j4}, and α_j such that the NN observer estimation errors $\tilde{X}_{j1}, \tilde{X}_{j2}$

and its weight estimation errors $\widetilde{W}_{jo}$, the origin $\hat{\bar{e}}_j = 0$, consisting of the position, orientation and estimated velocity tracking errors, and the control NN weight estimation errors $\widetilde{W}_j$, for follower j are all *UUB*.

Please see the Appendix for proof of *Theorem 2.*

Remark 6: Recalling the relationship between the actual velocity tracking error and the estimated velocity tracking error defined in (21), it is clear that the convergence of the observer estimation errors and the estimated velocity tracking error to a compact set guarantees the convergence of e_{jc}.

E. Leader Control Structure

The kinematics and dynamics of the formation leader i are defined similarly to (2) and (3), respectively, for follower j. From [12], the leader tracks a virtual reference robot, and the tracking error for the leader and its derivative are found to be

$$e_i = \begin{bmatrix} e_{i1} \\ e_{i2} \\ e_{i3} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_i \\ y_r - y_i \\ \theta_r - \theta_i \end{bmatrix} \text{ and } \dot{e}_i = \begin{bmatrix} \dot{e}_{i1} \\ \dot{e}_{i2} \\ \dot{e}_{i3} \end{bmatrix} = \begin{bmatrix} -v_i + v_{ir}\cos e_{i3} + \omega_i e_{i2} \\ -\omega_i e_{i1} + v_{ir}\sin e_{i3} \\ \omega_{ir} - \omega_i \end{bmatrix}$$

where x_{ir}, y_{ir}, θ_{ir} v_{ir} and ω_{ir} are the Cartesian position in the x and y direction, orientation and the linear and angular velocities, respectively, of a virtual reference robot for leader i.

Since leader i tracks a virtual robot, it has knowledge of the velocities of the reference robot; however, under *Assumption 1,* the leader cannot measure its own linear and angular velocities, and thus, an observer must be utilized. Similarly to the observer development for follower j, we define an auxiliary system consisting of a measurable

term $X_{i1} = [x_i \; y_i \; \theta_i]^T \in \Re^{3x1}$ and an immeasurable term $X_{i2} = [v_i \quad \omega_i]^T \in \Re^{2x1}$. The dynamics of X_{i1} and X_{i2} are then written as

$$\dot{X}_{i1} = S_i(X_{i1})X_{i2} + \zeta_{io1}$$
$$\dot{X}_{i2} = f_{io}(X_{i1}, X_{i2}) + g_{io}u_i + \zeta_{io2}$$

where $S_i(X_{i1}) = S_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -d_i \sin\theta_i & d_i \cos\theta_i & 1 \end{bmatrix}^T$, $f_{io}(X_{i1}, X_{i2}) = \overline{M}_i^{-1}\left(\overline{V}_{mi}\overline{v}_i + \overline{F}_i(\overline{v}_i)\right)$, $g_{io}u_i = \overline{M}_i^{-1}\overline{\tau}_i$ are formulated from the kinematics and dynamics of robot i and $\zeta_{io1} \in \Re^{3x1}$ and $\zeta_{io2} \in \Re^{2x1}$ represent unknown but bounded disturbances. It is useful to point out $c_3 I \pm S_i^T \dot{S}_i$ is positive definite for any choice of a positive constant c_3 and I being the identity matrix of appropriate dimension.

Similar to the observer design for follower j, a change of coordinates is defined as $z_{i1} \equiv X_{i1} \in \Re^{3x1}$ and $z_{i2} \equiv S_i X_{i2} \in \Re^{3x1}$ for convenience, and an observer for the leader i can be realized as

$$\begin{aligned} \dot{\hat{z}}_{i1} &= \hat{z}_{i2} + K_{io1}\tilde{z}_{i1} \\ \dot{\hat{z}}_{i2} &= \hat{\bar{f}}_{io} + S_i\left(g_{io}u_{io} + \gamma_i\right) + K_{io2}\tilde{z}_{j1} \end{aligned} \tag{28}$$

where K_{io1} are K_{io2} real positive design constants, $\tilde{z}_{i1} = z_{i1} - \hat{z}_{i1}$, and γ_i is a robustifying signal defined as $\gamma_i = -S_i^T K_{io3}\left(\left(K_{io3} - K_{io1}\right)\tilde{X}_{i1}\right)$. The NN *universal approximation property* is also utilized to estimate the unknown dynamics of leader i in $\bar{f}_{io} = S_i f_{io} + \dot{S}_i X_{i2} \in \Re^{3x1}$, and the NN takes the form of $\hat{\bar{f}}_{io} = \hat{W}_{io}^T \sigma(V_{io}^T \hat{z}_{io})$ where $\hat{W}_{io}^T$ is the estimate of the target observer weights W_{io}^T, and $\hat{z}_{io}$ is the input to the NN defined using the observer velocity estimates as $\hat{z}_{io} = [1 \; X_{i1}^T \quad \hat{X}_{i2}^T]^T$. The estimate of the

leader's velocities can now be defined as $\hat{X}_{i2} = S_i^T(\hat{z}_{i2} + K_{io3}\tilde{z}_{i1})$ where K_{io3} is a positive design constant.

From (28) and the definitions of z_{i1} and z_{i2}, the dynamics of the estimation errors $\tilde{X}_{i1}$ and $\tilde{X}_{i2}$ are found similarly to (14) as

$$\begin{aligned} \dot{\tilde{X}}_{i1} &= S_i\,\tilde{X}_{i2} - (K_{io1} - K_{io3})\tilde{X}_{i1} + \zeta_{io1} \\ \dot{\tilde{X}}_{i2} &= -(K_{io3} + S_i^T\dot{S}_i)\tilde{X}_{i2} + S_i^T\left(\tilde{f}_{io} - K_{io2}\tilde{X}_{i1}\right) + \delta_{io3} \end{aligned} \tag{29}$$

where $\delta_{io3} = S_i^T(\zeta_{io2} + \varepsilon_{io} + W_{io}^T\left(\sigma(V_{io}^T z_{io}) - \sigma(V_{io}^T\hat{z}_{io})\right) - K_{io3}\zeta_{io1}) \le \delta_{i3B}$ with δ_{i3B} being a computable constant, and ε_{io} is the NN approximation error.

Moving on, the control laws for the leader can now be presented. The control velocity $v_{ic}(t)$ can be defined similarly to [12] as

$$v_{ic} = \begin{bmatrix} v_{ir}\cos e_{i3} + k_{i1}e_{i1} \\ \omega_{ir} + k_{i2}v_{ir}e_{i2} + k_{i3}(v_{ir}+1)\sin e_{i3} \end{bmatrix} \tag{30}$$

where k_{i1}, k_{i2}, k_{i3} are design constants. Next, define the estimated velocity tracking error as

$$\hat{e}_{ic} = [\hat{e}_{i4} \quad \hat{e}_{i5}]^T = [v_{ic} \quad \omega_{ic}]^T - [\hat{v}_i \quad \hat{\omega}_i]^T = v_{ic} - \hat{\bar{v}}_i \tag{31}$$

Defining $\tilde{\bar{v}}_i = \bar{v}_i - \hat{\bar{v}}_i$, manipulating (31) to reveal $\bar{v}_i = v_{ic} - \hat{e}_{ic} + \tilde{\bar{v}}_i$ and substituting this relation into $\dot{e}_i$ above yields the closed loop kinematic tracking error dynamics written to include the estimated velocity tracking errors as well as the observer estimation errors as

$$\dot{e}_i = \begin{bmatrix} \dot{e}_{i1} \\ \dot{e}_{i2} \\ \dot{e}_{i3} \end{bmatrix} = \begin{bmatrix} \hat{e}_{i4} - k_{i1}e_{i1} + \omega_i e_{i2} - \tilde{\bar{v}}_i \\ -\omega_i e_{i1} + v_{ir}\sin e_{i3} \\ \hat{e}_{i5} - k_{i2}v_{ir}e_{i2} - k_{i3}(v_{ir}+1)\sin e_{i3} - \tilde{\omega}_i \end{bmatrix} \tag{32}$$

In the absence of observer estimation errors, $\hat{\bar{e}}_i = [e_i^T \;\; \hat{e}_{ic}^T] = 0$ is an equilibrium point of (32). Using similar steps as that of (23), the error system for the leader can be obtained similar to follower *j*. The torque input $\bar{\tau}_i$ for the leader is defined as

$$\bar{\tau}_i = K_{i4}\hat{e}_{ic} + \hat{f}_i(\hat{z}_i) + u_i \tag{33}$$

where $u_i \in \Re^{2x1}$ is a robustifying term defined as

$$u_i = \alpha_i \begin{bmatrix} e_{i1} \\ \sin(e_{i3})/k_{i2} \end{bmatrix} \tag{34}$$

with α_i being a positive design constant and $K_{i4} = k_{i4} I$ for a positive design constant k_{i4}. Here the NN *universal approximation property* has been used to estimate the nonlinear function $f_i(z_i) = W_i^T \sigma(V_i^T z_i) + \varepsilon_i$, which is defined similar to $f_j(z_j)$ for follower *j* (23). The NN estimate is then written as $\hat{f}_i(\hat{z}_i) = \hat{W}_i^T \sigma(V_i^T \hat{z}_i)$ where $\hat{W}_i$ is the approximation of the target NN weights, W_i, and $\hat{z}_i = [1 \; v_{ic}^T \; \dot{v}_{ic}^T \; \hat{\bar{v}}_i^T \; \theta_i \; \hat{e}_{ic}^T \; e_i^T]^T$ is the NN input written in terms of the observer state estimates.

Remark 7: Similarly to follower *j*, the stability of leader *i* can be proven without the robustifying term (34), but is included in (33) since it is a reliable signal whereas the other terms in (33) are all being estimated.

Using (33), the closed loop error system for leader *i* can be formed similarly to the closed loop error system for the follower (26) as

$$\bar{M}_i \dot{\hat{e}}_{ic} = -\left(K_{i4} + \bar{V}_{mi}\right)\hat{e}_{ic} - u_i + \tilde{W}_i^T \hat{\sigma}_i + \zeta_i \tag{35}$$

where $\|\zeta_i\| \le \zeta_{iM}$ for a computable constant ζ_{iM} defined similarly to ζ_{jM}. Examining (34) and (35), it can be concluded that $\hat{\bar{e}}_i = [e_i^T \;\; \hat{e}_{ic}^T] = 0$ is an equilibrium point (35) in the absence of disturbances and NN approximation errors.

Theorem 3 (Leader Robot Control): Given the kinematic and dynamic system for leader i (defined similar to (2) and (3), respectively), let $v_{ir} > 0$ and ω_{ir} be bounded. Consider the NN observer defined by (28) with its weight update defined by

$$\dot{\hat{W}}_{io} = F_{io}\hat{\sigma}_{io}\tilde{X}_{i1}^T - F_{io}\left(\kappa_{io1}\left\|\tilde{X}_{i1}\right\| + \kappa_{io2}\right)\hat{W}_{io} \tag{36}$$

where $F_{io} = F_{io}^T > 0$, $\kappa_{io1} > 0$ and $\kappa_{io2} > 0$ are design parameters. Let a smooth velocity control input $v_{ic}(t)$ (30) and NN torque control (33) be applied, and the NN controller weight update law be given by

$$\dot{\hat{W}}_i = F_i\sigma_i\hat{e}_{ic}^T - F_i\left(\kappa_{i1}\left\|\hat{\bar{e}}_i'\right\| + \kappa_{i2}\right)\hat{W}_i \tag{37}$$

where $F_i = F_i^T > 0$, $\kappa_{i1} > 0$ and $\kappa_{i2} > 0$ are small design parameters, and $\left\|\hat{\bar{e}}_i'\right\| = \left\|[e_{j1} \;\; \sin e_{j3} \;\; \hat{e}_{jc}^T]\right\|$ is an auxiliary error signal. Then there exists positive constants K_{io1}, K_{io2}, K_{io3}, $k_{i1}, k_{i2}, k_{i3}, k_{i4}$, and α_i such that the observer estimation errors $\tilde{X}_{i1}, \tilde{X}_{i2}$, the position, orientation and velocity tracking errors $\hat{\bar{e}}_i$, and the NN weight estimation errors of the observer and the dynamic controller, $\tilde{W}_{io}, \tilde{W}_i$, respectively, are all *UUB*.

Please see the Appendix for proof of *Theorem 3.*

Remark 8: Observing $e_{ic} = \hat{e}_{ic} - \tilde{\bar{v}}_i$, it is clear that the convergence of the observer estimation errors and the estimated velocity tracking error to a compact set guarantees the convergence of e_{ic}. Next the stability of the formation is introduced.

F. Formation Stability

It has been shown that the dynamics of leader *i* are incorporated into the control torque of follower *j*. Similarly, in a formation topology where follower *j* becomes a

leader to follower *j+1*, the dynamics of follower *j* become incorporated into the control torque of follower *j+1*, and since the dynamics of follower *j* incorporates the dynamics of leader *i,* follower *j+1* inherently brings in the dynamics of leader *i* by considering the dynamics of follower *j*. As a result, the formation error dynamics of a formation consisting of one leader and *N* followers can be captured by taking the sum of the individual Lyapunov candidates for leader *i* and follower *j,* $j = 1, 2, ...N$ as demonstrated in the following theorem.

Theorem 4 (Formation Stability): Let the hypotheses of *Theorems 2* and *3* hold. Then, the formation errors consisting of leader and follower states are UUB.

Please see the Appendix for proof of *Theorem 4.*

The overall formation controller is now presented in Fig. 2. In the figure, the complete control structures for follower *j* and leader *i* are labeled as (a) and (b), respectively, where the generalized functions $f_{ej}(\bullet), f_{ei}(\bullet), f_{vjc}(\bullet)$ and $f_{vic}(\bullet)$ describe the kinematic error system and the control velocity for follower *j* and leader *i* , respectively. For both the leader and follower control structures, the kinematic and dynamic control blocks along with the observer block were drawn according to the mathematical equations derived in this work. In the figure, the follower observer structure was drawn according equation (12), the follower kinematic controller was drawn according equation (17), and the torque control input for the follower was drawn according to equation (24). Similarly, the observer, kinematic controller, and dynamic controller for the leader were each drawn according to equations (28), (30), and (33), respectively. Using wireless communication, shown in (c), the leader communicates its orientation to its followers.

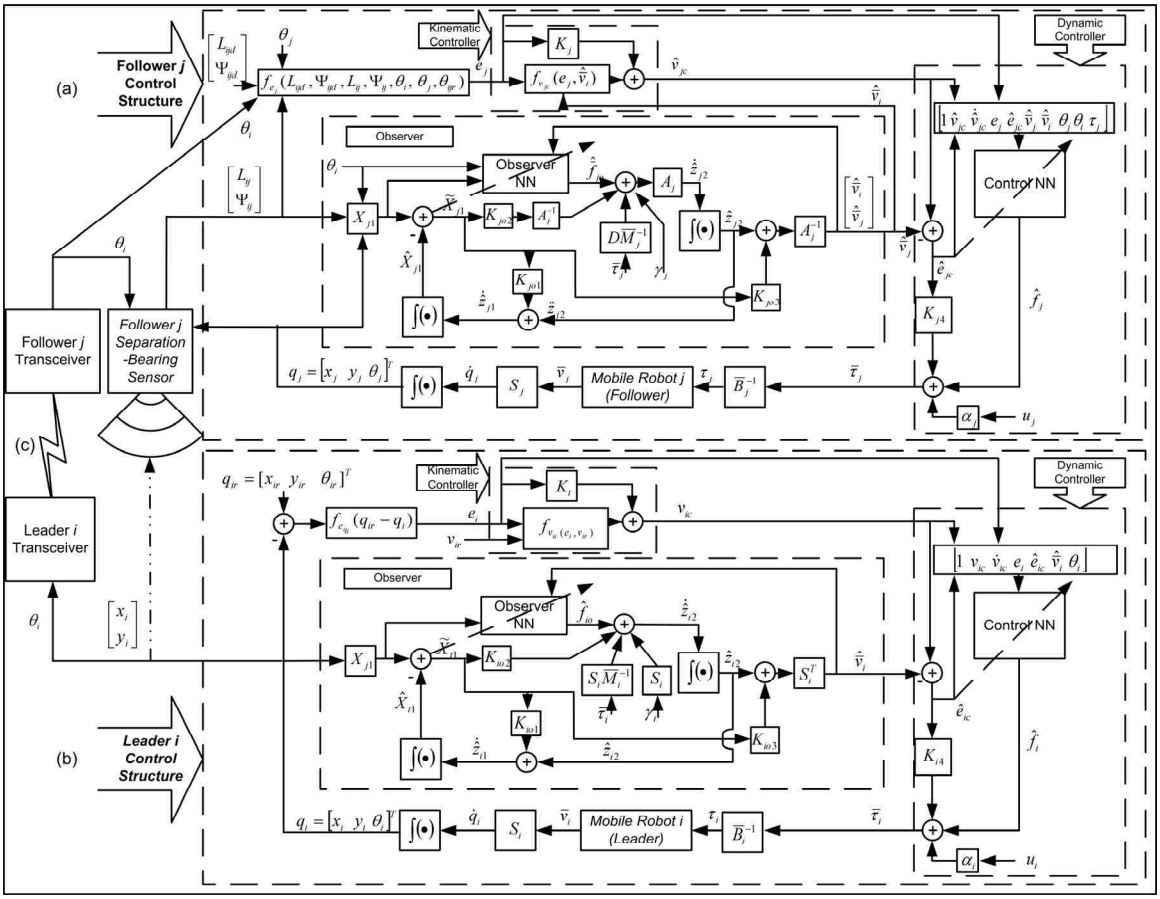


Fig. 2. Formation controller structure.

All other required information is obtained locally by the follower using the NN observer and local sensory information as shown.

***Remark 9*:** To implement the proposed output feedback control scheme, two NN's are required. Although this appears to be a computationally demanding algorithm, our previous work [18] on the control of spark ignition engines has demonstrated that three NN's can be successfully implemented in hardware simultaneously with promising results. In fact, it was found that the total time required to compute the controller calculations was less than $100\,\mu\text{sec}$.

III. LEADER-FOLLOWER OBSTACLE AVOIDANCE

In [11], an obstacle avoidance scheme was proposed that allowed follower j to track its leader while simultaneously avoiding obstacles. To accomplish this, the desired separation and bearing were no longer considered to be constants but were considered to be time varying. In this section, only an overview of the obstacle avoidance scheme is presented. The obstacle avoidance scheme from [11] has to be modified and the stability has to be revisited due to the addition of the observer and output feedback control. In this section, the time varying desired separation and bearing will be denoted as $L_{ijd}(t)$ and $\Psi_{ijd}(t)$ while the constant desired separation and bearing will be written as L_{ijd} and Ψ_{ijd}.

Furthermore, the distance, s_j, from follower j to an obstacle and relative angle of the obstacle, θ_{js}, are considered measurable while the velocity, $\bar{v}_o = [v_o \ \omega_o]^T$, and orientation, θ_o, of the obstacle are unavailable. It is also assumed that leader i utilizes a

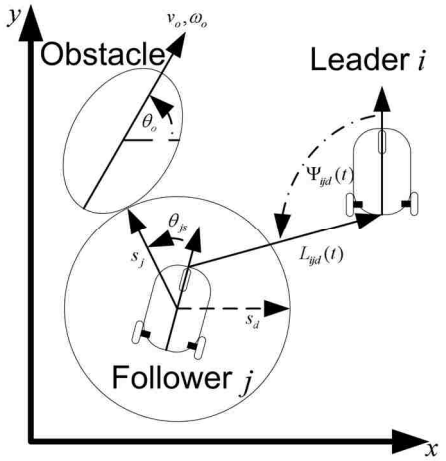


Fig. 3. Obstacle avoidance.

path planning scheme such that by tracking the virtual reference cart described in [12], the lead robot i navigates around any encountered obstacles.

In the configuration shown in Fig. 3, it is desirable that the follower robot j maintains a safe distance, s_d, from all obstacles. Therefore, when the nearest edge of an obstacle is detected at an angle θ_{js} and distance s_j relative to follower j such that $s_j < s_d$, the desired separation and bearing, $L_{ijd}(t)$ and $\Psi_{ijd}(t)$, are modified to ensure the follower is steered away from the obstacle by

$$L_{ijd}(t) = L_{ijd} - \frac{1}{2}K_L\left(\frac{1}{s_j} - \frac{1}{s_d}\right)^2 \operatorname{sgn}(\theta_{js}\Psi_{ijd}), \quad \Psi_{ijd}(t) = \Psi_{ijd} + \frac{1}{2}K_\Psi\left(\frac{1}{s_j} - \frac{1}{s_d}\right)^2 \xi_j \tag{38}$$

where $\xi_j = \operatorname{sgn}(\Psi_{ijd})\operatorname{sgn}(\theta_{js}\Psi_{ijd})$, *sgn* is the signum function, and K_L and K_ψ are design constants. In [11], the error dynamics in the presence of obstacles were found to be

$$\begin{bmatrix}\dot{e}_{jo1}\\ \dot{e}_{jo2}\end{bmatrix} = \begin{bmatrix}\dot{\hat{e}}_{jo1} - v_j + v_i\cos\theta_{ij} + \omega_j e_{jo2} - \omega_i L_{ijd}(t)\sin(\Psi_{ijd}(t) + \theta_{ij})\\ \dot{\hat{e}}_{jo2} - \omega_j e_{jo1} + v_i\sin\theta_{ij} - d_j\omega_j + \omega_i L_{ijd}(t)\cos(\Psi_{ijd}(t) + \theta_{ij})\end{bmatrix} \tag{39}$$

where

$$\begin{aligned}\dot{\hat{e}}_{jo1} &= \dot{\hat{L}}_{ijd}(t)\cos(\Psi_{ijd}(t) + \theta_{ij}) - \dot{\hat{\Psi}}_{ijd}(t)L_{ijd}(t)\sin(\Psi_{ijd}(t) + \theta_{ij})\\ \dot{\hat{e}}_{jo2} &= \dot{\hat{L}}_{ijd}(t)\sin(\Psi_{ijd}(t) + \theta_{ij}) + \dot{\hat{\Psi}}_{ijd}(t)L_{ijd}(t)\cos(\Psi_{ijd}(t) + \theta_{ij})\end{aligned}, \tag{40}$$

$$\dot{\hat{L}}_{ijd}(t) = \operatorname{sgn}(\theta_{js}\Psi_{ijd})K_L\left(\frac{1}{s_j} - \frac{1}{s_d}\right)\frac{1}{s_j^2}\dot{\hat{s}}_j, \quad \dot{\hat{\Psi}}_{ijd}(t) = -\xi_j K_\psi\left(\frac{1}{s_j} - \frac{1}{s_d}\right)\frac{1}{s_j^2}\dot{\hat{s}}_j, \quad \text{and}$$

$\dot{\hat{s}}_j = s_j(t) - s_j(t - \Delta t)$ is the estimate of $\dot{s}_j$ for an arbitrarily small time interval, Δt. The estimate $\dot{\hat{s}}_j$ is used because velocity vector $\bar{v}_o$ and orientation θ_o of the obstacle are not available to follower *j*. Utilizing the observer estimates $\hat{\bar{v}}_i$, the following velocity control inputs for follower robot *j* are proposed to stabilize the error dynamics (39) in the presence of an obstacle

$$\hat{v}_{jco}=\begin{bmatrix}(\hat{v}_i+K_{jo4}\widetilde{X}_{j11})\cos\theta_{ij}-(\hat{\omega}_i+K_{jo4}\widetilde{X}_{j12})L_{ijd}(t)\sin(\Psi_{ijd}(t)+\theta_{ij})+\bar{k}_{j1}e_{jo1}\\((\hat{v}_i+K_{jo4}\widetilde{X}_{j11})\sin\theta_{ij}+(\hat{\omega}_i+K_{jo4}\widetilde{X}_{j12})L_{ijd}(t)\cos(\Psi_{ijd}(t)+\theta_{ij})+\bar{k}_{j2}e_{jo2})/d_j\end{bmatrix}+\begin{bmatrix}\dot{\hat{e}}_{jo1}\\\dot{\hat{e}}_{jo2}/d_j\end{bmatrix} \tag{41}$$

where K_{jo4}, $\widetilde{X}_{j11}$, and $\widetilde{X}_{j12}$ were previously defined in *Section II.D* while $\bar{k}_{j1}$ and $\bar{k}_{j2}$ are positive design constants. Next, define the velocity tracking error in the presence of obstacles similarly to (19) as $\hat{e}_{jco}=\hat{v}_{jco}-\hat{\bar{v}}_j$. Then, substituting (41) into the error dynamics of (39) while observing $\bar{v}_j=\hat{v}_{jco}-\hat{e}_{jco}+\widetilde{\bar{v}}_j$ reveals

$$\begin{bmatrix}\dot{e}_{jo1}\\\dot{e}_{jo2}\end{bmatrix}=\begin{bmatrix}\hat{e}_{jo4}+\omega_j e_{jo2}-\bar{k}_{j1}e_{jo1}\\d_j\hat{e}_{jo5}-\omega_j e_{jo1}-\bar{k}_{j2}e_{jo2}\end{bmatrix}+\begin{bmatrix}\dot{\widetilde{e}}_{jo1}\\\dot{\widetilde{e}}_{jo2}\end{bmatrix}-\left(K_{jo4}\Omega_{jo1}\widetilde{X}_{j1}-\Omega_{jo2}\widetilde{X}_{j2}\right) \tag{42}$$

where Ω_{jo1} and Ω_{jo2} are matrices defined by

$$\Omega_{jo1}=\begin{bmatrix}\cos\theta_{ij} & -L_{ijd}\sin(\Psi_{ijd}+\theta_{ij}) & 0 & 0\\ \sin\theta_{ijr} & L_{ijd}\cos(\Psi_{ijd}+\theta_{ij}) & 0 & 0\end{bmatrix},\Omega_{jo2}=\begin{bmatrix}\cos\theta_{ij} & -L_{ijd}\sin(\Psi_{ijd}+\theta_{ij}) & -1 & 0\\ \sin\theta_{ijr} & L_{ijd}\cos(\Psi_{ijd}+\theta_{ij}) & 0 & -d_j\end{bmatrix},$$

which are shown to be upper bounded such that $\left\|\Omega_{jo1}\right\|_F\le\Omega_{jo1M}$ and $\left\|\Omega_{jo2}\right\|_F\le\Omega_{jo2M}$ for some computable constants Ω_{jo1M} and Ω_{jo2M}, respectively, and $\dot{\widetilde{e}}_{jo1}=\dot{\bar{e}}_{jo1}-\dot{\hat{e}}_{jo1}$ and $\dot{\widetilde{e}}_{jo2}=\dot{\bar{e}}_{jo2}-\dot{\hat{e}}_{jo2}$ are the bounded estimation errors for the estimates in (40) such that $\left\|\dot{\widetilde{e}}_{jo}\right\|=\left\|[\dot{\widetilde{e}}_{jo1}\ \ \dot{\widetilde{e}}_{jo2}]\right\|\le\varepsilon_{jMo}$ with $\varepsilon_{jMo}>0$ being a constant. Whenever there is zero estimation error, $\bar{e}_{jo}=[e_{jo}^T\ \hat{e}_{jco}^T]^T=0$ is an equilibrium point for (42). It is observed that the dynamic controller (24), error system (26), and NN weight update law (27) are valid in the presence of obstacles with $\hat{e}_{jc}$ replaced with $\hat{e}_{jco}$. Next the performance of the follower in the presence of obstacles is introduced.

Theorem 5 (Follower Obstacle Avoidance): Given the nonholonomic system consisting of (2) and (3), the leader follower criterion of (1), and the auxiliary system for

follower j given by (10), let *Assumptions 1-3* hold. In the presence of obstacles, consider the NN observer defined by (14), and let a smooth velocity control input, $\hat{v}_{jco}$, and torque control, $\bar{\tau}_j$, for the follower j be given by (41) and (24), respectively. Select the robustifying term as $u_{jo} = \alpha_{jo}[e_{jo1} \;\; d_j e_{jo2}]^T$ where α_{jo} is a positive constant, and let the update law for the NN observer and controller be given by (16) and (27), respectively. Then, there exists positive constants, $K_L, K_\psi, K_{jo1}, K_{jo2}, K_{jo3}, K_{jo4}, \bar{k}_{j1}, \bar{k}_{j2}, k_{j4}$, and α_{jo} such that the position and velocity tracking errors for the follower j, the NN weight estimation errors $\tilde{W}_j$, the observer estimation errors $\tilde{X}_{j1}, \tilde{X}_{j2}$ and the NN observer weight estimation errors, $\tilde{W}_{jo}$,are all *UUB* in the presence of obstacles.

Please see the Appendix for proof of *Theorem 5*.

IV. Simulation Results

A wedge formation of five identical nonholonomic mobile robots is considered where the trajectory of the leader is the desired formation trajectory, and simulations are carried out in MATLAB under two scenarios. First, in the absence of obstacles, the NN output feedback controllers developed in this work for the leader and its followers is considered with non-ideal sensor measurements. The linear velocity of the leader's reference robot is $v_{ir} = 0.5\, m/s$ while the reference angular velocity is selected as $\omega_{ir} = -0.025\cos(0.5t)\, rad/s$. In the second scenario, obstacles are added in the path of the follower robots, and the obstacle avoidance scheme of *Theorem 5* is demonstrated under both a static and dynamic obstacle environment.

A wedge formation is considered such that follower *j* should track its leader at separation of $L_{ijd} = 2$ meters with a bearing of $\Psi_{ijd} = \pm 120°$ depending on the follower's location, and the formation leader is located at the apex of the wedge as shown in Fig. 4. In the figure, followers 1 and 3 track the leader and followers 2 and 4 track followers 1 and 3, respectively. The following parameters are considered for the leader and its followers: $m = 5kg$, $I = 3kg^2$, $R = .175m$, $r = 0.08m$, and $d = 0.4m$, and the maximum achievable linear velocity of any robot in the formation is assumed to be $2m/s$. The control gains for the leader were selected as $k_{i1} = 10$, $k_{i2} = 5$, $k_{i3} = 4$, $K_{i4} = diag\{25\}$, and $\alpha_i = 0.1$, and for each follower, gains were selected as $k_{j1} = \bar{k}_{j1} = 5$, $k_{j2} = \bar{k}_{j2} = 5$, $k_{j3} = 15$, $K_{j4} = diag\{30\}$, and $\alpha_j = \alpha_{jo} = 0.1$. These controller gains were selected according to the constraints observed in *Theorem 2.* The observer gains for the leader were selected as $K_{io1} = 25$, $K_{io2} = 10$, $K_{io3} = 20$, and for each follower, the observer gains were selected as $K_{jo1} = 25$, $K_{jo2} = 6$, $K_{jo3} = 20$ and $K_{jo4} = 0.03$. The NN parameters for both the leader and each follower were selected as $F_{jo} = F_{io} = 10$,

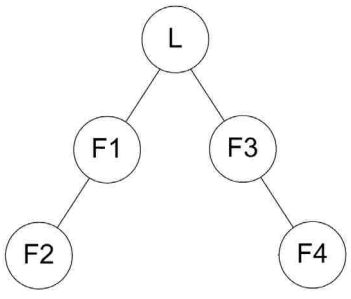
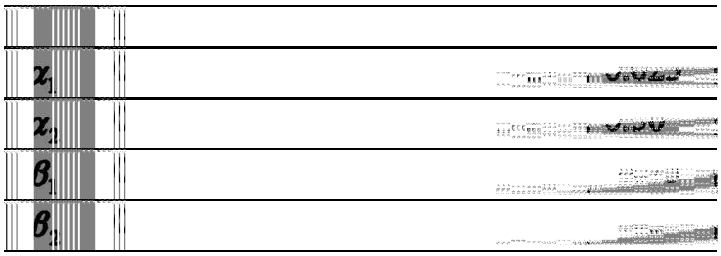


Fig. 4. Formation structure.

$\kappa_{jo1} = \kappa_{io1} = 5$, $\kappa_{jo2} = \kappa_{io2} = 0.4$, $F = F_i = 10$, and $\kappa_1 = \kappa_{i1} = \kappa_2 = \kappa_{i2} = .4$. In addition, five hidden layer neurons were considered in each NN for the leader and each follower. Friction is added to the dynamics of the leader and follower and modeled as $F = [\alpha_1 sign(v) + .\beta_1 v,\ \alpha_2 sign(\omega) + \beta_2 \omega]^T$, where α_k and $\beta_k, k = 1,2$, are the friction coefficients as given in *Table I.*

Remark 10: In the proceeding analysis, $L, F1, F2, F3,$ and $F4$ will be used to denote the *leader, follower 1*, *follower 2, follower 3,* and *follower 4,* respectively.

TABLE I. Friction Coefficients.



A. Scenario I: Obstacle Free Environment

Figure 5 shows the resulting trajectories for the NN output feedback controller of this work where the robots start in the bottom left corner of the figure and travel toward the top right corner of the figure. Examining Fig. 5, the NN output feedback controller achieves and maintains the formation in the presence of unknown dynamics, immeasurable velocities, and sensor noise. In the simulation, noise is generated from a normal distribution with a zero mean and standard deviation of one and is introduced to

the formation measurements as well as the position and orientation measurements. The variance of the noise signals were chosen to represent a 10 percent error in terms of the desired separation and bearing.

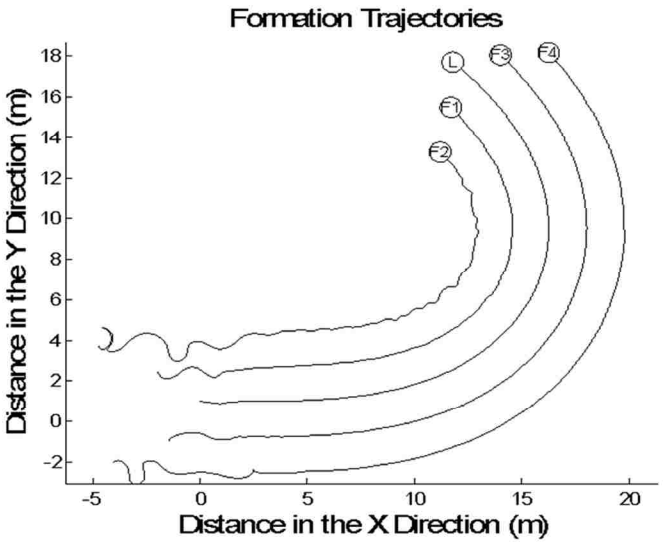


Fig. 5. Robot trajectories.

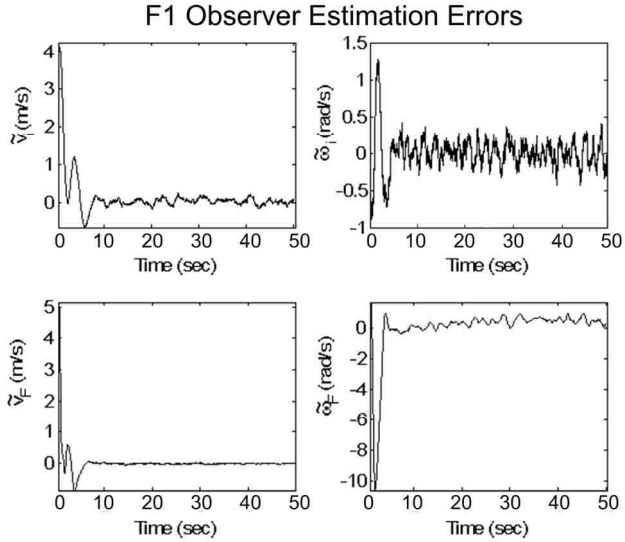


Fig. 6. Observer estimation error of F1.

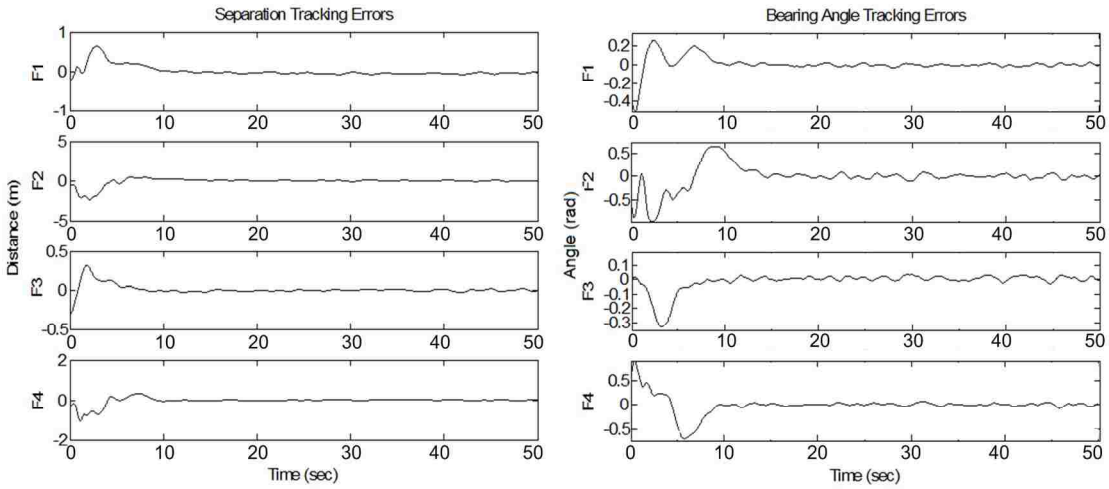


Fig. 7. Formation errors.

Figures 6 and 7 present the observer estimation errors for Follower 1 and all the formation tracking errors, respectively. Examining the observer estimation error plots for Follower 1, it is clear that the robot successfully recovers its linear and angular velocity as well as the linear and angular velocity of its leader with bounded error which is consistent with the theoretical results derived in this work. Furthermore, comparing the convergence of the formation tracking errors for Follower 1 with the convergence of the observer estimation errors for Follower 1, it is apparent that the observer errors converge to the origin before the formation errors. A similar phenomenon was observed for the other robots, but their observer error plots are not shown due to space constraints. The relationship between the convergence of the observer estimates and formation errors demonstrates that accurate velocity information is needed to maintain the formation, and that the proposed observer recovers the immeasurable velocities in a satisfactory manner.

The results presented in Figs. 6 and 7 also support the theoretical conjectures presented in *Theorem 4*.

B. Scenario II: Obstacle Ridden Environment

Now, the wedge formation of five robots is considered in an environment with stationary and moving obstacles, and the parameters and controller gains defined previously along with $K_L = 0.75$ and $K_\psi = 1.5$ were utilized. The gains were selected to be small in order to keep the desired separation and bearing changes small.

Figure 8 depicts the formation trajectories in the presence of both stationary and moving obstacles. The dotted lines represent the path of moving obstacles, and the connected circles denote the obstacles' final positions. Examining the formation trajectories, it is evident that the robots are able to maneuver around the encountered obstacle while simultaneously tracking their leaders with bounded errors as the result of *Theorem 5* suggests. Because the followers on the outside of the formation track the robots in the inner formation, the movements of the robots in the interior of the formation propagate to the followers on the exterior.

Thus, when a robot on the interior of the formation performs an obstacle avoidance maneuver, their movements are mimicked by their followers, as evident in Fig. 8. The observer estimation errors for Follower 1 are shown in Fig. 9, and examining the estimation error plots, one can see that a disturbance occurs at approximately 4 seconds corresponding to the time follower 1 encounters an obstacle. Thus, an encountered obstacle can be viewed as a disturbance to the formation. When the disturbance occurs, the NN quickly adapts, and the estimation errors return to a small bounded region around the origin as per *Theorem 5*.

Fig. 8. Trajectories in the presence of obstacles.

Fig. 9. Observer errors of F1 in the presence of obstacles.

V. Conclusions

A NN output feedback tracking controller for leader-follower based formation control was presented that considers the dynamics of the leader and the follower using backstepping technique and with limited communication between the leader and its followers. Further, the velocity vectors were considered to be immeasurable, and a novel NN observer was designed which allowed the follower robots to not only recover their own velocity vector, but also the velocity vector of their respective leader. It was shown using Lyapunov techniques that the entire formation is UUB in both the presence and absence of obstacles while relaxing the separation principle. Numerical results were presented and the stability of the system was verified. The formation control scheme was also shown to be effective in simulation not only in the presence of measurement noise, but also in both a static and dynamic obstacle environment, and the simulation results verify the theoretical conjecture.

References

[1] Y. Q. Chen and Z. Wang, "Formation control: a review and a new consideration," *Proc. IEEE International Conference on Intelligent Robots and Systems,* pp. 3181-3186, August 2005.

[2] G. L. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis, "Vision-based localization of leader-follower formations," *Proc. IEEE European Control Conference on Decision and Control*, pp 635-640, December 2005.

[3] A. Das, R. Fierro, V. Kumar, J. Ostrowski, J. Spletzer, and C. Taylor, "A vision-based formation control framework," *IEEE Trans. on Robotics and Automation*, vol. 18, pp 813-825, October 2002.

[4] X. Li, J. Xiao, and Z. Cai, "Backstepping based multiple mobile robots formation control," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp 887-892, August 2005.

[5] H. Hsu and A. Liu, "Multi-agent based formation control using a simple representation," *Proc. IEEE International Conference on Networking, Sensing & Control*, Taipei, Taiwan, pp 276-282, March 2004.

[6] J. Shao, G. Xie, J. Yu, and L. Wang, "A tracking controller for motion coordination of multiple mobile robots," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp 783-788, August 2005.

[7] Y. Li and X. Chen, "Dynamic control of multi-robot formation," P*roc. IEEE International Conference on Mechatronics*, Taipei, Taiwan, pp 352-357, July 2005.

[8] M. Breivik, M. Subbotin, and T. Fossen, "Guided formation control for wheeled mobile robots," *Proc. IEEE International Conference on Robotics and Automation*, pp. 1-7, Singapore, December 2006.

[9] K. D. Do, "Formation tracking control of unicycle-type mobile robots with limited sensing ranges," *IEEE Trans. On Control Systems Technology, vol. 16, pp. 527-538, May 2008.*

[10] C. De La Cruz and R. Carelli, "Dynamic modeling and centralized formation control of mobile Robots," *Proc. IEEE Conference on Industrial Electronics,* pp. 3880-3885, November 2006.

[11] T. Dierks and S. Jagannathan, "Neural Network Control of Mobile Robot Formations using RISE Feedback," *IEEE Transactions on Systems, Man, and Cybernetics—Part B,* vol. 39, pp. 332-346, April 2009.

[12] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. on Neural Networks*, vol. 8, pp. 589-600, July 1998.

[13] F.L. Lewis, S. Jagannathan, and A. Yesilderek, "*Neural network control of robot manipulators and nonlinear systems*," Taylor and Francis, London, UK, 1999.

[14] E. Kim, "Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic," *IEEE Trans. on Fuzzy System,* vol. 12, pp. 368-378, June 2004.

[15] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Trans. on Robotics and Automation,* vol. 15, pp. 301-309, April 1999.

[16] T. J. Tarn, A. K. Bejczy, X. Yun, and Z. Li, "Effect of motor dynamics on nonlinear feedback robot arm control," *IEEE Trans. On Robotics and Automation,* vol. 7, pp. 114-122, February 1991.

[17] D. DeVon and T. Bretl, "Kinematic and dynamic control of a wheeled mobile robot," *Proc. IEEE Intl. Conference on Intelligent Robots and System,* pp. 4065-4070, November 2007.

[18] J. Vance, B. Kaul, S. Jagannathan, and J. Drallmeier, "Output feedback controller for operation of spark ignition engines at lean conditions using neural networks," *IEEE Transactions on Control Systems Technology,* vol. 16, pp 214-228, March 2008.

APPENDIX

Proof of Theorem 1: (Follower Robot NN Observer): Consider the following positive definite Lyapunov candidate

$$V_{jo} = \frac{1}{2}\tilde{X}_{j1}^T\tilde{X}_{j1} + \frac{1}{2K_{jo2}}\tilde{X}_{j2}^T\tilde{X}_{j2} + \frac{1}{2K_{jo2}}tr\{\tilde{W}_{jo}^T F_o^{-1}\tilde{W}_{jo}\} + \frac{1}{K_{jo2}}\int_0^t \tilde{X}_{j2}^T\left(c_{j1}I + A_j^{-1}\dot{A}_j\right)\tilde{X}_{j2}ds \quad \text{(A.1)}$$

whose derivative is given by

$$\dot{V}_{jo} = \tilde{X}_{j1}^T\dot{\tilde{X}}_{j1} + \frac{1}{K_{jo2}}\tilde{X}_{j2}^T\dot{\tilde{X}}_{j2} + \frac{1}{K_{jo2}}tr\{\tilde{W}_{jo}^T F_o^{-1}\dot{\tilde{W}}_{jo}\} + \frac{1}{K_{jo2}}\tilde{X}_{j2}^T\left(c_{j1} + A_j^{-1}\dot{A}_j\right)\tilde{X}_{j2}$$

where $c_{j1} > 0$ is a positive constant. Recalling the properties of A_j presented in *Section II.C,* it is clear (A.1) is positive definite. Substitution of the observer error dynamics (14) and the robustifying signal (15) reveals

$$\dot{V}_{jo} = -(K_{jo1} - K_{jo3})\tilde{X}_{j1}^T\tilde{X}_{j1} - \frac{1}{K_{jo2}}\tilde{X}_{j2}^T\left(K_{jo3} - c_{j1}\right)\tilde{X}_{j2} + \tilde{X}_{j1}^T A_j\tilde{X}_{j2} - \tilde{X}_{j2}^T A_j^{-1}\tilde{X}_{j1}$$
$$+ \tilde{X}_{j1}^T\zeta_{jo1} + \frac{1}{K_{jo2}}\tilde{X}_{j2}^T\delta'_{jo} + \frac{1}{K_{jo2}}tr\{\tilde{W}_{jo}^T F_o^{-1}\dot{\tilde{W}}_{jo}\} + \frac{1}{K_{jo2}}\tilde{X}_{j2}^T\tilde{W}_{jo}^T\hat{\sigma}_{jo}.$$

Next, substituting the NN update law (16), adding and subtracting $\left\|\tilde{X}_{j2}\right\|^2\sqrt{N_o}/(2K_{jo2})$ where N_o is the number of hidden layer neurons of the NN observer, and taking the upper bound of $\dot{V}_{jo}$ yields

$$\dot{V}_{jo} \le -(K_{jo1}-K_{jo3})\left\|\tilde{X}_{j1}\right\|^2 - \frac{1}{K_{jo2}}\left(K_{jo3}-c_{j1}-\frac{\sqrt{N_0}}{2}\right)\left\|\tilde{X}_{j2}\right\|^2 + (A_{jM}+A_{jM}^I)\left\|\tilde{X}_{j1}\right\|\left\|\tilde{X}_{j2}\right\| + \left\|\tilde{X}_{j1}\right\|\zeta_{joB} + \frac{1}{K_{jo2}}\left\|\tilde{X}_{j2}\right\|\delta_{jB}$$
$$-\frac{\sqrt{N_0}}{2K_{jo2}}\left\|\tilde{X}_{j2}\right\|^2 + \frac{\sqrt{N_o}}{K_{jo2}}\left\|\tilde{W}_{jo}\right\|_F\left\|\tilde{X}_{j2}\right\| - \frac{\kappa_{o1}A_{jM}^I}{K_{jo2}}\left\|\tilde{X}_{j1}\right\|\left(\left\|\tilde{W}_{jo}\right\|_F^2 - \frac{\sqrt{N_o}+W_{oM}\kappa_{o1}}{\kappa_{o1}}\left\|\tilde{W}_{jo}\right\|_F\right) - \frac{\kappa_{o2}A_{jM}}{K_{jo2}}\left\|\tilde{W}_{jo}\right\|_F^2 + \frac{\kappa_{o2}A_{jM}W_{oM}}{K_{jo2}}\left\|\tilde{W}_{jo}\right\|_F$$

where computable constants A_{jM} and A_{jM}^I are bounds on A_j and its inverse, respectively, described in *Section II.C*. Further, the facts $\left\|\hat{\sigma}_{jo}\right\| \le \sqrt{N_o}$, $\left\|W_{jo}\right\|_F \le W_{oM}$ and $tr\{\tilde{W}_{jo}^T(W_{jo}-\tilde{W}_{jo})\} \le \left\|\tilde{W}_{jo}\right\|_F W_{oM} - \left\|\tilde{W}_{jo}\right\|_F^2$ were utilized with W_{oM} being the known positive constant upper bound of the ideal NN weights. Then, completing the squares with respect to the terms containing $\left\|\tilde{X}_{j1}\right\|\left\|\tilde{W}_{jo}\right\|_F$, $\left\|\tilde{X}_{j2}\right\|\left\|\tilde{W}_{jo}\right\|_F$, and $\left\|\tilde{X}_{j1}\right\|\left\|\tilde{X}_{j2}\right\|$, $\dot{V}_{jo}$ is rewritten as

$$\dot{V}_{jo} \le -\left(K_{jo1}-K_{jo3}-\frac{1}{2}\right)\left\|\tilde{X}_{j1}\right\|^2 - \frac{1}{K_{jo2}}\left(K_{jo3}-\Xi_{jo}\right)\left\|\tilde{X}_{j2}\right\|^2 + \left\|\tilde{X}_{j1}\right\|\left(\zeta_{joB}+\Phi_{jo}\right) + \frac{1}{K_{jo2}}\left\|\tilde{X}_{j2}\right\|\delta_{jB}$$
$$-\left(\frac{\kappa_{o2}A_{jM}}{K_{jo2}} - \frac{\sqrt{N_0}}{2K_{jo2}}\right)\left\|\tilde{W}_{jo}\right\|_F^2 + \frac{\kappa_{o2}A_{jM}W_{oM}}{K_{jo2}}\left\|\tilde{W}_{jo}\right\|_F$$

where $\Phi_{jo} = A_{jM}^I\left(\sqrt{N_o}+W_{oM}\kappa_{o1}\right)^2/(4K_{jo2}\kappa_{o1})$ and $\Xi_{jo} = c_{j1} + \sqrt{N_0}/2 + (A_{jM}+A_{jM}^I)^2/2$ are positive constants. Finally, we complete the square with respect to $\left\|\tilde{X}_{j1}\right\|, \left\|\tilde{X}_{j2}\right\|$ and $\left\|\tilde{W}_{jo}\right\|_F$ to get

$$\dot{V}_{jo} \le -\frac{1}{2}\left(K_{jo1}-K_{jo3}-\frac{1}{2}\right)\left\|\tilde{X}_{j1}\right\|^2 - \frac{1}{2K_{jo2}}\left(K_{jo3}-\Xi_{jo}\right)\left\|\tilde{X}_{j2}\right\|^2 - \frac{1}{2K_{jo2}}\left(\kappa_{o2}A_{jM}-\frac{\sqrt{N_0}}{2}\right)\left\|\tilde{W}_{jo}\right\|_F^2 + \eta_{jo} \quad \text{(A.2)}$$

where η_{jo} is a positive constant defined as

$$\eta_{jo} = \frac{1}{2}\left(\frac{\left(\zeta_{joB}+\Phi_{jo}\right)^2}{\left(K_{jo1}-K_{jo3}-1/2\right)} + \frac{\delta_{jB}'^2}{\left(K_{jo2}\left(K_{jo3}-\Xi_{jo}\right)\right)} + \frac{\left(\kappa_{o2}A_{jM}W_{oM}\right)^2}{\left(K_{jo2}\left(\kappa_{o2}A_{jM}/K_{jo2}-\sqrt{N_0}/(2K_{jo2})\right)\right)}\right).$$

The first derivative (A.2) is guaranteed to be less than zero as long as $K_{jo1} > K_{jo3} + 1/2$, $K_{jo3} > \Xi_{jo}$, $\kappa_{o2} > \sqrt{N_0}/(2A_{jM})$ and one of the following conditions holds

$$\left\|\tilde{X}_{j1}\right\| \geq \sqrt{\frac{2\eta_{jo}}{\left(K_{jo1} - K_{jo3} - 1/2\right)}} \quad \text{or} \quad \left\|\tilde{X}_{j2}\right\| \geq \sqrt{\frac{2K_{jo2}\eta_{jo}}{\left(K_{jo3} - \Xi_{jo}\right)}} \quad \text{or} \quad \left\|\tilde{W}_{jo}\right\|_F \geq \sqrt{\frac{2K_{jo2}\eta_{jo}}{\left(\kappa_{o2}A_{jM} - \sqrt{N_0}/2\right)}} \tag{A.3}$$

Observing the definition of η_{jo} above, it is clear that η_{jo} can be made arbitrarily small by increasing the design parameters K_{jo1}, K_{jo2}, K_{jo3}, κ_{o1} and κ_{o2}. Similarly, the error bounds in (A.3), are reduced by increasing the design parameters K_{jo1}, K_{jo3} and κ_{o2}. Thus $\dot{V}_{jo}$ is negative outside of a compact set, and therefore it can be concluded that the observer estimation errors $\tilde{X}_{j1}$ and $\tilde{X}_{j2}$ and NN observer weight estimation errors $\tilde{W}_{jo}$ are all UUB.

***Proof of Theorem 2: (Follower Output Feedback Control)*:** Consider the following Lyapunov function candidate

$$\overline{V}_j = V_j' + V_{jo} \tag{A.4}$$

where V_{jo} is defined in (A.1), $V_j' = V_j + V_{jNN}$, $V_j = \frac{k_{j2}}{2}(e_{j1}^2 + e_{j2}^2) + \frac{d_j k_{j3}}{2}e_{j3}^2$ and $V_{jNN} = \frac{1}{2\alpha_j}\hat{e}_{jc}^T\overline{M}_j\hat{e}_{jc} + \frac{1}{2\alpha_j}tr\{\tilde{W}_j^T F^{-1}\tilde{W}_j\}$. Differentiating V_j and substituting the closed-loop kinematic error dynamics (20) reveals

$$\begin{aligned}\dot{V}_j = &-k_{j2}k_{j1}e_{j1}^2 - k_{j2}^2e_{j2}^2 - k_{j3}^2e_{j3}^2 + 2k_{j2}e_{j2}v_i\sin\left(\frac{e_{j3}}{2}\right)\cos\left(\theta_i - \frac{\theta_{jr} + \theta_j}{2}\right) - k_{j3}k_{j2}e_{j2}e_{j3} \\ &+ k_{j2}e_{j1}\hat{e}_{j4} + d_j\left(k_{j2}e_{j2} + k_{j3}e_{j3}\right)\hat{e}_{j5} - e_j^T(K_{jo4}\Omega_{j1}\tilde{X}_{j1} - \Omega_{j2}\tilde{X}_{j2})\end{aligned}$$

where Ω_{j1} and Ω_{j2} are matrices defined as

$$\Omega_{j1}=\begin{bmatrix} k_{j2}\cos\theta_{ij} & -k_{j2}L_{ijd}\sin(\Psi_{ijd}+\theta_{ij}) & 0 & 0 \\ k_{j2}\sin(\theta_{ijr}) & k_{j2}L_{ijd}\cos(\Psi_{ijd}+\theta_{ij}) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$\Omega_{j2}=\begin{bmatrix} k_{j2}\cos\theta_{ij} & -k_{j2}L_{ijd}\sin(\Psi_{ijd}+\theta_{ij}) & -k_{j2} & 0 \\ k_{j2}\sin(\theta_{ijr}) & k_{j2}L_{ijd}\cos(\Psi_{ijd}+\theta_{ij}) & 0 & -k_{j2}d_j \\ 0 & 0 & 0 & -d_jk_{j3} \end{bmatrix}.$$

It is important to notice that the matrices are bounded by $\|\Omega_{j1}\|_F \le \Omega_{j1M}$ and $\|\Omega_{j2}\|_F \le \Omega_{j2M}$.

Noting that $|\sin(e_{j3}/2)| \le |e_{j3}|$ for all $e_{j3} \in [-\pi,\pi]$, $\dot{V}_j$ takes the form of

$$\dot{V}_j \le -k_{j2}k_{j1}e_{j1}^2 - k_{j2}^2e_{j2}^2 - k_{j3}^2e_{j3}^2 + k_{j2}|e_{j2}e_{j3}|(k_{j3}+2v_{i\max}) + k_{j2}e_{j1}\hat{e}_{j4} + d_j(k_{j2}e_{j2}+k_{j3}e_{j3})\hat{e}_{j5} - e_j^T(K_{jo4}\Omega_{j1}\tilde{X}_{j1} - \Omega_{j2}\tilde{X}_{j2})$$

where $|v_i| \le v_{i\max}$ [12] was utilized. In the next step, it is desired to select k_{j3} such that $(k_{j3}+2v_{i\max}) < 2k_{j3}$, and for any $\varepsilon_{vi} > 0$, selecting $k_{j3} = 2v_{i\max} + \varepsilon_{vi}$ ensures this inequality holds. Specifically, we select $\varepsilon_{vi} = 2\varepsilon_{k3}k_{j3}$ where $\varepsilon_{k3} \in (0,1/2)$ so that

$$k_{j3} = 2v_{i\max}/(1-2\varepsilon_{k3}). \tag{A.5}$$

Selecting k_{j3} as in (A.5) allows $\dot{V}_j$ to be written as

$$\begin{aligned} \dot{V}_j &\le -k_{j2}k_{j1}e_{j1}^2 - k_{j2}^2e_{j2}^2 - k_{j3}^2e_{j3}^2 + 2k_{j2}k_{j3}|e_{j2}e_{j3}|(1-\varepsilon_{k3}) + [k_{j2}e_{j1} \;\; d_j(k_{j2}e_{j2}+k_{j3}e_{j3})]\hat{e}_{jc} - e_j^T(K_{jo4}\Omega_{j1}\tilde{X}_{j1} - \Omega_{j2}\tilde{X}_{j2}) \quad \text{(A.6)} \\ &= -k_{j2}k_{j1}e_{j1}^2 - \varepsilon_{k3}k_{j2}^2e_{j2}^2 - \varepsilon_{k3}k_{j3}^2e_{j3}^2 - (1-\varepsilon_{k3})(k_{j3}|e_{j3}| - k_{j2}|e_{j2}|)^2 + [k_{j2}e_{j1} \;\; d_j(k_{j2}e_{j2}+k_{j3}e_{j3})]\hat{e}_{jc} - e_j^T(K_{jo4}\Omega_{j1}\tilde{X}_{j1} - \Omega_{j2}\tilde{X}_{j2}) \end{aligned}$$

Next, differentiating V_{jNN}, substituting the closed loop dynamics (26), the tuning law (27), and applying the *skew symmetric property* reveals

$$\dot{V}_{jNN} = -\frac{1}{\alpha_j}\hat{e}_{jc}^TK_{j4}\hat{e}_{jc} - \frac{1}{\alpha_j}\hat{e}_{jc}^Tu_j + \frac{\kappa_1}{\alpha_j}\|\hat{\bar{e}}_j\|tr\{\tilde{W}_j^T(W_j-\tilde{W}_j)\} + \frac{\kappa_2}{\alpha_j}tr\{\tilde{W}_j^T(W_j-\tilde{W}_j)\} + \frac{1}{\alpha_j}\hat{e}_{jc}^T\zeta_j \quad \text{(A.7)}$$

Now, combining (A.6) and (A.7), substituting the robustifying term (25), and recalling $\hat{\bar{e}}_j = [e_j^T \;\; \hat{e}_{jc}^T]^T$ along with noting $tr\{\widetilde{W}_j^T(W_j - \widetilde{W}_j)\} \le \left\|\widetilde{W}_j\right\|_F W_M - \left\|\widetilde{W}_j\right\|_F^2$ and *Assumption 3* reveals

$$\dot{V}_j' \le -\Gamma_j \left\|\hat{\bar{e}}_j\right\|^2 + \left\|\hat{\bar{e}}_j\right\| \left(\frac{\kappa_1}{\alpha_j} \left(\left\|\widetilde{W}_j\right\|_F W_M - \left\|\widetilde{W}_j\right\|_F^2 \right) + \frac{1}{\alpha_j}\zeta_{jM} + K_{jo4}\Omega_{j1M}\left\|\widetilde{X}_{j1}\right\| + \Omega_{j2M}\left\|\widetilde{X}_{j2}\right\| \right) + \frac{\kappa_2}{\alpha_j}\left(\left\|\widetilde{W}_j\right\|_F W_M - \left\|\widetilde{W}_j\right\|_F^2 \right) \}$$

where $\Gamma_j = \min\left\{k_{j2}k_{j1}, \varepsilon_{k3}k_{j2}^2, \varepsilon_{k3}k_{j3}^2, k_{j4}/\alpha_j\right\}$. Next, adding and subtracting $\left\|\hat{\bar{e}}_j\right\|^2 \left(\frac{K_{jo4}}{2} + \frac{K_{jo2}}{2} \right)$, $\left\|\widetilde{X}_{j1}\right\|^2 K_{jo4}\Omega_{j1M}^2/2$ and $\left\|\widetilde{X}_{j2}\right\|^2 \Omega_{j2M}^2 \big/ (2K_{jo2})$, along with completing the squares with respect to the terms containing to $\left\|\widetilde{W}_j\right\|_F \left\|\hat{\bar{e}}_j\right\|$, $\left\|\widetilde{X}_{j1}\right\| \left\|\hat{\bar{e}}_j\right\|$, and $\left\|\widetilde{X}_{j2}\right\| \left\|\hat{\bar{e}}_j\right\|$, $\dot{V}_j'$ can be rewritten as

$$\begin{aligned} \dot{V}_j' \le & -\frac{K_{jo4}}{2}\left(\left\|\hat{\bar{e}}_j\right\| - \Omega_{j1M}\left\|\widetilde{X}_{j1}\right\| \right)^2 - \frac{K_{jo2}}{2}\left(\left\|\hat{\bar{e}}_j\right\| - \left\|\widetilde{X}_{j2}\right\|\Omega_{j2M}/K_{jo2} \right)^2 - \frac{\kappa_2}{\alpha_j}\left\|\widetilde{W}_j\right\|_F^2 + \frac{\kappa_2 W_M}{\alpha_j}\left\|\widetilde{W}_j\right\|_F \\ & - \left(\Gamma_j - \frac{K_{jo4}}{2} - \frac{K_{jo2}}{2} \right)\left\|\hat{\bar{e}}_j\right\|^2 - \frac{\left\|\hat{\bar{e}}_j\right\|\kappa}{\alpha_j}\left(\left\|\widetilde{W}_j\right\|_F - \frac{W_M}{2} \right)^2 + \left(\frac{W_M^2\kappa_1 + 4\zeta_{jM}}{4\alpha_j} \right)\left\|\hat{\bar{e}}_j\right\| \\ & + \left\|\widetilde{X}_{j1}\right\|^2 \frac{K_{jo4}\Omega_{j1M}^2}{2} + \left\|\widetilde{X}_{j2}\right\|^2 \frac{\Omega_{j2M}^2}{2K_{jo2}} \end{aligned} \tag{A.8}$$

Finally, combining (A.8) and (A.2) and completing the squares with respect to $\left\|\hat{\bar{e}}_j\right\|$ and $\left\|\widetilde{W}_j\right\|_F$ reveals

$$\begin{aligned} \dot{\bar{V}}_j \le & -\frac{\bar{\Gamma}_j}{2}\left\|\hat{\bar{e}}_j\right\|^2 - \frac{1}{2}\left(K_{jo1} - K_{jo3} - \frac{1}{2} - \frac{K_{jo4}\Omega_{j1M}^2}{2} \right)\left\|\widetilde{X}_{j1}\right\|^2 - \frac{1}{2K_{jo2}}\left(K_{jo3} - \Xi_{jo} - \frac{\Omega_{j2M}^2}{2} \right)\left\|\widetilde{X}_{j2}\right\|^2 \\ & - \frac{\kappa_2}{2\alpha_j}\left\|\widetilde{W}_j\right\|_F^2 - \frac{1}{2K_{jo2}}\left(\kappa_{o2}A_{jM} - \frac{\sqrt{N_0}}{2} \right)\left\|\widetilde{W}_{jo}\right\|_F^2 + \eta_j \end{aligned} \tag{A.9}$$

where $\overline{\Gamma}_j = \Gamma_j - K_{jo4}/2 - K_{jo2}/2$ and $\eta_j = \eta_{jo} + \frac{\left(W_M^2 \kappa_1 + 4\zeta_{jM}\right)^2}{32\alpha_j^2 \overline{\Gamma}_j} + \frac{\kappa_2 W_M^2}{2\alpha_j}$ with η_{jo} defined earlier. Recall that η_{jo} can be made arbitrarily small by increasing the design parameters K_{jo1}, K_{jo2}, K_{jo3}, κ_{o1} and κ_{o2}. Observing the definition of η_j above, increasing $\overline{\Gamma}_j$ and α_j allows η_j to be further reduced. Therefore, (A.9) is guaranteed to be negative when $\Gamma_j > K_{jo4}/2 + K_{jo2}/2$, $K_{jo1} > K_{jo3} + 1/2 + K_{jo4}\Omega_{j1M}^2/2$, $K_{jo3} > \Xi_{jo} + \Omega_{j2M}^2/2$ and $\kappa_{o2} > \sqrt{N_0}/(2A_{jM})$ *one* of the following inequalities is satisfied

$$\left.\begin{array}{ll} \left\|\hat{\bar{e}}_j\right\| > \sqrt{\frac{2\eta_j}{\overline{\Gamma}_j}} \text{ or } \left\|\tilde{X}_{j1}\right\| > \sqrt{\frac{2\eta_j}{\left(K_{jo1} - K_{jo3} - 1/2 - K_{jo4}\Omega_{j1M}^2/2\right)}} & \text{or } \left\|\tilde{X}_{j2}\right\| > \sqrt{\frac{2K_{jo2}\eta_j}{\left(K_{jo3} - \Xi_{jo} - \Omega_{j2M}^2/2\right)}} \\ \text{or } \left\|\tilde{W}_j\right\|_F > \sqrt{\frac{2\alpha_j\eta_j}{\kappa_2}} & \text{or } \left\|\tilde{W}_{jo}\right\|_F > \sqrt{\frac{2K_{jo2}\eta_j}{\left(\kappa_{o2}A_{jM} - \sqrt{N_0}/2\right)}} \end{array}\right\} \quad \text{(A.10)}$$

Note that the bound on $\left\|\hat{\bar{e}}_j\right\|$ can be made arbitrarily small by increasing $\overline{\Gamma}_j$ through the design parameters. Therefore, $\dot{\bar{V}}_j < 0$ outside of the compact set, and it can be concluded that the kinematic and dynamic tracking errors $\hat{\bar{e}}_j$, the NN weight estimation errors $\tilde{W}_j$, the observer estimation errors $\tilde{X}_{j1}$, $\tilde{X}_{j2}$ and the NN observer weight estimation errors, $\tilde{W}_{jo}$ are all UUB.

Proof of Theorem 3 (Leader Robot Control): Consider the following positive definite Lyapunov candidate

$$\bar{V}_i = V_i + V_{io} \quad \text{(A.11)}$$

where $V_i = \frac{1}{2}(e_{i1}^2 + e_{i2}^2) + \frac{1}{k_{i2}}(1 - \cos e_{i3}) + \frac{1}{2\alpha_i}\hat{e}_{ic}^T \overline{M}_i \hat{e}_{ic} + \frac{1}{2\alpha_i} tr\{\tilde{W}_i^T F_i^{-1} \tilde{W}_i\}$ and $V_{io} = \frac{1}{2}\tilde{X}_{i1}^T K_{io2} \tilde{X}_{i1} + \frac{1}{2}\tilde{X}_{i2}^T \tilde{X}_{i2} + \frac{1}{2} tr\{\tilde{W}_{io}^T F_{io}^{-1} \tilde{W}_{io}\}$. Differentiating V_i and substituting the closed

loop kinematic and dynamic error systems (32) and (35), respectively, as well as the NN weight update law (37) for control reveals

$$\dot{V}_i \le -k_{i1}e_{i1}^2 - \frac{k_{i3}(v_{ir}+1)}{k_{i2}}\sin^2 e_{i3} - \frac{1}{\alpha_i}\hat{e}_{ic}^T K_{i4}\hat{e}_{ic} + \frac{1}{\alpha_i}\left(\kappa_{i1}\left\|\hat{\bar{e}}_i'\right\| + \kappa_{i2}\right)\left(\left\|\widetilde{W}_i\right\|_F W_{iM} - \left\|\widetilde{W}_i\right\|_F^2\right) + \frac{\zeta_{iM}}{\alpha_i}\left\|\hat{e}_{ic}\right\| + \left|e_{i1}\ \sin e_{i3}\right|\left\|\Omega_i\right\|_F\left\|\widetilde{X}_{i2}\right\|$$

where $\Omega_i = [1\ \ 0;0\ \ 1/k_{i2}]$, and by selecting $k_{i2} \ge 1$, clearly $\left\|\Omega_i\right\|_F \le \sqrt{2}$. Completing the square with respect to the terms containing $\left\|\hat{\bar{e}}_i'\right\|\left\|\widetilde{W}_i\right\|_F$ and applying the bounds of *Assumption 3* with $\left\|W_i\right\|_F \le W_{iM}$ for a known positive constant W_{iM}, $\dot{V}_i$ can be rewritten as

$$\dot{V}_i \le -\Gamma_i\left\|\hat{\bar{e}}_i'\right\|^2 - \frac{\left\|\hat{\bar{e}}_i'\right\|}{\alpha_i}\kappa_{i1}\left(\left\|\widetilde{W}_i\right\|_F - \frac{W_{iM}}{2}\right)^2 + \left\|\hat{\bar{e}}_i'\right\|\left(\frac{\kappa_{i1}W_{iM}^2 + 4\zeta_{iM}}{4\alpha_i}\right) + \sqrt{2}\left\|\widetilde{X}_{i2}\right\|\left\|\hat{\bar{e}}_i'\right\| - \frac{\kappa_{i2}}{\alpha_i}\left(\left\|\widetilde{W}_i\right\|_F^2 - W_{iM}\left\|\widetilde{W}_i\right\|_F\right)$$

where $\Gamma_i = \min\{k_{i1}, k_{i3}/k_{i2}, k_{i4}/\alpha_i\}$. Now, adding and subtracting $\left\|\hat{\bar{e}}_i'\right\|^2$ and $\left\|\widetilde{X}_{i2}\right\|^2/2$, and completing the square with respect to $\left\|\hat{\bar{e}}_i'\right\|$ and $\left\|\widetilde{W}_i\right\|_F$ reveals

$$\dot{V}_i \le -\frac{(\Gamma_i - 1)}{2}\left\|\hat{\bar{e}}_i'\right\|^2 - \frac{\kappa_{i2}}{2\alpha_i}\left\|\widetilde{W}_i\right\|_F^2 + \frac{\left\|\widetilde{X}_{i2}\right\|^2}{2} + \left(\frac{\kappa_{i1}W_{iM}^2 + 4\zeta_{iM}}{32\alpha_i^2(\Gamma_i - 1)}\right)^2 + \frac{\kappa_{i2}W_{iM}^2}{2\alpha_i} \qquad \text{(A.12)}$$

Next, differentiating V_{io} and substitution of the observer error dynamics (29) and NN tuning law (36) reveals

$$V_{io} \le -\frac{K_{io2}(K_{io1} - K_{io3})}{2}\left\|\widetilde{X}_{i1}\right\|^2 - \frac{1}{2}\left(K_{io3} - \frac{N_{io}(2+d_i)}{2\kappa_{io2}}\right)\left\|\widetilde{X}_{i2}\right\|^2 - \frac{\kappa_{io2}}{4}\left\|\widetilde{W}_{io}\right\|_F^2 + \eta_{io} \qquad \text{(A.13)}$$

where the facts $\left\|S_i\right\|_F \le \sqrt{2+d_i}$ and $\left\|W_{io}\right\|_F \le W_{ioM}$ for a known positive constant W_{ioM} were utilized and η_{io} is a defined as

$$\eta_{io} = \frac{\left(\left(\sqrt{N_{io}} + \kappa_{io1}W_{ioM}\right)^2 + 4\zeta_{ioB}\kappa_{io1}K_{io2}\right)^2}{32\kappa_{io1}^{\ 2}K_{io2}(K_{io1} - K_{io3})} + \frac{(2+d_i)\delta_{io3B}^2}{2\left(K_{jo3} - N_{io}(2+d_i)/(2\kappa_{o2})\right)} + \kappa_{io2}W_{ioM}^2 .$$

Remark A.1: The steps taken in the formulation of (A.13) are identical to the formulation of (A.2) and therefore, are not repeated here.

Finally, combining (A.12) and (A.13), $\dot{\overline{V}}_i$ can be written as

$$\dot{V}_i \le -\frac{(\Gamma_i - 1)}{2}\left\|\hat{\tilde{e}}_i'\right\|^2 - \frac{\kappa_{i2}}{2\alpha_i}\left\|\widetilde{W}_i\right\|_F^2 - \frac{K_{io2}(K_{io1} - K_{io3})}{2}\left\|\widetilde{X}_{i1}\right\|^2 - \frac{1}{2}\left(K_{io3} - \frac{N_{io}(2+d_i)}{2\kappa_{io2}} - 1\right)\left\|\widetilde{X}_{i2}\right\|^2 - \frac{\kappa_{io2}}{4}\left\|\widetilde{W}_{io}\right\|_F^2 + \eta_i \quad \text{(A.14)}$$

where $\eta_i = \eta_{io} + \left(\frac{\kappa_{i1}W_{iM}^2 + 4\zeta_{iM}}{32\alpha_i^2(\Gamma_i - 1)}\right)^2 + \frac{\kappa_{i2}W_{iM}^2}{2\alpha_i}$, and (A.14) is less than zero as long as the control parameters are selected as $K_{io1} > K_{io3}$, $K_{io3} > N_{io}(2+d_i)/(2\kappa_{io2}) + 1$, $\Gamma_i > 1$ and *one* of the following inequalities holds

$$\left.\begin{array}{lll} \left\|\hat{\tilde{e}}_i'\right\| > \sqrt{\frac{2\eta_i}{(\Gamma_i - 1)}} & \text{or} \quad \left\|\widetilde{X}_{i1}\right\| > \sqrt{\frac{2\eta_i}{K_{io2}(K_{io1} - K_{io3})}} & \text{or} \quad \left\|\widetilde{X}_{i2}\right\| > \sqrt{\frac{2\eta_i}{K_{io3} - (N_{io}(2+d_i))/(2\kappa_{io2}) - 1}} \\ \text{or} \quad \left\|\widetilde{W}_{io}\right\|_F > \sqrt{\frac{4\eta_i}{\kappa_{io2}}} & & \text{or} \quad \left\|\widetilde{W}_i\right\|_F > \sqrt{\frac{2\alpha_i\eta_i}{\kappa_{i2}}} \end{array}\right\} \quad \text{(A.15)}$$

Thus, (A.14) is negative outside of a compact set, and it can be concluded that the observer estimation errors $\widetilde{X}_{i1}, \widetilde{X}_{i2}$, the position, orientation and velocity tracking errors $\hat{\tilde{e}}_i$ and the NN weight estimation errors of the observer and the dynamic controller, $\widetilde{W}_{io}, \widetilde{W}_i$, respectively, are all *UUB*. Observing the bounds in (A.15), one can see that the system errors can be minimized by increasing controller gain parameters.

Proof of Theorem 4 (Formation Stability): Consider the following Lyapunov candidate

$$V_{ij} = \sum_1^N \overline{V}_j + \overline{V}_i$$

where $\overline{V}_j$ and $\overline{V}_i$ are defined by (A.4) and (A.11), respectively. In *Theorem 2*, sufficient conditions where found to ensure $\dot{\overline{V}}_j \le 0$; thus, it follows $\sum_1^N \dot{\overline{V}}_j \le 0$ if the provisions of *Theorem 2* are satisfied for every $j = 1,2,3,...N$. Similarly, *Theorem 3* derived adequate conditions to ensure $\dot{\overline{V}}_i$ is less than zero. Combining these results reveals $\dot{V}_{ij}$ is less than zero outside of a compact set, and the stability of the formation follows.

The stability of a formation for the case when follower *j* becomes a leader to follower *j+1* follows directly from *Theorem 2* and the positive definite Lyapunov candidate $V_j'' = \sum_j^{j+1} \overline{V}_j$ where $\overline{V}_j$ is defined in (A.4). In this case, follower *j* becomes the reference for follower *j+1,* and thus the dynamics of follower *j* must be considered by follower *j+1*. Since the dynamics of follower *j* incorporates the dynamics of leader *i,* follower *j+1* inherently brings in the dynamics of leader *i* by considering the dynamics of follower *j*.

Proof of Theorem 5 (Follower Obstacle Avoidance): Consider the Lyapunov candidate

$$\overline{V}_j' = \overline{V}_{jo} + \overline{V}_{jNN} + V_{jo} \tag{A.16}$$

with $\overline{V}_{jo} = \frac{1}{2}(e_{jo1}^2 + e_{jo2}^2)$, $\overline{V}_{jNN} = \frac{1}{2\alpha_{jo}} \hat{e}_{jco}^T \overline{M}_j \hat{e}_{jco} + \frac{1}{2\alpha_{jo}} tr\{\tilde{W}_j^T F^{-1} \tilde{W}_j\}$ and V_{jo} defined in (A.1). Differentiating $\overline{V}_j'$ and applying similar steps and justifications used to formulate (A.9) reveals

$$\dot{\overline{V}}_j' \le -\frac{\overline{\Gamma}_{jo}}{2}\left\|\hat{\tilde{e}}_{jco}\right\|^2 - \frac{1}{2}\left(K_{jo1} - K_{jo3} - \frac{1}{2} - \frac{K_{jo4}\Omega_{jo1M}^2}{2}\right)\left\|\tilde{X}_{j1}\right\|^2 - \frac{1}{2K_{jo2}}\left(K_{jo3} - \Xi_{jo} - \frac{\Omega_{jo2M}^2}{2K_{jo2}}\right)\left\|\tilde{X}_{j2}\right\|^2 - \frac{\kappa_2}{2\alpha_{jo}}\left\|\tilde{W}_j\right\|_F^2 - \frac{1}{2K_{jo2}}\left(\kappa_{o2}A_{jM} - \frac{\sqrt{N_0}}{2}\right)\left\|\tilde{W}_{jo}\right\|_F^2 + \overline{\eta}_{jo} \tag{A.17}$$

where $\overline{\eta}_{jo}$ is a computable constant defined as

$$\overline{\eta}_{jo} = \frac{1}{2}\left(\frac{(4\varepsilon_{jMo}\alpha_{jo} + 4\zeta_{jM} + W_M^2\kappa_1)^2}{32\alpha_{jo}^2\overline{\Gamma}_{jo}} + \frac{\kappa_2 W_M^2}{\alpha_{jo}} + \frac{\left(\zeta_{joB} + \Phi_{jo}\right)^2}{\left(K_{jo1} - K_{jo3} - 1/2\right)} + \frac{\delta_{jB}'^2}{\left(K_{jo2}\left(K_{jo3} - \Xi_{jo}\right)\right)} + \frac{(\kappa_{o2}A_{jM}W_{oM})^2}{\left(K_{jo2}\left(\kappa_{o2}A_{jM}/K_{jo2} - \sqrt{N_0}/(2K_{jo2})\right)\right)}\right)$$

where $\overline{\Gamma}_{jo} = \Gamma_{jo} - K_{jo4}/2 - K_{jo2}/2$ and $\Gamma_{j0} = \min\left\{\overline{k}_{j1}, \overline{k}_{j2}, k_{j4}/\alpha_{jo}\right\}$. The inequality (A.17) is less than zero provided $\Gamma_{jo} > K_{jo4}/2 + K_{jo2}/2$, $K_{jo1} > K_{jo3} + 1/2 + K_{jo4}\Omega_{j1M}^2/2$, $K_{jo3} > \Xi_{jo} + \Omega_{j2M}^2/2$ and $\kappa_{o2} > \sqrt{N_0}/(2A_{jM})$ and one of the inequalities in (A.10) holds with $\hat{\tilde{e}}_j$, α_j, η_j, and $\overline{\Gamma}_j$ replaced by $\hat{\tilde{e}}_{jo}$, α_{jo}, η_{jo}, and $\overline{\Gamma}_{jo}$, respectively. Therefore, $\dot{\overline{V}}_j' < 0$ outside of a compact set, demonstrating the stability of $\hat{\tilde{e}}_{jo}$, consisting of the position and estimated velocity tracking errors, the NN weight estimation errors $\tilde{W}_j$, the observer estimation errors $\tilde{X}_{j1}, \tilde{X}_{j2}$ and the NN observer weight estimation errors, $\tilde{W}_{jo}$, for the follower j in the presence of obstacles. Additionally, the convergence of the estimated velocity tracking error and observer estimation errors to a compact set implies that the actual velocity tracking error converges to a compact set as well.

Remark A.2: If the above conditions are satisfied for every follower $j = 1, 2, 3, \ldots N$, it follows $\sum_1^N \dot{\overline{V}}_j' \le 0$. Combining this result with *Theorem 3* reveals the entire formation is UUB in the presence of obstacles. Furthermore, the stability of a formation in the presence of obstacles for the case when follower *j* becomes

a leader to follower $j+1$ follows directly from the Lyapunov candidate $\overline{V}_j'' = \sum_{j}^{j+1} \overline{V}_j'$ where $\overline{V}_j'$ is defined in (A.16).

3. Output Feedback Control of a Quadrotor UAV using Neural Networks[1]

Travis Dierks and S. Jagannathan

Abstract*—A new nonlinear controller for a quadrotor unmanned aerial vehicle (UAV) is proposed using neural networks (NN) and output feedback. The assumption on the availability of UAV dynamics is not always practical, especially in an outdoor environment. Therefore, in this work, a NN is introduced to learn the complete dynamics of the UAV online, including uncertain nonlinear terms like aerodynamic friction and blade flapping. Although a quadrotor UAV is underactuated, a novel NN virtual control input scheme is proposed which allows all six degrees of freedom of the UAV to be controlled using only four control inputs. Furthermore, a NN observer is introduced to estimate the translational and angular velocities of the UAV, and an output feedback control law is developed in which only the position and attitude of the UAV are considered measurable. It is shown using Lyapunov theory that the position, orientation, and velocity tracking errors, the virtual control and observer estimation errors, and the NN weight estimation errors for each NN are all semi-globally uniformly ultimately bounded (SGUUB) in the presence of bounded disturbances and NN functional reconstruction errors while simultaneously relaxing the separation principle. The effectiveness of proposed output feedback control scheme is then demonstrated in the*

[1] Research Supported in part by GAANN Program through the Department of Education and Intelligent Systems Center. Authors are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology (formerly University of Missouri-Rolla), 1870 Miner Circle, Rolla, MO 65409. Contact author Email: tad5x4@mst.edu.

presence of unknown nonlinear dynamics and disturbances, and simulation results are included to demonstrate the theoretical conjecture.

Index Terms — Neural network, Quadrotor UAV, Lyapunov method, Output feedback, Observer

I. Introduction

Quadrotor helicopters have quickly emerged as a popular unmanned aerial vehicle (UAV) platform in the last several years. Besides applications like surveillance and search and rescue, the popularity of this platform has stemmed from its simple construction as compared with conventional helicopters. For example, a quadrotor UAV employs fixed pitch rotors so that its rotor speed can be adjusted to achieve control as opposed to mechanical control linkages used in conventional helicopters. Thus, a quadrotor UAV is easier to build and maintain [1].

The dynamics of the quadrotor UAV are not only nonlinear, but also coupled with each other and under actuated; characteristics which can make the platform difficult to control. In other words, the UAV has six degrees of freedom (DOF) with only four control inputs consisting of thrust and the three rotational torque inputs. To solve the quadrotor UAV tracking control problem, many techniques have been proposed [2-10] where the control objective is to track three desired Cartesian positions and a desired yaw angle.

In [2], a state-dependent Riccati equation-based control scheme was developed using the small angle approximation in order to derive the desired pitch and roll required for velocity tracking. In contrast, the authors of [3] design a controller using backstepping to track the three desired Cartesian positions and a yaw angle while stabilizing the pitch

and roll angles. Then, in [4], saturation functions are employed in the development of the control inputs obtained via backstepping approach. A drawback of these controllers [2-4] is the need for full state measurement and knowledge of the UAV dynamics a priori while the dynamics like aerodynamic friction are either simplified or ignored altogether. It was shown in [1] that the above simplifications are valid only at very low speeds such as hovering while the aerodynamic effects can become significant even at moderate velocities causing instability of the UAV.

On the other hand, in [5], a sliding mode observer is introduced to estimate the translational and angular velocities of the UAV. In addition to the UAV velocities, the authors in [6] propose a sliding mode estimator of external disturbances such as wind and model uncertainties. Then, using virtual control inputs and the *arcsin* function, the desired pitch and roll of the UAV were defined to track. Although the use of the *arcsin* function provides a natural saturation of the desired angles, *arcsin* becomes undefined when its argument is outside the range defined by $[-1,1]$, and provisions to ensure the aforementioned scenario does not happen are not guaranteed by [6]. In [7], an output feedback controller is achieved by strategically introducing a constant term into the filtered tracking error which is normally defined as a function of the position and translational velocity tracking errors, respectively. The introduction of the constant term is then utilized in the design of an auxiliary control input for the translational velocities whereas the system nonlinearities have to satisfy a linear in the unknown parameters (LIP) assumption. In [8], an adaptive observer is proposed to recover the speed of the UAV using accelerations, angle measurements, as well as measured angular velocities. Thus, the estimation relies on an accurate inertial measurement unit besides needing a

stringent persistency of excitation (PE) condition [12] to guarantee performance which is very hard to satisfy in practice.

In [9] and [10], the approximation property of NN [12] is applied to learn the dynamics of the quadrotor UAV. However, in both cases, the NN's are trained completely offline with experimentally collected data. A study evaluating the performance of NN's applied to UAV models which were trained offline and NN's with online learning was performed in [20]. This study verified several well-known properties of online learning versus offline training. Offline training allows for large amounts of data to be analyzed since computation time in not a critical issue although offline data collection is expensive. Moreover, models which are properly trained offline are often robust to small variations in the system but fail to adapt to larger changes in the system. Further, an offline scheme alone does not allow the NN to learn any new dynamics it encounters during a new maneuver. In other words, in dynamical environment, such as an outdoor setting with changing wind conditions, certain modes of the UAV dynamics may not be excited all the time (e.g. blade flapping, drag, etc.). Under this scenario, an offline trained NN may not render a satisfactory performance. In contrast, NN models which learn online quickly adapt to variations in the nonlinear behavior of the system in real time with no prior knowledge. Also, it is not practical to collect data for every operating scenario since UAV's often operate in dynamic environments.

Therefore, in [21]-[22], NN approaches are proposed to learn the dynamics of the UAV online while assuming full state feedback. In contrast, this paper seeks to remove the assumptions of full state measurement and knowledge of the UAV dynamics. First, by observing the natural constraints of the underactuated system [11], a novel NN virtual

control input is developed for the roll and pitch, which not only ensures that they remain within a stable operating region, but also guarantees that the UAV tracks the specific velocity command required to follow the desired trajectory. The virtual control input is well defined and provides a means of controlling all six DOF using only four control inputs. Additionally, the physical meaning of the virtual control inputs can be linked to the types of trajectories that can be successfully tracked. Next, the inputs of the dynamical system are calculated by utilizing the approximation properties of NN to learn the dynamics of the UAV online, including unmodeled dynamics like aerodynamic damping and blade flapping [1] while relaxing the LIP assumption.

Finally, a NN observer is utilized to estimate the translational and angular velocities of the UAV so that an output feedback control law can be realized. All NNs are tuned online to allow adaptations to changes of the UAV dynamics and the operating environment. It is shown using Lyapunov theory that the position, orientation, and velocity tracking errors, the virtual control observer estimation errors, and the NN weight estimation errors of each NN are all semi-globally uniformly ultimately bounded (SGUUB). Further by considering the NN observer errors in the same Lyapunov candidate as the UAV tracking errors, the separation principle is also relaxed. Simulation results are also presented to verify the controller in the presence of unmodeled nonlinear dynamics and random disturbances.

Linear models obtained from nonlinear systems are generally valid near a specific operating point [12], and for the UAV, the operating point is generally chosen near the hovering configuration [23] which may not be acceptable for dynamical outdoor setting with changing wind conditions. Therefore, the contribution of the proposed NN output

feedback controller include: 1) a novel nonlinear NN-based controller is developed for hovering or tracking time varying trajectories that are not near the hovering operating point; 2) explicit knowledge of the nonlinear dynamics is not required; 3) using output feedback, the number of sensors/states required to implement the controller is reduced while still guaranteeing performance and stability; and 4) the NN relaxes the LIP assumption which is required for adaptive controllers.

This paper is organized as follows. In Section II, the required background material is presented, and the dynamic representation of the UAV is identified along with the constraints associated with the underactuated system. Next, the novel dynamic output feedback tracking controller is developed and verified in Sections III and IV, respectively. Finally, concluding remarks are provided in Section V.

II. Background

A. Quadrotor UAV Dynamics

Consider the quadrotor UAV shown in Fig. 1 with six DOF defined in the inertial coordinate frame , E^a, as $[x, y, z, \phi, \theta, \psi]^T \in E^a$ where $\rho = [x, y, z]^T \in E^a$ are the position coordinates of the UAV and $\Theta = [\phi, \theta, \psi]^T \in E^a$ describe its orientation and are referred to as roll, pitch, and yaw, respectively. The translational and angular velocities are expressed in the body fixed frame attached to the center of mass of the UAV, E^b, and the kinematics of the UAV are written as

$$\dot{\rho} = Rv \tag{1}$$

and

$$\dot{\Theta} = T\omega . \tag{2}$$

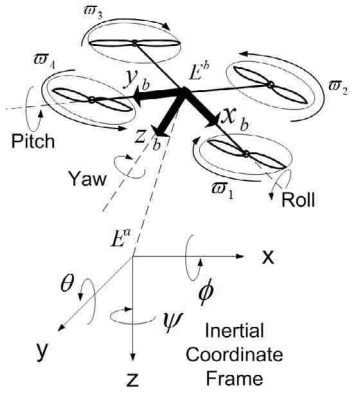


Fig. 1. Quadrotor helicopter.

The matrix $R(\Theta) \in \Re^{3x3}$ is the translational rotation matrix which is used to relate a vector in the body fixed frame to the inertial coordinate frame defined as [2]

$$R(\Theta) = R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{3}$$

where the abbreviations $s_{(\bullet)}$ and $c_{(\bullet)}$ have been used for $\sin(\bullet)$ and $\cos(\bullet)$, respectively. It is important to note that $\|R\|_F = R_{\max}$ for a known constant $R_{\max}$, $R^{-1} = R^T$, $\dot{R} = RS(\omega)$ and $\dot{R}^T = -S(\omega)R^T$ where $S(\bullet) \in \Re^{3x3}$ is the general form of a skew symmetric matrix defined in [7] which satisfies the *skew symmetric property* [12], $w^T S(\gamma) w = 0$, for any vector $w \in \Re^3$ and $\gamma \in \Re^3$. It is also necessary to define a rotational transformation matrix from the fixed body from to the inertial coordinate frame as [7]

$$T(\Theta) = T = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}, \qquad T^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \tag{4}$$

where the abbreviation $t_{(\bullet)}$ has been used for $\tan(\bullet)$. The transformation matrix T is bounded according to $\|T\|_F < T_{\max}$ for a known constant $T_{\max}$ provided $-(\pi/2) < \phi < (\pi/2)$ and $-(\pi/2) < \theta < (\pi/2)$ [15].

Next, the dynamics of the UAV in the body fixed frame can be written as [7]

$$M\begin{bmatrix}\dot{v}\\ \dot{\omega}\end{bmatrix} = \bar{S}(\omega)\begin{bmatrix}v\\ \omega\end{bmatrix} + \begin{bmatrix}N_1(v)\\ N_2(\omega)\end{bmatrix} + \begin{bmatrix}G(R)\\ 0_{3x1}\end{bmatrix} + U + \tau_d, \tag{5}$$

where $M = diag\{mI_3, J\} \in \Re^{6x6}$, $\bar{S}(\omega) = diag\{-mS(\omega), S(J\omega)\} \in \Re^{6x6}$, $U = [0 \;\; 0 \;\; u_1 \;\; u_2^T]^T \in \Re^6$ and m is a positive scalar that represents the total mass of the UAV, $J \in \Re^{3x3}$ represents the positive definite inertia matrix. The vector $v(t) = [v_{xb}, v_{yb}, v_{zb}]^T \in \Re^3$ represents the translational velocity, $\omega(t) = [\omega_{xb}, \omega_{yb}, \omega_{zb}]^T \in \Re^3$ represents the angular velocity, $N_i(\bullet) \in \Re^{3x1}$, $i = 1,2$, are the nonlinear aerodynamic effects, $u_1 \in \Re^1$ provides the thrust along the z-direction, $u_2 = [u_{21} \;\; u_{22} \;\; u_{23}]^T \in \Re^3$ provides the rotational torques, $\tau_d = [\tau_{d1}^T, \tau_{d2}^T]^T \in \Re^6$ and $\tau_{di} \in \Re^3$, $i = 1,2$ represents unknown, bounded disturbances such that $\|\tau_d\| < \tau_M$ for all time t, with τ_M being a known positive constant. Additionally, $I_{nxn} \in \Re^{nxn}$ is an nxn identity matrix, and $0_{mxl} \in \Re^{mxl}$ represents an mxl matrix of all zeros. Furthermore, $G(R) \in \Re^3$ represents the gravity vector defined as $G(R) = mgR^T(\Theta)E_z$ where $E_z = [0,0,1]^T$ is a unit vector in the inertial coordinate frame, and $g = 9.81 m/s^2$.

The control inputs to the UAV, u_1 and u_2, represent the thrust and torques, respectively, generated by the angular speeds of rotors, ϖ_i, $i = 1,2,3,4$, and are related to the thrust and drag factors by the following relationship [4]

$$u_1 = -c_t \sum_{i=1}^{4} \varpi_i^2 \ , \quad u_2 = [dc_t(\varpi_2^2 - \varpi_4^2),\ dc_t(\varpi_1^2 - \varpi_3^2),\ c_d \sum_{i=1}^{4} (-1)^{i+1} \varpi_i^2]^T \qquad (6)$$

where d is a positive scalar representing the distance from the epicenter of the quadrotor to the rotor axes, c_t is a positive scalar representing the thrust factor, and c_d is a positive scalar representing the drag factor.

Remark 1: Once the control inputs to the UAV have been determined, the relationship in (6) can be used to determine the required rotor speeds in order to achieve the desired thrust and rotational torques. Several authors, [3] and [6] for example, have considered the tracking control of the rotors assuming DC motors drive them. However, in this work, we are concerned with deriving the required thrust and rotational torques as in [2], [4], [7], and [9], respectively.

The nonlinear effects due to aerodynamic damping, $D_A(t) \in \Re^6$ [7] are modeled as

$$D_A(t) = \begin{bmatrix} d_1 + d_2|v_{xb}| & 0 & 0 & 0 & 0 & 0 \\ 0 & d_3 + d_4|v_{yb}| & 0 & 0 & 0 & 0 \\ 0 & 0 & d_5 + d_6|v_{zb}| & 0 & 0 & 0 \\ 0 & 0 & 0 & d_7 + d_8|\omega_{xb}| & 0 & 0 \\ 0 & 0 & 0 & 0 & d_9 + d_{10}|\omega_{yb}| & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{11} + d_{12}|\omega_{zx}| \end{bmatrix} \begin{bmatrix} v_{xb} \\ v_{yb} \\ v_{zb} \\ \omega_{xb} \\ \omega_{yb} \\ \omega_{zb} \end{bmatrix}$$

where d_i ,i=1,2,…12 are the damping coefficients. Additionally, blade flapping effects are considered which result from differences between the effective velocities of the rotor relative to the air. As a consequence, a difference in lift between rotors is observed causing the rotor blades to flap up and down once per revolution. Furthermore, flapping of the rotor blades tilts the rotor plane away from the direction of motion, thus affecting the thrust and rotational torques of the UAV and ultimately it's tracking ability. Specifically, blade flapping creates a longitude thrust $T_{Li} = T_i \sin\alpha_i$ where T_i is the thrust

generated at rotor *i* and α_i is the angle in *radians* [*rad*] by which the thrust is redirected at the rotor and is dependent the translational movements of the UAV as well as the wind conditions. Further, for stiff rotors which are commonly utilized in a quadrotor UAV, blade flapping results in a moment being generated at each rotor hub as $M_{Hi} = k_\beta \alpha_i$ where k_β is the stiffness of the rotor in *Newton-meters per radian* [*N-m/rad*]. For complete details on blade flapping and its full effects, please refer [1].

As shown in (6), the state of each rotor is related to the total thrust and rotational torques which drive the dynamics of (5). Therefore, in this work the effects of blade flapping at each rotor will be combined as single nonlinear disturbances to the quadrotor dynamics (5). For the translational velocities, blade flapping results in thrust being redirected longitudinally while the lifting force is reduced. Therefore, the disturbances in the *x* and *y* directions are modeled as $T_{Lx,y} = T\sin\alpha$, the reduction in thrust in the z direction is taken as $T_{L_z} = T(1-\cos\alpha)$, and the disturbance to the angular velocity vector is written as $M_H = k_\beta \alpha$ where $T = u_1$ and α is the angle at which the total thrust is redirected, respectively. The total nonlinear aerodynamic effects in (5) are then written as

$$\begin{bmatrix} N_1(v) \\ N_2(\omega) \end{bmatrix} = \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} = D_A(t) - [u_1\sin\alpha \quad u_1\sin\alpha \quad u_1(1-\cos\alpha) \quad k_\beta\alpha \quad k_\beta\alpha \quad k_\beta\alpha]^T \in \Re^6 \qquad (7)$$

B. Constraints of the Underactuated System

Examining the UAV dynamics (5), it becomes clear that the translational velocity dynamics form an underactuated system where only v_{zb} is controllable through u_1. Ignoring disturbances and the nonlinear dynamics $N_1(v)$, the constraints associated with

the pitch and roll for the underactuated translational velocity dynamics were derived in [11] by observing $\dot{v} = R^T \ddot{\rho} - S(\omega)v$ along with (5) and were found to be

$$\theta = a\tan\left((c_\psi \ddot{x} + s_\psi \ddot{y})/(\ddot{z} - g)\right), \qquad \phi = a\tan\left((s_\psi \ddot{x} - c_\psi \ddot{y})/(s_\theta c_\psi \ddot{x} + s_\theta s_\psi \ddot{y} + c_\theta(\ddot{z} - g))\right). \quad (8)$$

The above constraints reveal an important property about the UAV during stable flight conditions. First, note that for certain combinations of $\ddot{x}, \ddot{y}, \ddot{z}, \text{and} \psi$, the pitch and roll angles approach $\pm\pi/2$ and thus R and T become singular. Therefore, the UAV becomes unstable, and it can be concluded that certain maneuvers are not achievable during stable conditions. These natural constraints will be exploited in the upcoming development of the virtual control inputs which will allow the UAV to track the desired trajectory.

C. Neural Networks

In this work, two-layer NNs are considered consisting of one layer of randomly assigned constant weights $V_N \in \Re^{axL}$ in the first layer and one layer of tunable weights $W_N \in \Re^{Lxb}$ in the second with a inputs, b outputs, and L hidden neurons. A compromise is made here between tuning the number of layered weights with computational complexity. The *universal approximation property* for NN's [12] states that for any smooth function $f_N(x_N)$, there exists a NN such that $f_N(x_N) = W_N^T \sigma(V_N^T x_N) + \varepsilon_N$ where ε_N is the bounded NN functional approximation error such that $\|\varepsilon_N\| < \varepsilon_M$, and $\sigma(\cdot) : \Re^a \to \Re^L$ is the activation function in the hidden layers. It has been shown that by randomly selecting the input layer weights V_N, the activation function $\sigma(\bar{x}_N) = \sigma(V_N^T x_N)$ forms a stochastic basis, and thus the approximation property holds for all inputs, $x_N \in \Re^a$, in the compact

set S [12]. The sigmoid activation function is considered here. Furthermore, on any compact subset of $\Re^n$, the target NN weights are bounded by a known positive value, W_M, such that $\|W_N\|_F \leq W_M$ [12]. For complete details of the NN and its properties, refer to [12]. In this effort, $\|\cdot\|$ and $\|\cdot\|_F$ will be used as the vector and Frobenius norms [12]. Next the definition of the semi-global uniformly ultimately boundedness is introduced.

Definition 1: The equilibrium point x_e is said to be *semi-global uniformly ultimately bounded (SGUUB)* if there exists a ball centered around the origin with an arbitrary radius r $S(0,r) = S_r \subset \Re^n$ so that for all $x_0 \in S_r$ there exists a bound $B > 0$ and a time $T(B, x_0)$ such that $\|x(t) - x_e\| \leq B$ for all $t \geq t_0 + T$. Further, if $S_r = \Re^n$, the stability result becomes *global uniformly ultimately bounded (GUUB)* and holds for all $x_0 \in \Re^n$ [16].

III. Neural Network Output Feedback Tracking Control

The overall control objective for the UAV is to track a desired trajectory, $\rho_d = [x_d, y_d, z_d]^T \in E^a$, and a desired yaw $\psi_d \in E^a$ while maintaining a stable flight configuration. The complete knowledge of the UAV dynamics and velocity information is required to complete the control objective; however, in this work, the translational and angular velocities are considered to be not measurable and full knowledge of the dynamics is not available whereas the constant mass and moments of inertia of the UAV are assumed known similar to [2]-[9]. Therefore, the *universal approximation property* of NN is utilized in the design of the observer, virtual controller, and the dynamical controller. Knowledge of the mass is required for the dynamic control law whereas the

mass and moments of inertia are needed for the observer. Future effort will seek to relax these assumptions.

The proposed NN observer estimates the UAV velocity vector which is required by the control loop. The control loop, which consists of a kinematic controller, NN virtual controller and a NN dynamical controller, uses the information provided by the observer to generate the appropriate commands to complete the control objective. To begin the NN output feedback controller development, the NN observer design is considered first.

A. NN Observer Design

In this section, a NN observer is designed to estimate the UAV translational and angular velocity vector without explicit knowledge of the dynamics (5). To begin, define new augmented variables $X = [\rho^T \ \Theta^T]^T \in \Re^6$ and $V = [v^T \ \omega^T]^T \in \Re^6$ whose dynamics are given by (1) and (5), respectively, and rewritten as

$$\begin{aligned} \dot{X} &= A(t)V + \xi_1 \\ \dot{V} &= f_o(x_o) + \overline{G} + M^{-1}U + \overline{\tau}_d \end{aligned} \tag{9}$$

where $\xi_1 \in \Re^6$ represents bounded sensor measurement noise such that $\|\xi_1\| \le \xi_{1M}$ for a known constant ξ_{1M}, $f_o(x_o) = M^{-1}(\overline{S}(\omega)V + [N_1(v) \ N_2(\omega)]^T)$ with $x_o = V$, $\overline{G} = M^{-1}G(R) \in \Re^6$, $\overline{\tau}_d = [\overline{\tau}_{d1}^T, \overline{\tau}_{d2}^T]^T = [\tau_{d1}^T / m, (J^{-1}\tau_{d2})^T]^T \in \Re^6$ and

$$A(t) = A = \begin{bmatrix} R & 0_{3x3} \\ 0_{3x3} & T \end{bmatrix}. \tag{10}$$

Remark 2: In [13] and [14], observers were proposed utilizing adaptive fuzzy logic and a NN, respectively, and by defining a change of variables. In both these

approaches, A is the identity matrix, while in this work, A a time varying, nonlinear matrix as a result of the relationships observed in kinematic equations (1) and (2).

Next, define a change of variable as $Z = V$, whose derivative with respect to time is given by (9). Then, define the NN observer estimates of X and V as $\hat{X}$ and $\hat{Z}$, respectively, as well as the observer estimation error $\tilde{X} = X - \hat{X}$. The proposed observer then takes the form of

$$\begin{aligned} \dot{\hat{X}} &= A\hat{Z} + K_{o1}\tilde{X} \\ \dot{\hat{Z}} &= \hat{f}_{o1} + \overline{G} + K_{o2}A^{-1}\tilde{X} + M^{-1}U \end{aligned} \tag{11}$$

where K_{o1} and K_{o2} are positive design constants. From the definition of the transformation matrix A in (10), it is observed that A^{-1} can be calculated using $R^{-1} = R^T$ and T^{-1} in (4). Further, there exists a positive constant A_M^I such that $\left\|A^{-1}\right\|_F \le A_M^I$. The observer velocity estimate $\hat{V}$ is then written as

$$\hat{V} = [\hat{v}^T \quad \hat{\omega}^T]^T = \hat{Z} + K_{o3}A^{-1}\tilde{X} \tag{12}$$

where K_{o3} is another positive design constant. Noting $\hat{Z} = \hat{V} - K_{o3}A^{-1}\tilde{X}$ from (12) and the definition of $\dot{Z}$ above, the observer error dynamics of (11) can be formulated as

$$\begin{aligned} \dot{\tilde{X}} &= A\tilde{V} - (K_{o1} - K_{o3})\tilde{X} + \xi_1 \\ \dot{\tilde{Z}} &= (f_o + (A^T - K_{o3}\dot{A}^{-1})\tilde{X}) - \hat{f}_{o1} - K_{o2}A^{-1}\tilde{X} - (A^T - K_{o3}\dot{A}^{-1})\tilde{X} + \bar{\tau}_d \end{aligned} \tag{13}$$

after adding and subtracting $(A^T - K_{o3}\dot{A}^{-1})\tilde{X}$.

In (11), the *universal approximation property* of NN [12] has been utilized to estimate the unknown function $f_{o1}(x_o) = f_o(x_o) + (A^T - K_{o3}\dot{A}^{-1})\tilde{X}$ by constant ideal bounded weights W_o^T, V_o^T such that $\left\|W_o\right\|_F \le W_{Mo}$ for a known constant W_{Mo}, and written

as $f_{o1}(x_o) = W_o^T \sigma(V_o^T x_o) + \varepsilon_o$ where ε_o is the bounded NN approximation error such that $\|\varepsilon_o\| \le \varepsilon_{Mo}$ for a known constant ε_{Mo}. In addition, it was shown in [24] that if the number of hidden layer neurons is sufficiently large, the reconstruction error can be made arbitrarily small on a compact set. In practice, the values for W_{Mo} and ε_{Mo} are selected based on properties of the dynamics being approximated and the number of hidden layer neurons being used. Additionally, the values for W_{Mo} and ε_{Mo} are not required to be known for the controller design whereas if the values are available, one can calculate the error bounds which will subsequently be derived in *Theorem 1*. The NN estimate of $f_{o1}(x_o)$ is written as $\hat{f}_{o1} = \hat{W}_o^T \sigma(V_o^T \hat{x}_o) = \hat{W}_o^T \hat{\sigma}_o$ where $\hat{W}_o^T$ is the estimate of W_o^T, and $\hat{x}_o$ is the NN input written in terms of the observer velocity estimates as $\hat{x}_o = [1 \;\; \hat{X}^T \;\; \hat{V}^T \;\; \tilde{X}^T]^T$.

Moving on and noting $\tilde{V} = V - \hat{V} = \tilde{Z} - K_{o3} A^{-1} \tilde{X}$, adding and subtracting $W_o^T \sigma(V_o^T \hat{x}_o)$, and using (13), the observer estimation error dynamics of (12) take the form of

$$\begin{aligned} \dot{\tilde{V}} &= (f_o + (A^T - K_{o3}\dot{A}^{-1})\tilde{X}) - W_o^T \sigma(V_o^T \hat{x}_o) + W_o^T \sigma(V_o^T \hat{x}_o) - \hat{f}_{o1} - A^{-1}(K_{o2} - K_{o3}(K_{o1} - K_{o3}))\tilde{X} \\ &\quad - A^T \tilde{X} - K_{o3}\tilde{V} - A^{-1} K_{o3} \xi_1 + \bar{\tau}_d \\ &= -K_{o3}\tilde{V} + \tilde{f}_{o1} - A^{-1}(K_{o2} - K_{o3}(K_{o1} - K_{o3}))\tilde{X} - A^T \tilde{X} + \xi_2 \end{aligned} \tag{14}$$

where $\tilde{f}_o = \tilde{W}_o \hat{\sigma}_o$, $\tilde{W}_o = W_o - \hat{W}_o$, $\xi_2 = \varepsilon_o + \bar{\tau}_d - K_{o3} A^{-1} \xi_1 + W_o \tilde{\sigma}_o \in \Re^6$, and $\tilde{\sigma}_o = \sigma_o - \hat{\sigma}_o$. Further, $\|\xi_2\| \le \xi_{2M}$ where ξ_{2M} is a positive computable constant defined as $\xi_{2M} = \varepsilon_{Mo} + M_M \tau_M \quad + K_{o3} A_M^I \xi_{1M} + 2W_{Mo}\sqrt{N_o}$ where $M_M = \| M^{-1} \|_F$, a computable constant, N_o is the number of hidden layer neurons in the NN, and the

fact $\|\sigma_o\| \le \sqrt{N_o}$ was used. Examining the error dynamics of (13) and (14) reveals that $\tilde{X} = 0, \tilde{V} = 0,$ and $\tilde{f}_{o1} = 0$ are equilibrium points when $\|\xi_1\| = 0$ and $\|\bar{\tau}_d\| = 0$. Next the following theorem can be stated.

Theorem 1: (NN Observer Boundedness) Let the NN observer be defined by (11) and (12), respectively, with the NN observer weights be tuned by

$$\dot{\hat{W}}_o = F_o \hat{\sigma}_o \tilde{X}^T - \kappa_{o1} F_o \hat{W}_o \tag{15}$$

where $F_o = F_o^T > 0$ and $\kappa_{o1} > 0$ are design parameters. Then there exists constant positive design parameters $K_{o1}, K_{o2},$ and K_{o3} where $K_{o1} > K_{o3} + (2N_0)/\kappa_{o1}$, $K_{o3} > (2N_o)/\kappa_{o1}$, and $K_{o2} = K_{o3}(K_{o1} - K_{o3})$ with N_o the number of hidden layer neurons, such that the observer estimation errors $\tilde{X}$, $\tilde{V}$ and the NN observer weight estimation errors, $\tilde{W}_o$, are *SGUUB*.

Proof: Consider the following positive definite Lyapunov candidate

$$V_o = \frac{1}{2}\tilde{X}^T\tilde{X} + \frac{1}{2}\tilde{V}^T\tilde{V} + \frac{1}{2}tr\{\tilde{W}_o^T F_o^{-1}\tilde{W}_o\} \tag{16}$$

whose first derivative is given by $\dot{V}_o = \tilde{X}^T\dot{\tilde{X}} + \tilde{V}^T\dot{\tilde{V}} + tr\{\tilde{W}_o^T F_o^{-1}\dot{\tilde{W}}_o\}$. Substitution of the closed loop observer estimation error dynamics (13) and (14) as well as the NN update law (15) reveals

$$\dot{V}_o = -\tilde{X}^T(K_{o1} - K_{o3})\tilde{X} + \tilde{X}^T\xi_1 - \tilde{V}^T K_{o3}\tilde{V} + \tilde{V}^T\xi_2 - tr\{\tilde{W}_o^T(\hat{\sigma}_o\tilde{X}^T - \hat{\sigma}_o\tilde{V}^T - \kappa_{o1}\hat{W}_o\}.$$

Recalling $\|\hat{\sigma}_o\| \le \sqrt{N_o}$, $\|W_o\|_F \le W_{Mo}$, and noting $tr\{\tilde{W}_o^T(W_o - \tilde{W}_o)\} \le \|\tilde{W}_o\|_F W_{Mo} - \|\tilde{W}_o\|_F^2$, $\dot{V}_o$ can then be rewritten as

$$\dot{V}_o = -(K_{o1} - K_{o3})\|\tilde{X}\|^2 + \|\tilde{X}\|\xi_{1M} - K_{o3}\|\tilde{V}\|^2 + \|\tilde{V}\|\xi_{2M} - \kappa_{o1}\|\tilde{W}_o\|_F^2 + \|\tilde{W}_o\|_F\|\tilde{X}\|\sqrt{N_o} + \|\tilde{W}_o\|_F\|\tilde{V}\|\sqrt{N_o} + \kappa_{o1}\|\tilde{W}_o\|_F W_{Mo}. \tag{17}$$

Now, completing the squares with respect to $\|\tilde{W}_o\|_F$, $\|\tilde{X}\|$, and $\|\tilde{V}\|$, (17) can be rewritten as

$$\dot{V}_o \le -\left(\frac{(K_{o1}-K_{o3})}{2}-\frac{N_o}{\kappa_{o1}}\right)\left\|\tilde{X}\right\|^2-\left(\frac{K_{o3}}{2}-\frac{N_o}{\kappa_{o1}}\right)\left\|\tilde{V}\right\|^2-\frac{\kappa_{o1}}{4}\left\|\tilde{W}_o\right\|_F^2+\eta_o \tag{18}$$

where $\eta_o=\kappa_{o1}W_{Mo}^2+\xi_{1M}^2/(2(K_{o1}-K_{o3}))+\xi_{2M}^2/(2K_{o3})$ and is dependant on the bounds of the sensor measurement errors, NN reconstruction error, disturbances and design parameters.

Finally, (18) is less than zero provided $K_{o1}>K_{o3}+(2N_o)/\kappa_{o1}$, $K_{o3}>(2N_o)/\kappa_{o1}$, and the following inequalities hold:

$$\left\|\tilde{X}\right\|>\sqrt{\frac{\eta_o}{((K_{o1}-K_{o3})/2-N_o/\kappa_{o1})}}\ \text{ or }\ \left\|\tilde{V}\right\|>\sqrt{\frac{\eta_o}{(K_{o3}/2-N_o/\kappa_{o1})}}\ \text{ or }\ \left\|\tilde{W}_o\right\|_F>\sqrt{\frac{4\eta_o}{\kappa_{o1}}} \tag{19}$$

Therefore, it can be concluded [12] that $\dot{V}_o$ is less than zero outside a compact set, revealing the observer estimation errors $\tilde{X}$, $\tilde{V}$ and the NN observer weight estimation errors, $\tilde{W}_o$, are bounded.

Examining the definition of η_o, it is clear that the constant terms can be made arbitrarily small by selecting κ_{o1} small, K_{o3} large, and $K_{o1}-K_{o3}$ large. Therefore, since the initial compact set can be made arbitrarily large by proper selection of gains, the stability result becomes *SGUUB* [7].

B. Kinematic Control System

In this section, we derive the terms which will be used by the NN virtual controller in the following subsection (and illustrated in Fig. 2 shown in *Section III-D*). Namely, the desired translational velocity $v_d=[v_{dx}\ v_{dy}\ v_{dz}]^T\in E^b$ is found to ensure the UAV position converges to the desired trajectory ($\rho\to\rho_d$). Next, the desired pitch, θ_d, and roll, ϕ_d, are found to ensure the *x*- and *y*- components of the UAV translational velocity track their respective desired values ($v_{xb}\to v_{dx}$ and $v_{yb}\to v_{dy}$, respectively).

Then, given the desired orientation, $\Theta_d = [\phi_d \ \theta_d \ \psi_d]^T \in E^a$, the desired angular velocity $\omega_d \in E^b$ is then calculated such that $\Theta \to \Theta_d$.

To begin the development of the UAV tracking controller, we first define the tracking errors for the position and translational velocity. For the position, define

$$e_\rho = \rho_d - \rho \in E^a . \tag{20}$$

Differentiating (20) and substitution of (1) yields the position error dynamics

$$\dot{e}_\rho = \dot{\rho}_d - Rv . \tag{21}$$

Next, select the desired velocity to stabilize the position error dynamics as

$$v_d = [v_{dx} \ v_{dy} \ v_{dz}]^T = R^T(\dot{\rho}_d + K_\rho e_\rho) \in E^b \tag{22}$$

where $K_\rho = diag\{k_{\rho x}, k_{\rho y}, k_{\rho z}\} \in \Re^{3x3}$ is a diagonal positive definite design matrix all with positive design constants. The translational velocity tracking error system is then defined as

$$e_v = [e_{vx} \ e_{vy} \ e_{vz}]^T = [v_{dx} \ v_{dy} \ v_{dz}]^T - [v_{xb} \ v_{yb} \ v_{zb}]^T = v_d - v . \tag{23}$$

The desired velocity v_d is a virtual control input to (21), and substituting (22) into (21) while observing $v = v_d - e_v$, the closed loop position error dynamics can be rewritten as

$$\dot{e}_\rho = -K_\rho e_\rho + R\, e_v . \tag{24}$$

Next, observing $\dot{v}_d = -S(\omega)v_d + R^T(\ddot{\rho}_d + K_\rho(\dot{\rho}_d - Rv))$, the translational velocity tracking error dynamics are formed by differentiating (23), and substituting the translational velocity dynamics in (5) to obtain

$$\dot{e}_v = \dot{v}_d - \dot{v} = -\frac{1}{m}N_1(v) - S(\omega)e_v - \frac{1}{m}G(R) - \frac{1}{m}u_1 E_z + R^T(\ddot{\rho}_d + K_\rho\dot{\rho}_d - K_\rho Rv) - \bar{\tau}_{d1} . \tag{25}$$

Now write (3) in terms of the desired orientation angles, Θ_d, and define $R_d = R(\Theta_d)$. Finally, we add and subtract $G(R_d)/m$ and $R_d^T(\ddot{\rho}_d + K_\rho \dot{\rho}_d - K_\rho R\hat{v})$ where $\hat{v}$ is the observer estimate of the translational velocity to yield

$$\dot{e}_v = -S(\omega)e_v - \frac{1}{m}G(R_d) + R_d^T(\ddot{\rho}_d + K_\rho \dot{\rho}_d - K_\rho R\hat{v} + f_{c1}(x_{c1})) - \frac{1}{m}u_1 E_z - \overline{\tau}_{d1} \tag{26}$$

where

$$f_{c1}(x_{c1}) = R_d\left(-\frac{1}{m}N_1(v) - \frac{1}{m}(G(R) - G(R_d)) + (R - R_d)^T(\ddot{\rho}_d + K_\rho \dot{\rho}_d) - R^T K_\rho Rv + R_d^T K_\rho R\hat{v}\right) \tag{27}$$

is an unknown function rewritten as $f_{c1}(x_{c1}) = [f_{c11} \quad f_{c12} \quad f_{c13}]^T \in \Re^3$. In the next step, we expand the velocity error dynamics (26) to get

$$\begin{bmatrix} \dot{e}_{vx} \\ \dot{e}_{vy} \\ \dot{e}_{vz} \end{bmatrix} = -S(\omega)e_v - g\begin{bmatrix} -s_{\theta d} \\ c_{\theta d}s_{\phi d} \\ c_{\theta d}c_{\phi d} \end{bmatrix} \tag{28}$$

$$+\begin{bmatrix} c_{\theta d}c_{\psi d} & c_{\theta d}s_{\psi d} & -s_{\theta d} \\ s_{\phi d}s_{\theta d}c_{\psi d} - c_{\phi d}s_{\psi d} & s_{\phi d}s_{\theta d}s_{\psi d} + c_{\phi d}c_{\psi d} & s_{\phi d}c_{\theta d} \\ c_{\phi d}s_{\theta d}c_{\psi d} + s_{\phi d}s_{\psi d} & c_{\phi d}s_{\theta d}s_{\psi d} - s_{\phi d}c_{\psi d} & c_{\phi d}c_{\theta d} \end{bmatrix}\begin{bmatrix} \ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + f_{c11} \\ \ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + f_{c12} \\ \ddot{z}_d + k_{\rho z}\dot{z}_d - \hat{v}_{R3} + f_{c13} \end{bmatrix} - \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} - \overline{\tau}_{d1}$$

where $\hat{v}_R = [\hat{v}_{R1} \quad \hat{v}_{R2} \quad \hat{v}_{R3}]^T = K_\rho R\hat{v} \in \Re^3$.

Examining (28), the error states, $\dot{e}_{vx}$ and $\dot{e}_{vy}$, are not controllable by using the control input u_1. Thus, $\dot{e}_{vx}$ and $\dot{e}_{vy}$ must be controlled through the states that are influenced by the control inputs u_1 or u_2. In this work, the pitch and roll are used to control the translational movements of the UAV along x and y directions, respectfully, and thus, the pitch and roll angles are treated as virtual control inputs to the underactuated portion of the UAV error dynamics in (28).

The key step in the development of the virtual control input for the dynamic system is identifying the *desired* closed loop velocity tracking error dynamics. For convenience, the *desired* translational velocity closed loop system is selected as

$$\dot{e}_v = -(K_v + S(\omega))e_v - \bar{\tau}_{d1} \tag{29}$$

where $K_v = diag\{k_{v1}\cos(\theta_d), k_{v2}\cos(\phi_d),\ k_{v3}\}$ is a diagonal positive definite design matrix with each $k_{vi} > 0$, $i = 1,2,3$. In the following development, it will be shown that $\theta_d \in (-\pi/2, \pi/2)$ and $\phi_d \in (-\pi/2, \pi/2)$; therefore, $\cos(\theta_d) > 0$ and $\cos(\phi_d) > 0$ for all $\theta_d \in (-\pi/2, \pi/2)$ and $\phi_d \in (-\pi/2, \pi/2)$, respectively.

Moving on, equating (28) and (29) and considering only the first two error states renders

$$-g\begin{bmatrix} -s_{\theta d} \\ c_{\theta d}s_{\phi d} \end{bmatrix} + \begin{bmatrix} c_{\theta d}c_{\psi d} & c_{\theta d}s_{\psi d} & -s_{\theta d} \\ s_{\phi d}s_{\theta d}c_{\psi d} - c_{\phi d}s_{\psi d} & s_{\phi d}s_{\theta d}s_{\psi d} + c_{\phi d}c_{\psi d} & s_{\phi d}c_{\theta d} \end{bmatrix} \begin{bmatrix} \ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + f_{c11} \\ \ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + f_{c12} \\ \ddot{z}_d + k_{\rho z}\dot{z}_d - \hat{v}_{R3} + f_{c13} \end{bmatrix} + \begin{bmatrix} k_{v1}e_{vx}c_{\theta d} \\ k_{v2}e_{vy}c_{\phi d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{30}$$

Then, applying basic math operations, the following relations hold.

$$s_{\theta d}\left(\ddot{z}_d + k_{\rho z}\dot{z}_d - \hat{v}_{R3} - g + f_{c13}\right) = c_{\theta d}\left(c_{\psi d}\left(\ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + f_{c11}\right) + s_{\psi d}\left(\ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + f_{c12}\right) + k_{v1}e_{vx}\right) \tag{31}$$

$$\begin{aligned} &c_{\phi d}\left(c_{\psi d}(\ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + f_{c12}) - s_{\psi d}(\ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + f_{c11}) + k_{v2}e_{vy}\right) = \\ &s_{\phi d}(gc_{\theta d} - s_{\phi d}c_{\theta d}(\ddot{z}_d + k_{\rho z}\dot{z}_d - \hat{v}_{R3} + f_{c13}) - s_{\theta d}s_{\psi d}(\ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + f_{c12}) - s_{\theta d}c_{\psi d}(\ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + f_{c11})) \end{aligned} \tag{32}$$

Since the velocity vector is not measurable in this work, the velocity tracking error (23) cannot be used in the definition of the desired pitch and roll angles. Therefore, define the estimated velocity tracking error using the observer velocity estimates as

$$\hat{e}_v = [\hat{e}_{vx}\ \ \hat{e}_{vy}\ \ \hat{e}_{vz}]^T = v_d - \hat{v} = v_d - (v - \tilde{v}) = e_v + \tilde{v} \tag{33}$$

where $\tilde{v}$ is the observer estimation error for the translational velocity. Using (33) and considering (31), the desired pitch angle, θ_d, is found to be

$$\theta_d = a\tan\left(\frac{c_{\psi d}(\ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + \hat{f}_{c11}) + s_{\psi d}(\ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + \hat{f}_{c12}) + k_{v1}\hat{e}_{vx}}{\ddot{z}_d + k_{\rho z}\dot{z}_d - \hat{v}_{R3} - g + \hat{f}_{c13}}\right) \tag{34}$$

where $a\tan(\bullet)$ is the arctangent function. Next, using (32), the desired roll angle, ϕ_d can be written as

$$\phi_d = a\tan\left(\frac{c_{\psi d}(\ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + \hat{f}_{c12}) - s_{\psi d}(\ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + \hat{f}_{c11}) + k_{v2}\hat{e}_{vy}}{g c_{\theta d} - c_{\theta d}(\ddot{z}_d + k_{\rho z}\dot{z}_d - \hat{v}_{R3} + \hat{f}_{c13}) - s_{\theta d}s_{\psi d}(\ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + \hat{f}_{c12}) - s_{\theta d}c_{\psi d}(\ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + \hat{f}_{c11})}\right). \tag{35}$$

In (34) and (35), the approximation properties of NN were utilized to estimate the unknown function $f_{c1}(x_{c1}) = W_{c1}^T\sigma(V_{c1}^T x_{c1}) + \varepsilon_{c1}$ by bounded target weights W_{c1}^T, V_{c1}^T such that $\|W_{c1}\|_F \le W_{Mc1}$ with W_{Mc1} a known constant, and ε_{c1} is the NN approximation error that satisfies $\|\varepsilon_{c1}\| \le \varepsilon_{Mc1}$ for a constant ε_{Mc1}. The NN estimate of f_{c1} is written as $\hat{f}_{c1} = \hat{W}_{c1}^T\sigma(V_{c1}^T\hat{x}_{c1}) = \hat{W}_{c1}^T\hat{\sigma}_{c1} = [\hat{W}_{c11}^T\hat{\sigma}_{c1}\ \ \hat{W}_{c12}^T\hat{\sigma}_{c1}\ \ \hat{W}_{c13}^T\hat{\sigma}_{c1}]^T = [\hat{f}_{c11}\ \ \hat{f}_{c12}\ \ \hat{f}_{c13}]^T$ where $\hat{W}_{c1}^T$ is the NN estimate of W_{c1}^T, $\hat{W}_{c1i}^T, i = 1,2,3$ is the i^{th} row of $\hat{W}_{c1}^T$, and $\hat{x}_{c1}$ is the NN input written as $\hat{x}_{c1} = [1\ \dot{\rho}_d^T\ \ddot{\rho}_d^T\ \hat{V}^T\ \Theta^T\ \tilde{X}^T]^T$.

Remark 3: The expressions for the desired pitch and roll in (34) and (35), respectively, can be viewed as virtual control inputs, and they lend themselves very well to the control of quadrotor UAV. First, the expressions are well defined since $a\tan(\bullet)$ has a domain of $\Re$ compared to $a\sin(\bullet)$ which has a domain between -1 and +1. Second, the expressions in (34) and (35) are naturally saturating and will always produce desired values in the interval $(-\pi/2, \pi/2)$ of the UAV. Finally, the virtual control inputs provide the types of desired trajectories that can be tracked in the steady state. Examination of (34) and (35) reveals that there exist desired trajectories which will result in operating regions near the unstable operating points of the UAV since

$a\tan(\bullet)$ approaches $\pm\pi/2$ as the input of $a\tan(\bullet)$ increases. Additionally, large values of k_{px} and k_{py} can push an UAV toward instability.

Now that the desired orientation of the UAV has been specified, the desired angular velocity, ω_d, can be found to ensure the UAV orientation converges to the desired values ($\phi \to \phi_d$, $\theta \to \theta_d$, and $\psi \to \psi_d$) which is considered next.

To begin the development of ω_d, first define the attitude tracking error as

$$e_\Theta = \Theta_d - \Theta \in E^a \tag{36}$$

where dynamics are found using (2) as

$$\dot{e}_\Theta = \dot{\Theta}_d - T\omega. \tag{37}$$

In order to drive the orientation errors (36) to zero, the desired angular velocity, ω_d, is selected as

$$\omega_d = T^{-1}(\dot{\Theta}_d + K_\Theta e_\Theta) \tag{38}$$

where $K_\Theta = diag\{k_{\Theta 1}, k_{\Theta 2}, k_{\Theta 3}\} \in \Re^{3x3}$ is a diagonal positive definite design matrix with each $k_{\Theta i} > 0,\ i = 1,2,3$. Then, define the angular velocity tracking error as

$$e_\omega = \omega_d - \omega, \tag{39}$$

and observing $\omega = \omega_d - e_\omega$, the closed loop orientation tracking error system can be written as

$$\dot{e}_\Theta = -K_\Theta e_\Theta + Te_\omega. \tag{40}$$

Examining (38), calculation of the desired angular velocity requires the knowledge of $\dot{\Theta}_d$; however, $\dot{\Theta}_d$ is not known in view of the fact that $\dot{\hat{v}}$ and $\dot{\hat{f}}_{c1}$ are not available. Further, in the development of u_2 in the following section, it will be shown

that $\dot{\omega}_d$ is required which in turn implies $\ddot{\hat{v}}$ and $\ddot{\hat{f}}_{c1}$ must be known. Since these requirements are not practical, the *universal approximation property* of NN is invoked to estimate ω_d and $\dot{\omega}_d$ using a NN virtual controller which is considered in the following subsection.

Remark 4: In [22], the authors use inverse kinematics to define the desired pitch and roll angles which are found to be arctangents of NN estimates. To implement the control law derived in [22], a derivative of the NN estimate is required similar to our work; however, even a single derivative of the NN output is difficult to obtain since differentiating the NN input often introduces additional unknown dynamics. The effort in [22] does not provide insight into how this NN derivative is obtained. In contrast, our approach also defines the desired pitch and roll using the arctangent function whose argument contains a NN estimate, but the argument of the arctangent function here is fundamentally different from [22]. Further, the need to differentiate the NN output twice is avoided in this work by using the NN virtual controller.

C. NN Virtual Control Development

In this section, the information provided by the kinematic controller derived in *Section III-B* is used to calculate the desired angular velocity using a NN virtual controller. To begin the development, we rearrange (38) to observe the dynamics of the proposed virtual controller when the all dynamics are known as

$$\begin{aligned} \dot{\Theta}_d &= T(\omega_d - T^{-1}K_\Theta e_\Theta) \\ \dot{\omega}_d &= \dot{T}^{-1}(\dot{\Theta}_d + K_\Theta e_\Theta) + T^{-1}(\ddot{\Theta}_d + K_\Theta \dot{e}_\Theta) \end{aligned}. \tag{41}$$

For convenience, define a change of variable as $\Omega_d = \omega_d - T^{-1}K_\Theta e_\Theta$, and (41) becomes

$$\begin{aligned}\dot{\Theta}_d &= T\Omega_d \\ \dot{\Omega}_d &= \dot{T}^{-1}\dot{\Theta}_d + T^{-1}\ddot{\Theta}_d = f_\Omega(x_\Omega) = f_\Omega\end{aligned} \tag{42}$$

Next, define the estimates of Θ_d and Ω_d as $\hat{\Theta}_d$ and $\hat{\Omega}_d$, respectively, and the estimation error as $\tilde{\Theta}_d = \Theta_d - \hat{\Theta}_d$. Then, the dynamics of the proposed NN virtual control inputs become

$$\begin{aligned}\dot{\hat{\Theta}}_d &= T\hat{\Omega}_d + K_{\Omega 1}\tilde{\Theta}_d \\ \dot{\hat{\Omega}}_d &= \hat{f}_{\Omega 1} + K_{\Omega 2}T^{-1}\tilde{\Theta}_d\end{aligned} \tag{43}$$

where $K_{\Omega 1}$ and $K_{\Omega 2}$ are positive constants. The estimate $\hat{\omega}_d$ is then written as

$$\hat{\omega}_d = \hat{\Omega}_d + T^{-1}K_\Theta e_\Theta + K_{\Omega 3}T^{-1}\tilde{\Theta}_d \tag{44}$$

where $K_{\Omega 3}$ is another positive constant. Observing $\tilde{\omega}_d = \omega_d - \hat{\omega}_d = \tilde{\Omega}_d - K_{\Omega 3}T^{-1}\tilde{\Theta}_d$, subtracting (43) from (42), as well as adding and subtracting $T^T\tilde{\Theta}_d$ and $K_{\Omega 3}\dot{T}^{-1}\tilde{\Theta}_d$, the virtual controller estimation error dynamics are found to be

$$\begin{aligned}\dot{\tilde{\Theta}}_d &= T\tilde{\omega}_d - (K_{\Omega 1} - K_{\Omega 3})\tilde{\Theta}_d \\ \dot{\tilde{\Omega}}_d &= (f_\Omega + T^T\tilde{\Theta}_d - K_{\Omega 3}\dot{T}^{-1}\tilde{\Theta}_d) - \hat{f}_{\Omega 1} - K_{\Omega 2}T^{-1}\tilde{\Theta}_d - T^T\tilde{\Theta}_d + K_{\Omega 3}\dot{T}^{-1}\tilde{\Theta}_d\end{aligned}. \tag{45}$$

In (43), *universal approximation property* of NN has been utilized to estimate the unknown function $f_{\Omega 1}(x_\Omega) = f_\Omega + T^T\tilde{\Theta}_d - K_{\Omega 3}\dot{T}^{-1}\tilde{\Theta}_d$ as $f_{\Omega 1}(x_\Omega) = W_\Omega^T\sigma\left(V_\Omega^T x_\Omega\right) + \varepsilon_\Omega$ by target weights W_Ω^T, V_Ω^T such that $\left\|W_\Omega\right\|_F \le W_{M\Omega}$ for a known constant $W_{M\Omega}$ and ε_Ω is the NN approximation error such that $\left\|\varepsilon_\Omega\right\| \le \varepsilon_{\Omega M}$ for a constant $\varepsilon_{\Omega M}$. The NN estimate of f_Ω is written as $\hat{f}_{\Omega 1} = \hat{W}_\Omega^T\sigma\left(V_\Omega^T\hat{x}_\Omega\right) = \hat{W}_\Omega^T\hat{\sigma}_\Omega$ where $\hat{W}_\Omega^T$ is the NN estimate of W_Ω^T and $\hat{x}_\Omega$ is the NN input written in terms of the virtual control input estimates and the NN observer velocity estimates. The NN input is selected as $\hat{x}_\Omega = [1\ \rho_d\ \dot{\rho}_d^T\ \ddot{\rho}_d^T\ \dddot{\rho}_d^T\ \Theta_d^T\ \hat{\Omega}_d^T\ \hat{V}^T\ \tilde{\Theta}_d^T]^T$.

Next, adding and subtracting $W_\Omega^T\hat{\sigma}_\Omega$ to the derivative of $\widetilde{\omega}_d$, the estimation error dynamics of (44) are found to be

$$\dot{\widetilde{\omega}}_d = -K_{\Omega 3}\widetilde{\omega}_d + \widetilde{f}_{\Omega 1} - T^{-1}\left(K_{\Omega 2} - K_{\Omega 3}(K_{\Omega 1} - K_{\Omega 3})\right)\widetilde{\Theta}_d - T^T\widetilde{\Theta}_d + \xi_\Omega \tag{46}$$

where $\widetilde{f}_{\Omega 1} = \widetilde{W}_{\Omega 1}^T\hat{\sigma}_\Omega$, $\widetilde{W}_\Omega^T = W_\Omega^T - \hat{W}_\Omega^T$, $\xi_\Omega = \varepsilon_\Omega + W_\Omega^T\widetilde{\sigma}_\Omega$, and $\widetilde{\sigma}_\Omega = \sigma_\Omega - \hat{\sigma}_\Omega$. Furthermore, $\|\xi_\Omega\| \le \xi_{\Omega M}$ with $\xi_{\Omega M}$ a positive computable constant defined as $\xi_{\Omega M} = \varepsilon_{\Omega M} + 2W_{M\Omega}\sqrt{N_\Omega}$ where N_Ω is the number of hidden layer neurons and the fact $\|\sigma_\Omega\| \le \sqrt{N_\Omega}$ was used.

Examination of (45) and (46) reveals $\widetilde{\Theta}_d = 0, \widetilde{\omega}_d = 0$, and $\widetilde{f}_{\Omega 1} = 0$ to be equilibrium points of the estimation error dynamics. In the following theorem, it will be shown that the NN virtual controller successfully estimates the desired orientation Θ_d and angular velocity ω_d. The stability of the position, orientation and velocity error systems will be considered in the following section where the dynamical output feedback control law is designed.

Theorem 2: (Virtual Controller Stability) Let the NN virtual controller be defined by (43) and (44), respectively, with the NN update law provided by

$$\dot{\hat{W}}_\Omega = F_\Omega\hat{\sigma}_\Omega\widetilde{\Theta}_d^T - \kappa_{\Omega 1}F_\Omega\hat{W}_\Omega \tag{47}$$

where $F_\Omega = F_\Omega^T > 0$ and $\kappa_{\Omega 1} > 0$ are design parameters. Then there exists positive design constants $K_{\Omega 1} > K_{\Omega 3} + (N_\Omega)/\kappa_{\Omega 1}$, $K_{\Omega 2} = K_{\Omega 3}(K_{\Omega 1} - K_{\Omega 3})$, and $K_{\Omega 3} > (2N_\Omega)/\kappa_{\Omega 1}$ where N_Ω the number of hidden layer neurons, such that the virtual controller estimation errors $\widetilde{\Theta}_d$, $\widetilde{\omega}_d$ and the virtual control NN weight estimation errors, $\widetilde{W}_\Omega$, are *SGUUB*.

Proof: Consider the following positive definite Lyapunov candidate

$$V_\Omega = \frac{1}{2}\widetilde{\Theta}_d{}^T\widetilde{\Theta}_d + \frac{1}{2}\widetilde{\omega}_d{}^T\widetilde{\omega}_d + \frac{1}{2}tr\{\widetilde{W}_\Omega^T F_\Omega^{-1}\widetilde{W}_\Omega\} \tag{48}$$

whose first derivative with respect to time is given by $\dot{V}_\Omega = \widetilde{\Theta}_d{}^T\dot{\widetilde{\Theta}}_d + \widetilde{\omega}_d{}^T\dot{\widetilde{\omega}}_d + tr\{\widetilde{W}_\Omega^T F_\Omega^{-1}\dot{\widetilde{W}}_\Omega\}$. Substitution of the virtual control closed loop estimation error dynamics (45) and (46) as well as the NN update law (47) reveals

$$\dot{V}_\Omega = -\widetilde{\Theta}_d{}^T(K_{\Omega 1} - K_{\Omega 3})\widetilde{\Theta}_d - K_{\Omega 3}\widetilde{\omega}_d{}^T\widetilde{\omega}_d + \widetilde{\omega}_d{}^T\xi_\Omega - tr\{\widetilde{W}_\Omega^T(\hat{\sigma}_\Omega\widetilde{\Theta}_d^T - \hat{\sigma}_\Omega\widetilde{\omega}_d{}^T - \kappa_{\Omega 1}\hat{W}_\Omega)\}.$$

Observing $\|\hat{\sigma}_\Omega\| \le \sqrt{N_\Omega}$, $\|W_\Omega\|_F \le W_{M\Omega}$ for a known constant, $W_{M\Omega}$, and $tr\{\widetilde{W}_\Omega^T(W_\Omega - \widetilde{W}_\Omega)\} \le \|\widetilde{W}_\Omega\|_F W_{M\Omega} - \|\widetilde{W}_\Omega\|_F^2$, $\dot{V}_\Omega$ can then be rewritten as

$$\begin{aligned}\dot{V}_\Omega \le & -(K_{\Omega 1} - K_{\Omega 3})\|\widetilde{\Theta}_d\|^2 - K_{\Omega 3}\|\widetilde{\omega}_d\|^2 + \|\widetilde{\omega}_d\|\xi_{\Omega M} + \sqrt{N_\Omega}\|\widetilde{\Theta}_d\|\|\widetilde{W}_\Omega\|_F + \sqrt{N_\Omega}\|\widetilde{\omega}_d\|\|\widetilde{W}_\Omega\|_F \\ & + \kappa_{\Omega 1}\|\widetilde{W}_\Omega\|_F W_{M\Omega} - \kappa_{\Omega 1}\|\widetilde{W}_\Omega\|_F^2\end{aligned} \tag{49}$$

Now, completing the squares with respect to $\|\widetilde{W}_\Omega\|_F$, $\|\widetilde{\Theta}_d\|$, and $\|\widetilde{\omega}_d\|$, (49) can be rewritten as

$$\dot{V}_\Omega \le -\left(K_{\Omega 1} - K_{\Omega 3} - \frac{N_\Omega}{\kappa_{\Omega 1}}\right)\|\widetilde{\Theta}_d\|^2 - \left(\frac{K_{\Omega 3}}{2} - \frac{N_\Omega}{\kappa_{\Omega 1}}\right)\|\widetilde{\omega}_d\|^2 - \frac{\kappa_{\Omega 1}}{4}\|\widetilde{W}_\Omega\|_F^2 + \eta_\Omega \tag{50}$$

where $\eta_\Omega = \kappa_{\Omega 1}W_{M\Omega}^2 + \xi_{\Omega M}^2/(2K_{\Omega 3})$ which is dependent on the bounds of the NN reconstruction errors, target NN weights, and design parameters. Examining, η_o, it is clear that the constant terms can be made arbitrarily small by decreasing $\kappa_{\Omega 1}$ and increasing $K_{\Omega 3}$.

Finally, (50) is less than zero provided $K_{\Omega 1} > K_{\Omega 3} + (N_\Omega)/\kappa_{\Omega 1}$ and $K_{\Omega 3} > (2N_\Omega)/\kappa_{\Omega 1}$, and the following inequalities hold:

$$\|\widetilde{\Theta}_d\| > \sqrt{\frac{\eta_\Omega}{(K_{\Omega 1} - K_{\Omega 3} - N_\Omega/\kappa_{\Omega 1})}} \text{ or } \|\widetilde{\omega}_d\| > \frac{\eta_\Omega}{(K_{\Omega 3}/2 - N_\Omega/\kappa_{\Omega 1})} \text{ or } \|\widetilde{W}_\Omega\|_F > \sqrt{\frac{4\eta_\Omega}{\kappa_{\Omega 1}}}. \tag{51}$$

Therefore, it can be concluded using standard extensions of Lyapunov theory [12] that $\dot{V}_\Omega$ is less than zero outside of a compact set, revealing the virtual controller estimation errors, $\widetilde{\Theta}_d$, $\widetilde{\omega}_d$, and the NN weight estimation errors, $\widetilde{W}_\Omega$, to be bounded. Examining the inequalities in (51), it is clear that the error bounds can be made arbitrarily small by proper selection of the gains. In addition, the initial compact set can be made arbitrarily large by proper selection of the gains, and the stability result becomes *SGUUB* [7].

In the following section, the actual control inputs to the dynamics system (5) are derived.

D. NN Output Feedback Control Law

In this section, the information provided by the NN observer, kinematic controller, and NN virtual controller are used to derive the actual inputs u_1 and u_2 to the dynamic system (5). The inputs u_1 and u_2 are calculated so that the desired lift velocity v_{dz} and desired angular velocity ω_d are tracked and the overall control objective is met.

First, the thrust control input, u_1, will be addressed. Consider again the translational velocity tracking error dynamics written in terms of the observer velocity estimates (28). The thrust control input is found by considering the dynamics of the third error state $\dot{e}_{vz}$ in (28) as

$$\begin{aligned} u_1 = {} & mk_{v3}\hat{e}_{vz} + m(c_{\phi d}s_{\theta d}c_{\psi d} + s_{\phi d}s_{\psi d})(\ddot{x}_d + k_{\rho x}\dot{x}_d - \hat{v}_{R1} + \hat{f}_{c11}) \\ & + m(c_{\phi d}s_{\theta d}s_{\psi d} - s_{\phi d}c_{\psi d})(\ddot{y}_d + k_{\rho y}\dot{y}_d - \hat{v}_{R2} + \hat{f}_{c12}) + mc_{\phi d}c_{\theta d}(\ddot{z}_d + k_{\rho z}\dot{z}_d - \hat{v}_{R3} + \hat{f}_{c13} - g) \end{aligned} \tag{52}$$

where $[\hat{f}_{c11} \;\; \hat{f}_{c12} \;\; \hat{f}_{c13}]^T = [\hat{W}_{c11}^T\hat{\sigma}_{c1} \;\; \hat{W}_{c12}^T\hat{\sigma}_{c1} \;\; \hat{W}_{c13}^T\hat{\sigma}_{c1}]^T$ is the NN estimate previously defined in *Section III.B*. Next, the closed loop translational velocity tracking dynamics

are formed by substituting the virtual control inputs (34) and (35) as well as the thrust (52) into (26) and adding and subtracting $R_d^T W_{c1}^T \hat{\sigma}_{c1}^T$ to reveal

$$\begin{aligned} \dot{e}_v &= -S(\omega)e_v - K_v\hat{e}_v + R_d^T\tilde{W}_{c1}^T\hat{\sigma}_{c1}^T + \xi_{c1} \\ &= -(S(\omega)+K_v)e_v - K_v\tilde{v} + R_d^T\tilde{W}_{c1}^T\hat{\sigma}_{c1}^T + \xi_{c1} \end{aligned}. \tag{53}$$

where $\xi_{c1} = R_d^T W_{c1}^T \tilde{\sigma}_{c1}^T + R_d^T \varepsilon_{c1} - \bar{\tau}_{d1}$, $\tilde{W}_{c1} = W_{c1} - \hat{W}_{c1}$, and $\tilde{\sigma}_{c1} = \sigma_{c1} - \hat{\sigma}_{c1}$. Further $\|R_d\|_F = R_{d\max}$ for a known constant $R_{d\max}$, and $\|\xi_{c1}\| \le \xi_{Mc1}$ for a computable constant $\xi_{Mc1} = R_{d\max}\varepsilon_{Mc1} + 2R_{d\max}W_{Mc1}\sqrt{N_c} + M_M\tau_M$ where M_M was defined in *Section III.A*, with N_{c1} is the number of hidden layer neurons.

Remark 5: In the formulation of (53), the expressions for the desired pitch and roll (34) and (35), respectively, were first written in the form of (31) and (32), so that sine and cosine of the angles could be substituted as opposed to substituting the arctangent expressions directly into the sine or cosine function.

Next, the rotational torques, u_2, will be addressed. First, the open loop angular velocity tracking error system is formed by multiplying the angular velocity tracking error (39) by the inertial matrix *J,* taking the first derivative with respect to time, substitute the UAV dynamics (5) and adding and subtracting $T^T e_\Theta$ to get

$$\begin{aligned} J\dot{e}_\omega &= J\dot{\omega}_d - J\dot{\omega} = (J\dot{\omega}_d - S(J\omega)\omega - N_2(\omega) + T^T e_\Theta) - u_2 - \tau_{d2} - T^T e_\Theta \\ &= f_{c2}(x_{c2}) - u_2 - \tau_{d2} - T^T e_\Theta \end{aligned}. \tag{54}$$

where $f_{c2}(x_{c2}) = f_{c2} = J\dot{\omega}_d - S(J\omega)\omega - N_2(\omega) + T^T e_\Theta \in \Re^3$, and unknown. Therefore, the *universal approximation property* of NN is utilized to estimate the function $f_{c2}(x_{c2}) = W_{c2}^T\sigma(V_{c2}^T x_{c2}) + \varepsilon_{c2}$ by target weights W_{c2}^T, V_{c2}^T such that $\|W_{c2}\|_F \le W_{Mc2}$ for a known constant W_{Mc2} and ε_{c2} is the NN functional reconstruction error such that $\|\varepsilon_{c2}\| \le \varepsilon_{Mc2}$ for a

known constant ε_{Mc2}. The NN estimate is given by $\hat{f}_{c2} = \hat{W}_{c2}^T \sigma(V_{c2}^T \hat{x}_{c2}) = \hat{W}_{c2}^T \hat{\sigma}_{c2}$ where $\hat{W}_{c2}^T$ is the NN estimate of W_{c2}^T and $\hat{x}_{c2} = [1 \ \hat{\omega}^T \ \dot{\hat{\Omega}}_d^T \ \widetilde{\Theta}_d{}^T \ e_\Theta^T]^T$ is the NN input written in terms of the observer and virtual controller estimates. By the construction of the virtual controller, $\dot{\hat{\omega}}_d$ is not directly available; therefore, observing (44), the terms $\dot{\hat{\Omega}}_d^T$, $\widetilde{\Theta}_d^{b\,T}$, and e_Θ^T have been included instead.

Similarly to the translational velocity tracking error, the angular velocity tracking error is not measureable. Thus, the estimated angular velocity tracking error is defined in terms of the NN virtual control estimate of $\hat{\omega}_d$ in (44) and the NN observer estimate of $\hat{\omega}$ in (12) and written as

$$\hat{e}_\omega = \hat{\omega}_d - \hat{\omega} = e_\omega - \widetilde{\omega}_d + \widetilde{\omega}. \tag{55}$$

Using the NN estimate $\hat{f}_{c2}$ and (55), the rotational torque control input is calculated as

$$u_2 = \hat{f}_{c2} + K_\omega \hat{e}_\omega. \tag{56}$$

Substituting the control input (56) into the angular velocity dynamics (55) as well as adding and subtracting $W_{c2}^T \hat{\sigma}_c$, the closed loop dynamics become

$$J\dot{e}_\omega = \widetilde{W}_{c2}^T \hat{\sigma}_{c2} - K_\omega e_\omega + K_\omega \widetilde{\omega}_d - K_\omega \widetilde{\omega} - T^T e_\Theta + \xi_{c2} \tag{57}$$

where $\widetilde{W}_{c2}^T = W_{c2}^T - \hat{W}_{c2}^T$, $\xi_{c2} = \varepsilon_{c2} + W_{c2}^T \widetilde{\sigma}_c - \tau_{d2}$, and $\widetilde{\sigma}_{c2} = \sigma_{c2} - \hat{\sigma}_{c2}$. Further $\left\|\xi_{c2}\right\| \le \xi_{Mc2}$ for a computable constant $\xi_{Mc2} = \varepsilon_{Mc2} + 2W_{Mc2}\sqrt{N_{c2}} + \tau_{dM}$ where N_{c2} is the number of hidden layer neurons.

As a final step, we define an augmented translational and angular velocity error system as $e_S = [e_v^T \;\; e_\omega^T]^T$ whose closed loop dynamics are described by (53) and (57), respectively, and rewritten as

$$\bar{J}\dot{e}_S = A_d^T \tilde{f}_c - \left(K_S + S_S(\omega)\right)e_S - K_S\tilde{V} - \bar{T}^T\bar{e}_\Theta + K_S\tilde{\bar{\omega}}_d + \xi_c \tag{58}$$

where $\bar{J} = [I_{3x3} \;\; 0_{3x3}; 0_{3x3} \;\; J] \in \Re^{6x6}$ is a constant, $K_S = [K_v \;\; 0_{3x3}; 0_{3x3} \;\; K_\omega] > 0 \in \Re^{6x6}$, $S_S(\omega) = [S(\omega)\; 0_{3x3}; 0_{3x3}\; 0_{3x3}] \in \Re^{6x6}$, $e_S^T S_S(\omega) e_S = 0$, $\tilde{V}$ is the velocity tracking error vector defined previously in the observer development, $\bar{T} = [0_{3x6}; 0_{3x3} \;\; T]$, $\bar{e}_\Theta = [0_{1x3} \;\; e_\Theta^T]^T$, $\tilde{\bar{\omega}}_d = [0_{1x3} \;\; \tilde{\omega}_d^T]^T$, $\xi_c = [\xi_{c1}^T \;\; \xi_{c2}^T]^T \in \Re^6$, and $\|\xi_c\| \le \xi_{Mc}$ for a positive computable constant $\xi_{Mc} = \sqrt{\xi_{Mc1}^2 + \xi_{Mc2}^2}$. Additionally, $\tilde{f}_c = \tilde{W}_c^T \hat{\sigma}_c \in \Re^6$ with $A_d = diag\{R_d, I_{3x3}\} \in \Re^{6x6}$, $\tilde{W}_c = diag\{\tilde{W}_{c1}, \tilde{W}_{c2}\}$ and $\hat{\sigma}_c = [\hat{\sigma}_{c1}^T \;\; \hat{\sigma}_{c2}^T]^T$. Examining (58) reveals $e_S, \bar{e}_\Theta, \tilde{\bar{\omega}}_d$ and $\tilde{f}_c$ to be equilibrium points of the augmented error dynamics when $\|\xi_c\| = 0$. Further, a single NN is utilized to estimate $\hat{f}_c = [\hat{f}_{c1}^T \;\; \hat{f}_{c2}^T]^T \in \Re^6$.

Figure 2 now illustrates a general control structure for the proposed NN output feedback control law. Examining the figure, five connected systems are observed: the NN observer, kinematic controller, NN virtual controller, NN output feedback controller, and the UAV dynamic system.

The external inputs to the system are considered to be the desired position, ρ_d, and desired yaw, ψ_d. Based on the difference between the current UAV position (ρ) and the desired position, the kinematic controller generates the desired velocity v_d to ensure $\rho \to \rho_d$. Subsequently, the desired pitch, θ_d, and roll, ϕ_d, are calculated to ensure the x

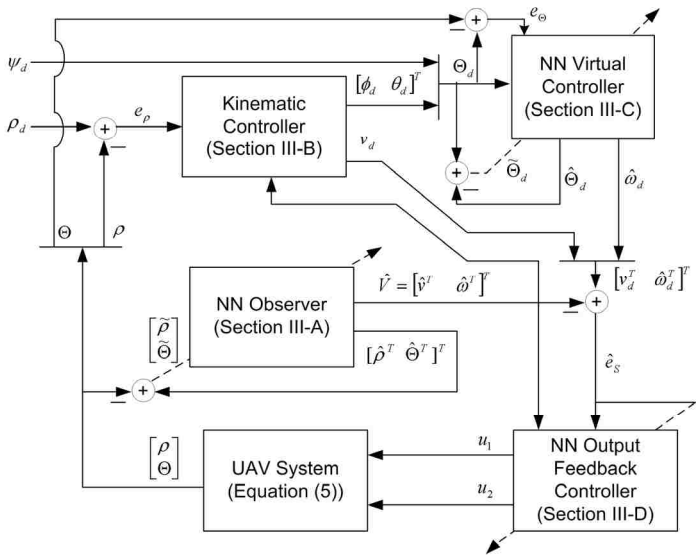


Fig. 2. NN output feedback control structure.

and y components of the desired velocity are tracked, respectively. Then, the NN virtual controller uses the information provided by the NN observer and kinematic controller to generate the desired angular velocity $\omega_d \in E^b$ which ensures $\Theta \to \Theta_d$. Then, the NN output feedback controller calculates the actual control inputs u_1 and u_1 based on the information provided by the kinematic controller, NN virtual controller, and the NN observer. The control inputs are then applied to the UAV system whose measurable output vector consists of the UAV position and orientation. The output vector is then fed back into the kinematic system as well as the NN observer.

In the final theorem, the stability of the entire system which includes position, orientation, and velocity tracking errors are considered along with the estimation errors of the observer and virtual controller and the NN weight estimation errors. Considering the entire system in a single Lyapunov candidate allows us to relax the separation principle.

Theorem 3: (Quadrotor UAV System Stability) Given the dynamic system of a quadrotor UAV in (5), let the NN observer be defined by (11) and (12), respectively, with the NN update law for the observer provided by (15). Given a smooth desired trajectory, let the desired translational velocity for the UAV to track be defined by (22) with the desired pitch and roll defined by (35) and (34), respectively. Let the NN virtual controller be provided by (43) and (44), respectively, with the NN update law given by (47). Let the dynamic NN controller be defined by (52) and (56), respectively, with the NN update given by

$$\dot{\hat{W}}_c = F_c \hat{\sigma}_c \left(A_d \hat{e}_S \right)^T - \kappa_{c1} F_c \hat{W}_c \tag{60}$$

where $F_c = F_c^T > 0$ and $\kappa_{c1} > 0$ are constant design parameters. Then there exists positive design constants $K_{o1}, K_{o2}, K_{o3}, K_{\Omega 1}, K_{\Omega 2}, K_{\Omega 3}$, and positive definite design matrices $K_\rho, K_\Theta, K_v, K_\omega$, such that observer estimation errors $\tilde{X}, \tilde{V}$ and the NN observer weight estimation errors, $\tilde{W}_o$, the virtual controller estimation errors $\tilde{\Theta}_d, \tilde{\omega}_d$ and the virtual control NN weight estimation errors, $\tilde{W}_\Omega$, the position, orientation, and translational and angular velocity tracking errors, e_ρ, e_Θ, e_S, respectively, and the dynamic controller NN weight estimation errors, $\tilde{W}_c$, are all *SGUUB*.

Proof: Consider the following positive definite Lyapunov candidate

$$V_{UAV} = K_{S\max}^2 V_o + K_{S\max}^2 V_\Omega + V_c$$

where V_o and V_Ω were defined in (16) and (48), respectively, $K_{S\max}$ is the maximum singular value of K_S, and

$$V_c = \frac{1}{2} e_\rho^T e_\rho + \frac{1}{2} e_\Theta^T e_\Theta + \frac{1}{2} e_S^T \bar{J} e_S + \frac{1}{2} tr\left\{ \tilde{W}_c^T F_c^{-1} \tilde{W}_c \right\}. \tag{61}$$

The first derivative of V_{UAV} with respect to time is given by $\dot{V}_{UAV} = K_{S\max}^2 \dot{V}_o + K_{S\max}^2 \dot{V}_\Omega + \dot{V}_c$. In *Theorem 1*, it was found that $\dot{V}_o$ could be upper bounded by (18) while in *Theorem 2*, the upper bound of $\dot{V}_\Omega$ was found to be (50). Now, observing $\dot{V}_c = e_\rho^T \dot{e}_\rho + e_\Theta^T \dot{e}_\Theta + e_S^T \bar{J} \dot{e}_S + tr\{\tilde{W}_c^T F_c^{-1} \dot{\tilde{W}}_c\}$, and substitution of the closed loop error dynamics (24), (40), and (58) yields

$$\dot{V}_c = -e_\rho^T K_\rho e_\rho + e_\rho^T R\, e_v - e_\Theta^T K_\Theta e_\Theta - e_S^T K_S e_S - e_S^T K_S \tilde{V} + e_S^T K_S \tilde{\bar{\omega}}_d + e_S^T \xi_c + tr\{\tilde{W}_c^T (F_c^{-1} \dot{\hat{W}}_c + \hat{\sigma}_c (A_d e_S)^T)\}$$

after simplification. Next defining $e_K = [e_\rho^T \;\; e_\Theta^T]^T$, $\Pi = [R \;\; 0_{3x3}; 0_{3x6}]$, and $K_K = [K_\rho \;\; 0_{3x3}; 0_{3x3} \;\; K_\Theta]$, and substituting the NN tuning law (60) into $\dot{V}_c$ reveals

$$\dot{V}_c = -e_K^T K_K e_K - e_S^T K_S e_S + e_K^T \Pi e_S - e_S^T K_S \tilde{V} + e_S^T K_S \tilde{\bar{\omega}}_d + e_S^T \xi_c + tr\{\tilde{W}_c^T (\hat{\sigma}_c (\tilde{\bar{\omega}}_d - \tilde{V}) A_d^T + \kappa_{c1} \hat{W}_c)\}.$$

Observing $\|\Pi\|_F < \Pi_{\max}$ and $\|W_c\|_F \le W_{Mc}$ for known constants, $\Pi_{\max}$ and $W_{c\Omega}$, $\|\tilde{\bar{\omega}}_d\| = \|\tilde{\omega}_d\|$, and $tr\{\tilde{W}_c^T (W_c - \tilde{W}_c)\} \le \|\tilde{W}_c\|_F W_{c\Omega} - \|\tilde{W}_c\|_F^2$, and an upper bound for $\dot{V}_c$ is written as

$$\begin{aligned} \dot{V}_c \le & -K_{K\min} \|e_K\|^2 - K_{S\min} \|e_S\|^2 + \Pi_{\max} \|e_K\| \|e_S\| + K_{S\max} \|\tilde{V}\| \|e_S\| + K_{S\max} \|e_S\| \|\tilde{\omega}_d\| + \|e_S\| \xi_{Mc} \\ & + \|\tilde{W}_c\|_F (\|\tilde{\omega}_d\| + \|\tilde{V}\|) A_{dM} \sqrt{N_c} + \kappa_{c1} \|\tilde{W}_c\|_F W_{c\Omega} - \kappa_{c1} \|\tilde{W}_c\|_F^2 \end{aligned}$$

where $K_{K\min}$ and $K_{S\min}$ are the minimum singular values of K_K and K_S, respectively, and $\|A_d\|_F \le A_{dM}$ for a known constant A_{dM}. Next, completing the squares with respect to $\|e_K\|$, $\|\tilde{W}_c\|_F$ and $\|e_S\|$, $\dot{V}_c$ can be upper bounded as

$$\begin{aligned} \dot{V}_c \le & -\frac{K_{K\min}}{2} \|e_K\|^2 - \frac{1}{2} \left(K_{S\min} - \frac{\Pi_{\max}^2}{K_{K\min}} - 2 \right) \|e_S\|^2 - \frac{\kappa_{c1}}{6} \|\tilde{W}_c\|_F^2 + \eta_c \\ & + \frac{1}{2} \left(K_{S\max}^2 + 3 \frac{(A_{dM} \sqrt{N_c})^2}{\kappa_{c1}} \right) \|\tilde{V}\|^2 + \frac{1}{2} \left(K_{S\max}^2 + 3 \frac{(A_{dM} \sqrt{N_c})^2}{\kappa_{c1}} \right) \|\tilde{\omega}_d\|^2 \end{aligned} \tag{62}$$

where $\eta_c = \kappa_{c1} W_{Mc}^2 / 2 + \xi_{cM}^2 / (2K_{S\min})$. Finally, combining the results of (18), (50), and (62), the upper bound of $\dot{V}_{UAV}$ can be written as

$$\begin{aligned}\dot{V}_{UAV} \le & -K_{S\max}^2 \left(\frac{(K_{o1} - K_{o3})}{2} - \frac{N_o}{\kappa_{o1}} \right) \left\| \tilde{X} \right\|^2 - \frac{K_{S\max}^2}{2} \left(K_{o3} - \frac{2N_o}{\kappa_{o1}} - 1 - \frac{3(A_{dM}\sqrt{N_c})^2}{\kappa_{c1} K_{S\max}^2} \right) \left\| \tilde{V} \right\|^2 \\ & - K_{S\max}^2 \left(K_{\Omega 1} - K_{\Omega 3} - \frac{N_\Omega}{\kappa_{\Omega 1}} \right) \left\| \tilde{\Theta}_d \right\|^2 - \frac{K_{S\max}^2}{2} \left(K_{\Omega 3} - \frac{2N_\Omega}{\kappa_{\Omega 1}} - 1 - \frac{3(A_{dM}\sqrt{N_c})^2}{\kappa_{c1} K_{S\max}^2} \right) \left\| \tilde{\omega}_d \right\|^2 \\ & - \frac{K_{K\min}}{2} \left\| e_K \right\|^2 - \frac{1}{2} \left(K_{S\min} - \frac{\Pi_{\max}^2}{K_{K\min}} - 2 \right) \left\| e_S \right\|^2 - \frac{\kappa_{c1}}{6} \left\| \tilde{W}_c \right\|_F^2 - \frac{\kappa_{o1} K_{S\max}^2}{4} \left\| \tilde{W}_o \right\|_F^2 - \frac{\kappa_{\Omega 1} K_{S\max}^2}{4} \left\| \tilde{W}_\Omega \right\|_F^2 + \eta_{UAV}\end{aligned} \tag{63}$$

where $\eta_{UAV} = K_{S\max}^2 \eta_o + K_{S\max}^2 \eta_\Omega + \eta_c$. The first nine terms in (63) are less than zero provided the controller gains are selected as

$$\begin{aligned} & K_{o1} > K_{o3} + \frac{2N_o}{\kappa_{o1}}, \ K_{o3} > \frac{2N_o}{\kappa_{o1}} + 1 + \frac{3(A_{dM}\sqrt{N_c})^2}{\kappa_{c1} K_{S\max}^2}, \ K_{\Omega 1} > K_{\Omega 3} + \frac{N_\Omega}{\kappa_{\Omega 1}}, \\ & K_{\Omega 3} > \frac{2N_\Omega}{\kappa_{\Omega 1}} + 1 + \frac{3(A_{dM}\sqrt{N_c})^2}{\kappa_{c1} K_{S\max}^2}, \ K_{S\min} > \frac{\Pi_{\max}^2}{K_{K\min}} + 2 \end{aligned} \tag{64}$$

Therefore, it can be concluded that $\dot{V}_{UAV}$ is less than zero provided the controller gains are selected according to (64) and the following inequalities holds:

$$\begin{aligned} & \left\| \tilde{X} \right\| > \sqrt{\frac{\eta_{UAV}}{K_{S\max}^2 ((K_{o1} - K_{o3})/2 - N_o/\kappa_{o1})}} \quad \text{or} \quad \left\| \tilde{\Theta}_d \right\| > \sqrt{\frac{\eta_{UAV}}{K_{S\max}^2 (K_{\Omega 1} - K_{\Omega 3} - N_\Omega/\kappa_{\Omega 1})}} \quad \text{or} \quad \left\| e_S \right\| > \sqrt{\frac{2\eta_{UAV}}{(K_{S\min} - \Pi_{\max}^2/K_{K\min} - 2)}} \\ & \text{or} \quad \left\| \tilde{V} \right\| > \sqrt{\frac{\eta_{UAV}}{K_{S\max}^2 \left(K_{o3} - 2N_o/\kappa_{o1} - 1 - 3(A_{dM}\sqrt{N_c})^2/(\kappa_{c1} K_{S\max}^2) \right)/2}} \quad \text{or} \quad \left\| \tilde{W}_\Omega \right\|_F > \sqrt{\frac{4\eta_{UAV}}{\kappa_{\Omega 1} K_{S\max}^2}} \quad \text{or} \quad \left\| e_K \right\| > \sqrt{\frac{2\eta_{UAV}}{K_{K\min}}} \\ & \text{or} \quad \left\| \tilde{\omega}_d \right\| > \sqrt{\frac{\eta_{UAV}}{K_{S\max}^2 \left(K_{\Omega 3} - 2N_\Omega/\kappa_{\Omega 1} - 1 - 3(A_{dM}\sqrt{N_c})^2/(\kappa_{c1} K_{S\max}^2) \right)/2}} \quad \text{or} \quad \left\| \tilde{W}_c \right\|_F > \sqrt{\frac{6\eta_{UAV}}{\kappa_{c1}}} \quad \text{or} \quad \left\| \tilde{W}_o \right\|_F > \sqrt{\frac{4\eta_{UAV}}{\kappa_{o1} K_{S\max}^2}} \end{aligned} \tag{65}$$

It can be concluded using standard Lyapunov extensions [12] that $\dot{V}_{UAV}$ is less than zero outside of a compact set revealing that the observer estimation errors $\tilde{X}$, $\tilde{V}$ and the NN observer weight estimation errors, $\tilde{W}_o$, the virtual controller estimation errors $\tilde{\Theta}_d$, $\tilde{\omega}_d$ and the virtual control NN weight estimation errors, $\tilde{W}_\Omega$, the position, orientation, and velocity tracking errors, e_ρ, e_Θ, e_S, respectively, and the controller NN

weight estimation errors, $\widetilde{W}_c$, are all bounded. Finally, the initial compact set can be made arbitrarily large through proper selection of the gains; thus, all signals are *SGUUB* [7].

Remark 6: Examining (65) reveals the error bounds can be reduced through the appropriate selection of the design parameters. The theoretical results of *Theorem 3* ensure that the estimation and tracking errors remain bounded in the presence of bounded disturbances. Further, examining the error bounds in (65), it is observed that the size of the estimation and error bounds is dependent on the magnitude of the disturbances. As a result, a very large disturbance will lead to potential large error bounds illustrating the relationship between the control system performance and the magnitude of the disturbances.

In the next section, the requirements and considerations for practical implementation of the proposed output feedback control scheme are presented.

E. Comments on Implementation and Practical Considerations

To implement the proposed output feedback control scheme, it is observed that three NN are required. Although this appears to be a computationally demanding algorithm, previous work on the control of spark ignition engines [17] has demonstrated that three NN can be successfully implemented in hardware simultaneously with promising results. In fact, it was found that the total time required to compute the controller calculations was less than $100\,\mu\text{sec}$ [17].

Further, the quantities required as inputs to each NN can be either measured or calculated using current technologies. The position and orientation of the UAV can be measured. The position of the UAV can be measured using global positioning systems

(GPS) enhanced with differential GPS to improve accuracy [18]. Additionally, several vision based approaches to measuring the UAV position have also been reported [19]. Measurement of the attitude of the UAV can easily be achieved using local measurements of an onboard attitude heading reference system (AHRS).

As a result of *Theorem 3*, $\Theta \to \Theta_d$ with small bounded error for $t \geq t_0 + T$. Recalling Θ_d in (34) and (35) is defined only within the interval of $(-\pi/2, -\pi/2)$ of the UAV, it can be concluded that the UAV maintains a stable flight configuration while in the steady state provided the desired trajectory ρ_d is feasible. As demonstrated in (8), certain trajectories are not achievable during stable flight. Thus, in order to guarantee stability in practice, one or both of the following considerations can be undertaken. First, applying the results of *Theorem 3,* the desired trajectory ρ_d should be achievable using steady state pitch and roll movements which satisfy $\theta_d, \phi_d \in (-\pi/2 + \sqrt{2\eta_{UAV}/K_{K\min}}, \pi/2 - \sqrt{2\eta_{UAV}/K_{K\min}})$. As a second consideration, the desired pitch and roll angles can be saturated before they reach $\pm\pi/2$. However, for very demanding trajectories, restricting the desired pitch and roll angles could result in an increase in the tracking error bounds.

During the time prior to $t_0 + T$, the bounds on e_Θ are potentially larger in magnitude than the steady state bounds derived in *Theorem 3.* To counter this fact, the following simple but effective solution is utilized based on practical observations. In practice, helicopters do not immediately begin aggressive maneuvers from the grounded position. They first rise to a safe operating height and then begin their flight pattern. Mimicking this observation, the desired trajectory can be chosen so that the desired pitch

and roll are close to zero for $t < t_0 + T$. For instance, given a desired trajectory ρ_{d1} starting at time t_1, the trajectory tracked by the UAV could be $\rho_d = \rho_{d1}(1 - \exp(-r(t - t_1)))$ where r is the decay rate which can be designed limit the initial aggressiveness of the UAV. Multiplying the desired trajectory ρ_{d1} by the exponential function limits the initial maneuver, but after a finite time period allows the UAV to track the original unconstrained trajectory ρ_{d1}. This approach will be employ in the following section which presents the numerical simulation results of the proposed output feedback control law. Using this technique, the results in the following section demonstrate that the pitch and roll of the UAV remains within the interval $(-\pi/2, \pi/2)$ throughout the duration of the test.

IV. Simulation Results

The quadrotor UAV (5) is now considered in the presence of unmodeled dynamics such as aerodynamic damping [7] and blade flapping [1], and the effectiveness of the NN output feedback control law developed in this work is verified. Additionally, random disturbances are added, and simulations are performed in MATLAB. The unknown aerodynamic effects are modeled as in (7) where the damping coefficients are selected as $[d_1, d_2, d_3, d_4, d_5, d_6]^T = [0.06,\ 0.1, 0.06, 0.1, 0.06, 0.1]^T$ and $[d_7, d_8, d_9, d_{10}, d_{11}, d_{12}]^T = [0.1,\ 0.15, 0.1, 0.15, 0.1, 0.15]^T$. Further, the blade flapping parameter k_β was selected as $k_\beta = 0.75\ N - m/rad$ and α is taken as zero at the beginning the simulation. Then, at $t = 20$ seconds, α steps from zero to twenty degrees, and the robustness of the control law is demonstrated. The sudden increase of α could

represent an occurrence such as a gust of wind. Additionally, a normally distributed noise signal with zero mean and variance of 0.01 is added to the UAV dynamics (5) through τ_d.

The inertial parameters of the UAV are taken to be as $m = 0.9\,kg$ and $J = diag\{0.32, 0.42, 0.63\}\ kg\ m^2$. The desired position ([m]) and yaw ([rad]) for the UAV to track is designated to be $\rho_d = [A_x \cos(\omega_x t)(1-\exp(-r_x t^2))\ \ A_y \sin(\omega_y t)(1-\exp(-r_y t^2))\ \ \ A_z(1-\exp(-r_z t))]^T$, $\psi_d = A_\psi \sin(\omega_\psi t)$ with $A_x = A_y = 10,\ \omega_x = \omega_y = 0.2\pi, r_x = r_y = 0.5,\ \ A_z = -10, r_z = 0.1,$ $A_\psi = 1,$ and $\omega_\psi = 0.3\pi$. Each NN employs 5 hidden layer neurons, and the control gains are selected to be $K_{o1} = 23,\ K_{o2} = 60,\ K_{o3} = 20$, $K_{\Omega 1} = 24,\ K_{\Omega 2} = 80,\ K_{\Omega 3} = 20,$ $K_\rho = diag\{10, 10, 30\}$, $k_{v1} = 10,\ k_{v2} = 10,\ k_{v3} = 30$, $K_\Theta = diag\{30, 30, 30\}$, and $K_\omega = diag\{25, 25, 25\}$ satisfying the constraints mentioned in the theorems. The NN parameters are selected as $F_o = 10, \kappa_{o1} = 0.1$, $F_\Omega = 40, \kappa_{\Omega 1} = 0.1$, and $F_c = 20, \kappa_{c1} = 0.1$.

In the simulation, all tunable NN weights are initialized to 0, while the initial observer estimates of the position and orientation are set to the UAV's initial position of $\hat{X}_o^T(0) = [\rho^T(0)\ \Theta^T(0)]^T = [-0.3\ \ 0.3\ \ 0\ \ 0\ \ 0\ \ 0.1]^T$.

Figure 3 displays the actual trajectory as well as the desired trajectory of the UAV. Additionally, the vector norm of the position error is also shown. Examining the trajectory plot, the desired trajectory starts from the origin while the UAV starts from the initial configuration denoted above, and the UAV quickly converges to the desired course and tracks with a small bounded as the theoretical results of *Theorem 3* suggest. At $t = 20$ seconds, a small peak in the error plot is observed corresponding to the external

disturbance being introduced. However, the NN controller quickly adapts to the changing conditions and the UAV returns to its desired path. The fact that the UAV successfully tracks the desired trajectory confirms that the orientations generated by the NN virtual controller correctly steers the UAV along the desired path as the results of *Theorem 2* imply.

Figure 4 displays the tracking errors for the position, orientation, translational velocity and the angular velocity, respectively, which are each observed to converge to a

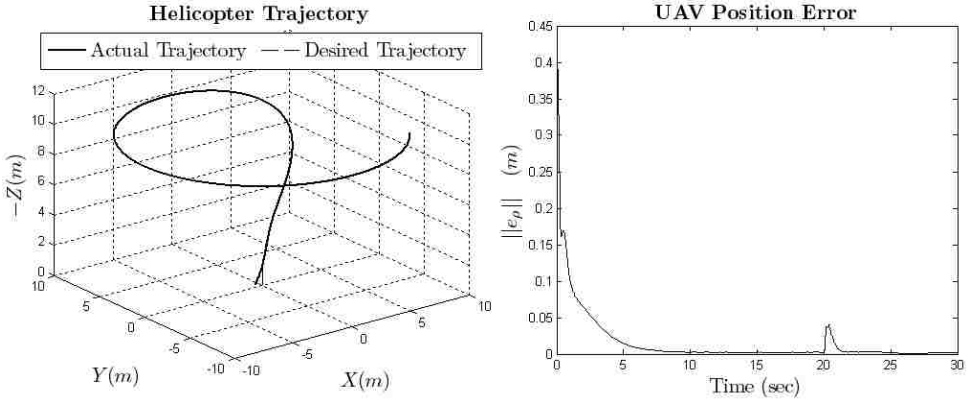


Fig. 3. UAV trajectory tracking.

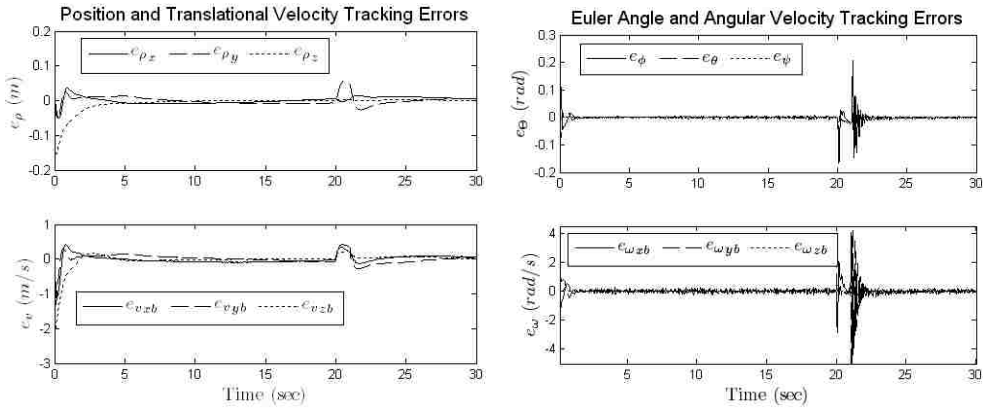


Fig. 4. UAV tracking errors.

small bounded region around the origin. Again, at $t = 20$ seconds, the effect of the external disturbance is visible, and it is observed that the each tracking error quickly returns to zero even though the disturbance itself does not vanish. Instead, the NN adapts so that acceptable tracking performance is regained. The tracking performance of the translational velocities again reinforce the ability of the virtual controller structure to calculate the appropriate pitch and roll angles necessary to achieve tracking.

Figure 5 displays the observer estimation errors for the position, orientation, translational velocities, and angular velocities, respectively, which are observed to converge to a small bounded region near the origin as the conjecture of *Theorem 1* suggested. For the observer position estimation errors in Fig. 5, recall that the position of the UAV is measureable; therefore the initial observer position states are selected as $\hat{\rho}(0) = \rho(0)$, and thus, $\tilde{\rho}(0) = \rho(0) - \hat{\rho}(0) = 0$.

For the remainder of the simulation, the maximum observer position error in Fig. 5 is observed to be less than 0.03. The maximum error of 0.03 is observed in the *y*-coordinate

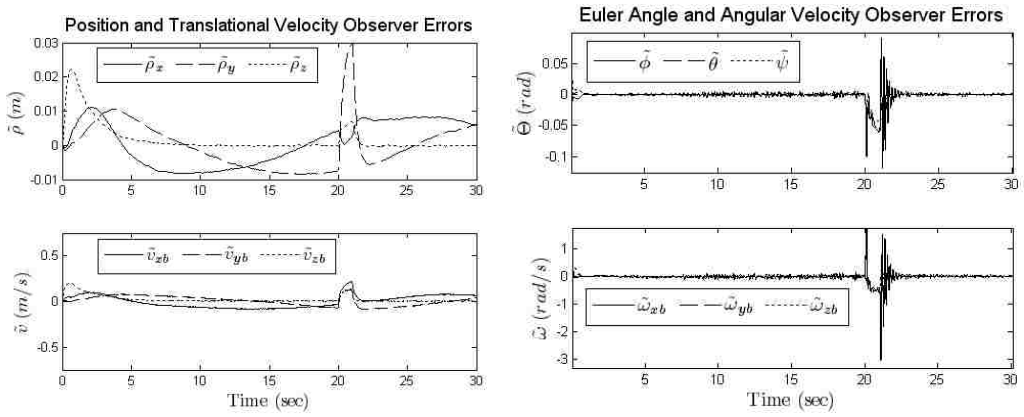


Fig. 5. UAV observer estimation errors.

estimate after introduction of the bounded disturbances. It is observed that the observer estimation errors initially increase when the unknown nonlinearities are introduced, and decrease as the NN observer begins to compensate for the nonlinearities. Moreover, the upper bound of the observer position estimation errors calculated from the simulation is given by $\|\tilde{\rho}(t)\| \le 0.0303$. Table 1 summarized the mean squared error ($MSE(\bullet)$) and maximum observed error ($\max(abs(\bullet))$) for each tracking error and observer estimation error. In each case, the mean squared error is observed to be small. This result is consistent with tracking and estimation performances observed in Fig. 5. Additionally, the maximum values observed for both the tracking and observer estimation errors occur either at the beginning of the simulation or directly after the external disturbance has been introduced. This phenomenon is also observed in the error plots of Fig. 5.

TABLE I: MEAN SQUARED ERRORS AND MAXIMUM ERRORS

Tracking Errors	$e_{\rho x}\ (m)$	$e_{\rho y}\ (m)$	$e_{\rho z}\ (m)$	$e_{vxb}\ (m/s)$	$e_{vyb}\ (m/s)$	$e_{vzb}\ (m/s)$
$MSE(\bullet)$	0.0012	0.0011	7.84×10^{-4}	0.1783	0.0656	0.1851
$\max(abs(\bullet))$	0.3	0.3	0.0054	0.6061	1.7781	3.2656
Observer Errors	$\tilde{\rho}_x\ (m)$	$\tilde{\rho}_y\ (m)$	$\tilde{\rho}_z\ (m)$	$\tilde{v}_{xb}\ (m/s)$	$\tilde{v}_{yb}\ (m/s)$	$\tilde{v}_{zb}\ (m/s)$
$MSE(\bullet)$	4.22×10^{-5}	5.18×10^{-5}	1.70×10^{-5}	0.0047	0.0026	0.0019
$\max(abs(\bullet))$	0.0112	0.0292	0.0220	0.2188	0.1192	0.2098
Tracking Errors	$e_{\phi}\ (rad)$	$e_{\theta}\ (rad)$	$e_{\psi}\ (rad)$	$e_{\omega xb}$ (rad/s)	$e_{\omega yb}$ (rad/s)	$e_{\omega zb}$ (rad/s)
$MSE(\bullet)$	4.94×10^{-4}	5.87×10^{-4}	5.58×10^{-4}	0.0845	0.0744	0.0277
$\max(abs(\bullet))$	0.2784	0.2285	0.1	5.8112	5.3300	2.5508
Observer Errors	$\tilde{\phi}\ (rad)$	$\tilde{\theta}\ (rad)$	$\tilde{\psi}\ (rad)$	$\tilde{\omega}_{xb}\left(\frac{rad}{s}\right)$	$\tilde{\omega}_{yb}\left(\frac{rad}{s}\right)$	$\tilde{\omega}_{zb}\left(\frac{rad}{s}\right)$
$MSE(\bullet)$	3.77×10^{-4}	2.73×10^{-4}	2.59×10^{-4}	0.0822	0.0683	0.0746
$\max(abs(\bullet))$	0.1321	0.0311	0.1828	2.7407	1.5156	2.2353

Finally, Fig. 6 shows the control inputs of the UAV as well as the time history of the unmodeled dynamics and the noise signal. Examining the time history of the unmodeled dynamics, the random noise signal is clearly visible for the entire simulation while the step disturbance is evident starting at 20 seconds. Additionally, the power of the NN controller is revealed when examining the control inputs time history. Starting at 20 seconds, the thrust as well as the rotational torques are clearly compensating for the newly added dynamics. Additionally, the system noise is observed to be most prevalent in the rotationally torques control inputs.

The simulation results verify that the UAV remains within the interval $(-\pi/2, \pi/2)$ throughout the duration of the test. While the noise and the external disturbance introduced at 20 seconds is observed in all of the error signals, the disturbances observed in the angular velocity tracking errors and observer estimation errors are more apparent since the angular velocities of the UAV generate much of the UAV's movements. To see this more clearly, recall that the desired velocity is calculated

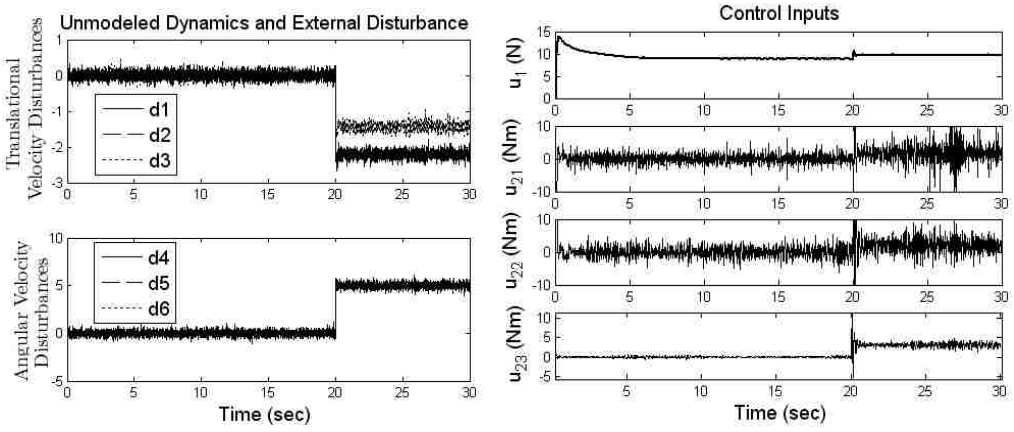


Fig. 6. UAV unmodeled dynamics, external disturbances and control signal.

from measured error values. Then, in order to achieve translational velocity tracking, the desired roll and pitch are calculated. Next, the angular velocity is found to ensure the desired orientation of the UAV is achieved. Finally, rotational torques guarantee that the desired angular velocity is tracked by the UAV. Each stage of the design process contains a proportional tracking controller, and thus, the disturbance and noise is amplified at each stage of the backstepping controller design. This phenomenon is also illustrated in Fig. 6.

As a final assessment of the output feedback control law developed in this work, a state feedback PID control law was implemented to control the translational and angular velocities and all NNs were removed. In the simulation, derivatives such as $\dot{e}_v$ and $\dot{e}_\omega$ that cannot be calculated due to uncertainty, were approximated using backward differences written as $\dot{e}_v \approx e_v(t) - e_v(t-\Delta t) \equiv \dot{\hat{e}}_v$ where $\Delta t = 10^{-6}$ seconds. The PID control laws take the form of $P_{ev} = [P_{ev1}\ P_{ev2}\ P_{ev3}]^T = K_{Pv}e_v + K_{Iv}\int_0^t e_v ds + K_{Dv}\dot{\hat{e}}_v$ and $P_{e\omega} = K_{P\omega}e_\omega + K_{I\omega}\int_0^t e_\omega ds + K_{D\omega}\dot{\hat{e}}_\omega$, and control gains were selected to be $K_{Pv} = diag\{[5\ 5\ 20]\}$, $K_{Iv} = diag\{[7\ 7\ 7]\}$, $K_{Dv} = diag\{[0.5\ 0.5\ 0.5]\}$, $K_{P\omega} = diag\{[20\ 20\ 20]\}$, $K_{I\omega} = diag\{[7\ 7\ 7]\}$, and $K_{D\omega} = diag\{[0.5\ 0.5\ 0.5]\}$. These gains were tuned to ensure acceptable tracking performance while minimizing the overshoot and undershoot of the error signals. Since the derivative terms are being approximated and discretized, $K_{D(\bullet)}$ gains less than one rendered the best performance. All other parameters used in the previous simulation remained unchanged.

To use the PID control law the desired pitch and roll in (34) and (35), respectively, and thrust and rotational torques (52) and (56), respectively, must be

modified by removing the NN estimates and proportional feedback terms and substituting the PID controllers in their place. For example, the rotational torques (56) becomes $u_2 = P_{e\omega}$ while the desired pitch (34) becomes $\theta_d = a\tan\left((c_{\psi d}(\ddot{x}_d + k_{\rho x}\dot{x}_d - v_{R1}) + s_{\psi d}(\ddot{y}_d + k_{\rho y}\dot{y}_d - v_{R2}) + P_{ev1})/(\ddot{z}_d + k_{\rho z}\dot{z}_d - v_{R3} - g)\right)$. The desired roll and thrust control input are modified in a similar manner.

Fig. 7 shows the norm of the position tracking error as well as the control effort used to achieve the tracking performance by a PID controller. Examining the position tracking error, it is evident that the PID controller achieves acceptable tracking performance after a significant amount of gain tuning in the presence of unmodeled dynamics and bounded disturbances. The orientation and the translational and angular velocity tracking errors are also satisfactory although the plots are not shown. However, comparing the control signal required to achieve tracking using PID control in Fig. 7 to the control signal of the NN output feedback controller in Fig. 6, it is clear the PID controller exerts significantly more effort to track the desired trajectory. It was found

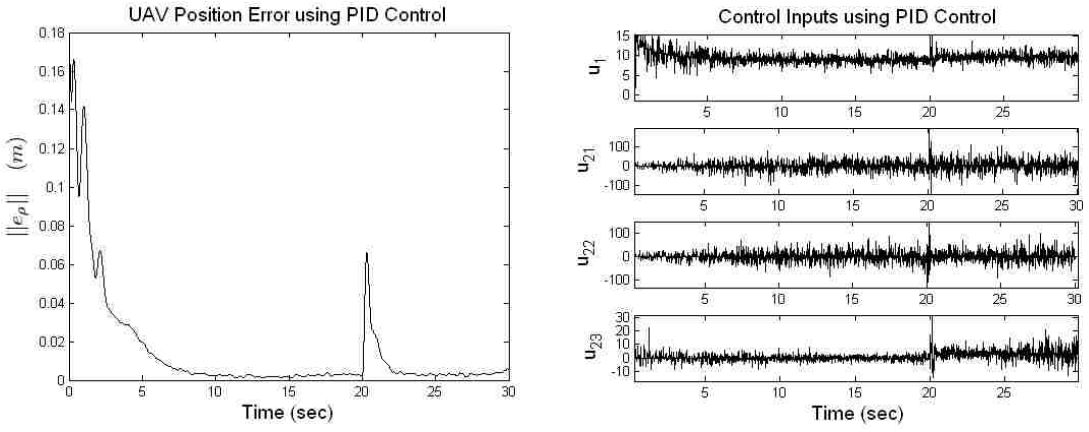


Fig. 7. UAV position error and control signals using PID control.

that that the rotational torque control signals u_{21}, u_{22}, and u_{23} from the PID controller where on average 4.3, 5.1, and 4 times larger the signals generated from the NN output feedback scheme, respectively. On the other hand, the thrust control signal generated by the PID controller was comparable to the thrust control signal of the proposed scheme.

The reason for the difference in control efforts is due to the fact that the PID gains were used to dominate the neglected dynamics in order to ensure an acceptable performance whereas the NN output feedback control law adapted online to learn the unknown dynamics and perform intelligent compensation. Additionally, significant noise amplification is observed in the control signal of the PID control law as a result of using large gains to dominate the UAV dynamics. Thus, the NN output feedback control renders the tracking of the desired trajectory using less control effort than the conventional PID control law while keeping the noise amplification small further demonstrating the effectiveness of the proposed approach. In addition, our approach does not require significant time to tune the controller gains which becomes necessary when using a PID controller with changing operating conditions.

Moreover, examining the control rate shown in Fig. 8, the NN approach generated less than the control rate required by the PID controller. Examining Figs. 6 and 7, noise is observed to be present in the UAV control inputs for both the proposed NN and the conventional PID controllers, respectively. Differentiating the noisy signals leads to large values of the derivatives in both the cases. However, the plots in Fig. 8 reinforce that the online learning based proposed NN controller performs better than a conventional PID controller.

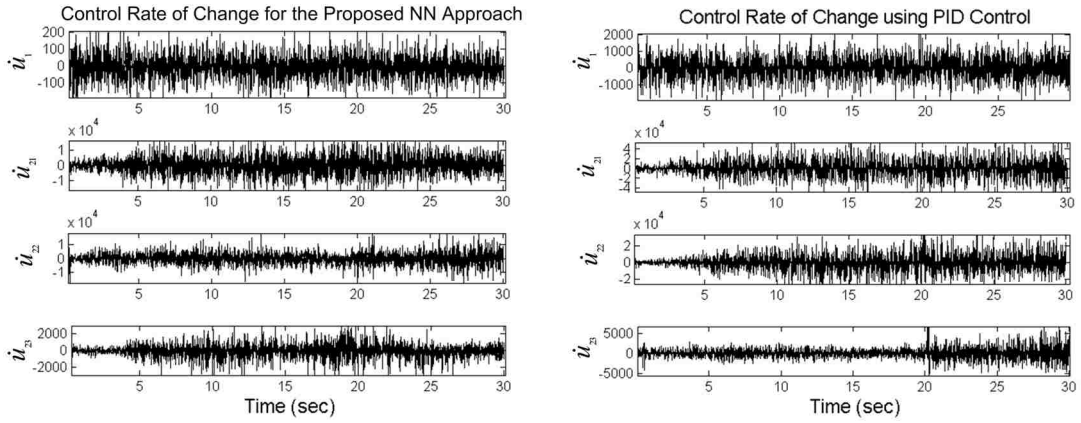


Fig. 8. Control rate of change for the proposed NN controller and a conventional PID controller.

V. CONCLUSIONS

A new NN output feedback control law was developed for an underactuated quadrotor UAV which utilizes the natural constraints of the underactuated system to generate virtual control inputs to guarantee the UAV tracks a desired trajectory. Using the adaptive backstepping technique, all six DOF are successfully tracked using only four control inputs while in the presence of unmodeled dynamics and bounded disturbances. Dynamics and velocity vectors were considered to be unavailable, thus a NN observer was designed to recover the immeasurable states. Then, a novel NN virtual control structure was proposed which allowed the desired translational velocities to be controlled using the pitch and roll of the UAV. Finally, a NN was utilized in the calculation of the actual control inputs for the UAV dynamic system. Using Lyapunov techniques, it was shown that the estimation errors of each NN, the observer, virtual controller, and the position, orientation, and velocity tracking errors were all *SGUUB* while relaxing the separation principle. Numerical results confirm the theoretical conjectures, and the

tracking ability of the UAV in the presence of unmodeled dynamics and bounded disturbances. The proposed controller outperforms a conventional linear controller.

VI. REFERENCES

[1] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: theory and experiment," *Proc. of the AIAA Guidance, Navigation and Control Conference and Exhibit*, South Carolina, pp. 1-20, August 2007.

[2] H. Voos, "Nonlinear state-dependent Riccati equation control of a quadrotor UAV," *Proc. of the IEEE Int. Conf. on Control Applications*, Munich, Germany, pp. 2547-2552, October 2006.

[3] T. Madani and A. Benallegue, "Control of a quadrotor mini-helicopter via full state backstepping technique," *Proc. of the IEEE Conf. on Decision and Control*, San Diego, CA, pp. 1515-1520, December 2006.

[4] N. Guenard, T. Hamel, and V. Moreau, "Dynamic modeling and intuitive control strategy for an X4-flyer," *Proc. of the IEEE Int. Conf. on Control and Automation*, Budapest, Hungary, pp. 141-146, June 2005.

[5] T. Madani and A. Benalleque, "Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles," *Proc. of the American Control Conference*, New York, NY, pp. 5887-5892, July 2007.

[6] L. Besnard, Y. Shtessel, and B. Landrum, "Control of a quadrotor vehicle using sliding mode disturbance observer," *Proc. of the American Control Conference*, New York, NY, pp. 5230-5235, July 2007.

[7] D. Timothy, T. Burg, B. Xian, and D. Dawson, "Output feedback tracking control of an underactuated quad-Rotor UAV," *Proc. of the American Control Conference*, New York, NY, pp. 1775-1780 July 2007.

[8] K. Benzemrane, G. Santosuosso, and G. Damm, "Unmanned aerial vehicle speed estimation via nonlinear adaptive observers," *Proc. of the American Control Conference*, New York, NY, pp. 985-990 July 2007.

[9] H. Voos, "Nonlinear and neural network-based control of a small four-rotor aerial robot," *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp.1-6, September 2007.

[10] J. Dunfied, M. Tarbouchi, and G. Labonte, "Neural network based control of a four rotor helicopter," *Proc. of the IEEE Int. Conf. on Industrial Technology*, pp. 1543-1548, Dec. 2004.

[11] Y. Bestaoui, and R. Slim, "Maneuvers for a quad-rotor autonomous helicopter," *Proc. of the AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, California, pp. 1-18, May 2007.

[12] F.L. Lewis, S. Jagannathan, and A. Yesilderek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, London, 1999.

[13] E. Kim, "Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic," *IEEE Transactions on Fuzzy System,* vol. 12, pp. 368-378, June 2004.

[14] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Transactions on Robotics and Automation,* vol. 15, pp. 301-309, April 1999.

[15] A.E. Neff, L. DongBin, V.K. Chitrakaran, D.M. Dawson, and T.C. Burg, "Velocity control for a quad-rotor uav fly-by-camera interface," *Proc. of the SoutheastCon,* pp. 273-278, March 2007.

[16] S. S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*, Springer Verlag, New York, 1999.

[17] J. Vance, B. Kaul, S. Jagannathan, and J. Drallmeier, "Output feedback controller for operation of spark ignition engines at lean conditions using neural networks," *IEEE Transactions on Control Systems Technology,* vol. 16, pp 214-228, March 2008.

[18] J. Jangy and C.J. Tomlin, "Longitudinal stability augmentation system design for the dragonfly UAV using a single GPS receiver," *Proc. AIAA Guidance, Navigation, and Control Conf. and Exhibit,* August 2003.

[19] E. Altug, J. Ostrowski, R. Mahony, "Control of a quadrotor helicopter using visual feedback," *Proc. IEEE Int. Conf. on Robotics and Automation,* pp. 72-77, May 2002.

[20] V. Puttige and S. Anavatti, "Comparison of real-time online and offline neural network models for a UAV," *Proc. IEEE Int. Conf. on Neural Networks,* pp. 412 – 417, August 2007.

[21] C. Nicol, C. Macnab, and A. Ramirez-Serrano, "Robust neural network control of a quadrotor helicopter," *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering,* pp. 1233-1238, May 2008.

[22] A. Das, F. Lewis, and K. Subbarao, "Neural network based robust backstepping control approach for quadrotors," *Proc. of AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2008.

[23] Y. S. Suh, "Robust control of a quad-rotor aerial vehicle," *Int. Journal of Applied Electromagnetics and Mechanics,* vol. 18, pp. 103-114, 2003.

[24] B. Igelnik and Y. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Networks*, vol. 6, pp. 1320-1329, 1995.

4. Leader-Follower Formation Control of Multiple Quadrotor Unmanned Aerial Vehicles using Neural Networks[1]

Travis Dierks and S. Jagannathan

***Abstract*—*In this paper, a novel framework for leader-follower formation control is developed for the control of multiple unmanned aerial vehicles (UAVs) such as underactuated quadrotor UAVs in three dimensions. Using alternate coordinate system and a desired separation, angle of incidence, and bearing relative to their leader, a desired trajectory is generated online for the follower UAVs through an auxiliary kinematic velocity control thus converting the formation control into an equivalent tracking problem. Then, novel neural network (NN) based virtual and dynamic control laws are introduced to learn the dynamics of the UAVs online including unmodeled dynamics like aerodynamic friction. The NN virtual control input scheme allows all six degrees of freedom of the UAVs to be controlled using only four control inputs while the dynamic control input generates the actual control signals for the UAVs in order to fly in formation. Additionally, the interconnection dynamical effects between the leader and its followers are explicitly considered and compensated, and the stability of the entire formation is demonstrated using Lyapunov theory. Numerical results are presented to verify the theoretical conjectures.*

Keywords: Formation Control, Leader-Follower, Quadrotor UAV, Neural Networks, Lyapunov Stability.

[1] Research Supported in part by GAANN Program and Intelligent Systems Center. This paper was not presented at any IFAC meeting. Contact author Email: tad5x4@mst.edu.

NOMENCLATURE

$(\bullet)_i$ Denotes a term for the leader UAV

$(\bullet)_j$ Denotes a term for the follower UAV

E^a Inertial coordinate frame

$E^b_{(\bullet)}$ UAV body fixed coordinate frame

$x_{(\bullet)}$ UAV x- coordinate in E^a

$y_{(\bullet)}$ UAV y- coordinate in E^a

$z_{(\bullet)}$ UAV z- coordinate in E^a

$\rho_{(\bullet)}$ UAV position vector in E^a

$\phi_{(\bullet)}$ UAV roll angle in E^a

$\theta_{(\bullet)}$ UAV pitch angle in E^a

$\psi_{(\bullet)}$ UAV yaw angle in E^a

$\Theta_{(\bullet)}$ UAV orientation vector in E^a

$E^a_{(\bullet)}$ Inertial coordinate frame rotated about $\psi_{(\bullet)}$

$R_{(\bullet)}$ Translational rotation matrix

$T_{(\bullet)}$ Rotational transformation matrix

$v_{(\bullet)}$ UAV translational velocity vector in $E^b_{(\bullet)}$

$\omega_{(\bullet)}$ UAV angular velocity vector in $E^b_{(\bullet)}$

$M_{(\bullet)}$ UAV mass and inertia matrix

$m_{(\bullet)}$ UAV total mass

$J_{(\bullet)}$ UAV moment of inertia matrix

$S(\bullet)$ General form of the skew symmetric matrix

$N_{(\bullet)k}$ UAV nonlinear aerodynamics effects, *k=1,2*

$\tau_{(\bullet)dk}$ External bounded disturbance, *k=1,2*

$u_{(\bullet)1}$ UAV thrust control input

$u_{(\bullet)2}$ UAV rotational torque vector control input

s_{jid} Desired separation between the follower and leader

α_{jid} Desired angle of incidence between the follower and the leader

β_{jid} Desired bearing angle between the follower and the leader

s_{ji} Measured separation between the follower and leader

α_{ji} Measured angle of incidence between the follower and the leader

β_{ji} Measured bearing angle between the follower and the leader

$x_{(\bullet)d}$ Desired UAV *x*- coordinate in E^a

$y_{(\bullet)d}$ Desired UAV *y*- coordinate in E^a

$z_{(\bullet)d}$ Desired UAV *z*- coordinate in E^a

$\rho_{(\bullet)d}$ Desired UAV position vector in E^a

$\phi_{(\bullet)d}$ Desired UAV roll angle in E^a

$\theta_{(\bullet)d}$ Desired UAV pitch angle in E^a

$\psi_{(\bullet)d}$ Desired UAV yaw angle in E^a

$\Theta_{(\bullet)d}$ Desired UAV orientation vector in E^a

$v_{(\bullet)d}$ Desired UAV translational velocity vector in $E^b_{(\bullet)}$

$\omega_{(\bullet)d}$ Desired UAV angular velocity vector in $E^b_{(\bullet)}$

R_{aj} Auxiliary transformation matrix (function of $\psi_{(\bullet)}$)

R_{ajd} Desired auxiliary transformation matrix (function of $\psi_{(\bullet)d}$)

Ξ_{ji} Separation transformation matrix (function of α_{ji} and β_{ji})

Ξ_{jid} Desired separation transformation matrix (function of α_{jid} and β_{jid})

$e_{(\bullet)\rho}$ UAV position tracking error vector

$e_{(\bullet)\Theta}$ UAV orientation tracking error vector

$e_{(\bullet)v}$ UAV translational velocity tracking error vector

$e_{(\bullet)\omega}$ UAV angular velocity tracking error vector

I. INTRODUCTION

In recent years, quadrotor helicopters have become a popular unmanned aerial vehicle (UAV) platform. The dynamics of the quadrotor UAV are not only nonlinear, but also coupled with each other and underactuated; characteristics which can make the platform difficult to control. In other words, the UAV has six degrees of freedom (DOF) with only four control inputs consisting of thrust and the three rotational torque inputs to control the six DOF. Recently, the control of single quadrotor UAVs has been undertaken by many researchers (Timothy, Burg, Xian & Dawson, 2007; Nicol, Macnab &

Ramirez-Serrano, 2008; Das, Lewis, & Subbarao, 2008; and Dierks & Jagannathan, 2008). However, a team of UAVs working together is often more effective than a single UAV in scenarios like surveillance, search and rescue, and perimeter security. Therefore, the formation control of UAVs has been proposed in the literature.

The work by Saffarian and Fahimi (2008) presents a modified leader-follower framework and proposes a model predictive nonlinear control algorithm to achieve the formation. Although the approach is verified via numerical simulations, proof of convergence and stability is not provided. Van der Walle, Fidan, Sutton, Yu and Anderson (2008) present a kinematic-based formation control law by assuming each UAV travels at a constant velocity while ignoring the UAV and the formation dynamics. Additionally, mathematical proof of stability is not provided. The work of Kingston, Beard and Holt (2008) offers a stable algorithm for perimeter security although the UAVs are restricted to travel at constant velocities ignoring UAV and formation dynamics.

By contrast in the work of Fierro, Belta, Desai and Kumar (2001), cylindrical coordinates and contributions from wheeled mobile robot leader follower formation control (Desai, Ostrowski & Kumar, 1998) are extended for aircrafts by assuming the dynamics are known. The work of Gu, Seanor, Campa, Napolitano, Rowe, Gururajan and Wan (2006) proposes a solution to the leader-follower formation control problem involving a linear inner and nonlinear outer-loop control structure, and experimental results are provided. However, an accurate dynamic model is needed and the measured position and velocity of the leader has to be communicated to its followers. In Xie, Zhang, Fierro and Motter (2005), the UAVs are assumed to be flying at a constant altitude, and the authors present two nonlinear robust formation controllers for UAVs.

The first approach assumes that the velocities and accelerations of the leader UAV are known while the second approach relaxes this assumption using robust control methodologies. In both the designs, the dynamics of the UAVs are assumed to be available. On the other hand, in the work of Galzi and Shtessel (2006), a robust formation controller is proposed based on higher order sliding mode controllers in the presence of bounded disturbances.

To overcome the assumption of known dynamics which are difficult to calculate, neural networks (NNs) have been considered in several works to control single quadrotor UAVs (Nicol, Macnab & Ramirez-Serrano, 2008; Das, Lewis, & Subbarao, 2008; Dierks & Jagannathan, 2008; Voos, 2007; Dunsfied, Tarbouchi & Labonte, 2004; and Puttige & Anavatti, 2007). On the other hand, in (Voos, 2007; and Dunsfied, Tarbouchi & Labonte, 2004), NN-based control laws are presented where the NN's are trained offline using experimentally collected data. A study performed by Puttige and Anavatti (2007) verified several well-known properties of online learning versus offline training and concluded that NN's which are properly trained offline are often robust to small variations in the system but fail to adapt to larger changes in the system. In contrast, NN models which learn online quickly adapt to variations in the nonlinear behavior of the system in real time with no prior knowledge are introduced in (Nicol, Macnab & Ramirez-Serrano, 2008; Das, Lewis, & Subbarao, 2008; and Dierks & Jagannathan, 2008).

On the other hand, linear models obtained from nonlinear systems are generally valid near a specific operating point (Lewis, Jagannathan & Yesilderek, 1999) and for the UAV, the operating point is generally chosen near the hovering configuration (Suh, 2003)

which may not be acceptable for dynamical outdoor setting with changing wind conditions. Under these outdoor conditions, more modes of the UAV dynamics will be excited more of the time (e.g. drag, etc.). As a consequence, an offline trained NN may not render a satisfactory performance since it is not practical or always possible to collect training data to account for every scenario the UAV may encounter. Similarly, linear controllers may not render satisfactory performance.

Therefore, in this work, a new leader-follower formation control framework is proposed for UAVs based on spherical coordinates where the desired trajectory of a follower UAV is generated online using a desired- separation, angle of incidence, and bearing , s_d, α_d, β_d, respectively, relative to its leader. Then, a new control law for leader-follower formation control is derived using NNs to learn the dynamics of the UAV online, including unmodeled dynamics like aerodynamic friction in the presence of bounded disturbances. Although a quadrotor UAV is underactuated, a novel NN virtual control input scheme for leader follower formation control is proposed which allows all six degrees of freedom of the UAV to be controlled using only four control inputs. The NN utilized in the follower control law compensates not only its own dynamics but also the formation dynamics. Thus, the framework of this paper effectively converts the leader-follower formation control for UAVs into a tracking control problem.

Therefore, the contribution of the proposed formation controller include: 1) a novel nonlinear NN-based controller is developed for follower UAVs and its leader where the objective of the formation is to achieve hovering or tracking time varying trajectories that are not near the hovering operating point; 2) explicit knowledge of the nonlinear dynamics of individual UAV and formation is not required while the linear in

the unknown parameters (LIP) assumption is not required; 3) the kinematic control law that translates the desired separation, angle of incidence and bearing into a trajectory online for successful formation control.

This paper is organized as follows. First, in Section II, the leader-follower formation control problem for UAVs is introduced, and required background information is presented. Then, the NN control law is developed for the follower UAVs as well as the formation leader, and the stability of the overall formation is presented in Section III. Section IV presents numerical simulations, and Section V provides some concluding remarks.

II. BACKGROUND

A. Quadrotor UAV Dynamics

Consider the quadrotor UAV shown in Fig 1. with six DOF defined in the inertial coordinate frame , E^a, as $[x, y, z, \phi, \theta, \psi]^T \in E^a$ where $\rho = [x, y, z]^T \in E^a$ are the position coordinates of the UAV and $\Theta = [\phi, \theta, \psi]^T \in E^a$ describe its orientation referred to as roll, pitch, and yaw, respectively. The kinematics of the UAV can be written as

$$\begin{aligned} \dot{\rho} &= Rv \\ \dot{\Theta} &= T\omega \end{aligned} \tag{1}$$

where $R(\Theta) \in \Re^{3x3}$ is the translational rotation matrix which is used to relate the translational velocity vector in the body fixed frame to derivative of the position vector in the inertial coordinate frame defined as (Dierks and Jagannathan, 2008)

$$R(\Theta) = R = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{2}$$

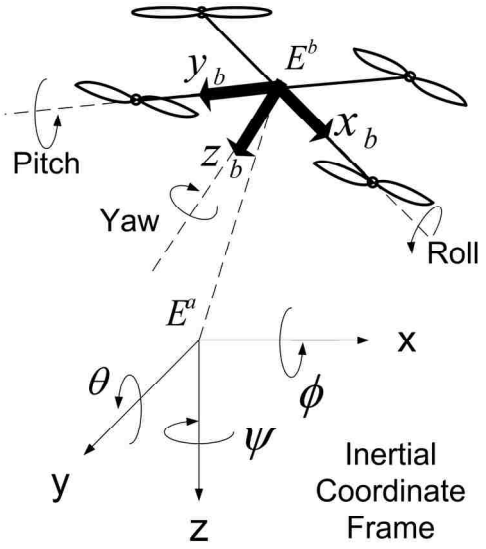


Fig. 1. Quadrotor UAV.

where the abbreviations $s_{(\bullet)}$ and $c_{(\bullet)}$ have been used for $\sin(\bullet)$ and $\cos(\bullet)$, respectively. It is useful to note that $\|R\|_F = R_{\max}$ for a known constant $R_{\max}$, $R^{-1} = R^T$, $\dot{R} = RS(\omega)$, $\dot{R}^T = -S(\omega)R^T$, and $S(\bullet) \in \Re^{3x3}$ is the general form of a skew symmetric matrix defined as

$$S(\gamma) = \begin{bmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{bmatrix}, \gamma = [\gamma_1, \gamma_2, \gamma_3]^T. \quad (3)$$

It is important to note that (3) satisfies the *skew symmetric property* (Lewis, Jagannathan, Yesilderek, 1999), $w^T S(\gamma) w = 0$, for any vector $w \in \Re^3$. The rotational transformation matrix from the fixed body to the inertial coordinate frame is defined as (Dierks and Jagannathan, 2008)

$$T(\Theta) = T = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}, \quad T^{-1} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \tag{4}$$

where the abbreviation $t_{(\bullet)}$ has been used for $\tan(\bullet)$. The transformation matrix T is bounded according to $\|T\|_F < T_{\max}$ for a known constant $T_{\max}$ as long as $-(\pi/2) < \theta < (\pi/2)$ (Neff, DongBin, Chitrakaran, Dawson & Burg, 2007). This region along with the regions $-(\pi/2) < \phi < (\pi/2)$ and $-\pi \le \psi \le \pi$ will be referred to as the *stable operating regions* of the UAV.

The translational and angular velocities are expressed in the body fixed frame attached to the center of mass of the UAV, E^b, and the dynamics of the UAV in the body fixed frame can be written as (Timothy, Burg, Xian & Dawson, 2007)

$$M \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \bar{S}(\omega) \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} N_1(v) \\ N_2(\omega) \end{bmatrix} + \begin{bmatrix} G(R) \\ 0_{3x1} \end{bmatrix} + U + \tau_d, \tag{5}$$

where $U = [0 \ 0 \ u_1 \ u_2^T]^T \in \Re^6$,

$$M = \begin{bmatrix} mI_3 & 0_{3x3} \\ 0_{3x3} & J \end{bmatrix} \in \Re^{6x6}, \quad \bar{S}(\omega) = \begin{bmatrix} -mS(\omega) & 0_{3x3} \\ 0_{3x3} & S(J\omega) \end{bmatrix} \in \Re^{6x6}$$

where m is a positive scalar that represents the total mass of the UAV and $J \in \Re^{3x3}$ represents the positive definite inertia matrix. The vector $v(t) = [v_{xb}, \ v_{yb}, \ v_{zb}]^T \in \Re^3$ represents the translational velocity, $\omega(t) = [\omega_{xb}, \omega_{yb}, \omega_{zb}]^T \in \Re^3$ represents the angular velocity, $N_k(\bullet) \in \Re^{3x1}, k = 1, 2$, are the nonlinear aerodynamic effects, $u_1 \in \Re$ provides the thrust along the z-direction, $u_2 = [u_{21} \ \ u_{22} \ \ u_{23}]^T \in \Re^3$ provides the three rotational torques to control the angular

velocities, $\tau_d = [\tau_{d1}^T, \tau_{d2}^T]^T \in \Re^6$ for $\tau_{dk} \in \Re^3, k = 1,2$ represents unknown, but bounded disturbances such that $\|\tau_d\| < \tau_M$ for all time t, with τ_M being an unknown positive constant. Additionally, $I_{nxn} \in \Re^{nxn}$ is an *nxn* identity matrix, and $0_{mxl} \in \Re^{mxl}$ represents an *mxl* matrix of all zeros. Furthermore, $G(R) \in \Re^3$ represents the gravity vector defined as $G(R) = mgR^T(\Theta)E_z$ where $E_z = [0,0,1]^T$ is a unit vector in the inertial coordinate frame, and $g = 9.81 m/s^2$.

B. Neural Networks

In this work, two-layer feedforward NNs are considered consisting of one layer of randomly assigned constant weights $V_N \in \Re^{axL}$ in the first layer and one layer of tunable weights $W_N \in \Re^{Lxb}$ in the second with a inputs, b outputs, and L hidden neurons. A compromise is made here between tuning the number of layered weights with computational complexity. The *universal approximation property* for NNs (Lewis, Jagannathan & Yesilderek, 1999) states that for any smooth function $f_N(x_N)$, there exists a NN such that $f_N(x_N) = W_N^T \sigma(V_N^T x_N) + \varepsilon_N$ where $\sigma(\cdot) : \Re^a \to \Re^L$ is the activation function in the hidden layers and ε_N is the bounded NN functional approximation error satisfying $\|\varepsilon_N\| < \varepsilon_M$ for a known constant ε_M. It has been shown that by randomly selecting the input layer weights V_N, the activation function vector $\sigma(\bar{x}_N) = \sigma(V_N^T x_N)$ forms a stochastic basis, and thus the approximation property holds for all inputs, $x_N \in \Re^a$, in the compact set S (Lewis, Jagannathan & Yesilderek, 1999). The sigmoid activation function is considered here. Furthermore, on any compact subset of $\Re^n$, the target NN weights are bounded by a known positive value, W_M, such

that $\|W_N\|_F \le W_M$. For complete details of the NN and its properties, see (Lewis, Jagannathan & Yesilderek, 1999). In this effort, $\|\cdot\|$ and $\|\cdot\|_F$ will be used as the vector and Frobenius norms (Lewis, Jagannathan & Yesilderek, 1999).

Next the definition of the semi-global uniformly ultimately boundedness is introduced.

Definition 1: The equilibrium point x_e is said to be semi-global uniformly ultimately bounded (SGUUB) if there exists a ball centered around the origin with an arbitrary radius *r,* $S(0,r) = S_r \subset \Re^n$ so that for all $x_0 \in S_r$ there exists a bound $B > 0$ and a time $T(B, x_0)$ such that $\|x(t) - x_e\| \le B$ for all $t \ge t_0 + T$ (Sastry, 1999).

C. A Novel Three Dimensional Leader-Follower UAV Formation Control Framework

Throughout this leader-follower development, the follower UAVs will be denoted with a subscript *'j'* while the formation leader will be denoted by the subscript *'i'*. To begin the development of this novel framework, an alternate reference frame denoted by E_j^a is introduced by rotating the inertial coordinate frame E^a about the z-axis by the yaw angle of follower *j,* ψ_j. In order to relate a vector in E^a to E_j^a, the transformation matrix is given by

$$R_{aj} = \begin{bmatrix} \cos(\psi_j) & \sin(\psi_j) & 0 \\ -\sin(\psi_j) & \cos(\psi_j) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $R_{aj}^T = R_{aj}^{-1}$.

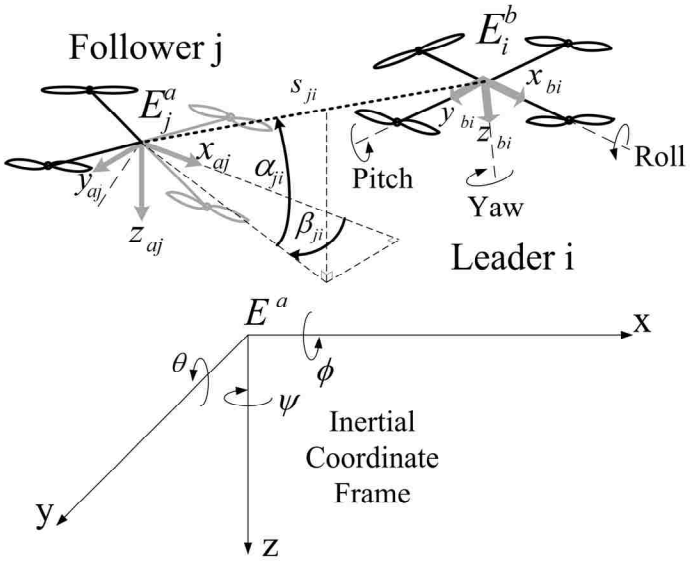


Fig. 2. UAV leader-follower formation control.

The objective of the proposed leader-follower formation control approach is for the follower UAV to maintain a desired separation, $s_{jid} \in \Re$, at a desired angle of incidence, $\alpha_{jid} \in E_j^a$, and bearing, $\beta_{jid} \in E_j^a$, with respect to its leader. The incidence angle is measured from the $x_{aj} - y_{aj}$ plane of follower *j* while the bearing angle is measured from the positive x_{aj}-axis as shown in Fig. 2. It is important to observe that each quantity is defined relative to the follower *j* instead of the leader *i* (Fierro, Belta, Desai & Kumar, 2001; and Desai, Ostrowski and Kumar, 1998). Additionally, in order to specify a unique configuration of follower *j* with respect to its leader, the desired yaw of follower *j* is selected to be the yaw angle of leader *i*, $\psi_i \in E^a$ as in (Saffarian & Fahimi, 2008). Using this approach, the relative distance between follower *j* and leader *i* is written as

$$\rho_i - \rho_j = R_{aj}^T s_{ji} \Xi_{ji} \tag{6}$$

where

$$\Xi_{ji} = [\cos(\alpha_{ji})\cos(\beta_{ji}) \quad \cos(\alpha_{ji})\sin(\beta_{ji}) \quad \sin(\alpha_{ji})]^T \tag{7}$$

Thus, to solve the leader-follower formation control problem in the proposed framework, a control velocity for the follower UAV must be derived to ensure

$$\left.\begin{array}{l} \lim_{t\to\infty}(s_{jid} - s_{ji}) = 0, \ \lim_{t\to\infty}(\beta_{jid} - \beta_{ji}) = 0, \\ \lim_{t\to\infty}(\alpha_{jid} - \alpha_{ji}) = 0, \ \lim_{t\to\infty}(\psi_{jd} - \psi_j) = 0 \end{array}\right\}. \tag{8}$$

Throughout the development, the desired separation, angle of incidence and bearing s_{jid}, α_{jid} and β_{jid}, respectively, will be taken as constants, while it is assumed that each UAV has knowledge of its own constant total mass, $m_{(\bullet)}$, where $(\bullet)$ is i for the leader and j for the follower. Additionally, it will be assumed that leader communicates its measured orientation and angular rate vectors, Θ_i and ω_i, respectively, and its *desired* states, $\psi_{id}, \dot{\psi}_{id}, \ddot{\psi}_{id}$, $v_{id}, \dot{v}_{id}$ reliably to its followers. This assumption will be relaxed in the future.

Further, the benefit of considering the desired instead of the measured states of the leader to its followers (Gu, Seanor, Campa, Napolitano, Rowe, Gururajan & Wan, 2006) in the design of the follower UAVs' control laws is significant when compensating for the formation dynamics which become incorporated in the follower UAVs dynamic controller design. Finally, communicating the desired states in fact reduces the reliance on noisy sensor measurements and thus reduces errors due to the noise from propagating

throughout the formation. Next, contributions from single UAV control will be considered and extended to the leader-follower formation control of UAVs.

III. Leader-Follower Formation Tracking Control

In this work, the formation leader control law is drawn from our previous work in single UAV control (Dierks & Jagannathan, 2008). The control objective of leader UAV i is to track a prescribed desired trajectory, and a desired yaw angle while maintaining a stable flight configuration. The z- component of the translation velocity vector is directly controllable with the thrust input. However, in order to control the x- and y- components of translational velocities, the pitch and roll must be controlled, respectively, thus redirecting the thrust. Complete consideration of the leader's controller design will be addressed in *Section III-B*.

To design the follower UAVs' control laws, frameworks for single UAV control (Dierks & Jagannathan, 2008) are extended to UAV formation to convert the formation control objective (8) into a tracking control problem as follows. In the proposed formation control formulation, the desired separation, angle of incidence and bearing angle will be utilized to define a desired trajectory of a follower UAV relative to its leader online while solving the formation control problem (8). Thus, by tracking the prescribed trajectory, the formation control problem (8) is converted into a tracking control problem.

Remark 1: The trajectory generated by the follower UAV relative to its leader should not be confused with the desired trajectory used to control the single UAVs without formation. For the case of single UAV control, the desired trajectory for the UAV is typically prescribed offline and can be tracked using only local information. In

contrast, the desired trajectories for the follower UAVs used in leader-follower formation control in this work are generated online, changes as a result of leader's maneuvers in real-time, in order to maintain the desired separation, incidence and bearing.

Moving on, once the desired trajectory has been specified online for the follower with respect its leader, a translational control velocity is calculated to ensure that the current position of the follower converges to its desired position. Then, the desired pitch, roll, and control thrust for the follower are designed such that the translational velocity of the UAV approaches the target translational control velocity. Next, given the designed desired attitude, the desired angular velocity is calculated along with the rotational torque vector which ensures the orientation and angular velocity of the follower UAV approaches their designed target values. The follower UAV controller design is considered next.

A. Follower UAV Control Law

Without loss of generality, it will be assumed throughout the development that follower *j* is following its formation leader *i.* However, in a formation where each UAV follows the UAV directly in front of it, this need not be the case. To begin the development of the follower control law, the desired position of the follower UAV is first defined relative to its leader. Then, the position error dynamics are derived and the translational control velocity for stabilization is designed.

Given a leader *i* subject to the kinematics and dynamics (1), and (5), respectively, define a reference trajectory for follower *j* to track at a desired separation s_{jid}, a desired angle of incidence, α_{jid}, and bearing, β_{jid} relative to the leader by

$$\rho_{jd} = \rho_i - s_{jid} R_{ajd}^T \Xi_{jid} \tag{9}$$

where

$$R_{ajd}^T \Xi_{jid} = \begin{bmatrix} \cos(\psi_{jd}) & \sin(\psi_{jd}) & 0 \\ -\sin(\psi_{jd}) & \cos(\psi_{jd}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha_{jid})\cos(\beta_{jid}) \\ \cos(\alpha_{jid})\sin(\beta_{jid}) \\ \sin(\alpha_{jid}) \end{bmatrix}.$$

Next, using (6) and (9), define the position tracking error as

$$e_{j\rho} = \rho_{jd} - \rho_j = s_{ji} R_{aj}^T \Xi_{ji} - s_{jid} R_{ajd}^T \Xi_{jid} \in E^a \tag{10}$$

which can be measured using local sensor information. To form the position tracking error dynamics, it is convenient to rewrite (10) as $e_{j\rho} = \rho_i - \rho_j - R_{ajd}^T s_{jid} \Xi_{jid}$ and use (1) to give

$$\dot{e}_{j\rho} = R_i v_i - R_j v_j - s_{jid} \dot{R}_{ajd}^T \Xi_{jid} . \tag{11}$$

Next, the desired translational velocity of follower j $v_{jd} = [v_{jdx} \; v_{jdy} \; v_{jdz}]^T \in E^b$, selected to stabilize (11) is written as

$$v_{jd} = R_j^T (R_i v_{id} - s_{jid} \dot{R}_{ajd}^T \Xi_{jid} + K_{j\rho} e_{j\rho}) \tag{12}$$

where $K_{j\rho} = diag\{k_{j\rho x}, k_{j\rho y}, k_{j\rho z}\} \in \Re^{3x3}$ is a diagonal positive definite design matrix all with positive design constants and v_{id} is the desired translation velocity of leader *i*. Next, the translational velocity tracking error for follower *j* and leader *i* is defined as

$$e_{jv} = [e_{jvx} \; e_{jvy} \; e_{jvz}]^T = v_{jd} - v_j \tag{13}$$

and

$$e_{iv} = [e_{ivx} \; e_{ivy} \; e_{ivz}]^T = v_{id} - v_i , \tag{14}$$

respectively. Applying (12) to (11) while observing $v_j = v_{jd} - e_{jv}$ and $v_i = v_{id} - e_{iv}$, reveals the closed loop position error dynamics to be rewritten as

$$\dot{e}_{j\rho} = -K_{j\rho} e_{j\rho} + R_j e_{jv} - R_i e_{iv} . \tag{15}$$

Next, the translational velocity tracking error dynamics for follower *j* are developed so that the desired pitch, roll, and control thrust can be found. Differentiating (13), observing

$$\dot{v}_{jd} = -S(\omega_j) v_{jd} + R_j^T \left(R_i S(\omega_i) v_{id} + R_i \dot{v}_{id} - \ddot{R}_{ajd}^T s_{jid} \Xi_{jid} \right) + R_j^T K_{j\rho} \left(R_i v_i - R_j v_j - \dot{R}_{ajd}^T s_{jid} \Xi_{jid} \right),$$

and substituting the translational velocity dynamics in (5) allows the velocity tracking error dynamics to be written as

$$\begin{aligned} \dot{e}_{jv} = \dot{v}_{jd} - \dot{v}_j = & -S(\omega_j) e_{jv} - N_{j1}(v_j)/m_j - G(R_j)/m_j \\ & - u_{j1} E_{jz}/m_j - \bar{\tau}_{jd1} + R_j^T (R_i S(\omega_i) v_{id} + R_i \dot{v}_{id} - \ddot{R}_{ajd}^T s_{jid} \Xi_{jid}) \\ & - R_j^T K_{j\rho} (R_j v_j + \dot{R}_{ajd}^T s_{jid} \Xi_{jid}) + R_j^T K_{j\rho} R_i v_i \end{aligned} \tag{16}$$

where $\bar{\tau}_{jd1} = \tau_{jd1} / m_j$. Next, adding and subtracting $R_j^T K_{j\rho} R_i v_{id}$ to (16) reveals

$$\begin{aligned} \dot{e}_{jv} = & -S(\omega_j) e_{jv} - N_{j1}(v_j)/m_j - G(R_j)/m_j \\ & - u_{j1} E_{jz}/m_j - \bar{\tau}_{jd1} + R_j^T (R_i S(\omega_i) v_{id} + R_i \dot{v}_{id} - \ddot{R}_{ajd}^T s_{jid} \Xi_{jid}) \\ & - R_j^T K_{j\rho} (R_j v_j + \dot{R}_{ajd}^T s_{jid} \Xi_{jid}) + R_j^T K_{j\rho} R_i v_{id} - R_j^T K_{j\rho} R_i e_{vi} \end{aligned} \tag{17}$$

which can be rewritten as

$$\begin{aligned} \dot{e}_{jv} = & -S(\omega_j) e_{jv} - N_{j1}(v_j)/m_j - G(R_j)/m_j - u_{j1} E_{jz}/m_j \\ & - \bar{\tau}_{jd1} - R_j^T K_{j\rho} R_i e_{vi} + R_j^T \Lambda_j \end{aligned} \tag{18}$$

with

$$\Lambda_j = R_i \dot{v}_{id} + R_i S(\omega_i) v_{id} - (\ddot{R}_{ajd}^T + K_{j\rho} \dot{R}_{ajd}^T) s_{jid} \Xi_{jid} + K_{j\rho} (R_i v_{id} - R_j v_j) . \tag{19}$$

Remark 2: Examining the velocity tracking error dynamics (18) of the follower, it is observed that the derivative of the control velocity, $\dot{v}_{id}$, of the leader is required as a result of using v_{id} in (12). If the measured velocity of the leader, v_i, had been used instead of v_{id} in (12), the tracking error dynamics (17) would be dependent on $\dot{v}_i$ which are considered to be unknown by the follower j in this work. In the following development, a NN is introduced to learn the unknown quantities of (17); however, to effectively approximate the leader's dynamics, $\dot{v}_i$, terms like the leader's control thrust and rotational torques would be required to be communicated to each follower in addition to the leader's measured linear and angular velocities so that the terms could be included in the NN input of the follower.

Moving on, we now seek to find expressions for the desired pitch, θ_{jd}, and roll, ϕ_{jd}, required to control the translational velocity components v_{jxb} and v_{jyb}, respectively. Moreover, it is desirable to specify the maximum desired pitch and roll angles, respectively, to be tracked by the follower UAV.

To accomplish these design objectives, we first define the *scaled* desired orientation vector, $\overline{\Theta}_{jd} = [\overline{\theta}_{jd} \ \overline{\phi}_{jd} \ \psi_{jd}]^T$ where $\overline{\theta}_{jd} = \pi\theta_{jd} \big/ (2\theta_{d\max})$, $\overline{\phi}_{jd} = \pi\phi_{jd} \big/ (2\phi_{d\max})$, where $\theta_{d\max} \in (0, \pi/2)$ and $\phi_{d\max} \in (0, \pi/2)$ are design constants used to specify the maximum desired roll and pitch, respectively. Next, we rewrite the translational rotation matrix (2) in terms of $\overline{\Theta}_{jd}$, and define $R_{jd} = R_j(\overline{\Theta}_{jd})$. Then, add and subtract $G(R_{jd})/m_j$ and $R_{jd}^T \Lambda_j$ to (18) to yield

$$\dot{e}_{jv} = -S(\omega_j)e_{jv} - G(R_{jd})\big/m_j + R_{jd}^T(\Lambda_j + f_{jc1}(x_{cj1})) - u_{j1}E_{jz}\big/m_j - R_j^T K_{j\rho} R_i e_{iv} - \overline{\tau}_{jd1} \qquad (20)$$

where

$$f_{jc1}(x_{cj1}) = R_{jd}(G(R_{jd})/m_j - G(R_j)/m_j + (R_j^T - R_{jd}^T)\Lambda_j) + R_{jd}(R_j^T R_i S(\omega_i) v_{id} - N_{j1}(v_j)/m_j) \quad (21)$$

is an unknown function which can be rewritten as $f_{jc1}(x_{jc1}) = \left[f_{jc11} \;\; f_{jc12} \;\; f_{jc13}\right]^T \in \Re^3$. In the forthcoming development, the approximation properties of NN will be utilized to estimate the unknown function $f_{jc1}(x_{jc1})$ by bounded ideal weights W_{jc1}^T, V_{jc1}^T such that $\left\|W_{jc1}\right\|_F \le W_{jMc1}$ for a known constant W_{jMc1}, and written as $f_{jc1}(x_{jc1}) = W_{jc1}^T \sigma(V_{jc1}^T x_{jc1}) + \varepsilon_{jc1}$ where $\varepsilon_{jc1} \le \varepsilon_{Mc1}$ is the bounded NN approximation error where ε_{Mc1} is a known constant. The NN estimate of f_{jc1} is written as $\hat{f}_{jc1} = \hat{W}_{jc1}^T \sigma(V_{jc1}^T x_{jc1}) = \hat{W}_{jc1}^T \sigma_{jc1}$ $= [\hat{W}_{jc11}^T \sigma_{jc1} \;\; \hat{W}_{jc12}^T \sigma_{jc1} \;\; \hat{W}_{jc13}^T \sigma_{jc1}]^T$ where $\hat{W}_{jc1}^T$ is the NN estimate of W_{jc1}^T, $\hat{W}_{jc1k}^T, k = 1,2,3$ is the i^{th} row of $\hat{W}_{jc1}^T$, and x_{jc1} is the NN input $x_{jc1} = [1 \; \Theta_j^T \;\; \Theta_i^T \;\; \omega_i^T \;\; \Lambda_j^T \;\; v_{jd}^T \;\; v_{id}^T \;\; \dot{v}_{id}^T \;\; \psi_{jd} \;\; \dot{\psi}_{jd} \;\; \ddot{\psi}_{jd} \;\; \omega_j^T \;\; v_j^T \;\; e_{jv}^T \;\; e_{j\rho}^T]^T$.

The key step in designing the desired pitch and roll is identifying the *desired* closed loop velocity tracking error dynamics. For convenience, the *desired* translational velocity closed loop system is selected as

$$\dot{e}_{jv} = -S(\omega_j) e_{jv} - K_{jv} e_{jv} - \bar{\tau}_{jd1} - R_j^T K_{j\rho} R_i e_{iv} \quad (22)$$

where $K_{jv} = diag\{k_{jv1} \cos(\bar{\theta}_{jd}), k_{jv2} \cos(\bar{\phi}_{jd}), k_{jv3}\}$ is a diagonal positive definite design matrix with each $k_{jvk} > 0$, $k = 1,2,3$. In the following development, it will be shown that $\bar{\theta}_{jd} \in (-\pi/2, \pi/2)$, $\bar{\phi}_{jd} \in (-\pi/2, \pi/2)$; therefore, it is clear that $K_{jv} > 0$. Then, equating (20) and (22) while considering only the first two velocity error states reveals

$$\begin{bmatrix}0\\0\end{bmatrix} = -g\begin{bmatrix}-s_{\bar{\theta}jd}\\ c_{\bar{\theta}jd}s_{\bar{\phi}jd}\end{bmatrix} + \begin{bmatrix}c_{\bar{\theta}jd}k_{jv1}e_{jvx}\\ c_{\bar{\phi}jd}k_{jv2}e_{jvy}\end{bmatrix} + \begin{bmatrix} c_{\bar{\theta}jd}c_{\psi jd} & c_{\bar{\theta}jd}s_{\psi jd} & -s_{\bar{\theta}jd}\\ s_{\bar{\phi}jd}s_{\bar{\theta}jd}c_{\psi jd} - c_{\bar{\phi}jd}s_{\psi jd} & s_{\bar{\phi}jd}s_{\bar{\theta}jd}s_{\psi jd} + c_{\bar{\phi}jd}c_{\psi jd} & s_{\bar{\phi}jd}c_{\bar{\theta}jd}\end{bmatrix}\begin{bmatrix}\Lambda_{j1} + f_{jc11}\\ \Lambda_{j2} + f_{jc12}\\ \Lambda_{j3} + f_{jc13}\end{bmatrix} \tag{23}$$

where $\Lambda_j = [\Lambda_{j1} \;\; \Lambda_{j2} \;\; \Lambda_{j3}]^T$ was utilized. Then, applying basic math operations, the first line of (23) can be rewritten as

$$c_{\bar{\theta}jd}(c_{\psi jd}(\Lambda_{j1} + f_{jc11}) + s_{\psi jd}(\Lambda_{j2} + f_{jc12}) + k_{jv1}e_{jvx}) = s_{\bar{\theta}jd}(\Lambda_{j3} + f_{jc13} - g). \tag{24}$$

Similarly, the second line of (23) can be rewritten as

$$\begin{aligned} &c_{\bar{\phi}jd}(c_{\psi jd}(\Lambda_{j2} + f_{jc12}) - s_{\psi jd}(\Lambda_{j1} + f_{jc11}) + k_{jv2}e_{jvy}) = \\ &\quad s_{\bar{\phi}jd}\left(gc_{\bar{\theta}jd} - s_{\bar{\theta}jd}c_{\psi jd}(\Lambda_{j1} + f_{jc11}) - s_{\bar{\theta}jd}s_{\psi jd}(\Lambda_{j2} + f_{jc12}) - c_{\bar{\theta}jd}(\Lambda_{j3} + f_{jc13})\right) \end{aligned} \tag{25}$$

Next, (24) is solved for the desired pitch θ_{jd} while (25) can be solved for the desired roll ϕ_{jd}. Using the NN estimates, $\hat{f}_{jc1}$, the desired pitch θ_{jd} can be written as

$$\theta_{jd} = \frac{2\theta_{\max}}{\pi} a\tan\left(\frac{N_{\theta jd}}{D_{\theta jd}}\right) \tag{26}$$

where $N_{\theta jd} = c_{\psi jd}(\Lambda_{j1} + \hat{f}_{jc11}) + s_{\psi jd}(\Lambda_{j2} + \hat{f}_{jc12}) + k_{jv1}e_{jvx}$ and $D_{\theta jd} = \Lambda_{j3} + \hat{f}_{jc13} - g$. Similarly, the desired roll angle, ϕ_{jd}, is found to be

$$\phi_{jd} = \frac{2\phi_{\max}}{\pi} a\tan\left(\frac{N_{\phi jd}}{D_{\phi jd}}\right) \tag{27}$$

where $N_{\phi jd} = c_{\psi jd}(\Lambda_{j2} + \hat{f}_{jc12}) - s_{\psi jd}(\Lambda_{j1} + \hat{f}_{jc11}) + k_{jv2}e_{jvy}$ and $D_{\phi jd} = gc_{\bar{\theta}jd} - s_{\bar{\theta}jd}c_{\psi jd}(\Lambda_{j1} + \hat{f}_{jc11})$ $- s_{\bar{\theta}jd}s_{\psi jd}(\Lambda_{j2} + \hat{f}_{jc12}) - c_{\bar{\theta}jd}(\Lambda_{j3} + \hat{f}_{jc13})$.

Remark 3: The expressions for the desired pitch and roll in (26) and (27) lend themselves very well to the formation control of quadrotor UAVs. The expressions will always produce desired values in the *stable operation regions* of the UAV since $a\tan(\bullet)$ approaches $\pm\pi/2$ as its argument increases. Thus, introducing the scaling factors in $\bar{\theta}_{jd}$ and $\bar{\phi}_{jd}$ results in $\theta_{jd} \in (-\theta_{\max}, \theta_{\max})$ and $\phi_{jd} \in (-\phi_{\max}, \phi_{\max})$, and the aggressiveness of the UAVs maneuvers can be managed. Further, if the un-scaled desired orientation vector were used in the development of (20), the maximum desired pitch and roll would still remain within the stable operating regions. It is observed that too conservative maximum values could lead to degraded tracking performance for very aggressive trajectories.

Now that the desired orientation of the UAV has been found, we now derive the orientation error dynamics and find the stabilizing angular velocity control. Next define the attitude tracking error as

$$e_{j\Theta} = \Theta_{jd} - \Theta_j \in E^a \tag{28}$$

where dynamics are found using (1) to be $\dot{e}_{j\Theta} = \dot{\Theta}_{jd} - T_j \omega_j$. In order to drive the orientation errors (28) to zero, the desired angular velocity, ω_{jd}, is selected as

$$\omega_{jd} = T_j^{-1}(\dot{\Theta}_{jd} + K_{j\Theta} e_{j\Theta}) \tag{29}$$

where $K_{j\Theta} = diag\{k_{j\Theta 1}, k_{j\Theta 2}, k_{j\Theta 3}\} \in \Re^{3x3}$ is a diagonal matrix of positive design constants. Define the angular velocity tracking error as

$$e_{j\omega} = \omega_{jd} - \omega_j, \tag{30}$$

and observing $\omega_j = \omega_{jd} - e_{j\omega}$, the closed loop orientation error system dynamics can be written as

$$\dot{e}_{j\Theta} = -K_{j\Theta} e_{j\Theta} + T_j e_{j\omega}. \tag{31}$$

Examining (29), calculation of the desired angular velocity requires knowledge of $\dot{\Theta}_{jd}$; however, $\dot{\Theta}_{jd}$ is not known in view of the fact $\dot{\Lambda}_j$ and $\dot{\hat{f}}_{jc1}$ are not available. Further, development of u_{j2} in the following section will reveal $\dot{\omega}_{jd}$ is required which in turn implies $\ddot{\Lambda}_j$ and $\ddot{\hat{f}}_{jc1}$ must be known. Since these requirements are not practical, the *universal approximation property* of NN is invoked to estimate ω_{jd} and $\dot{\omega}_{jd}$ by using a nonlinear virtual control structure.

To begin the NN virtual control development, we rearrange (29) to observe the dynamics of the ideal virtual controller to be

$$\begin{aligned} \dot{\Theta}_{jd} &= T_j(\omega_{jd} - T_j^{-1} K_{j\Theta} e_{j\Theta}) \\ \dot{\omega}_{jd} &= \dot{T}_j^{-1}(\dot{\Theta}_{jd} + K_{j\Theta} e_{j\Theta}) + T_j^{-1}(\ddot{\Theta}_{jd} + K_{j\Theta} \dot{e}_{j\Theta}) \end{aligned} \tag{32}$$

For convenience, we define a change of variable as $\Omega_{jd} = \omega_{jd} - T_j^{-1} K_{j\Theta} e_{j\Theta}$, and the dynamics (32) become

$$\begin{aligned} \dot{\Theta}_{jd} &= T_j \Omega_{jd} \\ \dot{\Omega}_{jd} &= \dot{T}_j^{-1} \dot{\Theta}_{jd} + T_j^{-1} \ddot{\Theta}_{jd} = f_{j\Omega}(x_{j\Omega}) = f_{j\Omega} \end{aligned} \tag{33}$$

Defining the estimates of Θ_{jd} and Ω_{jd} to be $\hat{\Theta}_{jd}$ and $\hat{\Omega}_{jd}$, respectively, and the estimation error $\tilde{\Theta}_{jd} = \Theta_{jd} - \hat{\Theta}_{jd}$, the dynamics of the proposed NN virtual controller become

$$\begin{aligned}\dot{\hat{\Theta}}_{jd} &= T_j \hat{\Omega}_{jd} + K_{j\Omega 1}\widetilde{\Theta}_{jd} \\ \dot{\hat{\Omega}}_{jd} &= \hat{f}_{j\Omega 1}(\hat{x}_{j\Omega}) + K_{j\Omega 2} T_j^{-1}\widetilde{\Theta}_{jd}\end{aligned} \tag{34}$$

where $K_{j\Omega 1}$ and $K_{j\Omega 2}$ are positive constants. The estimate $\hat{\omega}_{jd}$ is then written as

$$\hat{\omega}_{jd} = \hat{\Omega}_{jd} + T_j^{-1} K_{j\Theta} e_{j\Theta} + K_{j\Omega 3} T_j^{-1}\widetilde{\Theta}_{jd} \tag{35}$$

where $K_{j\Omega 3}$ is another positive constant. Observing

$$\widetilde{\omega}_{jd} = \omega_{jd} - \hat{\omega}_{jd} = \widetilde{\Omega}_{jd} - K_{j\Omega 3} T_j^{-1}\widetilde{\Theta}_{jd}, \tag{36}$$

with $\widetilde{\Omega}_{jd} = \Omega_{jd} - \hat{\Omega}_{jd}$, subtracting (34) from (33), as well as adding and subtracting $T_j^T \widetilde{\Theta}_{jd} + K_{j\Omega 3}\dot{T}_j^{-1}\widetilde{\Theta}_{jd}$, the virtual controller estimation error dynamics are found to be

$$\begin{aligned}\dot{\widetilde{\Theta}}_{jd} &= T_j \widetilde{\omega}_{jd} - (K_{j\Omega 1} - K_{j\Omega 3})\widetilde{\Theta}_{jd} \\ \dot{\widetilde{\Omega}}_{jd} &= f_{j\Omega 1}(x_{j\Omega}) - \hat{f}_{j\Omega 1}(\hat{x}_{j\Omega}) - K_{j\Omega 2} T_j^{-1}\widetilde{\Theta}_{jd} - T_j^T \widetilde{\Theta}_{jd} + K_{j\Omega 3}\dot{T}_j^{-1}\widetilde{\Theta}_{jd}\end{aligned} \tag{37}$$

where $f_{j\Omega 1}(x_{j\Omega}) = f_{j\Omega} + T_j^T \widetilde{\Theta}_{jd} - K_{j\Omega 3}\dot{T}_j^{-1}\widetilde{\Theta}_{jd}$ is an unknown function.

In (34), *universal approximation property* of NN has been utilized to estimate the unknown function $f_{j\Omega 1}(x_{j\Omega})$ by bounded ideal weights $W_{j\Omega}^T, V_{j\Omega}^T$ such that $\|W_{j\Omega}\|_F \le W_{jM\Omega}$ for a known constant $W_{jM\Omega}$, and written as $f_{1j\Omega}(x_{j\Omega}) = W_{j\Omega}^T \sigma(V_{j\Omega}^T x_{j\Omega}) + \varepsilon_{j\Omega}$ where $\varepsilon_{j\Omega}$ is the bounded NN approximation error such that $\|\varepsilon_{j\Omega}\| \le \varepsilon_{\Omega M}$ for a known constant $\varepsilon_{\Omega M}$. The NN estimate of $f_{j\Omega}$ is written as $\hat{f}_{j\Omega}(\hat{x}_{j\Omega}) = \hat{f}_{j\Omega} = \hat{W}_{j\Omega}^T \sigma(V_{j\Omega}^T \hat{x}_{j\Omega}) = \hat{W}_{j\Omega}^T \hat{\sigma}_{j\Omega}$ where $\hat{W}_{j\Omega}^T$ is the NN estimate of $W_{j\Omega}^T$ and $\hat{x}_{j\Omega}$ is

the NN input written in terms of the virtual control estimates, desired trajectory, and the UAV velocity. The NN input is defined as $\hat{x}_{j\Omega} = [1 \ \Lambda_j^T \ \ \Theta_{jd}^T \ \ \hat{\Omega}_{jd}^T \ \tilde{\Theta}_{jd}^T \ \ v_j^T \ \ \omega_j^T]^T$.

Next, differentiating (36), using (37) as well as adding and subtracting $W_{j\Omega}^T \hat{\sigma}_{j\Omega}$ reveals

$$\dot{\tilde{\omega}}_{jd} = -K_{j\Omega 3}\tilde{\omega}_{jd} + \tilde{f}_{j\Omega 1}(\hat{x}_{j\Omega}) - T_j^T \tilde{\Theta}_{jd} - T_j^{-1}(K_{j\Omega 2} - K_{j\Omega 3}(K_{j\Omega 1} - K_{j\Omega 3}))\tilde{\Theta}_{jd} + \xi_{j\Omega} \quad (38)$$

where $\tilde{f}_{j\Omega} = \tilde{W}_{j\Omega}^T \hat{\sigma}_{j\Omega}, \tilde{W}_{j\Omega}^T = W_{j\Omega}^T - \hat{W}_{j\Omega}^T, \quad \xi_{j\Omega} = \varepsilon_{j\Omega} + W_{j\Omega}^T \tilde{\sigma}_{j\Omega}$, and $\tilde{\sigma}_{j\Omega} = \sigma_{j\Omega} - \hat{\sigma}_{j\Omega}$. Furthermore, $\|\xi_{j\Omega}\| \le \xi_{j\Omega M}$ with $\xi_{j\Omega M} = \varepsilon_{\Omega M} + 2W_{jM\Omega}\sqrt{N_{j\Omega}}$ a computable constant with $N_{j\Omega}$ the constant number of hidden layer neurons in the virtual control NN and the fact $\|\sigma_{j\Omega}\| \le \sqrt{N_{j\Omega}}$ was used. Examination of (30) and (31) reveals that $\tilde{\Theta}_{jd} = 0, \tilde{\omega}_{jd} = 0$, and $\tilde{f}_{j\Omega} = 0$ to be equilibrium points of the estimation error dynamics when $\|\xi_{j\Omega}\| = 0$.

To this point, the desired translational velocity for follower *j* has been identified to ensure the leader-follower objective (8) is achieved. Then, the desired pitch and roll were derived to drive $v_{jxb} \to v_{jdx}$ and $v_{jyb} \to v_{jdy}$, respectively. Then, the desired angular velocity was found to ensure $\Theta_j \to \Theta_{jd}$. What remains is to identify the UAV thrust to guarantee $v_{jzb} \to v_{jdz}$ and rotational torque vector to ensure $\omega_j \to \hat{\omega}_{jd}$. First, the thrust is derived.

Consider again the translational velocity tracking error dynamics (20), as well as the *desired* velocity tracking error dynamics (22). Equating (20) and (22) and manipulating the third error state, the required thrust is found to be

$$\begin{aligned} u_{j1} = {} & m_j(c_{\bar{\phi}jd}s_{\bar{\theta}jd}c_{\psi jd} + s_{\bar{\phi}jd}s_{\psi jd})(\Lambda_{j1} + \hat{f}_{jc11}) + m_j k_{jvz} e_{vj3} + m_j c_{\bar{\theta}jd}c_{\bar{\phi}jd}(\Lambda_{j3} + \hat{f}_{jc13} - g) \\ & + m_j(c_{\bar{\phi}jd}s_{\bar{\theta}jd}s_{\psi jd} - s_{\bar{\phi}jd}c_{\psi jd})(\Lambda_{j2} + \hat{f}_{jc12}) \end{aligned} \quad (39)$$

where $\hat{f}_{jc1}$ is the NN estimate in (21) previously defined. Substituting the desired pitch (26), roll (27), and the thrust (39) into the translational velocity tracking error dynamics (20) yields

$$\dot{e}_{jv} = -K_{jv} e_{jv} + R_{jd}^T \tilde{W}_{jc1}^T \sigma_{jc1} - R_j^T K_{j\rho} R_i e_{iv} + \xi_{jc1}, \tag{40}$$

with $\xi_{jc1} = R_{jd}^T \varepsilon_{jc} - \bar{\tau}_{jd1}$, $\tilde{W}_{jc1} = W_{jc1} - \hat{W}_{jc1}$ and, $\|\xi_{jc1}\| \le \xi_{jMc1}$ for a computable constant $\xi_{jMc1} = R_{\max} \varepsilon_{Mc1} + \tau_M / m_j$. In the formulation of (40), the expressions for the desired pitch and roll (26) and (27), respectively, were first written in the form of (24) and (25), so that sine and cosine of the angles could be substituted as opposed to substituting the arctangent expressions directly into the sine or cosine function.

Next, the rotational torque vector, u_{j2}, will be addressed. First, multiply the angular velocity tracking error (30) by the constant inertia matrix J_j, take the first derivative with respect to time, substitute the UAV dynamics (5) and add and subtract $T_j^T e_{j\Theta}$ to reveal

$$J_j \dot{e}_{j\omega} = J_j \dot{\omega}_{jd} - J_j \dot{\omega}_j = f_{jc2}(x_{jc2}) - u_{j2} - T_j^T e_{j\Theta} - \tau_{jd2} \tag{41}$$

with $f_{jc2}(x_{jc2}) = J_j \dot{\omega}_{jd} - S(J_j \omega_j)\omega_j - N_{j2}(\omega_j) + T_j^T e_{j\Theta}$.

Examining $f_{jc2}(x_{jc2})$, it is clear that the function is nonlinear and contains unknown terms; therefore, the *universal approximation property* of NN is utilized to estimate the function $f_{jc2}(x_{jc2})$ by bounded ideal weights W_{jc2}^T, V_{jc2}^T such that $\|W_{jc2}\|_F \le W_{jMc2}$ for a known constant W_{jMc2}. The ideal NN representation is written as $f_{jc2}(x_{jc2}) = W_{jc2}^T \sigma(V_{jc2}^T x_{jc2}) + \varepsilon_{jc2}$ where ε_{jc2} is the bounded NN functional reconstruction

error such that $\|\varepsilon_{jc2}\| \le \varepsilon_{Mc2}$ for a known constant ε_{Mc2}. The NN estimate of f_{jc2} is given by $\hat{f}_{jc2}(\hat{x}_{jc2}) = \hat{W}_{jc2}^T \sigma(V_{jc2}^T \hat{x}_{jc2}) = \hat{W}_{jc2}^T \hat{\sigma}_{jc2}$ where $\hat{W}_{jc2}^T$ is the NN estimate of W_{jc2}^T and $\hat{x}_{jc2} = [1 \ \omega_j^T \ \dot{\hat{\Omega}}_{jd}^T \ \widetilde{\Theta}_{jd}{}^T \ e_{j\Theta}^T]^T$ is the input to the NN written in terms of the virtual controller estimates. By the construction of the virtual controller, $\dot{\hat{\omega}}_{jd}$ is not directly available; therefore, observing (35), the terms $\dot{\hat{\Omega}}_{jd}^T$, $\widetilde{\Theta}_{jd}{}^T$, and $e_{j\Theta}^T$ have been included instead.

Moving on, the angular velocity tracking error $e_{j\omega}$ cannot be calculated due to its dependence on the unknown vector ω_{jd}. Thus, using the desired angular velocity (35), we define the estimated angular velocity tracking error as $\hat{e}_{j\omega} = \hat{\omega}_{jd} - \omega_j$. Now, using the NN estimate $\hat{f}_{jc2}$ and $\hat{e}_{j\omega}$, the rotational torque control input is written as

$$u_{j2} = \hat{f}_{jc2} + K_{j\omega} \hat{e}_{j\omega}, \tag{42}$$

and substituting the control input (42) into the angular velocity dynamics (41) yields

$$J_j \dot{e}_{j\omega} = f_{jc2} - \hat{f}_{jc2} - K_{j\omega} \hat{e}_{j\omega} - T_j^T e_{j\Theta} - \tau_{jd2} \ . \tag{43}$$

Now, adding and subtracting $W_{jc2}^T \hat{\sigma}_{jc}$ and observing $\hat{e}_{j\omega} = e_{j\Omega} - \widetilde{\omega}_{jd}$, the closed loop dynamics (43) become

$$J_j \dot{e}_{j\omega} = -K_{j\omega} e_{j\omega} + \widetilde{W}_{jc2}^T \hat{\sigma}_{jc2} + K_{j\omega} \widetilde{\omega}_{jd} - T_j^T e_{j\Theta} + \xi_{jc2} \tag{44}$$

where $\tilde{W}_{jc2}^T = W_{jc2}^T - \hat{W}_{jc2}^T$, $\xi_{jc2} = \varepsilon_{jc2} + W_{jc2}^T \tilde{\sigma}_{jc} - \tau_{jd2}$, and $\tilde{\sigma}_{jc2} = \sigma_{jc2} - \hat{\sigma}_{jc2}$. Further, $\|\xi_{jc2}\| \le \xi_{jMc2}$ for a computable constant $\xi_{jMc2} = \varepsilon_{Mc2} + 2W_{jMc2}\sqrt{N_{jc2}} + \tau_{dM}$ where N_{jc2} is the number of hidden layer neurons.

As a final step, we define the augmented variables $e_{jD} = [e_{jv}^T \;\; e_{j\omega}^T]^T$, $\hat{e}_{jD} = [e_{jv}^T \;\; \hat{e}_{j\omega}^T]^T$, $\tilde{W}_{jc} = [\tilde{W}_{jc1}\; 0; 0\; \tilde{W}_{jc2}]$ and $\hat{\sigma}_{jc} = [\hat{\sigma}_{jc1}^T \;\; \hat{\sigma}_{jc2}^T]^T$. In the following theorem, the stability of the follower *j* is shown while considering $e_{iv} = 0$. In other words, the position, orientation, and velocity tracking errors are considered along with the estimation errors of the virtual controller and the NN weight estimation errors of each NN for follower *j* while ignoring the interconnection errors (e_{iv}) between the leader and its followers. This assumption will be relaxed in the following section.

Theorem 1: (Follower UAV System Stability) Given the dynamic nonlinear system of follower *j* in the form of (5), let the desired translational velocity, pitch and roll for follower *j* be defined by (12), (26) and (27), respectively. Let the NN virtual controller be defined by (34) and (35), respectively, with the NN update law given by

$$\dot{\hat{W}}_{j\Omega} = F_{j\Omega}\hat{\sigma}_{j\Omega}\tilde{\Theta}_{jd}^T - \kappa_{j\Omega}F_{j\Omega}\hat{W}_{j\Omega} \tag{45}$$

where $F_{j\Omega} = F_{j\Omega}^T > 0$ and $\kappa_{j\Omega} > 0$ are design parameters. Let the dynamic NN controller for follower *j* be defined by (39) and (42), respectively, with the NN update given by

$$\dot{\hat{W}}_{jc} = F_{jc}\hat{\sigma}_{jc}(A_{jd}\hat{e}_{jD})^T - \kappa_{jc}F_{jc}\hat{W}_{jc} \tag{46}$$

where $A_{jd} = [R_{jd}\; 0_{3x3}; 0_{3x3}\; I_{3x3}] \in \Re^{6x6}$, and $F_{jc} = F_{jc}^T > 0$ and $\kappa_{jc} > 0$ are constant design parameters. Then there exists positive design constants $K_{j\Omega 1}, K_{j\Omega 2}$, $K_{j\Omega 3}$, and positive

definite design matrices $K_{j\rho}, K_{j\Theta}, K_{jv}, K_{j\omega}$, such that the virtual controller estimation errors $\tilde{\Theta}_{jd}$, $\tilde{\omega}_{jd}$ and the virtual control NN weight estimation errors, $\tilde{W}_{j\Omega}$, the position, orientation, and translational and angular velocity tracking errors, $e_{j\rho}, e_{j\Theta}, e_{jv}, e_{j\omega}$, respectively, and the dynamic controller NN weight estimation errors, $\tilde{W}_{jc}$, are all *SGUUB*.

Proof: Consider the following Lyapunov candidate

$$V_j = K_{j\omega Max}^2 V_{j\Omega} + V_{jc} \tag{47}$$

where $K_{j\omega Max} > 0$ is the maximum singular value of $K_{j\omega}$,

$$V_{j\Omega} = \frac{1}{2}\tilde{\Theta}_{jd}^T \tilde{\Theta}_{jd} + \frac{1}{2}\tilde{\omega}_{jd}{}^T \tilde{\omega}_{jd} + \frac{1}{2} tr\{\tilde{W}_{j\Omega}^T F_{j\Omega}^{-1} \tilde{W}_{j\Omega}\},$$

and

$$V_{jc} = \frac{1}{2} e_{j\rho}^T e_{j\rho} + \frac{1}{2} e_{j\Theta}^T e_{j\Theta} + \frac{1}{2} e_{jv}^T e_{jv} + \frac{1}{2} e_{j\omega}^T J e_{j\omega} + \frac{1}{2} tr\{\tilde{W}_{jc}^T F_{jc}^{-1} \tilde{W}_{jc}\}$$

whose first derivative with respect to time is given by $\dot{V}_j = K_{j\omega Max}^2 \dot{V}_{j\Omega} + \dot{V}_{jc}$. Considering first, $\dot{V}_{j\Omega}$ and substituting the closed loop virtual control estimation error dynamics (30) and (31) as well as the NN tuning law (45), yields

$$\begin{aligned} \dot{V}_{j\Omega} = & -\tilde{\Theta}_{jd}^T (K_{j\Omega 1} - K_{j\Omega 3}) \tilde{\Theta}_{jd} - \tilde{\omega}_{jd}{}^T K_{j\Omega 3} \tilde{\omega}_{jd} + \\ & \tilde{\omega}_{jd}{}^T (T_j^{-1} (K_{j\Omega 3} (K_{j\Omega 1} - K_{j\Omega 3}) - K_{j\Omega 2}) \tilde{\Theta}_{jd}) - \tilde{\omega}_{jd}{}^T T_j^T \tilde{\Theta}_{jd} + \tilde{\Theta}_{jd}^T T_j \tilde{\omega}_{jd} \\ & -tr\{\tilde{W}_{j\Omega}^T \hat{\sigma}_{j\Omega} (\tilde{\Theta}_{jd} + \tilde{\omega}_{jd})^T\} + \kappa_{j\Omega} tr\{\tilde{W}_{j\Omega}^T \hat{W}_{j\Omega}\} - \tilde{\omega}_{jd}{}^T \xi_{j\Omega} \end{aligned} \tag{48}$$

Next, selecting $K_{j\Omega 2} = K_{j\Omega 3}(K_{j\Omega 1} - K_{j\Omega 3})$ and observing $\tilde{\omega}_{jd}{}^T T_j^T \tilde{\Theta}_{jd} = \tilde{\Theta}_{jd}^T T_j \tilde{\omega}_{jd}$ and $tr\{\tilde{W}_{j\Omega}^T \hat{W}_{j\Omega}\} = tr\{\tilde{W}_{j\Omega}^T (W_{j\Omega} - \tilde{W}_{j\Omega})\}$, (48) can be upper bounded by

$$\dot{V}_{j\Omega} \le -(K_{j\Omega 1} - K_{j\Omega 3})\left\|\tilde{\Theta}_{jd}\right\|^2 - K_{j\Omega 3}\left\|\tilde{\omega}_{jd}\right\|^2 + \sqrt{N_{j\Omega}}\left\|\tilde{W}_{j\Omega}\right\|_F\left\|\tilde{\Theta}_{jd}\right\| + \sqrt{N_{j\Omega}}\left\|\tilde{W}_{j\Omega}\right\|_F\left\|\tilde{\omega}_{jd}\right\| + \kappa_{j\Omega}\left\|\tilde{W}_{j\Omega}\right\|_F W_{j\Omega M} - \kappa_{j\Omega}\left\|\tilde{W}_{j\Omega}\right\|_F^2 + \left\|\tilde{\omega}_{jd}\right\|\xi_{j\Omega M}$$

Now, completing the squares with respect to $\left\|\tilde{W}_{j\Omega}\right\|_F$, $\left\|\tilde{\Theta}_{jd}\right\|$ and $\left\|\tilde{\omega}_{jd}\right\|$ allows the upper bound of $V_{j\Omega}$ to be written as

$$\dot{V}_{j\Omega} \le -\left(K_{j\Omega 1} - K_{j\Omega 3} - \frac{N_{j\Omega}}{\kappa_{j\Omega}}\right)\left\|\tilde{\Theta}_{jd}\right\|^2 - \left(\frac{K_{j\Omega 3}}{2} - \frac{N_{j\Omega}}{\kappa_{j\Omega}}\right)\left\|\tilde{\omega}_{jd}\right\|^2 - \frac{\kappa_{j\Omega}}{4}\left\|\tilde{W}_{j\Omega}\right\|_F^2 + \eta_{j\Omega} \qquad (49)$$

where $\eta_{j\Omega} = \kappa_{j\Omega}W_{jM\Omega}^2 + \xi_{j\Omega M}^2/(2K_{j\Omega 3})$. Next, considering $\dot{V}_{jc}$, and substituting the closed loop kinematics (15) and (31), dynamics (40) and (44), and NN tuning law (46) while considering $e_{iv} = 0$ reveals

$$\dot{V}_{jc} = -e_{j\rho}^T K_{j\rho} e_{j\rho} - e_{j\Theta}^T K_{j\Theta} e_{j\Theta} - e_{jv}^T K_{jv} e_{jv} - e_{j\omega}^T K_{j\omega} e_{j\omega} + e_{j\rho}^T R_j e_{jv} + e_{j\omega}^T K_{j\omega}\tilde{\omega}_{jd} + e_{jv}^T \xi_{jc1} + e_{j\omega}^T \xi_{jc2} + tr\left\{\tilde{W}_{jc}^T \hat{\sigma}_{jc}\left(A_{jd}(e_{jD} - \hat{e}_{jD})\right)^T\right\} + \kappa_{jc} tr\{\tilde{W}_{jc}^T \hat{W}_{jc}\}$$

Now, observing $\left\|A_{jd}(e_{jD} - \hat{e}_{jD})\right\| = \left\|\tilde{\omega}_d\right\|$, $\left\|\hat{\sigma}_{jc}\right\| \le \sqrt{N_{jc1} + N_{jc2}} \equiv \sqrt{N_{jc}}$, $\left\|W_{jc}\right\|_F \le W_{jcM}$ for a known positive constant W_{jcM}, and $tr\{\tilde{W}_{jc}^T \hat{W}_{jc}\} = tr\{\tilde{W}_{jc}^T (W_{jc} - \tilde{W}_{jc})\}$, $\dot{V}_{jc}$ is upper bounded by

$$\dot{V}_{jc} \le -K_{j\rho Min}\left\|e_{j\rho}\right\|^2 - K_{j\Theta Min}\left\|e_{j\Theta}\right\|^2 - K_{jvMin}\left\|e_{jv}\right\|^2 - K_{j\omega Min}\left\|e_{j\omega}\right\|^2 + R_{Max}\left\|e_{j\rho}\right\|\left\|e_{jv}\right\| + K_{j\omega Max}\left\|\tilde{\omega}_{jd}\right\|\left\|e_{j\omega}\right\| + \left\|e_{jv}\right\|\xi_{Mc1} + \left\|e_{j\omega}\right\|\xi_{Mc2} + \sqrt{N_{jc}}\left\|\tilde{W}_{jc}\right\|_F\left\|\tilde{\omega}_{jd}\right\| + \kappa_{jc}W_{jcM}\left\|\tilde{W}_{jc}\right\|_F - \kappa_{jc}\left\|\tilde{W}_{jc}\right\|_F^2$$

where $K_{j\rho\min}$, $K_{j\Theta\min}$, $K_{jv\min}$, and $K_{j\omega\min}$ are the minimum singular values of $K_{j\rho}$, $K_{j\Theta}$, K_{jv}, and $K_{j\omega}$, respectively, and greater than zero. Next, completing the squares with respect to $\left\|e_{j\rho}\right\|, \left\|e_{j\Theta}\right\|, \left\|e_{jv}\right\|$, and $\left\|e_{j\omega}\right\|$ yields

$$\dot{V}_{jc} \le -\left(K_{j\rho Min} - \frac{3R_{Max}^2}{4K_{jvMin}}\right)\left\|e_{j\rho}\right\|^2 - K_{j\Theta Min}\left\|e_{j\Theta}\right\|^2 - \frac{K_{jvMin}}{3}\left\|e_{jv}\right\|^2 - \frac{K_{j\omega Min}}{3}\left\|e_{j\omega}\right\|^2 - \frac{\kappa_{jc}}{3}\left\|\widetilde{W}_{jc}\right\|_F^2 + \frac{3N_c\left\|\widetilde{\omega}_{jd}\right\|^2}{4\kappa_{jc}} + \frac{3K_{j\omega Max}^2\left\|e_{j\omega}\right\|^2}{4K_{j\omega Min}} + \frac{\eta_{jc}}{4} \tag{50}$$

where $\eta_{jc} = 3\kappa_{jc}W_{jcM}^2 + 3\xi_{jMc1}^2 / K_{jvMin} + 3\xi_{jMc2}^2 / K_{j\omega Min}$. Now, combining (49) and (50), an upper bound for $\dot{V}_j$ is written as

$$\begin{aligned}\dot{V}_j \le & -K_{j\omega Max}^2\left(K_{j\Omega 1} - K_{j\Omega 3} - \frac{N_\Omega}{\kappa_{j\Omega}}\right)\left\|\widetilde{\Theta}_{jd}\right\|^2 - \frac{K_{j\omega Max}^2\kappa_{j\Omega}}{4}\left\|\widetilde{W}_{j\Omega}\right\|_F^2 \\ & -K_{j\omega Max}^2\left(\frac{K_{j\Omega 3}}{2} - \frac{N_\Omega}{\kappa_{j\Omega}} - \frac{3N_c}{4\kappa_{jc}K_{j\omega Max}^2} - \frac{3}{4K_{j\omega Min}}\right)\left\|\widetilde{\omega}_{jd}\right\|^2 - \frac{K_{j\omega Min}}{3}\left\|e_{j\omega}\right\|^2 \\ & -\left(K_{j\rho Min} - \frac{3R_{Max}^2}{4K_{jvMin}}\right)\left\|e_{j\rho}\right\|^2 - K_{j\Theta Min}\left\|e_{j\Theta}\right\|^2 - \frac{K_{jvMin}}{3}\left\|e_{jv}\right\|^2 - \frac{\kappa_{jc}}{3}\left\|\widetilde{W}_{jc}\right\|_F^2 + \eta_j\end{aligned} \tag{51}$$

where $\eta_j = \eta_{jc} / 4 + K_{j\omega Max}^2\eta_{j\Omega}$. Finally, (51) is less than zero provided

$$K_{j\Omega 1} > K_{j\Omega 3} + \frac{N_{j\Omega}}{\kappa_{j\Omega}},\ K_{j\Omega 3} > \frac{2N_{j\Omega}}{\kappa_{j\Omega}} + \frac{3N_{jc}}{2\kappa_{jc}K_{j\omega Max}^2} + \frac{3}{2K_{j\omega Min}}\quad K_{j\rho Min} > \frac{3R_{Max}^2}{4K_{jvMin}} \tag{52}$$

and the following inequalities hold

$$\begin{aligned}& \left\|\widetilde{\omega}_{jd}\right\| > \sqrt{\frac{\eta_j}{K_{j\omega Max}^2\left(\frac{K_{j\Omega 3}}{2} - \frac{N_{j\Omega}}{\kappa_{j\Omega}} - \frac{3N_{jc}}{4\kappa_{jc}K_{j\omega Max}^2} - \frac{3}{4K_{j\omega Min}}\right)}} \\ \text{or}\quad & \left\|e_{j\rho}\right\| > \sqrt{\frac{\eta_j}{\left(K_{j\rho Min} - \frac{3R_{Max}^2}{4K_{jvMin}}\right)}} \quad \text{or} \quad \left\|e_{j\Theta}\right\| > \sqrt{\frac{\eta_j}{K_{j\Theta Min}}} \\ \text{or}\quad & \left\|\widetilde{W}_{j\Omega}\right\|_F > \sqrt{\frac{4\eta_j}{K_{j\omega Max}^2\kappa_{j\Omega}}} \quad \text{or} \quad \left\|\widetilde{\Theta}_{jd}\right\| > \sqrt{\frac{\eta_j}{K_{j\omega Max}^2\left(K_{j\Omega 1} - K_{j\Omega 3} - \frac{N_{j\Omega}}{\kappa_{j\Omega}}\right)}} \\ \text{or}\quad & \left\|\widetilde{W}_{jc}\right\|_F > \sqrt{\frac{3\eta_j}{\kappa_{jc}}} \quad \text{or} \quad \left\|e_{j\omega}\right\| > \sqrt{\frac{3\eta_j}{K_{j\omega Min}}} \quad \text{or} \quad \left\|e_{jv}\right\| > \sqrt{\frac{3\eta_j}{K_{jvMin}}}.\end{aligned} \tag{53}$$

Therefore, it can be concluded using standard extensions of Lyapunov theory (Lewis, Jagannathan & Yesilderek, 1999) that $\dot{V}_j$ is less than zero outside of a compact set, revealing the virtual controller estimation errors, $\widetilde{\Theta}_{jd}, \widetilde{\omega}_{jd}$, and the NN weight estimation errors, $\widetilde{W}_{j\Omega}$, the position, orientation, and translational and angular velocity tracking errors, $e_{j\rho}, e_{j\Theta}, e_{jv}, e_{j\omega}$, respectively, and the dynamic controller NN weight estimation errors, $\widetilde{W}_{jc}$, are all bounded. Finally, the initial compact set can be made arbitrarily large through proper selection of the gains; thus, all signals are *SGUUB* (Timothy, Burg, Xian & Dawson, 2007).

In the next section, results from our previous work (Dierks & Jagannathan, 2008) are revisited in the design of the formation leader control laws.

B. Formation Leader Control Law

The kinematics and dynamics for the formation leader are defined similar to (1) and (5), respectively. In our previous work (Dierks & Jagannathan, 2008), an output feedback control law for a single quadrotor UAV was designed to ensure the UAV tracks a desired path, $\rho_{id} = [x_{id}, y_{id}, z_{id}]^T$, and desired yaw angle, ψ_{id}. Using a similar approach to (10)-(15), the translational control velocity for leader *i* was found to be (Dierks & Jagannathan, 2008)

$$v_{id} = [v_{idx} \; v_{idy} \; v_{idz}]^T = R_i^T (\dot{\rho}_{id} + K_{i\rho} e_{i\rho}) \in E^b, \tag{54}$$

and the closed loop position tracking error then takes the form of

$$\dot{e}_{i\rho} = -K_{i\rho} e_{i\rho} + R_i \, e_{iv} \tag{55}$$

Then, using the leader's velocity tracking error (14) and following steps similar to (17)-(27), the desired pitch angle is given by

$$\theta_{id} = \frac{2\theta_{\max}}{\pi} a\tan\left(\frac{N_{i\theta d}}{D_{i\theta d}}\right) \tag{56}$$

where $N_{i\theta d} = c_{i\psi d}\Lambda_{i1} + s_{i\psi d}\Lambda_{i2} + k_{iv1}e_{ivx}$ and $D_{i\theta d} = \Lambda_{i3} - g$. Similarly, the desired roll angle for the leader is found to be

$$\phi_{id} = \frac{2\phi_{\max}}{\pi} a\tan\left(\frac{N_{i\phi d}}{D_{i\phi d}}\right) \tag{57}$$

where $N_{i\phi d} = c_{\psi id}\Lambda_{i2} - s_{\psi id}\Lambda_{i1} + k_{iv2}e_{ivy}$ and $D_{i\phi d} = c_{\bar{\theta}id}(g - \Lambda_{i3}) - s_{\bar{\theta}id}s_{\psi id}\Lambda_{i2} - s_{\bar{\theta}id}c_{\psi id}\Lambda_{i1}$ with $\Lambda_{i1} = \ddot{x}_{id} + k_{i\rho x}\dot{x}_{id} - v_{iR1} + \hat{f}_{ic11}$, $\Lambda_{i2} = \ddot{y}_{id} + k_{i\rho y}\dot{y}_{id} - v_{iR2} + \hat{f}_{ic12}$, $\Lambda_{i3} = \ddot{z}_{id} + k_{i\rho z}\dot{z}_{id} - v_{iR3} + \hat{f}_{ic13}$, $v_{iR} = [v_{iR1}\ v_{iR2}\ v_{iR3}]^T = K_{i\rho}R_i v_i$ and $\hat{f}_{ic1} = [\hat{f}_{ic11}\ \hat{f}_{ic12}\ \hat{f}_{ic13}]^T$ is a NN estimate of the unknown function $f_{ic1}(x_{ic1})$ (Dierks & Jagannathan, 2008). The development of the desired angular velocity as well as the NN virtual controller for the formation leader follows similar to (28)-(38), and finally, the thrust and rotation torque vector for the leader were found to be (Dierks & Jagannathan, 2008)

$$\begin{aligned} u_{i1} = {} & m_i k_{iv3}e_{ivz} + m_i(c_{\bar{\phi}id}s_{\bar{\theta}id}c_{\psi id} + s_{\bar{\phi}id}s_{\psi id})\Lambda_{i1} + m_i(c_{\bar{\phi}id}s_{\bar{\theta}id}s_{\psi id} - s_{\bar{\phi}id}c_{\psi id})\Lambda_{i2} \\ & + m_i c_{\bar{\phi}id}c_{\bar{\theta}id}(\Lambda_{i3} - g) \end{aligned} \tag{58}$$

and

$$u_{i2} = \hat{f}_{ic2} + K_{i\omega}\hat{e}_{i\omega}, \tag{59}$$

respectively, where $\hat{f}_{ic2} \in \Re^3$ is a NN estimate of an unknown function $f_{ic2}(x_{ic2})$ and $\hat{e}_{i\omega} = \hat{\omega}_{id} - \omega_i$. The closed loop orientation, virtual control, and velocity tracking error

dynamics for the formation leader are found to take a form similar to (31), (37) and (38), and (40) and (44), respectively (Dierks & Jagannathan, 2008).

A general controller structure for the follower UAV as well as the formation leader is now shown in Fig. 3 where the subscripts *'i'* and *'j'* have been omitted. In the figure, four connected systems are observed: a kinematic controller, NN virtual controller, NN dynamic controller, and the UAV dynamic system. The kinematic controller refers to the calculation of the translational control velocity and desired pitch and roll (12), (26), and (27), respectively, for the follower and (54), (56), and (57), respectively, for the leader.

The external inputs to the system are considered to be the desired position, ρ_d, and desired yaw, ψ_d. For the leader, ρ_d and ψ_d are known values. In contrast, the follower UAV calculates ρ_d according to (9) and receives ψ_d from the leader via wireless

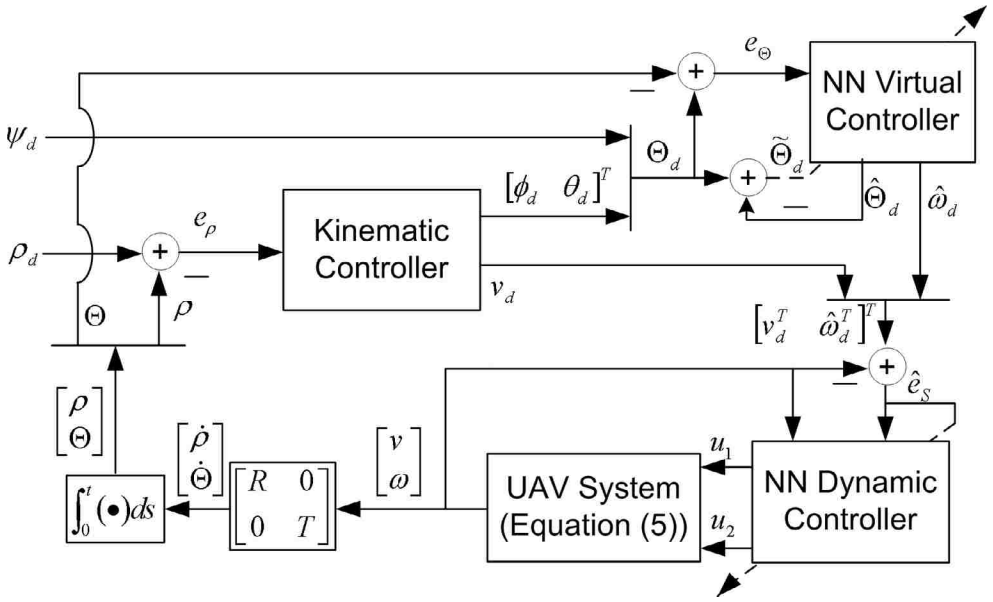


Fig. 3. Control structure for the follower and leader UAV.

communication so that the leader-follower formation control problem (8) is effectively converted into a tracking control problem. Based on the difference between the current UAV position (ρ) and the desired position, the kinematic controller generates the desired velocity v_d to ensure $\rho \to \rho_d$. Subsequently, the desired pitch, θ_d, and roll, ϕ_d, are calculated to ensure the *x* and *y* components of the desired velocity are tracked, respectively. Then, the NN virtual controller uses the information provided by the kinematic controller to generate the desired angular velocity $\hat{\omega}_d \in E^b$ which ensures $\Theta \to \Theta_d$. Then, the NN dynamic controller calculates the actual control inputs u_1 and u_2 based on the information provided by the kinematic controller and NN virtual controller.

Next, the stability of the formation leader is investigated in the following theorem.

Theorem 2 (Formation Leader Stability): Given a smooth trajectory ρ_{id} and desired yaw angle ψ_{id} for the leader *i*, let control velocity and desire pitch and roll for the leader be given by (54), (56), and (57), respectively. Let the virtual controller for the leader *i* be defined similar to (34) and (35) with the virtual control NN update law defined similar to (45). Let the thrust and rotation torque vector defined by (58) and (59), respectively, and let the control NN update law be defined similarly to (46). Then, the position, orientation, and velocity tracking errors, the virtual control estimation errors, and the virtual controller and the dynamic controller NN weight estimation errors for the formation leader *i* are all *SGUUB*.

Proof of *Theorem 2* is addressed in the following section where the stability of the formation consisting of 1 leader and N followers is shown while considering the interconnection errors between the leader and its followers.

C. Quadrotor UAV Formation Stability

Before proceeding, it is convenient to define the following augmented error systems. First, the position and translational velocity tracking errors of leader i and N follower UAVs are written as $e_\rho = [e_{i\rho}^T \; e_{j\rho}^T\big|_{j=1} .... \; e_{j\rho}^T\big|_{j=N}]^T \in \Re^{3(N+1)}$ and $e_v = [e_{iv}^T \; e_{jv}^T\big|_{j=1} .... e_{jv}^T\big|_{j=N}]^T \in \Re^{3(N+1)}$. Next, the transformation matrix (2) is augmented as

$$R_F = diag\{R_i, R_j\big|_{j=1}, ..., R_j\big|_{j=N}\} \in \Re^{3(N+1)x3(N+1)} \tag{60}$$

with $\|R_F\|_F = R_{FMax}$ for a computable constant R_{FMax} while the NN weights and activation functions for the translational velocity error system are augmented as $\hat{W}_{c1} = diag\{\hat{W}_{ic1}, \hat{W}_{jc1}\big|_{j=1}, ..., \hat{W}_{jc1}\big|_{j=N}\} \in \Re^{(N\cdot N_{jc1}+N_{ic1})}$ and $\sigma_{c1} = [\sigma_{ic1}^T \; \sigma_{jc1}^T\big|_{j=1}, ..., \sigma_{jc1}^T\big|_{j=N}]^T \in \Re^{(N\cdot N_{jc1}+N_{ic1})}$. Now, using the augmented variables above, the augmented closed loop position and translational velocity error dynamics for the entire formation are written as

$$\dot{e}_\rho = -K_\rho e_\rho + (I_{3(N+1)x3(N+1)} - G_F)R_F \, e_v \tag{61}$$

and

$$\dot{e}_v = -K_v e_v + A_{dF}\tilde{W}_{c1}^T\hat{\sigma}_{c1} - R_F^T K_\rho G_F R_F e_v + \xi_{c1}, \tag{62}$$

respectively, where $A_{dF} = diag\{A_{id}, \; A_{jd}\big|_{j=1}, ..., A_{jd}\big|_{j=N}\}$ with A_{id} defined similarly to A_{jd} in terms of $\overline{\Theta}_{id}$, ξ_{c1} is an appropriately defined vector consisting of $\xi_{ic1}, \xi_{jc1}\big|_{j=1}$, etc.,

$K_\rho = diag\{K_{i\rho}, K_{j\rho}\big|_{j=1}, ..., K_{j\rho}\big|_{j=N}\}$, $K_v = diag\{K_{iv}, K_{jv}\big|_{j=1}, \ ..., K_{jv}\big|_{j=N}\}$, G_F is a constant matrix relating to the formation interconnection errors defined as

$$G_F = \begin{bmatrix} 0_{3x3N} & 0_{3x3} \\ F_T & 0_{3Nx3} \end{bmatrix} \in \Re^{3(N+1)x3(N+1)} \tag{63}$$

and $F_T \in \Re^{3Nx3N}$ is a matrix of ones and zeros and is dependent on the specific formation topology. For instance, in a string formation where each follower follows the UAV directly in front of it, follower *1* tracks leader *i,* follower *2* tracks follower *1*, etc., and F_T becomes the identity matrix. Further, it is observed that $\|G_F\|_F = \sqrt{3N}$.

Next, augmented variables for the orientation and angular velocity tracking errors are written as $e_\Theta = [e_{i\Theta}^T \ e_{j\Theta}^T\big|_{j=1} .... \ e_{j\Theta}^T\big|_{j=N}]^T \in \Re^{3(N+1)}$ and $e_\omega = [e_{i\omega}^T \ e_{j\omega}^T\big|_{j=1} .... \ e_{j\omega}^T\big|_{j=N}]^T \in \Re^{3(N+1)}$, and the rotational transformation matrix (4) is augmented as $T_F = diag\{T_i, T_j\big|_{j=1}, \ ..., T_j\big|_{j=N}\} \in \Re^{3(N+1)x3(N+1)}$. The NN weights and activation functions for the angular velocity error system are augmented as $\hat{W}_{c2} = diag\{\hat{W}_{ic2}, \hat{W}_{jc2}\big|_{j=1}, ..., \hat{W}_{jc2}\big|_{j=N}\} \in \Re^{(N \cdot N_{jc2} + N_{ic2})}$ and $\hat{\sigma}_{c2} = [\hat{\sigma}_{ic2}^T \ \hat{\sigma}_{jc2}^T\big|_{j=1}, ..., \ \hat{\sigma}_{jc2}^T\big|_{j=N}]^T \in \Re^{(N \cdot N_{jc2} + N_{ic2})}$, and the augmented closed loop orientation and angular velocity error dynamics for the entire formation are written as

$$\dot{e}_\Theta = -K_\Theta e_\Theta + T_F \, e_\omega \tag{64}$$

and

$$J\dot{e}_\omega = -K_\omega e_\omega + \tilde{W}_{c2}^T \hat{\sigma}_{c2} - T_F^T e_\Theta + K_\omega \tilde{\omega}_d + \xi_{c2}, \tag{65}$$

respectievely, with $K_\Theta = diag\{K_{i\Theta}, K_{j\Theta}\big|_{j=1}, ..., K_{j\Theta}\big|_{j=N}\}$, $K_\omega = diag\{K_{i\omega}, \ K_{j\omega}\big|_{j=1}, ..., K_{j\omega}\big|_{j=N}\}$, $J = diag\{J_i, J_j\big|_{j=1}, ..., J_j\big|_{j=N}\}$ and ξ_{c2} is an appropriately defined vector consisting of ξ_{ic2}, $\xi_{jc2}\big|_{j=1}$, etc. The vectors $\widetilde{\omega}_d$ and $\widetilde{\Theta}_d$ are the augmented virtual control estimation errors written as $\widetilde{\omega}_d = [\widetilde{\omega}_{id}^T, \ \widetilde{\omega}_{jd}^T\big|_{j=1} ... \widetilde{\omega}_{jd}^T\big|_{j=N}]^T \in \Re^{3(N+1)}$ and $\widetilde{\Theta}_d = [\widetilde{\Theta}_{id}^T, \ \widetilde{\Theta}_{jd}^T\big|_{j=1} ... \ \widetilde{\Theta}_{jd}^T\big|_{j=N}]^T \in \Re^{3(N+1)}$.

From (37) and (38), the dynamics of the augmented virtual controller are

$$\dot{\widetilde{\Theta}}_d = T_F \widetilde{\omega}_d - (K_{\Omega 1} - K_{\Omega 3})\widetilde{\Theta}_d \tag{66}$$

and

$$\dot{\widetilde{\omega}}_d = -K_{\Omega 3}\widetilde{\omega}_d + \widetilde{W}_\Omega^T \hat{\sigma}_\Omega - T_F^T \widetilde{\Theta}_d + \xi_\Omega, \tag{67}$$

respectively, where $K_{\Omega 1} = diag\{K_{\Omega 1} I_{3x3}, \ K_{\Omega 1} I_{3x3}\big|_{j=1} ... K_{\Omega 1} I_{3x3}\big|_{j=N}\}$, $K_{\Omega 3} = diag\{K_{\Omega 3} I_{3x3}, \ K_{\Omega 3} I_{3x3}\big|_{j=1}$ $... K_{\Omega 3} I_{3x3}\big|_{j=N}\}$, and $K_{\Omega 2} = \ K_{\Omega 3}(K_{\Omega 1} - K_{\Omega 3})$, and $\xi_\Omega = diag\{\xi_{i\Omega}^T, \xi_{j\Omega}^T\big|_{j=1} ... \xi_{j\Omega}^T\big|_{j=N}\}$. The augmented NN variables for the augmented virtual controller are given by $\hat{W}_\Omega = diag\{\hat{W}_{i\Omega}, \hat{W}_{j\Omega}\big|_{j=1}, ..., \ \hat{W}_{j\Omega}\big|_{j=N}\} \in \Re^{(N \cdot N_{j\Omega} + N_{i\Omega})}$ and $\hat{\sigma}_\Omega = [\hat{\sigma}_{i\Omega}^T \ \hat{\sigma}_{j\Omega}^T\big|_{j=1}, ..., \hat{\sigma}_{j\Omega}^T\big|_{j=N}]^T \in \Re^{(N \cdot N_{j\Omega} + N_{i\Omega})}$.

As a final step in defining the augmented error systems, we define the augmented NN weight updates for the virtual control and dynamic controller to be

$$\dot{\hat{W}}_\Omega = F_\Omega \hat{\sigma}_\Omega \widetilde{\Theta}_d^T - F_\Omega \kappa_\Omega \hat{W}_\Omega \tag{68}$$

and

$$\dot{\hat{W}}_c = F_c \hat{\sigma}_c (A_d \hat{e}_D)^T - F_c \kappa_c \hat{W}_c, \tag{69}$$

respectively, where $F_\Omega = diag\{F_{i\Omega}, F_{j\Omega}\big|_{j=1} ... F_{j\Omega}\big|_{j=N}\}$, $F_c = diag\{F_{ic}, F_{jc}\big|_{j=1} ... \ F_{jc}\big|_{j=N}\}$, $\hat{W}_c = diag\{\hat{W}_{ic,} \ \hat{W}_{jc}\big|_{j=1} ... \ \hat{W}_{jc}\big|_{j=N}\}$, $\hat{\sigma}_c = [\sigma_{c1}^T \ \hat{\sigma}_{c2}^T]^T$, $\kappa_c = diag\{\kappa_{ic} I, \kappa_{jc} I\big|_{j=1} \ \ ... \kappa_{jc} I\big|_{j=N}\}$,

$\kappa_\Omega = diag\{\kappa_{i\Omega} I, \kappa_{j\Omega} I\big|_{j=1} \ldots \kappa_{j\Omega} I\big|_{j=N}\}$ with each I being an appropriately dimensioned identity matrix, $e_D = [e_v^T \;\; e_\omega^T]^T$ and $\hat{e}_D = [e_v^T \;\; \hat{e}_\omega^T]^T$.

A general formation controller structure is now shown in Fig. 4 where each UAV control block contains the controller structure shown in Fig. 3. Additionally, communication links have been illustrated. In the figure, each UAV can have multiple follower UAVs, and local sensors (not shown) are utilized by the follower UAVs to measure their locations relative to their respective leaders. Starting from the top of the figure, the formation leader i has $P+1$ followers and communicates its measured

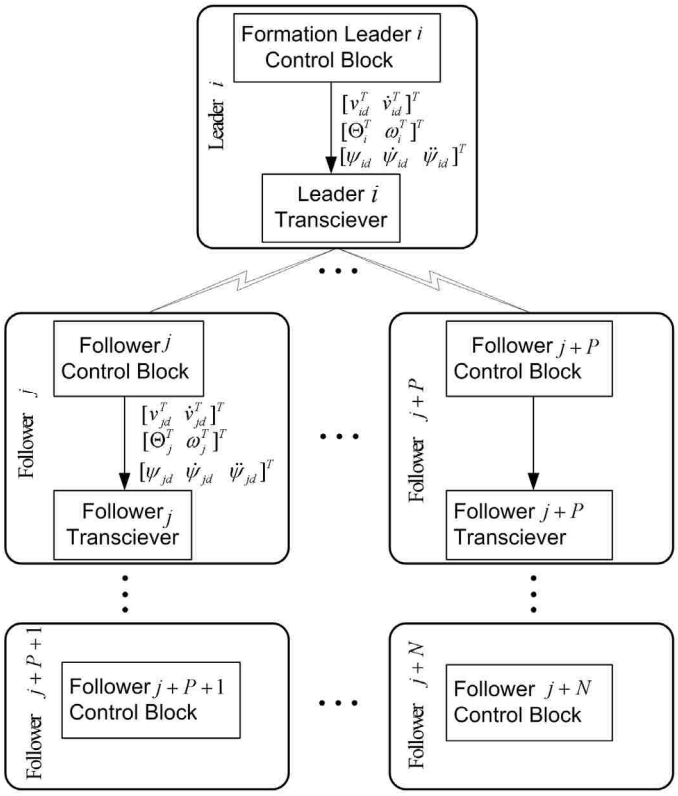


Fig. 4. Formation control structure.

orientation and angular rate vectors, Θ_i and ω_i, respectively, and its *desired* states, $\psi_{id}, \dot{\psi}_{id}, \ddot{\psi}_{id}$, $v_{id}, \dot{v}_{id}$ to each follower. Next, in the second layer of UAVs, followers *j* through *j+P* become leaders to followers *j+P+1* through *j+N,* respectively.

Note that follower *j* does not explicitly communicate the states of the leader *i* to its followers. However, by construction, the desired states v_{jd} and $\dot{v}_{jd}$ contain the states of the formation leader *i*. Thus, followers *j+P+1* through *N* inherently bring in the dynamics of leader *i* by considering the dynamics of followers *j* through *j+P,* respectively.

Now, the following theorem can be stated regarding the stability of the entire UAV formation.

Theorem 3: (UAV Formation Stability) Given the leader-follower criterion of (8) with one leader and *N* followers, let the hypotheses of *Theorem 1* and *Theorem 2* hold. Then, the position, orientation, and velocity tracking errors, the virtual control estimation errors and the virtual controller and the dynamic controller NN weight estimation errors for the entire formation are all *SGUUB*.

Proof: Consider the following positive definite Lyapunov candidate

$$V_F = \Gamma_{c1}^2 K_{\omega Max} V_\Omega + V_c \tag{70}$$

where $\Gamma_{c1} = \sqrt{3(N+1)} + \sqrt{3N} R_{FMax} > 0$ and $K_{\omega Max} > 0$ is the maximum singular value of K_ω,

$$V_\Omega = \frac{1}{2}\tilde{\Theta}_d^T \tilde{\Theta}_d + \frac{1}{2}\tilde{\omega}_d{}^T \tilde{\omega}_d + \frac{1}{2} tr\{\tilde{W}_\Omega^T F_\Omega^{-1} \tilde{W}_\Omega\}, \tag{71}$$

and

$$V_c = \frac{1}{2} e_\rho^T e_\rho + \frac{\Gamma_{c1}^2}{2}(e_\Theta^T e_\Theta + e_v^T e_v + e_\omega^T J e_\omega + tr\{\tilde{W}_c^T F_c^{-1} \tilde{W}_c\}) \tag{72}$$

where $\widetilde{W}_c = diag\{\widetilde{W}_{ic}, \widetilde{W}_{jc}\big|_{j=1} ... \widetilde{W}_{jc}\big|_{j=N}\}$ $F_\Omega = diag\{F_{i\Omega}, F_{j\Omega}\big|_{j=1} ... F_{j\Omega}\big|_{j=N}\}$, and $F_c = diag\{F_{ic}, F_{jc}\big|_{j=1} ... F_{jc}\big|_{j=N}\}$. The derivative (70) with respect to time is given by $\dot{V}_F = K_{\omega Max}\dot{V}_\Omega + \dot{V}_c$. Considering $\dot{V}_\Omega$ and using (66), (67) and (68) while applying similar steps used in the formulation of (48)-(49) allows $\dot{V}_\Omega$ to be upper bounded as

$$\dot{V}_\Omega \le -\left(K_{\Omega 1\min} - K_{\Omega 3\max} - \frac{N_\Omega}{\kappa_{\Omega\min}}\right)\left\|\widetilde{\Theta}_d\right\|^2 - \left(\frac{K_{\Omega 3\min}}{2} - \frac{N_\Omega}{\kappa_{\Omega\min}}\right)\left\|\widetilde{\omega}_{jd}\right\|^2 - \frac{\kappa_{\Omega\min}}{4}\left\|\widetilde{W}_\Omega\right\|_F^2 + \eta_\Omega \quad (73)$$

where $K_{\Omega 1\min}$, $K_{\Omega 3\min}$ and $\kappa_{\Omega\min}$ are the minimum singular values of $K_{\Omega 1}, K_{\Omega 3}$, and κ_Ω, respectively, $K_{\Omega 3\max}$ is the maximum singular value of $K_{\Omega 3}$, $\eta_\Omega = \sqrt{\eta_{i\Omega}^2 + \sum_{j=1}^{N}\eta_{j\Omega}^2}$ with $\eta_{i\Omega}$ defined similarly to $\eta_{j\Omega}$, and $N_\Omega = \sqrt{N_{i\Omega}^2 + \sum_{j=1}^{N}N_{j\Omega}^2}$. Next, considering $\dot{V}_c$ and using (61), (62), (64), (65), and (69) yields

$$\begin{aligned}\dot{V}_c = &-e_\rho^T K_\rho e_\rho - \Gamma_{c1}^2 e_\Theta^T K_\Theta e_\Theta - \Gamma_{c1}^2 e_v^T K_v e_v - \Gamma_{c1}^2 e_\omega^T K_\omega e_\omega + e_\rho^T (I_{3(N+1)x3(N+1)} - G_F) R_F\, e_v \\ &+\Gamma_{c1}^2 e_\omega^T K_\omega \widetilde{\omega}_d - \Gamma_{c1}^2 e_v^T R_F^T K_\rho G_F R_F e_v + \Gamma_{c1}^2 e_v^T \xi_{c1} + \Gamma_{c1}^2 e_\omega^T \xi_{c2} \\ &+\Gamma_{c1}^2 tr\left\{\widetilde{W}_c^T (\hat{\sigma}_c (A_d e_D - A_d \hat{e}_D)^T + \kappa_c \hat{W})\right\}\end{aligned} \quad (74)$$

Next, (74) can be upper bounded as

$$\begin{aligned}\dot{V}_c \le &-K_{\rho\min}\left\|e_\rho\right\|^2 - K_{\Theta\min}\Gamma_{c1}^2\left\|e_\Theta\right\|^2 - \overline{K}_{v\min}\Gamma_{c1}^2\left\|e_v\right\|^2 - K_{\omega\min}\Gamma_{c1}^2\left\|e_\omega\right\|^2 - \kappa_{c\min}\Gamma_{c1}^2\left\|\widetilde{W}_c\right\|_F^2 \\ &+\left\|e_v\right\|\Gamma_{c1}^2\xi_{Mc1} + \left\|e_\omega\right\|\Gamma_{c1}^2\xi_{Mc2} + \Gamma_{c1}\left\|e_v\right\|\left\|e_\rho\right\| + K_{\omega\max}\Gamma_{c1}^2\left\|\widetilde{\omega}_d\right\|\left\|e_\omega\right\| \\ &+\Gamma_{c1}^2\sqrt{N_c}\left\|\widetilde{W}_c\right\|_F\left\|e_\omega\right\| + \kappa_{c\max}W_{cM}\Gamma_{c1}^2\left\|\widetilde{W}_c\right\|_F\end{aligned}$$

where $K_{\rho\min}, K_{\Theta\min}, K_{\omega\min}$, and $\kappa_{c\min}$ are the minimum singular values of $K_\rho, K_\Theta, K_\omega$, and κ_c, respectively, $\left\|\hat{\sigma}_c\right\| \le N_c$ for a known constant N_c, and $\overline{K}_{v\min}$ is the minimum singular value of $\overline{K}_v = K_v + R_F^T K_\rho G_F R_F$ with K_v selected to ensure $\overline{K}_v > 0$.

Now, completing the squares with respect to $\|e_\rho\|, \|e_\Theta\|, \|e_v\|, \|e_\omega\|$, and $\|\widetilde{W}_c\|_F$ yields

$$\begin{aligned}\dot{V}_c \le & -(K_{\rho\min} - 1/2)\|e_\rho\|^2 - K_{\Theta\min}\Gamma_{c1}^2\|e_\Theta\|^2 - \Gamma_{c1}^2(\overline{K}_{v\min} - 1)\|e_v\|^2 \\ & -\Gamma_{c1}^2(K_{\omega\min} - (1+\sqrt{N_c})/2 - K_{\omega\max}/2)\|e_\omega\|^2 \\ & -\Gamma_{c1}^2(\kappa_{c\min} - \sqrt{N_c}/2 - \kappa_{c\max}/2)\|\widetilde{W}_c\|_F^2 \\ & + \frac{K_{\omega\max}\Gamma_{c1}^2}{2}\|\widetilde{\omega}_d\|^2 + \frac{\Gamma_{c1}^2}{2}(\kappa_{c\max}W_{cM}^2 + \xi_{Mc1}^2 + \xi_{Mc2}^2).\end{aligned} \tag{75}$$

Next, using (73) and (75), $\dot{V}_F$ is formed as

$$\begin{aligned}\dot{V}_F \le & -\Gamma_{c1}^2 K_{\omega Max}\left(K_{\Omega 1\min} - K_{\Omega 3\max} - \frac{N_\Omega}{\kappa_{\Omega\min}}\right)\|\widetilde{\Theta}_d\|^2 - \frac{\Gamma_{c1}^2 K_{\omega Max}\kappa_{\Omega\min}}{4}\|\widetilde{W}_\Omega\|_F^2 \\ & -\Gamma_{c1}^2 K_{\omega Max}\left(\frac{K_{\Omega 3\min}}{2} - \frac{N_\Omega}{\kappa_{\Omega\min}} - \frac{1}{2}\right)\|\widetilde{\omega}_{jd}\|^2 \\ & -(K_{\rho\min} - 1/2)\|e_\rho\|^2 - K_{\Theta\min}\Gamma_{c1}^2\|e_\Theta\|^2 - \Gamma_{c1}^2(\overline{K}_{v\min} - 1)\|e_v\|^2 \\ & -\Gamma_{c1}^2(K_{\omega\min} - (1+\sqrt{N_c})/2 - K_{\omega\max}/2)\|e_\omega\|^2 \\ & -\Gamma_{c1}^2(\kappa_{c\min} - \sqrt{N_c}/2 - \kappa_{c\max}/2)\|\widetilde{W}_c\|_F^2 + \eta_F\end{aligned} \tag{76}$$

where $\eta_F = \Gamma_{c1}^2(\kappa_{c\max}W_{cM}^2 + \xi_{Mc1}^2 + \xi_{Mc2}^2 + 2K_{\omega Max}\eta_\Omega)/2$. The first eight terms of (76) are less than zero provided the augmented gains are selected according to

$$\begin{aligned}& K_{\Omega 1\min} > K_{\Omega 3\max} + \frac{N_\Omega}{\kappa_{\Omega\min}},\ K_{\Omega 3\min} > \frac{2N_\Omega}{\kappa_{\Omega\min}} + 1, K_{\rho\min} > \frac{1}{2}, \\ & \overline{K}_{v\min} > 1, K_{\omega\min} > (1+\sqrt{N_c})/2 + K_{\omega\max}/2 \\ & \kappa_{c\min} > \sqrt{N_c}/2 + \kappa_{c\max}/2\end{aligned}, \tag{77}$$

and $\dot{V}_F$ is less than zero provided the gains are selected according to (77) and the following inequalities hold:

$$
\begin{aligned}
&\left\|\widetilde{\Theta}_d\right\| > \sqrt{\frac{\eta_F}{\Gamma_{c1}^2 K_{\omega Max}\left(K_{\Omega 1\min} - K_{\Omega 3\max} - \frac{N_\Omega}{\kappa_{\Omega\min}}\right)}} \\
\text{or} \quad &\left\|\widetilde{W}_\Omega\right\|_F > \sqrt{\frac{4\eta_F}{\Gamma_{c1}^2 K_{\omega Max}\kappa_{\Omega\min}}} \quad \text{or} \quad \|e_v\| > \sqrt{\frac{\eta_F}{\Gamma_{c1}^2(\overline{K}_{v\min} - 1)}} \\
\text{or} \quad &\|\widetilde{\omega}_d\| > \sqrt{\frac{\eta_F}{\Gamma_{c1}^2 K_{\omega Max}\left(\frac{K_{\Omega 3\min}}{2} - \frac{N_\Omega}{\kappa_{\Omega\min}} - \frac{1}{2}\right)}} \\
\text{or} \quad &\|e_\rho\| > \sqrt{\frac{\eta_F}{(K_{\rho\min} - 1/2)}} \quad \text{or} \quad \|e_\Theta\| > \sqrt{\frac{\eta_F}{K_{\Theta\min}\Gamma_{c1}^2}} \\
\text{or} \quad &\|e_\omega\| > \sqrt{\frac{\eta_F}{\Gamma_{c1}^2(K_{\omega\min} - (1+\sqrt{N_c})/2 - K_{\omega\max}/2)}} \\
\text{or} \quad &\left\|\widetilde{W}_c\right\|_F > \sqrt{\frac{\eta_F}{\Gamma_{c1}^2(\kappa_{c\min} - \sqrt{N_c}/2 - \kappa_{c\max}/2)}}.
\end{aligned}
\tag{78}
$$

Therefore, it can be concluded using standard extensions of Lyapunov theory (Lewis, Jagannathan & Yesilderek, 1999) that $\dot{V}_F$ is less than zero outside of a compact set, revealing the augmented virtual controller estimation errors, $\widetilde{\Theta}_d$, $\widetilde{\omega}_d$, and the NN weight estimation errors, $\widetilde{W}_\Omega$, the augmented position, orientation, and translational and angular velocity tracking errors, $e_\rho, e_\Theta, e_v, e_\omega$, respectively, and the augmented dynamic controller NN weight estimation errors, $\widetilde{W}_c$, are all bounded. Finally, the initial compact set can be made arbitrarily large through proper selection of the gains; thus, the formation errors are all *SGUUB* (Timothy, Burg, Xian & Dawson, 2007).

Remark 4: The conclusions of *Theorem 3* are independent of any specific formation topology, and the Lyapunov candidate (70) represents the most general form required to show the stability of the entire formation. Examining (77) and (78), the minimum value of the controller gains and the error bounds increases with the number of follower UAV's, *N*. These results are not surprising since increasing the number of UAV's will increase the sources of errors propagated throughout the formation.

Remark 5: Once a specific formation topology has been decided and set in the form of F_T, the results of *Theorem 3* can be reformulated more precisely. For this case, the stability of the formation is proven using the sum of the individual Lyapunov candidates of each UAV as opposed to using the augmented error systems.

IV. Simulation Results

A wedge formation of five heterogeneous quadrotor UAVs is now considered in MATLAB with the formation leader located at the apex of the wedge as shown in Fig. 5 where the abbreviations *L, F1, F2, F3,* and *F4* have been used to denote the formation

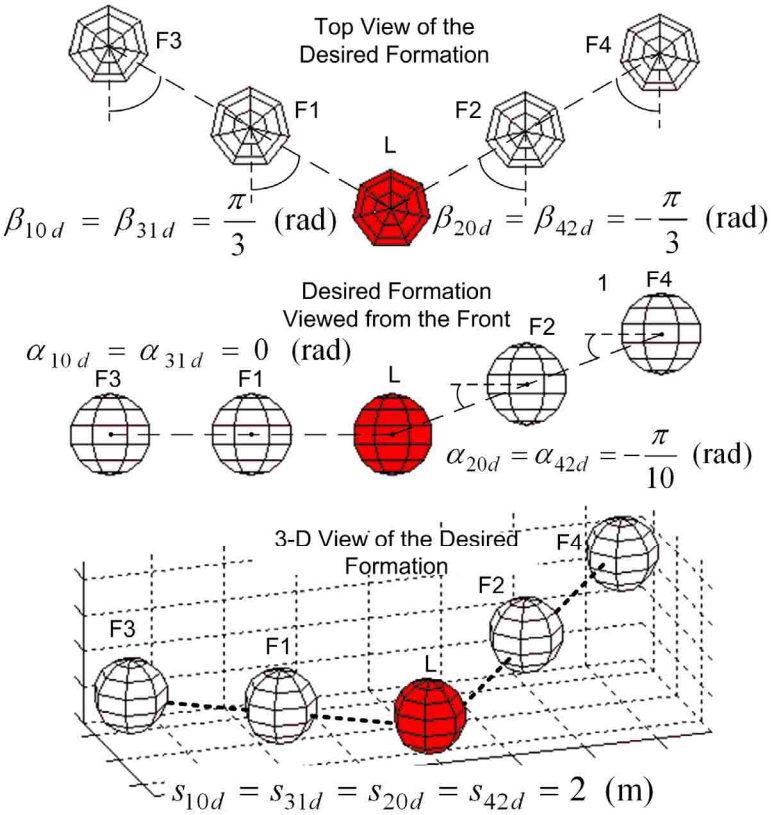


Fig. 5. Desired formation topology.

leader, follower 1, follower 2, follower 3, and follower 4, respectively. In addition, the leader UAV will be numbered as UAV *0*. In the figure, follower 1 should track the leader while follower 3 should track follower 1 at a desired separation $s_{jid} = 2\ m$, desired angle of incidence $\alpha_{jid} = 0\ (rad)$, and desired bearing $\beta_{jid} = \pi/3\ (rad)$, respectively. On the right side of the formation, follower 2 tracks the leader while follower 4 tracks follower 2 at a desired separation $s_{jid} = 2\ m$ desired angle of incidence, $\alpha_{jid} = -\pi/10\ (rad)$, and desired bearing $\beta_{jid} = -\pi/3\ (rad)$, respectively.

The desired position ([m]) and yaw ([rad]) for the leader to track is designated to be $\rho_d = [A_x \cos(\omega_x t)(1-\exp(-r_x t^2))\quad A_y \sin(\omega_y t)(1-\exp(-r_y t^2))\qquad A_z(1-\exp(-r_z t))]^T$, and $\psi_d = 0$ with $A_x = A_y = 10,\ \omega_x = \omega_y = 0.1\pi,\ \ r_x = r_y = 0.05,\ A_z = -10,$ and $r_z = 0.25$.

The inertial parameters of each UAV in the formation are summarized in Table I. In addition, a normally distributed noise signal with zero mean and variance of 0.01 is added to each UAV's dynamic model (5) through τ_d. Unmodeled dynamics in the form of aerodynamic friction are also added to each UAV system and modeled as shown below (Dierks & Jagannathan, 2008)

$$\begin{bmatrix} N_1(v_{(\bullet)}) \\ N_2(\omega_{(\bullet)}) \end{bmatrix} = \begin{bmatrix} d_1 + d_2|v_{(\bullet)xb}| & 0 & 0 & 0 & 0 & 0 \\ 0 & d_3 + d_4|v_{(\bullet)yb}| & 0 & 0 & 0 & 0 \\ 0 & 0 & d_5 + d_6|v_{(\bullet)zb}| & 0 & 0 & 0 \\ 0 & 0 & 0 & d_7 + d_8|\omega_{(\bullet)xb}| & 0 & 0 \\ 0 & 0 & 0 & 0 & d_9 + d_{10}|\omega_{(\bullet)yb}| & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{11} + d_{12}|\omega_{(\bullet)zx}| \end{bmatrix} \begin{bmatrix} v_{(\bullet)xb} \\ v_{(\bullet)yb} \\ v_{(\bullet)zb} \\ \omega_{(\bullet)xb} \\ \omega_{(\bullet)yb} \\ \omega_{(\bullet)zb} \end{bmatrix}$$

TABLE I. UAV Dynamic Parameters

	Leader	F1	F2	F3	F4
$d_k, k=1,2,\ldots 6$	$\begin{bmatrix}0.06\\0.1\\0.06\\0.1\\0.06\\0.1\end{bmatrix}$	$\begin{bmatrix}0.1\\0.9\\0.05\\0.07\\0.07\\0.08\end{bmatrix}$	$\begin{bmatrix}0.09\\0.8\\0.07\\0.8\\0.09\\0.07\end{bmatrix}$	$\begin{bmatrix}0.1\\0.7\\0.05\\0.07\\0.05\\0.08\end{bmatrix}$	$\begin{bmatrix}0.15\\0.3\\0.17\\0.08\\0.1\\0.09\end{bmatrix}$
$d_k, k=7,8,\ldots 12$	$\begin{bmatrix}0.1\\0.15\\0.1\\0.15\\0.1\\0.15\end{bmatrix}$	$\begin{bmatrix}0.06\\0.1\\0.06\\0.1\\0.06\\0.1\end{bmatrix}$	$\begin{bmatrix}0.1\\0.9\\0.05\\0.07\\0.07\\0.08\end{bmatrix}$	$\begin{bmatrix}0.06\\0.3\\0.06\\0.25\\0.06\\0.25\end{bmatrix}$	$\begin{bmatrix}0.1\\0.5\\0.15\\0.17\\0.17\\0.18\end{bmatrix}$
$m_{(\bullet)}$ (kg)	0.9	1	1.1	1.25	1.15
$J_{(\bullet)}$ $(kg\cdot m^2)$	$diag\left\{\begin{bmatrix}0.3\\0.4\\0.6\end{bmatrix}\right\}$	$diag\left\{\begin{bmatrix}0.5\\0.5\\0.5\end{bmatrix}\right\}$	$diag\left\{\begin{bmatrix}0.5\\0.6\\0.7\end{bmatrix}\right\}$	$diag\left\{\begin{bmatrix}0.5\\0.5\\0.5\end{bmatrix}\right\}$	$diag\left\{\begin{bmatrix}0.5\\0.6\\0.7\end{bmatrix}\right\}$

where each $d_k, k=1,2,\ldots 12$ are the damping coefficients summarized in Table I. At t=10 seconds, a step disturbance is added to the translational and angular velocity dynamics with magnitudes of 2.5 and 0.25, respectively.

Each NN employed by the leader and its followers consists of 10 hidden layer neurons, and for each UAV, the control gains are selected to be, $K_{\Omega 1}=24,\ K_{\Omega 2}=80,\ K_{\Omega 3}=20$, $K_\rho = diag\{10,10,30\}$, $k_{v1}=10,\ k_{v2}=10, k_{v3}=30,$ $K_\Theta = diag\{30,30,30\}$, and $K_\omega = diag\{45,45,45\}$ based on the theorems. The NN parameters are selected as, $F_\Omega = 10, \kappa_\Omega = 1$, and $F_c = 10, \kappa_c = 0.1$, and the maximum desired pitch and roll values are both selected as $2\pi/5$ for each UAV.

Fig. 6 displays the quadrotor UAV formation trajectories. Examining the trajectories in this figure, it is important to recall that the bearing angle, β_{ji}, is measured in the inertial reference frame of the follower rotated about its yaw angle. Examining the figure, each UAV begins from the ground, and quickly tracks its respective leader upon takeoff.

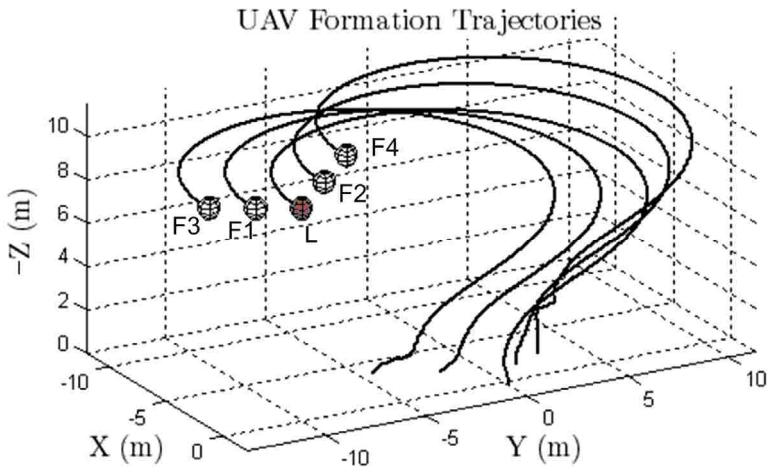


Fig. 6. Quadrotor UAV formation trajectories.

Comparing the final configuration of the UAVs shown in Fig. 6 to the desired formation topology shown in Fig. 5, one can see that the desired formation was achieved. Figures 7 through 16 show the position, orientation and the translation and angular velocity tracking errors for the leader and its followers. Examining the tracking errors for the leader and its followers in these figures, it is clear that all states track their desired values with small bounded errors consistent with the results of *Theorem 3*. Initially,

errors are observed in each state for each UAV, but these errors quickly vanish as the virtual control NN and the NN in the actual control law learns the nonlinear UAV dynamics. At t=10 seconds, a small peak in the error plots of each UAV is observed corresponding to the external step disturbance being introduced. However, the NN controllers of the UAVs quickly adapt to the changing conditions and the UAVs return to track their desired paths with small bounded errors. Additionally, the tracking performance of the underactuated states $v_{(\bullet)x}$ and $v_{(\bullet)y}$ implies that the desired pitch and roll, respectively, as well as the desired angular velocities generated by the virtual control system are satisfactory for the leader, and each follower. Further, the tracking performance confirms the theoretical conjectures derived in *Theorem 3*.

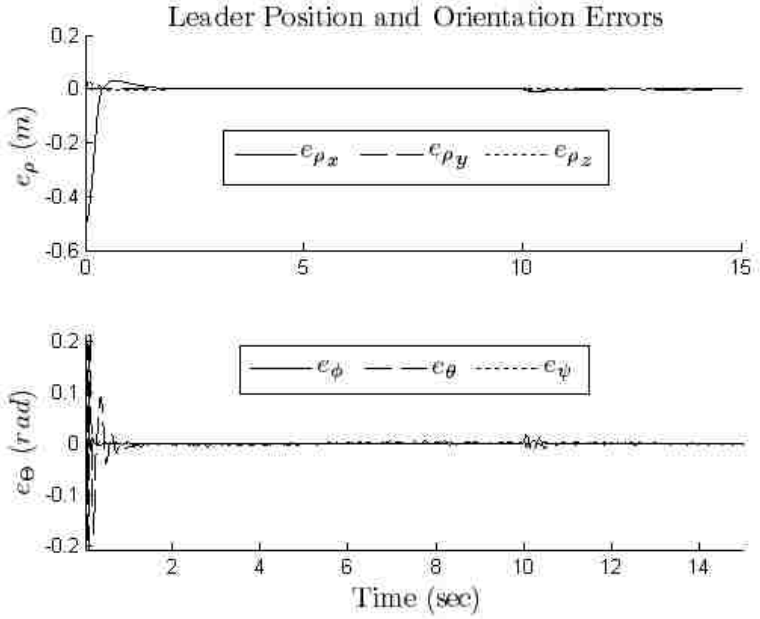


Fig. 7. Leader position and orientation tracking errors.

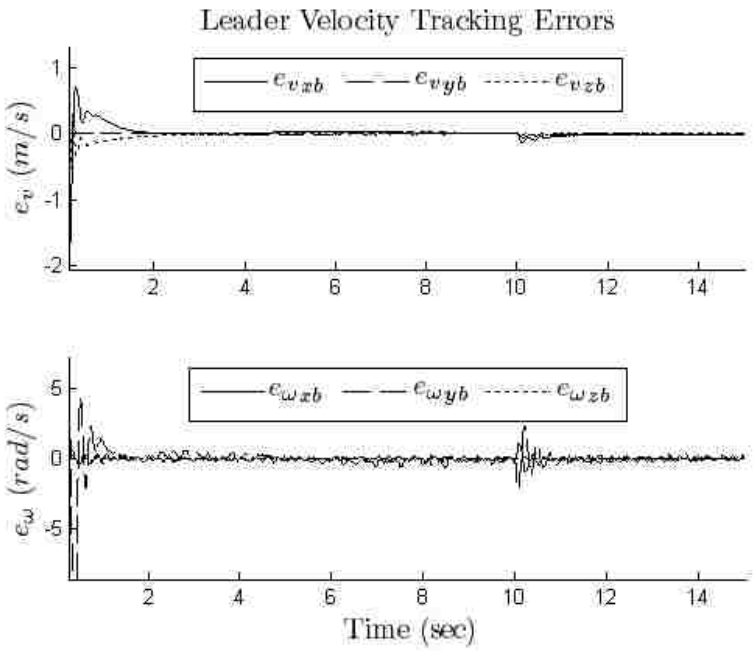


Fig. 8. Leader translational and angular velocity tracking errors.

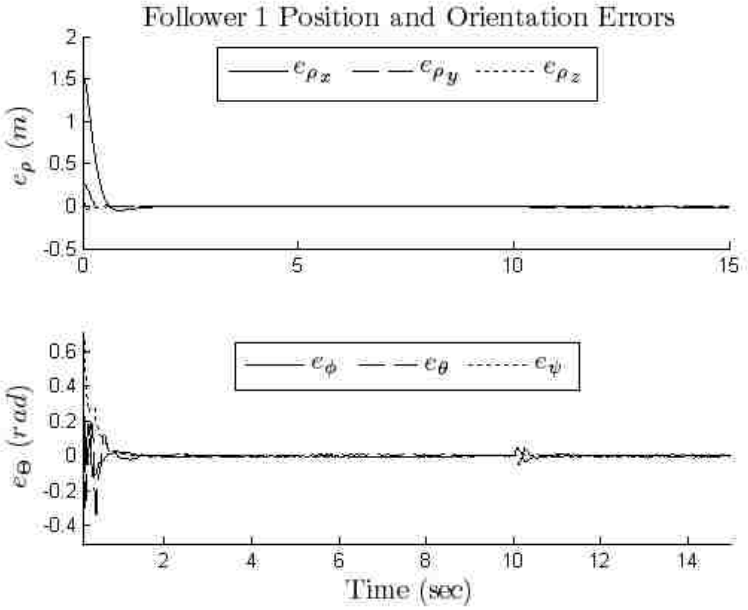


Fig. 9. Position and orientation tracking errors for follower 1.

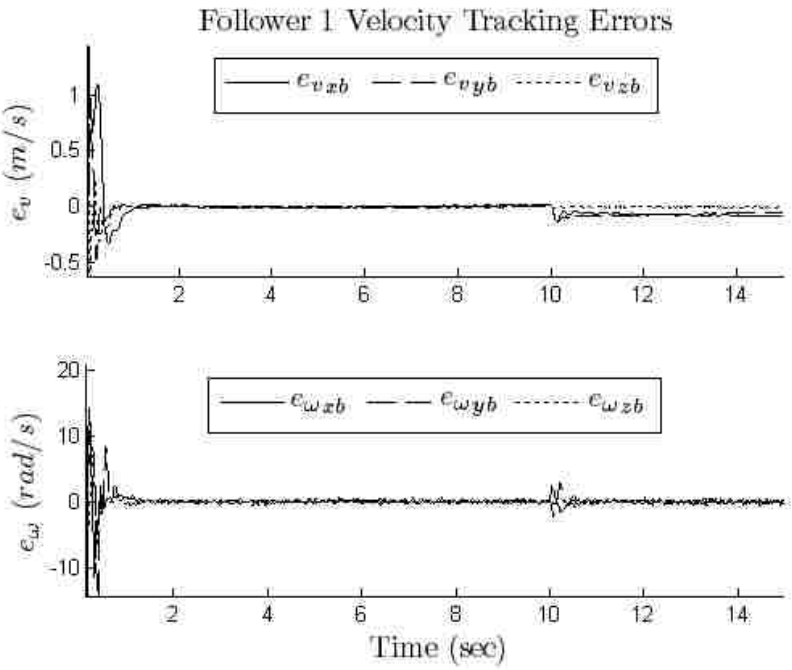


Fig. 10. Velocity tracking errors for follower 1.

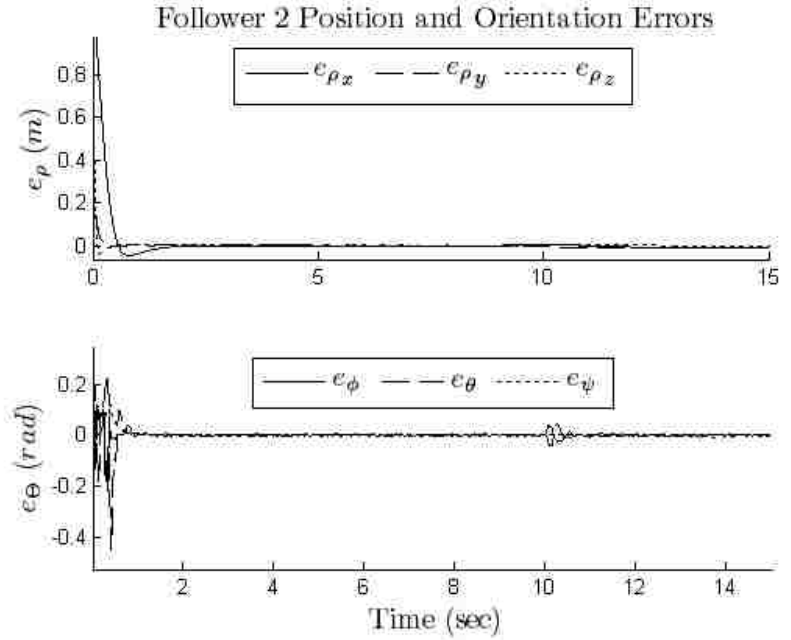


Fig. 11. Position and orientation tracking errors for follower 2.

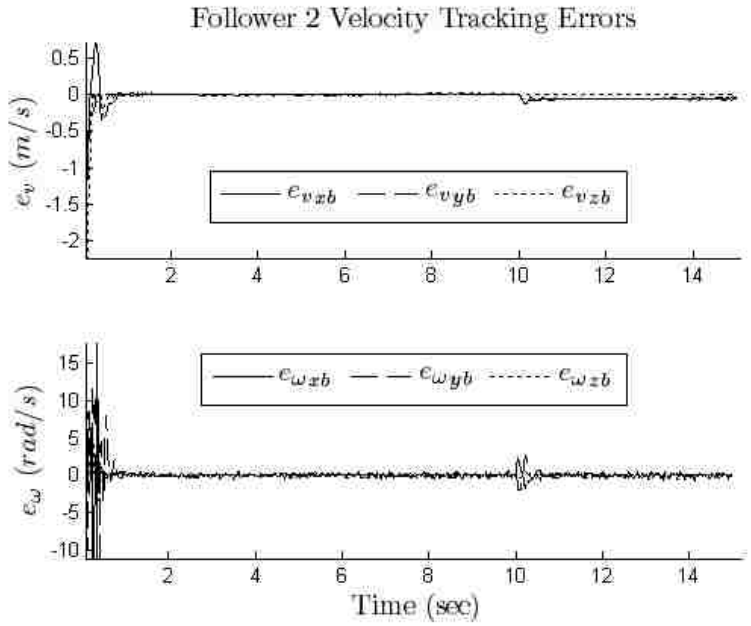


Fig. 12. Velocity tracking errors for follower 2.

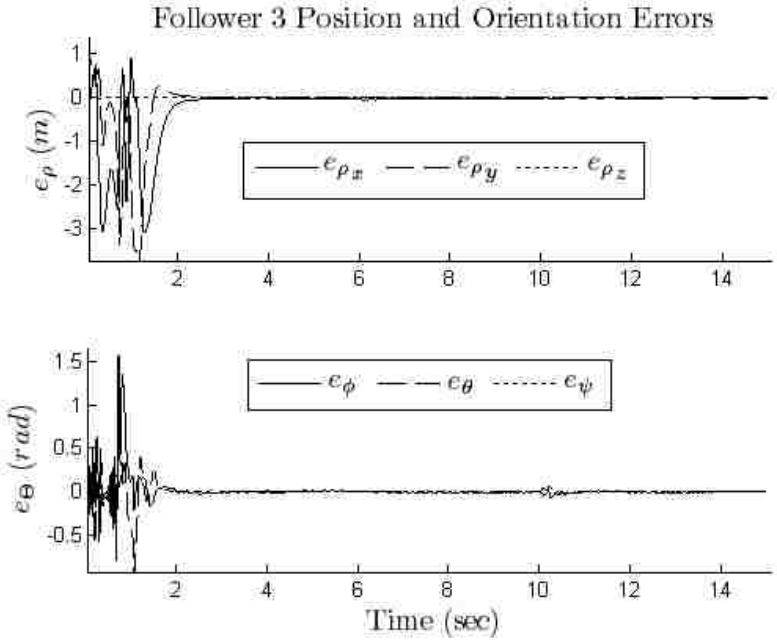


Fig. 13. Position and orientation tracking errors for follower 3.

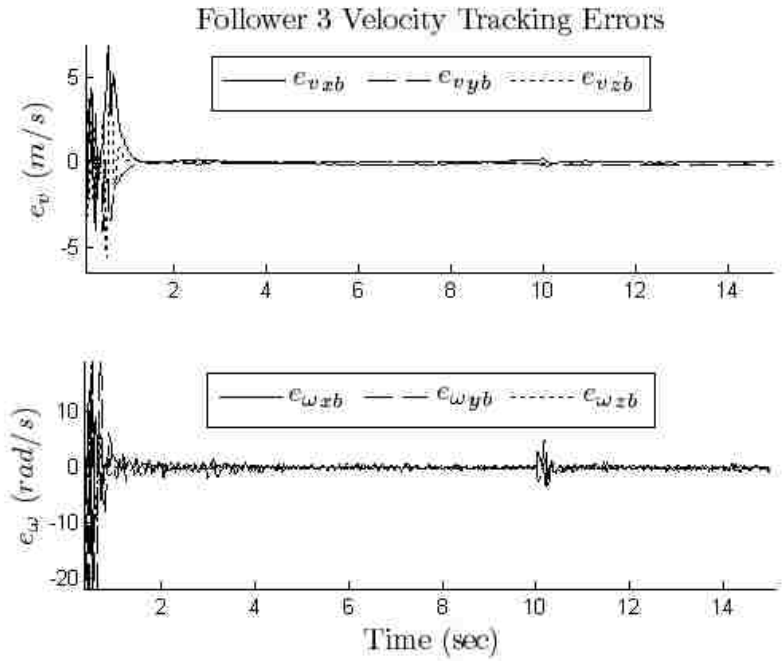


Fig. 14. Velocity tracking errors for follower 3.

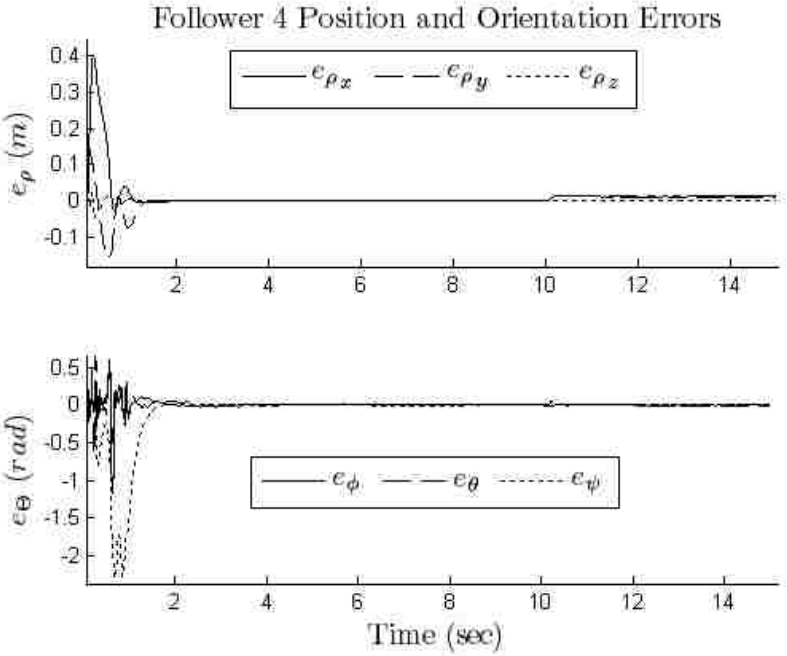


Fig. 15. Position and orientation tracking errors for follower 4.

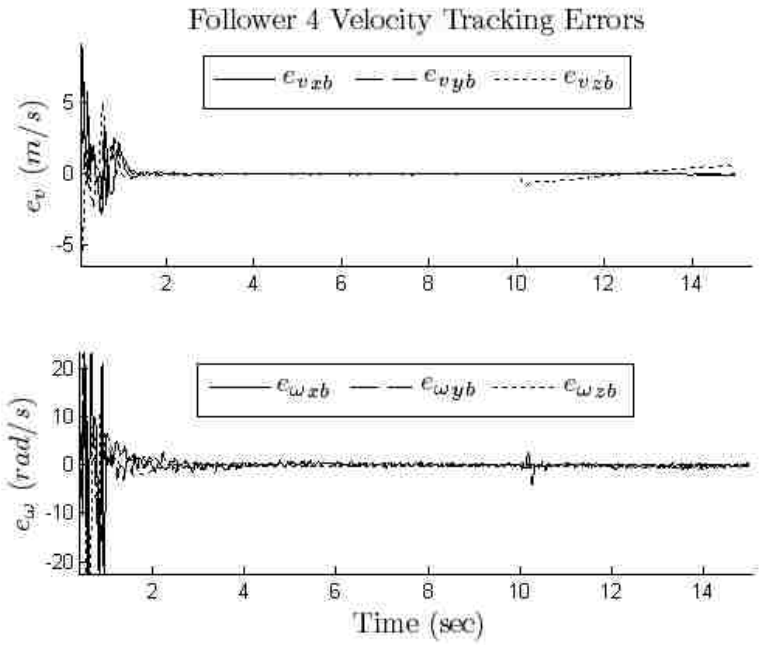


Fig. 16. Velocity tracking errors for follower 4.

Next, we investigate the importance of the formation dynamics by employing the assumption that the formation is traveling at a constant velocity (Van der Walle, Fidan, Sutton, Yu & Anderson 2008; and Kingston, Beard & Holt, 2008). In the experiment, each UAV tracks its respective leader under the assumption that its leader is traveling at a constant velocity, and thus, each UAV does not account for the formation dynamics.

The resulting formation trajectories are similar to the trajectories shown Fig. 6. Although the formation is achieved, the importance of the formation dynamics is observed by examining the velocity tracking errors for the followers. Fig. 17 displays the dynamic errors for follower 3, and it is observed that the transient response of the errors not only lasts longer, but the size of the bound on the error has increased when compared to Fig. 14 as a result of ignoring the formation dynamics. Similar results were observed for the other follower UAVs.

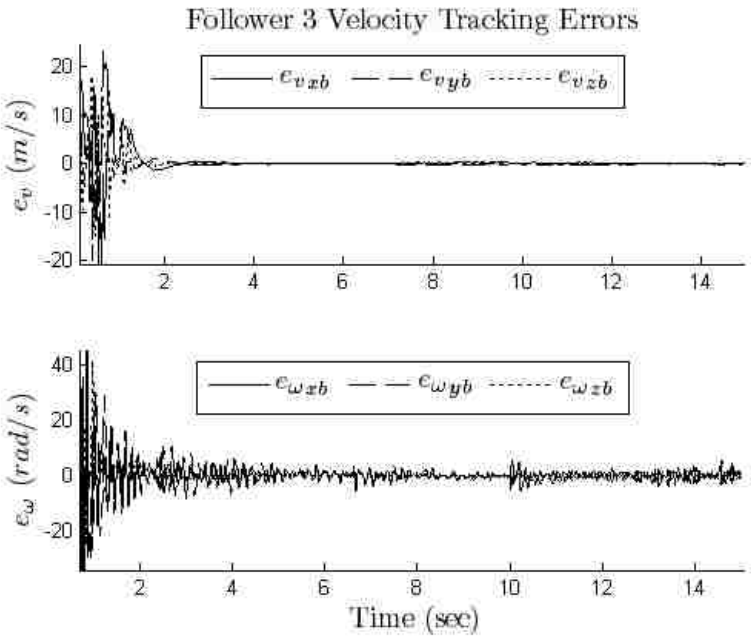


Fig. 17. Velocity tracking errors for follower 3 when the formation dynamics are ignored.

V. Conclusions

The proposed framework for quadrotor UAV leader-follower formation control using NNs for each UAV allows the follower UAVs to track their leader without the knowledge of its own and formation dynamics. By converting the formation control into a tracking control problem, and designing a NN virtual control structure, all six DOF of an underactuated UAV are successfully controlled using only four inputs while in the presence of unmodeled dynamics and bounded disturbances. Lyapunov analysis guarantees *SGUUB* of the entire formation, and numerical results confirm the theoretical conjectures.

REFERENCES

Das, A., Lewis, F., & Subbarao, K. (2008). Neural network based robust backstepping control approach for quadrotors. *Proc. of AIAA Guidance, Navigation and Control Conference and Exhibit*.

Desai, J., Ostrowski, J. P., & Kumar, V. (1998). Controlling formations of multiple mobile robots. *Proc. IEEE Int. Conf. Robot. Automat.*, 2964-2869.

Dierks, T. & Jagannathan, S. (2008). Neural network output feedback control of a quadrotor UAV. *Proc. of the IEEE Conf. on Decision and Control*, 3633-3639.

Dunfied, J., Tarbouchi, M., & Labonte, G. (2004). Neural network based control of a four rotor helicopter. *Proc. of the IEEE Int. Conf. on Industrial Technology*, 1543-1548.

Fierro, R., Belta, C., Desai, J. P., & Kumar V. (2001). On controlling aircraft formations. *Proc. IEEE Conf. on Decision and Control,* 1065-1079.

Galzi, D., & Shtessel, Y. (2006), UAV formations control using high order sliding modes. *Proc. IEEE American Control Conference,* 4249-4254.

Gu, Y., Seanor, B., Campa, G., Napolitano, M., Rowe, L., Gururajan, S., & Wan, S. (2006). Design and flight testing evaluation of formation control laws. *IEEE Trans. on Control Systems Technology,* 14, 1105-1112.

Kingston, D., Beard, R. W., & Holt, R. S. (2008). Decentralized perimeter surveillance using a team of UAVs. *IEEE Trans. on Robotics and Automation,* 24, 1394-1404.

Lewis, F.L., Jagannathan, S. & Yesilderek, A. (1999). *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, London.

Neff, A.E., DongBin, L., Chitrakaran, V.K., Dawson, D.M., & Burg, T.C. (2007). Velocity control for a quad-rotor uav fly-by-camera interface. *Proc. of the SoutheastCon,* 273-278.

Nicol, C., Macnab, C., & Ramirez-Serrano, A. (2008). Robust neural network control of a quadrotor helicopter. *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering,* 1233-1238.

Puttige, V. & Anavatti, S. (2007). Comparison of real-time online and offline neural network models for a UAV. *Proc. IEEE Int. Conf. on Neural Networks,* 412 – 417.

Saffarian, M. & Fahimi, F. (2008). Control of helicopters' formation using non-iterative nonlinear model predictive approach. *Proc. of IEEE American Control Conference,* 3707-3712.

Sastry, S. S. (1999). *Nonlinear Systems: Analysis, Stability and Control*, Springer Verlag, New York.

Suh, Y. S. (2003). Robust control of a quad-rotor aerial vehicle. *Int. Journal of Applied Electromagnetics and Mechanics,* 18, 103-114.

Timothy, D., Burg, T., Xian, B., & Dawson, D. (2007). Output feedback tracking control of an underactuated quad-Rotor UAV. *Proc. of the American Control Conference*, 1775-1780.

Van der Walle, D., Fidan, B., Sutton, A., Yu, C., & Anderson, B. (2008). Non-hierarchical UAV formation control for surveillance tasks. *Proc. IEEE American Control Conference,* 777-782.

Voos, H. (2007). Nonlinear and neural network-based control of a small four-rotor aerial robot. *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 1-6.

Xie, F., Zhang, X., Fierro, R., & Motter, M. (2005). Autopilot based nonlinear UAV formation controller with extremum-seeking. *Proc. IEEE Conf. on Decision and Control,* 4933-4938.

5. Optimal Control of Affine Nonlinear Discrete-time Systems with Unknown Internal Dynamics using Online Approximators[1]

Travis Dierks and S. Jagannathan

Abstract— *In this paper, direct dynamic programming techniques are utilized to solve the infinite-horizon Hamilton Jacobi-Bellman equation forward-in-time time for the optimal control of general affine nonlinear discrete-time systems. The proposed approach, referred normally as adaptive dynamic programming, uses online approximators (OLA's) to solve the infinite horizon optimal regulation and tracking control of affine nonlinear discrete-time systems in the presence of unknown internal dynamics and a known control coefficient matrix. For both regulation and tracking, the controller designs are implemented using OLA's to obtain the optimal feedback control signal and its associated cost function. Additionally, the tracking controller design entails a feedforward portion which is derived and approximated using an additional OLA for steady state conditions. Novel update laws for tuning the unknown parameters of the OLA's online are derived. Lyapunov techniques are used to show that all signals are uniformly ultimately bounded (UUB) and that the approximated control signals approach the optimal control inputs with small bounded error. In the absence of disturbances, an optimal control is demonstrated. Simulation results are included to show the effectiveness of the approach.*

[1] Research Supported in part by NSF ECCS#0621924 and Intelligent Systems Center. Authors are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology (formerly University of Missouri-Rolla), 1870 Miner Circle, Rolla, MO 65409. Contact author Email: tad5x4@mst.edu.

Index Terms —Online nonlinear optimal control; Hamilton Jacobi-Bellman; tracking; online approximators

I. INTRODUCTION

Online approximators (OLAs) have been widely used in the controller designs for discrete time nonlinear systems; however, stability is typically the only consideration for the resulting control laws [1]. In many cases, it is desirable that the control law not only stabilizes the system, but also minimizes a pre-defined cost function to achieve optimality. Traditionally, the optimal control of linear systems accompanied by quadratic cost functions can be achieved by solving the well known Riccati equation [2]. However, the optimal control of nonlinear discrete time systems is a much more challenging task that often requires solving the nonlinear Hamilton-Jacobi-Bellman (HJB) equation.

Although nonlinear optimal control and nonlinear H_∞ optimal control have been extensively studied for both discrete and continuous time systems [2]-[6], solving the HJB and Hamilton-Jacobi-Isaacs (HJI) equations still remain challenges. In practice, the HJB and HJI equations are more difficult to work with because they involve solving either nonlinear partial difference or differential equations [7]; therefore, several works in literature have attempted to solve the discrete time nonlinear optimal regulation problem using dynamic programming based approaches and neural networks (NN's) [7]-[8] by assuming that there are no NN reconstruction errors; however, the optimal solutions are obtained via offline training of the online approximators such as NN's.

Specifically, the authors in [7] propose an iterative solution to the generalized HJB equation and present a nearly optimal state feedback control law for affine nonlinear discrete time systems derived using a Taylor series expansion. In [8], the authors present

an iteration-based offline solution with convergence proof to the HJB equation using heuristic dynamic programming (HDP) [13]. While proof of convergence is shown in [7] and [8], the NN reconstruction errors are considered negligible in both cases. In addition to NN's, Taylor series expansions and Galerkin approximation techniques have also been used to estimate the solution to the HJI equation [9]-[10].

To overcome the iterative offline training methodology, several online approximator-based controller designs were presented in [11]-[13], and are often referred to as forward dynamic programming (FDP) or adaptive critic designs (ACD). The central theme of the approaches [11] and [12] as well as several works in [13] is that the optimal control law and cost function are approximated by online parametric structures, such as NN's. Although the techniques [11]-[13] are verified via numerical simulations, the reconstruction or approximation errors are not considered and mathematical proofs of convergence are not offered.

In addition to the optimal regulation problem, the optimal tracking control problem has been considered in recent literature through linearization of the tracking error equations [16], receding horizon optimal control [17], inverse optimal control [19], and directly calculating the infinite horizon HJB equation via offline scheme [20]. In [16], the authors consider the H_∞ optimal tracking control by linearizing the error equations about the origin yielding a locally optimal control law. The effort in [17] considers the receding horizon optimal tracking control by linearizing the nonlinear error dynamics about the origin [16].

To extend the results of linear optimal control theory to nonlinear systems, the state dependent Riccati equation (SDRE) [18] was proposed; however, the optimal

control is developed under some tight assumptions including the need for full knowledge of the system dynamics. To overcome linearization, the authors in [20] consider the HJB equation and employ similar techniques as [8] to find an offline solution to the optimal tracking control problem via HDP. Besides ignoring the online approximator (OLA) reconstruction errors, complete system dynamics are needed to implement offline training.

In this work, a novel direct dynamic programming (DDP) approach to the optimal regulation of nonlinear discrete-time affine systems is first undertaken to solve the HJB equation online. Using an initial stabilizing control, an OLA is tuned online to learn the HJB equation. Then, a second OLA is utilized that minimizes the cost (HJB) function based on the information provided by the first OLA. For the regulation problem, knowledge of the internal system dynamics is not required while the control coefficient matrix alone is needed. In addition, this novel DDP approach is extended to the optimal tracking control of affine nonlinear discrete-time systems when the internal dynamics of the system are unknown using OLA's. The proposed tracking controller utilized three OLAs- one for approximating the cost function, a second for generating a feedback portion of the control input whereas a third OLA is used for approximating the feedforward part of the control input. It is useful to observe that for both linear and nonlinear systems, the overall control input for the tracking problem normally contains a feedback term as well as a feedforward portion. Since the internal dynamics are considered to be unknown in this work, the third OLA is required to estimate the feedforward portion of the control input.

Novel online parameter tuning laws for the OLA's are derived. Further, Lyapunov theory is utilized to demonstrate the stability of the system while explicitly considering the approximation errors resulting from the use of the OLA's in contrast to the other works [7], [8], [20]. The OLA's considered in this work are NN's although any nonlinear approximator such as radial basis functions, splines, polynomials, and linear in the tunable parameter (LIP) adaptive control technique can be utilized.

The near optimal control laws proposed in this work are obtained without linearizing the equations about the origin [16]-[17] and are accomplished using the infinite horizon cost function in contrast with [17]. Additionally, the knowledge of the internal system dynamics are not required in contrast to [7], [8], [16], [18] and [20], and the proposed approach is solved online and forward-in-time; thus, it does not require offline NN training as in [7], [8], and [20].

This paper is organized as follows. First, background information for the discrete time nonlinear optimal regulation problem is presented in Section II. In Section III, the nearly optimal regulation control law is derived, and the stability is verified using Lyapunov theory. The nearly optimal tracking control law is developed in Section IV and the stability of the proposed scheme is verified using Lyapunov theory. Then, Section V illustrates the effectiveness of the proposed regulation and tracking schemes via numerical simulations, and Section VI provides concluding remarks.

II. BACKGROUND

Consider the affine nonlinear discrete-time system in the absence of disturbances described by

$$\begin{aligned} x(k+1) &= f(x(k)) + g(x(k))u(x(k)) \\ &= f(k) + g(k)u(k) \end{aligned} \tag{1}$$

where $x(k) \in \Re^n$, $f(k) \in \Re^n$, $g(k) \in \Re^{nxm}$ satisfies $\| g(k) \|_F \leq g_M$ where the Frobenius norm is applied, and $u(k) \in \Re^m$ is the control input. Without loss of generality, assume that the system is observable and controllable, sufficiently smooth, drift free, with $x = 0$ a unique equilibrium point on a compact set Ω. Under these conditions, the optimal control input for the nonlinear system (1) can be calculated [2]. In order to control (1) in an optimal manner, select the control sequence $u(k)$ that minimizes the infinite horizon cost function as [8]

$$J(k) = \sum_{i=0}^{\infty} r(k+i) = r(k) + \sum_{i=0}^{\infty} r(k+i+1) = r(k) + J(k+1) \tag{2}$$

for all $x(k)$, where $r(k) = Q(x(k)) + u(k)^T Ru(k)$ with $Q(x(k)) > 0$ and $R \in \Re^{mxm}$ is a symmetric positive definite matrix. Further, it is required that the control policy $u(k)$ guarantees that (2) is finite; or $u(k)$ must be admissible.

Definition 1: Admissible Control [7]. A control action $u(k)$ is admissible with respect to the infinite horizon cost function (2) on a compact set Ω provided the control action $u(k)$ is continuous on a compact set Ω, the control $u(k)$ stabilizes (1) on Ω with $u(k)|_{x(k)=0} = 0$, and $J(x(0))$ is finite for all $x(0) \in \Omega$.

The optimal control policy for (1) that minimizes (2) is found by applying the stationary condition [2]

$$\frac{\partial J(x(k))}{\partial u(k)} = 2Ru(k) + g(k)^T \frac{\partial J(x(k+1))}{\partial x(k+1)} = 0$$

and is shown to be [2]

$$u^*(k) = -\frac{1}{2} R^{-1} g(k)^T \frac{\partial J^*(x(k+1))}{\partial x(k+1)}. \quad (3)$$

where $u^*(k)$ and $J^*(\bullet)$ are the optimal control policy and optimal cost function, respectively.

Even in the presence of known dynamics, the optimal control (3) is generally unavailable for nonlinear discrete time systems. To circumvent this problem, several approaches [7]-[8] find (3) via offline iterative training while others [11]-[13] approximate (3) using online learning. In the following section, a new approach to online optimal control is presented which guarantees the optimal control policy (3) for the nonlinear system (1) is found with small bounded error while ensuring the OLA parameter estimates remain bounded close to their target values using Lyapunov theory. The authors in [11]-[13] do not provide these guarantees.

III. Near Optimal Regulation of Nonlinear Systems

The nearly optimal nonlinear regulator design entails two steps: an OLA designed to learn the HJB equation online and forward-in-time, and a second OLA designed to learn the control signal that minimizes the cost (HJB) function based on the information provided by the first OLA. Using the approximation property of OLA's [1], the cost function (2), feedback and control policy (3) have OLA representations on a compact set expressed as

$$J(x(k)) = \Phi_c^T \sigma(x(k)) + \varepsilon_c \quad (4)$$

and

$$u(x(k)) = \Phi_A^T \vartheta(x(k)) + \varepsilon_A \quad (5)$$

respectively, where Φ_c and Φ_A are the constant target OLA parameters, ε_c and ε_A are the bounded approximation errors, and $\sigma(\bullet)$ and $\vartheta(\bullet)$ are the vector activation functions for the cost and control signal OLA schemes, respectively.

The following assumptions which are common in OLA literature [1],[21] regarding the boundedness of the ideal OLA parameters are required.

Assumption 1. The upper bounds for the ideal OLA parameters are taken as $\|\Phi_c\| \le \Phi_{cM}$ and $\|\Phi_A\|_F \le \Phi_{AM}$ where Φ_{cM}, Φ_{AM} are positive constants [1].

Assumption 2. The approximation errors are upper bounded as $\|\varepsilon_c\| \le \varepsilon_{cM}$ and $\|\varepsilon_A\| \le \varepsilon_{AM}$ where ε_{cM} and ε_{AM} are positive constants [1].

Assumption 3. The gradient of the approximation error is upper bounded as $\|\partial \varepsilon_c / (\partial x(k+1))\|_F \le \varepsilon'_{cM}$ where ε'_{cM} is also a positive constant [21].

To begin the optimal regulator design, the cost function will be approximated first.

A. Cost Function Approximation for Optimal Regulator Design

The objective of the optimal control law is to stabilize the system (1) while minimizing the cost function (2). The cost function (2) will be approximated by an OLA and written as

$$\hat{J}(k) = \hat{J}(x(k)) = \hat{\Phi}_c^T(k)\sigma(x(k)) = \hat{\Phi}_c^T(k)\sigma(k) \tag{6}$$

where $\hat{J}(k)$ represents an approximated value of the original cost function $J(k)$, $\hat{\Phi}_c$ is the vector of actual parameter vector for the target OLA parameter vector, Φ_c, and $\sigma(k) = \{\sigma_\ell(k)\}_1^{L:}$ is set of activation functions which are each chosen to be basis sets and

thus are linearly independent. The basis function should satisfy $\|\sigma(x_0)\| = 0$ for $\|x_0\| = 0$ with $x_0 \in \Re^n$. Selection of $\sigma(\bullet)$ in this way ensures $J(0) = 0$ can be satisfied [2].

For convenience, define the error in the cost function as

$$e_c(k) = r(k-1) + \hat{\Phi}_c^T(k)\sigma(k) - \hat{\Phi}_c^T(k)\sigma(k-1) \tag{7}$$

whose dynamics are given by

$$e_c(k+1) = r(k) + \hat{\Phi}_c^T(k+1)\big(\sigma(k+1) - \sigma(k)\big). \tag{8}$$

Next, we define an auxiliary cost error vector as

$$E_c(k) = Y(k-1) + \hat{\Phi}_c^T(k)X(k-1) \in \Re^{1x(1+j)} \tag{9}$$

where $Y(k-1)=[r(k-1)\ \ r(k-2)\ldots r(k-1-j)]$ and $X(k-1) = [\Delta\sigma(k) \quad \Delta\sigma(k-1)\ldots\Delta\sigma(k-j)]$ with $\Delta\sigma(k) = \sigma(k) - \sigma(k-1)$, $0 < j < k-1 \in \mathbf{N}$ and $\mathbf{N}$ being the set of natural real numbers. It is useful to observe that (9) can be rewritten as $E_c(k) = [e_c(k \mid k) \ \ e_c(k \mid k-1) \cdots e_c(k \mid k-j)]$ where the notation $e_c(k \mid k-1)$ means the cost error $e_c(k-1)$ re-evaluated at time k using the actual cost parameter matrix $\hat{\Phi}_c^T(k)$. The dynamics of the auxiliary vector (9) are formed similar to (8) and revealed to be

$$E_c^T(k+1) = Y^T(k) + X^T(k)\hat{\Phi}_c(k+1). \tag{10}$$

Examining the error dynamics (10), it is observed that they closely resemble a nonlinear affine system with $\hat{\Phi}_c(k+1)$ being the control input, and $Y^T(k)$ and $\mathrm{X^T}(k)$ being nonlinear vector fields. To proceed, the following technical results are needed.

Definition 2: Linear Independent Functions [14]. A set of functions $\sigma(k) = \{\sigma_\ell(k)\}_1^{L:}$ is said to be linearly independent if $\sum_{\ell=1}^{L} c_\ell \sigma_\ell(x) = 0$ implies that $c_1 = \cdots = c_L = 0$.

Lemma 1. Let $\mu(k)$ be an admissible control such that $x(k+1) = f(k) + g(k)\mu(k)$ is asymptotically stable. If the set $\sigma(k) = \{\sigma_\ell(k)\}_1^{L:}$ is linearly independent, then the set $\Delta\sigma(k+1) = \{\sigma_\ell(k+1) - \sigma_\ell(k)\}_1^{L}$ is also linearly independent.

Proof: Consider the expression

$$\sigma(x(\infty)) - \sigma(x(j)) = \sum_{k=j}^{\infty} \big(\sigma(x(k+1)) - \sigma(x(k))\big). \tag{11}$$

Since, $\mu(k)$ is an admissible control $x(\infty) = 0$, and thus, $\sigma(x(\infty)) = 0$ allowing (11) to be rewritten as

$$-\sigma(x(j)) = \sum_{k=j}^{\infty} \big(\sigma(x(k+1)) - \sigma(x(k))\big). \tag{12}$$

Now, suppose that the *Lemma 1* is not true. Then there exists a nonzero constant vector $C_1 \in \Re^L$ such that

$$C_1^T \big(\sigma(x(k+1)) - \sigma(x(k))\big) \equiv 0. \tag{13}$$

From (12) and (13), we have $-C_1^T \sigma(x(j)) = \sum_{k=j}^{\infty} C_1^T \big(\sigma(x(k+1)) - \sigma(x(k))\big) \equiv 0$ which contradicts the hypothesis of linear independence of $\sigma(x(j))$ so that $\Delta\sigma(k+1) = \{\sigma_\ell(k+1) - \sigma_\ell(k)\}_1^{L}$ must be linearly independent. ■

Now define the cost function OLA parameter update to be

$$\hat{\Phi}_c(k+1) = X(k)\big(X^T(k)X(k)\big)^{-1}\big(\alpha_c E_c^T(k) - Y^T(k)\big) \tag{14}$$

where $0 < \alpha_c < 1$, and substituting (14) into (10) reveals

$$E_c^T(k+1) = \alpha_c E_c^T(k). \tag{15}$$

Remark 1: It is interesting to observe that the parameter update law (14) resembles the least squares update rule commonly used in offline ADP [7]-[8]; however, instead of summing over a mesh of training points [7]-[8], the update (14) represents a sum over the system's time history stored in $E_c(k)$. Thus, the update (14) uses data collected in real time instead of data formed offline [7],[8].

Remark 2: As a result of *Lemma 1*, the matrix $X^T(k)X(k)$ is invertible provided $x(k) \neq 0$. Observing the definition of the cost function (2) and OLA approximation (6), it is evident that both become zero only when $x(k) = 0$. Thus, once the system states have converged to zero, the cost function approximation is no longer be updated. This can be viewed as a persistency of excitation (PE) requirement for the inputs to the cost function OLA wherein the system states must be persistently exiting long enough for the OLA to learn the optimal cost function.

As a final step in the cost function OLA design, we define the parameter estimation error to be $\tilde{\Phi}_c(k) = \Phi_c - \hat{\Phi}_c(k)$, and rewrite (2) using the ideal OLA representation (4) revealing $\Phi_c^T\sigma(x(k)) + \varepsilon_c(k) = r(k) + \Phi_c^T\sigma(x(k+1)) + \varepsilon_c(k+1)$ which can be rewritten as

$$r(k) = -\Phi_c^T\Delta\sigma(x(k)) - \Delta\varepsilon_c(k) \tag{16}$$

where $\Delta\varepsilon_c(k) = \varepsilon_c(k+1) - \varepsilon_c(k)$. Substituting (16) into (8) as well as utilizing (7) and $e_c(k+1) = \alpha_c e_c(k)$ from (15) yields

$$\tilde{\Phi}_c^T(k+1)\Delta\sigma(x(k)) = -\alpha_c(r(k-1) + \hat{\Phi}_c^T(k)\Delta\sigma(k-1)) - \Delta\varepsilon_c(k). \tag{17}$$

In a similar manner as (16), we now form $r(k-1) = -\Phi_c^T \Delta\sigma(x(k-1)) - \Delta\varepsilon_c(k-1)$ and substitute this expression into (17) revealing $\Delta\sigma^T(x(k))\widetilde{\Phi}_c(k+1) = \alpha_c \Delta\sigma^T(x(k-1))\widetilde{\Phi}_c(k) + \alpha_c \Delta\varepsilon_c(k-1) - \Delta\varepsilon_c(k)$, and the OLA parameter estimation error dynamics are revealed to be

$$\begin{aligned}\widetilde{\Phi}_c(k+1) = \alpha_c \Delta\sigma(x(k))\left(\Delta\sigma^T(x(k))\Delta\sigma(x(k))\right)^{-1}\Delta\sigma^T(x(k-1))\widetilde{\Phi}_c(k) \\ +\Delta\sigma(x(k))\left(\Delta\sigma^T(x(k))\Delta\sigma(x(k))\right)^{-1}(\alpha_c \Delta\varepsilon_c(k-1) - \Delta\varepsilon_c(k)).\end{aligned} \tag{18}$$

Next, the boundedness of the cost function error (7) and OLA estimation error (18) is demonstrated, but first, the following definition is needed.

Definition 3 [1]*:* An equilibrium point x_e is said to be *uniformly ultimately bounded (UUB)* if there exists a compact set $S \subset \Re^{n*}$ so that for all initial states $x_0 \in S$ there exists a bound B and a time $T(B, x_o)$ such that $\|x(k) - x_e\| \le B$ for all $k \ge k_0 + T$.

Theorem 1: (Boundedness of the Cost OLA Errors). Let $\mu(k)$ be any admissible control for the controllable system (1), and let the cost OLA parameter update law be given by (14). Then, there exists a positive constant, α_c, such that the cost errors (7) and (18) are *UUB* with bounds given by $|e_c(k)| \le b'_{ec}$ and $\|\widetilde{\Phi}_c(k)\|_F \le b'_{\Phi c}$.

Proof: Consider the positive definite Lyapunov candidate

$$V_C(k) = e_c(k)^2 + \Delta\sigma_{\min}^2 \widetilde{\Phi}_c^T(k)\widetilde{\Phi}_c(k) \tag{19}$$

where $\Delta\sigma_{\min}^2$ is a positive constant given by $1/\|\Delta\sigma^T(x(k))\Delta\sigma(x(k))\| \le 1/\Delta\sigma_{\min}^2$. The existence of $\Delta\sigma_{\min}^2 > 0$ is ensured by the PE condition described in *Remark 2.* The first

difference of (19) is given by $\Delta V_C(k) = e_c(k+1)^2 + \Delta\sigma_{\min}^2 \widetilde{\Phi}_c^T(k+1)\widetilde{\Phi}_c(k) - e_c(k)^2 - \Delta\sigma_{\min}^2 \widetilde{\Phi}_c^T(k)\widetilde{\Phi}_c(k)$, and using (18) and the fact $e_c(k+1) = \alpha_c e_c(k)$ from (15) yields

$$\begin{aligned}\Delta V_C(k) = &-(1-\alpha_c^2)e_c(k)^2 - \Delta\sigma_{\min}^2 \widetilde{\Phi}_c^T(k)\widetilde{\Phi}_c(k) \\ &+\Delta\sigma_{\min}^2 \left(\left(\alpha_c \Delta\sigma^T(x(k-1))\widetilde{\Phi}_c(k) - \Delta\varepsilon_c(k) + \alpha_c \Delta\varepsilon_c(k-1)\right)\right)^T \times \\ &\left(\left(\Delta\sigma^T(x(k))\Delta\sigma(x(k))\right)^{-1}\left(\alpha_c \Delta\sigma^T(x(k-1))\widetilde{\Phi}_c(k) - \Delta\varepsilon_c(k) + \alpha_c \Delta\varepsilon_c(k-1)\right)\right)\end{aligned} \tag{20}$$

Since $\mu(k)$ is admissible, then $x(k)$ is asymptotically stable, and there exists a computable positive constant $\Delta\sigma_M$ such that $\|\Delta\sigma(x(k-1))\| \le \Delta\sigma_M$. Then, (20) is rewritten as

$$\Delta V_C(k) \le -(1-\alpha_c^2)e_c(k)^2 - \left(\Delta\sigma_{\min}^2 - 2\alpha_c^2\Delta\sigma_M^2\right)\left\|\widetilde{\Phi}_c(k)\right\|_F^2 + 8(1+\alpha_c)^2\varepsilon_{cM}^2, \tag{21}$$

and selecting

$$\alpha_c^2 < \min\{1, \Delta\sigma_{\min}^2/(2\Delta\sigma_M^2)\}, \tag{22}$$

the first two terms of (21) are less than zero. Further, $\Delta V_C(k)$ is less than zero provided the gain is selected according to (22) and the following inequalities hold

$$|e_c(k)| > \sqrt{\frac{8(1+\alpha_c)^2\varepsilon_{cM}^2}{(1-\alpha_c)^2}} \equiv b'_{ec} \quad \text{or} \quad \left\|\widetilde{\Phi}_c(k)\right\|_F > \sqrt{\frac{8(1+\alpha_c)^2\varepsilon_{cM}^2}{\Delta\sigma_{\min}^2 - 2\alpha_c^2\Delta\sigma_M^2}} \equiv b'_{\Phi c}.$$

Thus, using standard Lyapunov theory [1], it can be concluded that $\Delta V_C(k)$ is less than zero outside of a compact set rendering the cost error and cost OLA parameter estimation errors are *UUB*. ■

Next, the optimal control signal estimation scheme is presented for the optimal regulator.

B. Estimation of the Optimal Feedback Control Signal

The objective of this section is to find the control policy which minimizes the approximated cost function (6). To begin the development of the feedback control policy, we define the OLA approximation of (5) to be

$$\hat{u}(k) = \hat{u}(x(k)) = \hat{\Phi}_A^T(k)\vartheta(x(k)) \tag{23}$$

where $\hat{\Phi}_A(k)$ is the estimated value of the ideal parameter matrix Φ_A and $\vartheta(\bullet)$ denotes the basis function.

Next, the optimal control signal error is defined to be the difference between the feedback control applied to (1) and the control signal that minimizes the estimated cost function (6), which is denoted as

$$e_a(k) = \hat{\Phi}_A^T(k)\vartheta(x(k)) + \frac{1}{2}R^{-1}g^T(k)\frac{\partial\sigma(x(k+1))}{\partial x(k+1)}\hat{\Phi}_c(k) \tag{24}$$

and

$$e_a(k+1) = \hat{\Phi}_A^T(k+1)\vartheta(x(k+1)) + \frac{R^{-1}g^T(k+1)}{2}\frac{\partial\sigma(x(k+2))}{\partial x(k+2)}\hat{\Phi}_c(k+1). \tag{25}$$

Similar to (10), the control signal error dynamics resemble a nonlinear affine system controlled through $\hat{\Phi}_A^T(k+1)$. Thus, the control OLA parameter update is defined to be

$$\hat{\Phi}_A(k+1) = \hat{\Phi}_A(k) - \alpha_a\frac{\vartheta(k)e_a^T(k)}{\vartheta^T(k)\vartheta(k)+1} \tag{26}$$

where $0 < \alpha_a < 1$ is a small positive design parameter. Substituting the parameter update (24) into (25) yields

$$e_a(k+1) = -\alpha_a e_a(k)\frac{\vartheta^T(k)\vartheta(x(k+1))}{\vartheta^T(k)\vartheta(k)+1} + \hat{\Phi}_A^T(k)\vartheta(x(k+1)) + \frac{R^{-1}g^T(k+1)}{2}\frac{\partial\sigma(x(k+2))}{\partial x(k+2)}\hat{\Phi}_c(k+1). \tag{27}$$

Since the control policy $u(x(k))$ in (5) minimizes the cost function (4), from (3) we can write

$$0 = \varepsilon_A(k+1) + \frac{1}{2}R^{-1}g^T(k+1)\frac{\partial\varepsilon_c(k+2)}{\partial x(k+2)} + \Phi_A^T\vartheta(k+1)) + \frac{1}{2}R^{-1}g^T(k+1)\frac{\partial\sigma(x(k+2))}{\partial x(k+2)}\Phi_c. \tag{28}$$

Subtracting (28) from (27) along with defining the control OLA parameter estimation error as $\tilde{\Phi}_A(k) = \Phi_A - \hat{\Phi}_A(k)$ while recalling $\tilde{\Phi}_c(k+1) = \Phi_c - \hat{\Phi}_c(k+1)$ yields

$$e_a(k+1) = -\alpha_a e_a(k)\frac{\vartheta^T(k)\vartheta(x(k+1))}{\vartheta^T(k)\vartheta(k)+1} - \varepsilon_A(k+1) - \frac{1}{2}R^{-1}g^T(k+1)\frac{\partial\varepsilon_c(k+2)}{\partial x(k+2)} - \frac{R^{-1}g^T(k+1)}{2}\frac{\partial\sigma(x(k+2))}{\partial x(k+2)}\tilde{\Phi}_c(k+1) - \tilde{\Phi}_A^T(k)\vartheta(x(k+1)). \tag{29}$$

As a final step, we form the parameter estimation error dynamics as

$$\tilde{\Phi}_A(k+1) = \tilde{\Phi}_A(k) + \alpha_a\frac{\vartheta(k)e_a^T(k)}{\vartheta^T(k)\vartheta(k)+1}. \tag{30}$$

Remark 3: To calculate the control signal error (24) and implement the OLA parameter update (26), knowledge of the input transformation matrix $g(k)$ is required. However, the internal dynamics $f(k)$ is not required for the cost or control signal OLA schemes.

In the following theorem, it will be shown that by starting with an initial stabilizing control, the control OLA update (26) ensures all future control inputs are also admissible.

Theorem 2: (Admissibility). Let $u_0(k)$ be an initial admissible control input for the controllable system (1). Then, there exists a positive constant α_a such that the control OLA parameter update (26) renders that the future control sequence provides stabilizing policies for the nonlinear system (1).

Proof: Suppose Φ_{A0} is a constant OLA parameter matrix such that $u_0(k_0) = \Phi_{A0}^T \vartheta(x(k_0))$ is an initial stabilizing control policy starting at time k_0, and let $\hat{\Phi}_{c0}(k_0)$ be the corresponding cost function OLA estimate. Using the control OLA update law (26), the control input at the next time $k_0 + 1$ can be written as

$$u_1(k_0+1) = \hat{\Phi}_A^T(k_0+1)\vartheta(x(k_0+1)) = u_0(k_0+1)) - \alpha_a \Delta u(k_0) \tag{31}$$

with $\Delta u(k_0) = (u_0(k_0) - \hat{u}_0^*(k))\vartheta^T(k_0)\vartheta(x(k_0+1))/(\vartheta^T(k_0)\vartheta(k_0)+1)$, $u_0(k_0+1)) = \Phi_{A0}^T \vartheta(x(k_0+1))$, $\hat{u}_0^*(k) = -R^{-1}g^T(k_0)\nabla\hat{J}(k_0+1)/2$, and $\nabla\hat{J}(k_0+1) = (\partial\sigma(x(k_0+1))/\partial x(k_0+1))\hat{\Phi}_{c0}(k_0)$. Observe that $\hat{u}_0^*(k)$ is bounded as a result of *Theorem 1*. Next, we evaluate the cost function (2) at time k_0+1 using the stabilizing policy, $u_0(k_0+1)$, and then using improved policy (31) where $J_0(\bullet)$ will denote the cost function corresponding to $u_0(\bullet)$ and $J_1(\bullet)$ denotes the cost function corresponding to $u_1(\bullet)$, respectively. First, using $u_0(k_0)$, we observe

$$J_0(k_0+1) = Q(x(k_0+1)) + u_0(k_0+1)^T R u_0(k_0+1) + J_0(k_0+2). \tag{32}$$

Next, using the policy update (31), the cost function (2) is

$$J_1(k_0+1) = J_0(k_0+1) - J_0(k_0+2) - u_0(k_0+1)^T R u_0(k_0+1) + u_1(k_0+1)^T R u_1(k_0+1) + J_1(k_0+2). \tag{33}$$

Now, we manipulate (33) and use (31) to get $J_1(k_0+2) - J_1(k_0+1) = J_0(k_0+2) - J_0(k_0+1) +$

$u_0(k_0+1)^T R u_0(k_0+1) - (u_0(k_0+1) - \alpha_a \Delta u(k_0))^T R(u_0(k_0+1) - \alpha_a \Delta u(k_0))$, and substituting (32) yields

$$\begin{aligned} \Delta J_1(k_0+1) = & -Q(x(k_0+1)) - u_0(k_0+1)^T R u_0(k_0+1) \\ & -\alpha_a^2 \Delta u(k_0)^T R \Delta u(k_0) + 2u_0(k_0+1)^T R(\alpha_a \Delta u(k_0)) \end{aligned}. \tag{34}$$

where $\Delta J_1(k_0+1) = J_1(k_0+2) - J_1(k_0+1)$. Finally, taking the upper bound of (34) and observing $2u_0(k_0+1)^T R(\alpha_a \Delta u(k_0)) \leq u_0(k_0+1)^T R u_0(k_0+1) + \alpha_a^2 \Delta u(k_0)^T R \Delta u(k_0)$ yields

$$\Delta J_1(k_0+1) \leq -Q(x(k_0+1)), \tag{35}$$

and it can be concluded that $\Delta J_1(k_0+1) < 0$. Thus, $u_1(k_0+1)$ is a stabilizing control. Now, repeating the process (31)-(35) by starting with the stabilizing control $u_1(k_0+1)$ reveals that $u_2(k_0+2)$ is also stabilizing, and by continuing in this way, it can be shown that each subsequent control policy is stabilizing for the nonlinear system (1). ■

The results of *Theorem 2* conclude that by starting with a stabilizing control policy, each subsequent control policy is also stabilizing. Next the following corollary can be stated.

Corollary 1: (Boundedness of OLA Basis Functions). Let $\mu(k)$ be an admissible control for the controllable system (1). Then, there exists a positive constant $x_0 = x(0)$ such that $x_0 \geq \|x(k)\|$ for all $k > 0$. Moreover, there exists positive constants $\vartheta_M = \| \vartheta(x_0) \|$ and $\sigma_M = \| \sigma(x_0) \|$ such that $\vartheta_M \geq \| \vartheta(x(k)) \|$ and $\sigma_M \geq \| \sigma(x(k)) \|$ for all $k > 0$.

Proof: Proof of *Corollary 1* is straight forward using the positive define Lyapunov candidate $V_0(x) = \|h(x(k))\|$ whose first difference is given by $\Delta V_0(x) = \|h(x(k+1))\| - \|h(x(k))\|$ while considering the separate cases of $h(x(k)) = x(k)$,

$h(x(k)) = \vartheta(x(k))$, and $h(x(k)) = \sigma(x(k))$ as well as recalling each control input is admissible (*Theorem 2)* and applying *Definition 1.* ■

A block diagram illustrating the proposed optimal regulation scheme is now presented in Fig. 1. Next, the stability of the cost estimation error, control estimation error, and the OLA estimations errors are considered.

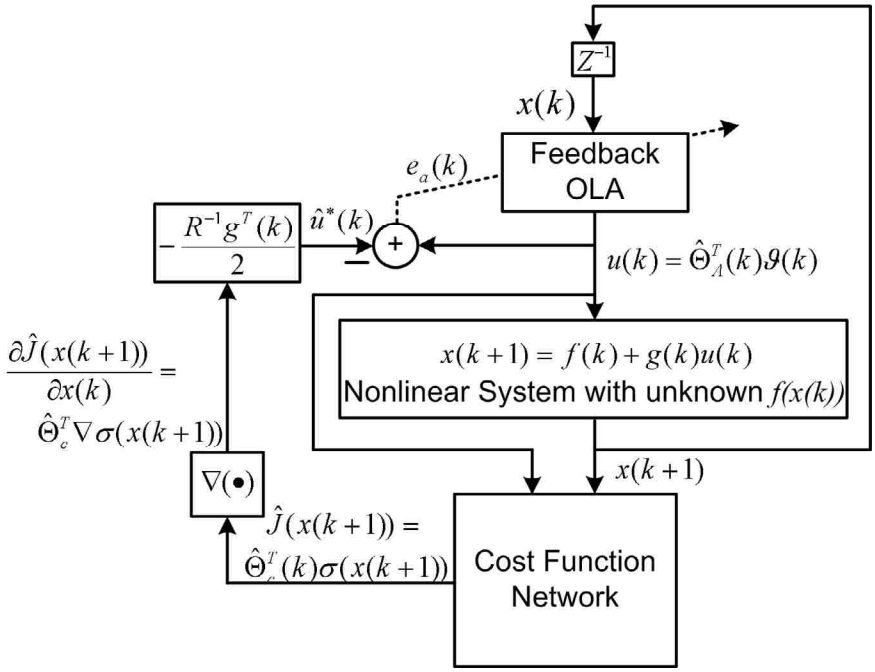


Fig. 1. Near optimal regulator block diagram.

C. *Convergence Proof*

In this section, it will be shown that the cost error (7), control error (24), as well as the OLA parameter estimation errors are *UUB.* Additionally, it will be shown that the estimated control input (23) approaches the optimal control signal with small bounded

error which is function of the OLA reconstruction errors ε_c and ε_A. If the OLA approximation errors are considered to be negligible [7]-[8], the estimated control policy approaches the optimal control asymptotically.

Theorem 3: (Convergence of the Optimal Control Signal). Let $u_0(k)$ be any initial admissible control policy for the nonlinear controllable system (1). Let the OLA parameter tuning for the cost estimator and the control input estimator be provided by (14) and (26), respectively. Then, there exists positive constants α_c and α_a such that the cost error (7) and control error (24) along with the cost and control signal OLA parameter estimates are all *UUB* for all $k \geq k_0 + T$ with bounds given by $\|e_c(k)\| \leq b_{ec}$, $\left\|\widetilde{\Phi}_c^T(k)\right\| \leq b_{\Phi c}$, $\|e_a(k)\| \leq b_{ea}$, $\left\|\Xi_A(k)\right\| \leq b_{\Xi A}$ for computable positive constants b_{ea}, b_{ec}, $b_{\Phi c}$, and $b_{\Xi A}$, respectively. Further, the system (1) is regulated in a near optimal manner. That is, $\left\|\hat{u} - u^*\right\| \leq \varepsilon_r$ for a small positive constant ε_r.

Proof: Consider the following positive definite Lyapunov candidate

$$V_{AC}(k) = V_C(k) + V_A(k) \tag{36}$$

where $V_C(k)$ is given by (19) and

$$V_A(k) = \alpha_a \alpha_c^2 \frac{\Delta\sigma_{\min}^2}{\vartheta_M^2 + 1} e_A^T(k) e_A(k) + \Delta\sigma_{\min}^2 \alpha_c^2 5tr\{\widetilde{\Phi}_A^T(k)\widetilde{\Phi}_A(k)\}. \tag{37}$$

The first difference of (36) is given by $\Delta V_{AC}(k) = \Delta V_C(k) + \Delta V_A(k)$, and $\Delta V_C(k)$ is given by (21). Next, taking the first difference of (37), substituting the control error dynamics (29), control OLA parameter estimation error dynamics (30), and applying the Cauchy-Schwartz (C-S) inequality yields

$$
\begin{aligned}
\Delta V_A(k) \le\ & 5\alpha_a^3\alpha_c^2\frac{\Delta\sigma_{\min}^2}{\vartheta_M^2+1}\left(\frac{\vartheta^T(k)\vartheta(x(k+1))}{\vartheta^T(k)\vartheta(k)+1}\right)^2 e_a^T(k)e_a(k)\\
&+5\alpha_a\alpha_c^2\left(\Delta\sigma_{\min}^2/(\vartheta_M^2+1)\right)\left(\varepsilon_A^T(k+1)\varepsilon_A(k+1)+\vartheta(x(k+1))^T\widetilde{\Phi}_A(k)\widetilde{\Phi}_A^T(k)\vartheta(x(k+1))\right)\\
&+\frac{5}{4}\alpha_a\alpha_c^2\frac{\Delta\sigma_{\min}^2}{\vartheta_M^2+1}\frac{\partial\varepsilon_c(k+2)}{\partial x(k+2)}^T\ g(k+1)(R^{-1})^2g^T(k+1)\frac{\partial\varepsilon_c(k+2)}{\partial x(k+2)}\\
&+\frac{5}{4}\alpha_a\alpha_c^2\frac{\Delta\sigma_{\min}^2}{\vartheta_M^2+1}\widetilde{\Phi}_c^T(k+1)\frac{\partial\sigma(x(k+2))}{\partial x(k+2)}^T\ g(k+1)(R^{-1})^2g^T(k+1)\frac{\partial\sigma(x(k+2))}{\partial x(k+2)}\widetilde{\Phi}_c(k+1)\\
&+5\alpha_a^2\alpha_c^2\Delta\sigma_{\min}^2\frac{\vartheta^T(k)\vartheta(k)e_a^T(k)e_a(k)}{\left(\vartheta^T(k)\vartheta(k)+1\right)^2}-\alpha_a\alpha_c^2\frac{\Delta\sigma_{\min}^2}{\vartheta_M^2+1}e_A^T(k)e_A(k)\\
&+\alpha_a\alpha_c^2\Delta\sigma_{\min}^2 10tr\{\frac{\widetilde{\Phi}_A(k)\vartheta(k)e_a^T(k)}{\vartheta^T(k)\vartheta(k)+1}\}.
\end{aligned}
\tag{38}
$$

Next, using a similar relationship as the one derived in (29) we rewrite the control error (24) as

$$
\begin{aligned}
e_a(k) = & -\widetilde{\Phi}_A^T(k)\vartheta(x(k))-\frac{1}{2}R^{-1}g^T(k)\frac{\partial\sigma(x(k+1))}{\partial x(k+1)}\widetilde{\Phi}_c(k)-\varepsilon_A(k)\\
& -\frac{1}{2}R^{-1}g^T(k)\frac{\partial\varepsilon_c(k+1)}{\partial x(k+1)}.
\end{aligned}
\tag{39}
$$

Then, substituting (39) into the last term of (38), using $\widetilde{\Phi}_c(k+1)$ from (18), and defining $\Xi_A(k)=\widetilde{\Phi}_A^T\vartheta(k)$ allows (38) to be rewritten as

$$
\begin{aligned}
\Delta V_A(k) \le\ & \frac{5\alpha_a^3\alpha_c^2\Delta\sigma_{\min}^2}{\vartheta_M^2+1}\left(\frac{\vartheta^T(k)\vartheta(x(k+1))}{\vartheta^T(k)\vartheta(k)+1}\right)^2\left\|e_a(k)\right\|^2\\
&-5\alpha_a\alpha_c^2\Delta\sigma_{\min}^2\left(\frac{1}{5(\vartheta_M^2+1)}-\frac{\alpha_a\vartheta^T(k)\vartheta(k)}{\left(\vartheta^T(k)\vartheta(k)+1\right)^2}\right)\left\|e_a(k)\right\|^2\\
&-\frac{5}{2}\frac{5\alpha_a\alpha_c^2\Delta\sigma_{\min}^2}{\vartheta^T(k)\vartheta(k)+1}\left(1-\frac{\alpha_a(2\varepsilon_{AM}+\varepsilon'_{cM}g_M\left\|R^{-1}\right\|_F)}{\vartheta^T(k)\vartheta(k)+1}\right)\left\|\Xi_A(k)\right\|^2\\
&+\frac{5}{2}\alpha_a\Delta\sigma_{\min}^2g_M^2\left\|R^{-1}\right\|_F^2\left(\frac{\alpha_c^2\sigma_M'^2}{\vartheta^T(k)\vartheta(k)+1}+\frac{\alpha_c^4\Delta\sigma_M^2}{\vartheta_M^2+1}\right)\left\|\widetilde{\Phi}_c^T(k)\right\|^2\\
&+\frac{5}{2}\alpha_c^2\Delta\sigma_{\min}^2(2\varepsilon_{AM}+\varepsilon'_{cM}g_M\left\|R^{-1}\right\|_F)+\frac{5\alpha_a\alpha_c^2}{4}\sigma_M'^2g_M^2\left\|R^{-1}\right\|_F^2\bar{\varepsilon}_{cM}^2\\
&+5\alpha_a\alpha_c^2\Delta\sigma_{\min}^2\varepsilon_{AM}^2+\frac{5\alpha_a\alpha_c^2}{4}\frac{\Delta\sigma_{\min}^2}{\vartheta_M^2+1}g_M^2\left\|R^{-1}\right\|_F^2\varepsilon_{cM}'^2
\end{aligned}
\tag{40}
$$

where $\bar{\varepsilon}_c^2 = (\Delta\varepsilon_c(k) + \alpha_c \Delta\varepsilon_c(k-1))^2$, $\bar{\varepsilon}_c^2 \le \bar{\varepsilon}_{cM}^2$ for a positive constant $\bar{\varepsilon}_{cM}^2$, and σ'_M is a positive constant such that $\sigma'_M \ge \| \nabla\sigma(x(k)) \|$. Now combining (21) and (40) reveals

$$\begin{aligned}\Delta V_{AC}(k) \le \eta(\varepsilon_A, \varepsilon_c) - (1-\alpha_c^2) e_c(k)^2 - \Pi \left\| \tilde{\Phi}_c^T(k) \right\|^2 - 5\alpha_a \alpha_c^2 \Delta\sigma_{\min}^2 \Lambda \left\| e_a(k) \right\|^2 \\ - \frac{5}{2} \frac{\alpha_a \alpha_c^2 \Delta\sigma_{\min}^2}{\vartheta^T(k)\vartheta(k)+1} \Omega \left\| \Xi_A(k) \right\|^2\end{aligned} \tag{41}$$

where

$$\begin{aligned}\eta_{AC}(\varepsilon_A, \varepsilon_c) = \frac{5}{2}\alpha_c \Delta\sigma_{\min}^2 (2\varepsilon_{AM} + \varepsilon'_{cM} g_M \left\| R^{-1} \right\|_F + 10\alpha_a \varepsilon_{AM}^2) + 8(1+\alpha_c)^2 \varepsilon_{cM}^2 \\ + \frac{5\alpha_a \alpha_c}{4} \left(\frac{\Delta\sigma_{\min}^2}{\vartheta_M^2 + 1} g_M^2 \left\| R^{-1} \right\|_F^2 \varepsilon'^2_{cM} + \sigma'^2_M g_M^2 \left\| R^{-1} \right\|_F^2 \bar{\varepsilon}_{cM}^2 \right)\end{aligned}, \tag{42}$$

$$\Pi = \Delta\sigma_{\min}^2 - \alpha_c^4 \frac{5\alpha_a}{2} \frac{\Delta\sigma_{\min}^2}{\vartheta_M^2 + 1} g_M^2 \left\| R^{-1} \right\|_F^2 \Delta\sigma_M^2 - \alpha_c^2 \left(\frac{5}{2} \frac{\alpha_a \sigma'^2_M g_M^2 \left\| R^{-1} \right\|_F^2}{\vartheta^T(k)\vartheta(k)+1} + 2\Delta\sigma_M^2 \right),$$

$$\Lambda = \frac{1}{5(\vartheta_M^2 + 1)} - \frac{\alpha_a \vartheta^T(k)\vartheta(k)}{\left(\vartheta^T(k)\vartheta(k)+1\right)^2} - \frac{\alpha_a^2}{\vartheta_M^2 + 1} \left(\frac{\vartheta^T(k)\vartheta(x(k+1))}{\vartheta^T(k)\vartheta(k)+1} \right)^2, \text{ and.}$$

$$\Omega = 1 - \frac{\alpha_a (2\varepsilon_{AM} + \varepsilon'_{cM} g_M \left\| R^{-1} \right\|_F}{\vartheta^T(k)\vartheta(k)+1}.$$

It is observed that the last four terms of (41) are less than zero provided the design parameters are selected according to

$$0 < \alpha_c^2 < \min(1, \mathrm{A}_{C\max}),$$

$$0 < \alpha_a < \min\left(\frac{-(\vartheta_M^2 + 1) + \sqrt{(\vartheta_M^2 + 1)^2 + 4/5}}{2}, \frac{1}{2\varepsilon_{AM} + \varepsilon'_{cM} g_M \left\| R^{-1} \right\|_F}, 1 \right) \tag{43}$$

where $\mathrm{A}_{C\max} = (B + \sqrt{B^2 + 4AC}) / (2A)$, $B = (5\alpha_a \sigma'^2_M g_M^2 \left\| R^{-1} \right\|_F^2) / 2(\vartheta^T(k)\vartheta(k)+1) + 2\Delta\sigma_M^2$

$A = (5\alpha_a \Delta\sigma_{\min}^2)/2(\vartheta_M^2 + 1)\, g_M^2 \left\|R^{-1}\right\|_F^2 \Delta\sigma_M^2$ and $C = \Delta\sigma_{\min}^2$. Finally, it is observed that (41) is less than zero provided the design parameters are selected according to (43) and the following inequalities hold

$$\begin{aligned} &e_c(k) \geq \sqrt{\frac{\eta(\varepsilon_A, \varepsilon_c)}{(1-\alpha_c^2)}} \equiv b_{ec} \quad \text{or} \quad \left\|\tilde{\Phi}_c^T(k)\right\| \geq \sqrt{\frac{\eta(\varepsilon_A, \varepsilon_c)}{\Pi}} \equiv b_{\Phi c ec} \quad \text{or} \\ &\left\|e_a(k)\right\| \geq \sqrt{\frac{\eta(\varepsilon_A, \varepsilon_c)}{5\alpha_a \alpha_c^2 \Delta\sigma_{\min}^2 \Lambda}} \equiv b_{ac} \quad \text{or} \quad \left\|\Xi_A(k)\right\| \geq \sqrt{\frac{\eta(\varepsilon_A, \varepsilon_c)}{\frac{5}{2}\frac{\alpha_a \alpha_c^2 \Delta\sigma_{\min}^2}{\vartheta^T(k)\vartheta(k)+1}\Omega}} \equiv b_{\Xi A}. \end{aligned} \tag{44}$$

Thus, using standard Lyapunov extensions [1], it can be concluded that $\Delta V_{AC}(k)$ is less than zero outside of a compact set revealing the cost and control errors as well as the cost and control OLA parameter estimates to be UUB. To show $\left\|\hat{u} - u^*\right\| \leq \varepsilon_r$, use (5) and (23) to observe $\hat{u}(k) - u^* = -\tilde{\Phi}_A^T(k)\vartheta(x(k)) - \varepsilon_A$. Then, taking the limit as $k \to \infty$ and taking the upper bound of $\hat{u}(k) - u^*$ shows

$$\left\|\hat{u}(k) - u^*\right\| \leq \left\|\Xi_A(k)\right\| + \varepsilon_{AM} \leq b_{\Xi A} + \varepsilon_{AM} \equiv \varepsilon_r \tag{45}$$

where $b_{\Xi A}$ is defined in (44). ■

Remark 4: If the OLA approximation errors ε_A and ε_c are considered to be negligible as in [7] and [8], it is clear that $\eta(\varepsilon_A, \varepsilon_c)$ in (44) and ε_r in (45) both become zero. For this scenario, it can be shown that the control and cost estimation errors and the control and cost OLA parameter estimates converge to zero asymptotically. That is, $\hat{u} \to u^*$.

Remark 5: The results of *Theorem 3* are drawn under the assumption of an initial admissible control, $u_0(k)$. This assumption is required to ensure that the initial cost function evaluated at $x(0)$ is finite. That is, $u_0(k)$ ensures $J(x(0)) < \infty$.

IV. Near Optimal Tracking Control of Nonlinear Discrete-time Systems with Partially Unknown Dynamics

In the previous section, the optimal regulator design was addressed for affine nonlinear discrete-time systems. The optimal tracking control problem can be considered as an extension of the regulation problem consistent with the other works in the literature [2]. The objective for the infinite-time optimal tracking problem is to find the optimal control sequence, $u^*(k)$, so as to make the nonlinear system in (1) to track a desired trajectory $x_d(k)$ in an optimal manner. To achieve our objective, the infinite-horizon cost function (2) must be modified accordingly to ensure it remains finite. To begin the development, define the dynamics of the desired trajectory as

$$\begin{aligned} x_d(k+1) &= f(x_d(k)) + g(x(k))u_d(k) \\ &= f_d(k) + g(k)u_d(k) \end{aligned} \tag{46}$$

where $f(x_d(k))$, or simply $f_d(k)$ for convenience, is the internal dynamics of the nonlinear system (1) rewritten in terms of the desired state $x_d(k)$, $g(k)$ is the input transformation matrix presented in (1), and $u_d(k)$ is the control input to the desired system. Next, define the tracking error as

$$e(k) = x(k) - x_d(k) \tag{47}$$

By using (1) and (46), the tracking error dynamics of (47) are given by

$$\begin{aligned} e(k+1) &= f(k) + g(k)u(k) - x_d(k+1) \\ &= f_e(k) + g(k)u_e(k) \end{aligned} \tag{48}$$

where $f_e(k) = f(k) - f_d(k)$ and

$$u_e(k) = u(k) - u_d(k). \tag{49}$$

Considering $u_e(k)$ as the control input for (48), it can be shown that $u_e(k)$ is an admissible control policy with $e(k) = 0$ being an equilibrium point of (48). To convert

the nonlinear tracking into a regulation problem, the infinite horizon cost function (2) is rewritten in terms of $e(k)$ and $u_e(k)$ as

$$J_e(k) = \sum_{i=0}^{\infty} r_e(k+i) = r_e(k) + \sum_{i=0}^{\infty} r_e(k+i+1) = r_e(k) + J_e(k+1) \tag{50}$$

where $r_e(k) = Q_e(e(k)) + u_e(k)^T R_e u_e(k)$ with $Q_e(e(k)) > 0$ and $R_e \in \Re^{mxm}$ is positive definite. Since $u_e(k)$ is admissible, (50) is finite. The optimal control input that minimizes (50) is found by solving $\partial J_e(k) / \partial u_e(k) = 0$ as

$$u_e^*(k) = -\frac{1}{2} R_e^{-1} g^T(k) \frac{\partial J_e^*(e(k+1))}{\partial e(k+1)} \tag{51}$$

or

$$u^*(k) = u_d(k) - \frac{1}{2} R_e^{-1} g^T(k) \frac{\partial J_e^*(e(k+1))}{\partial e(k+1)}. \tag{52}$$

The feedforward control input $u_d(k)$ obtained from (46) is given by

$$u_d(k) = g(k)^{-1}(x_d(k+1) - f_d(k)). \tag{53}$$

It is observed that the optimal tracking control input (52) consists of a predetermined feedforward term, $u_d(k)$, and an optimal feedback term that is a function of the gradient of the optimal cost function consistent with the linear control case [2]. Additionally, implementation of the feedforward term requires knowledge of the internal dynamics $f(k)$ and control coefficient matrix $g(k)$. In this effort, the infinite horizon optimal tracking control problem is solved without the knowledge of $f(k)$. Additionally, in this section it is assumed that there exists a matrix $g(k)^I \in \Re^{mxn}$ such that $g(k)g(k)^I = I \in \Re^{nxn}$ where I is the identity matrix. Note that when $n = m$, $g(k)^I = g(k)^{-1}$.

Before proceeding, the following technical Lemma is needed.

Lemma 1: Let $u_e(k)$ be an admissible control policy for the controllable error system (48). Then, the internal dynamics $f_e(k)$ is bounded above satisfying

$$\|f_e(k)\|^2 \leq \frac{\Gamma}{2}\|Q_e(e(k))\| + \frac{1}{2}(\Gamma\lambda_{\min}(R_e) - 2g_M^2)\|u_e(k)\|^2 \tag{54}$$

where $\Gamma > 2g_M^2 / \lambda_{\min}(R_e)$ is a known positive constant with $\lambda_{\min}(R_e)$ being the minimum eigenvalue of R_e, and $Q_e(e(k))$ is defined as in (50).

Proof: Consider the positive definite Lyapunov candidate

$$V(k) = e^T(k)e(k) + \Gamma J_e(k) \tag{55}$$

where $J_e(k)$ is the cost function (50). The first difference of (55) is given by $\Delta V(k) = e^T(k+1)e(k+1) \quad - e^T(k)e(k) + \Gamma\Delta J_e(k)$. Using (48) and (50) as well as applying the CS inequality, an upper bound for $\Delta V(k)$ is

$$\Delta V(k) \leq 2\|f_e(k)\|^2 - \Gamma\|Q_e(e(k))\| - (\Gamma\lambda_{\min}(R_e) - 2g_M^2)\|u_e(k)\|^2 - \|e(k)\|^2 \tag{56}$$

Since $u_e(k)$ is admissible, the tracking error system (48) is asymptotically stable and $\Delta V(k) < 0$. Then, applying this to (56), the bound in (54) results with $\Delta V(k) \leq -\|e(k)\|^2$ [22]. ■

Moving on, the proposed DDP design for tracking entails three portions: a feedback system that is designed to produce a nearly optimal portion of the control signal, a HJB function estimator which evaluates the performance of the error system, and a feedforward design to produce the feedforward control input (52). Using the approximation property of OLA's [1], the cost function (50), feedback control policy

(51), and feedforward control policy (53) have OLA representations on a compact set expressed as

$$J_e(e(k)) = \Theta_c^T \sigma_e(e(k)) + \varepsilon_{ec} \tag{57}$$

$$u_e(e(k)) = \Theta_A^T \vartheta_e(e(k)) + \varepsilon_{eA} \tag{58}$$

and

$$u_d(k) = g(k)^{-1}(x_d(k+1) - \Theta_d^T \phi(x_d(k)) + \varepsilon_d) \tag{59}$$

respectively, where Θ_c, Θ_A, and Θ_d are the constant target OLA parameters, ε_c, ε_A, and ε_d are the bounded approximation errors, and $\sigma_e(\bullet) \in \Re^{\ell_c}$, $\vartheta_e(\bullet) \in \Re^{\ell_a}$ and $\phi(\bullet) \in \Re^{\ell_d}$ are the bounded vector activation functions for the cost, feedback, and feedforward control networks, respectively [1].

Next, the following assumptions which are common in OLA literature [1],[21] are stated regarding the boundedness of the OLA parameters for the tracking problem.

Assumption 4: The upper bounds for the ideal OLA parameters are taken as $\|\Theta_c\| \le \Theta_{cM}$, $\|\Theta_A\|_F \le \Theta_{AM}$, and $\|\Theta_d\|_F \le \Theta_{dM}$ where Θ_{cM}, Θ_{AM}, and Θ_{dM} are positive constants [1].

Assumption 5: The approximation errors are considered to be bounded above such that $\|\varepsilon_{ec}\| \le \varepsilon_{ecM}$, $\|\varepsilon_{eA}\| \le \varepsilon_{eAM}$, and $\|\varepsilon_d\| \le \varepsilon_{dM}$ where $\varepsilon_{ecM}, \varepsilon_{eAM}$ and ε_{dM} are positive constants [1].

Assumption 6: Upper bounds for the basis functions are taken as $\|\sigma_e(\bullet)\| \le \sigma_{eM}$, $\|\vartheta_e(\bullet)\| \le \vartheta_{eM}$, and $\|\phi(\bullet)\| \le \phi_M$ for known constants σ_{eM}, ϑ_{eM}, and ϕ_M, respectively [1].

Assumption 7: The gradient of approximation errors and activation functions are considered to be bounded according to $\|\partial\varepsilon_{ec}/(\partial e(k+1))\|_F \le \varepsilon'_{ecM}$ and $\|\partial\sigma_e(\bullet)/(\partial(\bullet)\|_F \le \sigma_{deM}$, respectively, where ε'_{ecM} and σ_{deM} are also known positive constants consistent with the work of [21].

To begin, the design of the HJB function approximator for the tracking problem will be considered first.

A. Cost Function Approximator Design for Tracking

The objective of the optimal tracking control law is to stabilize the system (48) while minimizing the cost function (50). This cost function will be approximated by an OLA as

$$\hat{J}_e(k) = \hat{J}_e(e(k)) = \hat{\Theta}_c^T(k)\sigma_e(e(k)) = \hat{\Theta}_c^T(k)\sigma_e(k) \tag{60}$$

where $\hat{\Theta}_c$ is the approximation for the ideal parameters Θ_c and $\sigma_e(k) = \{\sigma_{e\ell}(e(k))\}_1^{L:}$ is set of activation functions selected to be linearly independent. Similarly to the regulation case, the basis vector $\sigma_e(\bullet)$ is selected to satisfy $\|\sigma_e(e_0)\| = 0$ for $\|e_0\| = 0$ with $e_0 \in \Re^n$ to facilitate $J_e(0) = 0$.

For convenience, we define the cost error to be

$$e_{ec}(k) = r_e(k) + \hat{\Theta}_c^T(k)\sigma_e(k+1) - \hat{\Theta}_c^T(k)\sigma_e(k) \tag{61}$$

Next, we define an auxiliary cost error vector as

$$E_{ec}(k) = Y_e(k) + \hat{\Theta}_c^T(k)X_e(k) \in \Re^{1x(1+j)} \tag{62}$$

where $Y_e(k)=[r_e(k)\ \ r_e(k-1)\ldots r_e(k-j)]\in\Re^{1xj+1}$ and

$X_e(k)=[\Delta\sigma_e(k+1)\ \ \Delta\sigma_e(k)\ldots\Delta\sigma_e(k+1-j)]\in\Re^{\ell_c xj+1}$ with $\Delta\sigma_e(k+1)=\sigma_e(k+1)-\sigma_e(k)$,

$0 < j < k-1 \in \mathbf{N}$ and $\mathbf{N}$ being the set of natural real numbers. It is useful to observe that (62) can be rewritten as $E_{ec}(k) = [e_{ec}(k \mid k) \;\; e_{ec}(k \mid k-1) \cdots e_{ec}(k \mid k-j)]$ where the notation $e_{ec}(k \mid k-1)$ means the cost error $e_{ec}(k-1)$ re-evaluated at time k using the actual cost parameter matrix $\hat{\Theta}_c^T(k)$.

Remark 6: Using a similar approach presented in *Lemma 1,* it can be shown that $\Delta\sigma_e(k+1)$ is linearly independent if $\sigma_e(k)$ is linearly independent. As a result, the matrix $X_e(k)$ is linearly independent provided $e(k) \neq 0$. Observing the definition of the cost function (50), and OLA approximation (60), it is evident that both become zero when $e(k) = 0$. Thus, once the tracking error has converged to zero, the parameter matrix associated with the estimator that approximates the cost function can no longer be updated. This can be viewed as a persistency of excitation (PE) requirement [1] for the inputs to the OLA that approximates the cost function. That is, the tracking error states must be persistently exiting long enough for the cost function and optimal control policy to be obtained. Further, the persistency of excitation condition ensures the existence of a nonzero lower bound $X_{eMin} \leq \|X_e(k)\|$.

Moving on, define the OLA parameter update law as

$$\hat{\Theta}_c(k+1) = \hat{\Theta}_c - \alpha_{ec} X_e E_c^T \tag{63}$$

where $0 < \alpha_{ec} < 1$ is a small positive design parameter.

Remark 7: In the previous section, the auxiliary cost error (10) was considered to be a dynamical system whereas the auxiliary cost error (62) is not given dynamics in this section. Additionally, it is observed that the cost OLA parameter update law (14) used in the regulation problem is quite different than the cost OLA update utilized for tracking in

(63). The reason for these differences can be linked to the added uncertainty introduced by approximating the feedforward term in (52) for tracking. The added complexities introduced in the stability analysis due to the added uncertainty does not yield a compact set for which a suitable Lyapunov candidate is less than zero when considering the auxiliary cost error dynamics (62) in the form of (10). However, the theoretical results of this section will prove that the cost error (62) remains bounded for all time.

To obtain the OLA parameter estimation error dynamics, rewrite (50) using the target OLA representation (57) as

$$\Theta_c^T \sigma(e(k)) + \varepsilon_{ec}(k) = r_e(k) + \Theta_c^T \sigma_e(e(k+1)) + \varepsilon_{ec}(k+1) \tag{64}$$

Rearranging (64) renders

$$r_e(k) = -\Theta_c^T \Delta\sigma_e(e(k+1)) - \Delta\varepsilon_{ec}(k) \tag{65}$$

where $\Delta\varepsilon_{ec}(k) = \varepsilon_{ec}(k+1) - \varepsilon_{ec}(k)$. Substituting (65) into (61) results in

$$e_c(k) = -\tilde{\Theta}_c^T(k)\Delta\sigma(e(k+1)) - \Delta\varepsilon_c(k) \tag{66}$$

where $\tilde{\Theta}_c(k) = \Theta_c - \hat{\Theta}_c(k)$ is the cost parameter estimation error. Similarly, (62) can be rewritten as

$$E_c(k) = -\tilde{\Theta}_c^T(k) X_e - \Psi_c(k) \tag{67}$$

where $\Psi_c(k) = [\Delta\varepsilon_{ec}(k)\ \Delta\varepsilon_{ec}(k-1)\ldots\Delta\varepsilon_{ec}(k-j)]$ and $\|\Psi_c(k)\|^2 \le \Psi_{cM}^2$. Now, observing $\tilde{\Theta}_c(k+1) = \Theta_c - \hat{\Theta}_c(k+1)$ and using (63) and (67) results in the OLA parameter estimation error dynamics to be expressed as

$$\tilde{\Theta}_c(k+1) = \left(I - \alpha_{ec} X_e X_e^T(k)\right)\tilde{\Theta}_c(k) - \alpha_{ec} X_e \Psi_c^T(k) \tag{68}$$

where I is the identity matrix of appropriate dimension.

The following theorem demonstrates the stability of the OLA cost function approximator given a fixed admissible control policy, $u_{e0}(k)$.

Theorem 4 (Cost Function OLA Stability): Let $u_{e0}(k)$ be any initial admissible control input for the system (48), and let the parameter tuning for the cost function OLA be provided by (63). Then, there exists a positive constant α_{ec} such that the OLA parameter estimation error for the cost function approximator is *UUB* for all time $k \geq k_0 + T$ with bounds given by $\left\|\widetilde{\Theta}_c^T(k)\right\| \leq \sqrt{(2\alpha_{ec}X_{eM}^2\Psi_{cM}^2 + \Phi_{cM}^2)/\left(X_{eMin}^2 - 2\alpha_{ec}X_{eM}^4\right)}$ where $X_{eMin} \leq \|X_e(k)\|_F \leq X_{eM}$ with X_{eMin} and X_{eM} are known positive constants given by *Remark 6* and *Assumption 6*, respectively.

Proof: Consider the positive definite Lyapunov candidate

$$V_{ec}(k) = \frac{1}{\alpha_{ec}} tr\{\widetilde{\Theta}_c^T(k)\widetilde{\Theta}_c(k)\} \tag{69}$$

whose first difference is given by

$$\Delta V_{ec}(k) = \frac{1}{\alpha_{ec}} (tr\{\widetilde{\Theta}_c^T(k+1)\widetilde{\Theta}_c(k+1)\} - tr\{\widetilde{\Theta}_c^T(k)\widetilde{\Theta}_c(k)\}). \tag{70}$$

Substituting the closed-loop estimation error dynamics (68) into (70) and applying some manipulations reveals

$$\begin{aligned}\Delta V_{ec}(k) = \frac{1}{\alpha_{ec}} tr\{&(\alpha_{ec}X_e X_e^T\widetilde{\Theta}_c(k) + \alpha_{ec}X_e\Psi_c^T(k))^T(\alpha_{ec}X_e X_e^T\widetilde{\Theta}_c(k) + \alpha_{ec}X_e\Psi_c^T(k))\} \\ &- 2tr\{\widetilde{\Theta}_c^T(k)(X_e X_e^T\widetilde{\Theta}_c(k) + X_e\Psi_c^T(k))\}\end{aligned} \tag{71}$$

Next, applying the CS inequality renders

$$\Delta V_{ec}(k) \leq -X_{eMin}^2\left(1 - 2\alpha_{ec}\frac{X_{eM}^4}{X_{eMin}^2}\right)\left\|\widetilde{\Theta}_c(k)\right\|^2 + 2\alpha_{ec}X_{eM}^2\Psi_{cM}^2 + \Psi_{cM}^2. \tag{72}$$

Examining (72), it can be concluded that $\Delta V_{ec}(k) < 0$ provided $\alpha_{ec} < X_{eMin}^2/(2X_{eM}^4)$ and

$$\left\|\tilde{\Theta}_c^T(k)\right\| > \sqrt{(2\alpha_{ec}X_{eM}^2\Psi_{cM}^2 + \Psi_{cM}^2)/\left(X_{eMin}^2 - 2\alpha_{ec}X_{eM}^4\right)}.$$

It can be concluded using standard Lyapunov extension [1], that $\Delta V_{ec}(k)$ is less than zero outside of a compact set so that the cost estimation errors are UUB. ■

Remark 8: Examining the cost error written in (66), it is clear that the boundedness of the cost parameter estimation error ensures the boundedness of the cost error. Additionally, the results of *Theorem 4*, are drawn under the assumption that that $u_{e0}(k)$ is a fixed control policy. This assumption will be relaxed in the following section.

B. Feedback Control Signal Design for Tracking

The objective of this section is to find the feedback control policy that minimizes the approximated cost function (60). To begin the development of the feedback control policy, define the OLA approximation of (58) as

$$\hat{u}_e(k) = \hat{u}_e(e(k)) = \hat{\Theta}_A^T(k)\vartheta_e(e(k)) \tag{73}$$

where $\hat{\Theta}_A(k)$ is the actual OLA parameters with Θ_A being the target, and the basis function $\vartheta_e(\bullet)$ is the basis function. Further, the PE condition described in *Remark 6* also guarantees the existence of a nonzero lower bound $\vartheta_{eMin} \le \|\vartheta_e(k)\|$.

Next, the feedback control signal error is defined to be the difference between the feedback control applied to the error system (48) and the control signal which minimizes the estimated cost function (60), which is denoted as

$$e_{ea}(k) = \hat{\Theta}_A^T(k)\vartheta(e(k)) + \frac{1}{2}R_e^{-1}g^T(k)\frac{\partial\sigma_e(e(k+1))}{\partial e(k+1)}\hat{\Theta}_c(k). \tag{74}$$

The feedback control parameter update is now defined to be

$$\hat{\Theta}_A(k+1) = \hat{\Theta}_A(k) - \alpha_{ea}\vartheta_e(k)e_{ea}^T(k) \tag{75}$$

where $0 < \alpha_{ea} < 1$ is a small positive design parameter.

The feedback control policy $u_e(e(k))$ in (58) minimizes the cost function (57). Therefore, it can be deduced that

$$0 = \varepsilon_{eA}(k) + \frac{1}{2}R_e^{-1}g^T(k)\frac{\partial \varepsilon_{ec}(k+1)}{\partial e(k+1)} + \Theta_A^T\vartheta_e(k)) + \frac{1}{2}R_e^{-1}g^T(k)\frac{\partial \sigma_e(e(k+1))}{\partial e(k+1)}\Theta_c . \tag{76}$$

Subtracting (76) from (74) reveals

$$e_a(k) = -\tilde{\Theta}_A^T(k)\vartheta_e(e(k)) - \frac{1}{2}R_e^{-1}g^T(k)\frac{\partial \sigma_e(e(k+1))}{\partial e(k+1)}\tilde{\Theta}_c(k) - \varepsilon_{eAC} \tag{77}$$

where $\tilde{\Theta}_A(k) = \Theta_A - \hat{\Theta}_A(k)$ is the feedback control parameter estimation error, $\tilde{\Theta}_c(k)$ is the cost function parameter estimation error previously defined with $\varepsilon_{eAC} = \varepsilon_{eA}(k) + R_e^{-1}g^T(k)(\partial \varepsilon_{ec}(k+1)/\partial e(k+1))/2$ and $\|\varepsilon_{eAC}\| \le \varepsilon_{eACM}$.

As a final step, we observe that $\tilde{\Theta}_a(k+1) = \Theta_a - \hat{\Theta}_a(k+1)$, and using (75) and (77), the feedback control parameter estimation error dynamics are found to be

$$\begin{aligned}\tilde{\Theta}_A(k+1) = &\left(I - \alpha_{ea}\vartheta_e(k)\vartheta_e^T(k)\right)\tilde{\Theta}_A(k) - \alpha_{ea}\vartheta_e(k)\varepsilon_{eAC}^T \\ &- \frac{\alpha_a}{2}\vartheta_e(k)\left(R_e^{-1}g^T(k)\frac{\partial \sigma_e(e(k+1))}{\partial e(k+1)}\tilde{\Theta}_c(k)\right)^T\end{aligned} \tag{78}$$

where *I* is the identity matrix of appropriate dimension.

Next, the stability of the cost and feedback OLA system is presented.

Theorem 5: (Cost and Feedback OLA Stability). Let $u_{e0}(k)$ be any initial admissible control for the nonlinear system (48), and let the parameter tuning for the cost and feedback control OLA systems be provided by (63) and (75), respectively. Then,

there exists positive constants α_{ec} and α_{ea} such that the OLA parameter estimation errors $\widetilde{\Theta}_A(k)$ and $\widetilde{\Theta}_c(k)$ are *UUB* for all $k \geq k_0 + T$ with bounds given by $\left\|\widetilde{\Theta}_A(k)\right\| \leq b_A$ and $\left\|\widetilde{\Theta}_c^T(k)\right\| \leq b_C$ for computable positive constants b_A and b_C, respectively. In addition, $\left\|\hat{u}_e - u_e^*\right\| \leq \varepsilon_{ue}$ for a small positive constant ε_{ue}.

Proof: Consider the positive definite Lyapunov candidate

$$V_{eAC}(k) = V_{eA}(k) + V_{ec}(k) \tag{79}$$

where

$$V_{eA}(k) = \frac{\alpha_{ec}}{\alpha_{ea}} tr\{\widetilde{\Theta}_A^T(k)\widetilde{\Theta}_A(k)\} \tag{80}$$

and $V_{ec}(k)$ is defined in (69). Taking the first difference of (79) gives $\Delta V_{eAC}(k) = \Delta V_{eA}(k) + \Delta V_{ec}(k)$. Now, considering $\Delta V_{eA}(k)$ and using (78) yields

$$\begin{aligned}\Delta V_{eA}(k) = &-2\alpha_{ec} tr\left\{\widetilde{\Theta}_A^T(k)\left(\vartheta_e(k)\vartheta_e^T(k)\widetilde{\Theta}_A(k) + \vartheta_e(k)\varepsilon_{AC}^T + \frac{\vartheta_e(k)\widetilde{\Theta}_c^T(k)}{2}\left(\frac{\partial\sigma_e(e(k+1))}{\partial e(k+1)}\right)^T g(k)R_e^{-T}\right)\right\} \\ &+ \frac{\alpha_{ec}}{\alpha_{ea}} tr\left\{\begin{array}{l}\left(\alpha_{ea}\vartheta_e(k)\vartheta_e^T(k)\widetilde{\Theta}_A(k) + \alpha_{ea}\vartheta_e(k)\varepsilon_{eAC}^T + \frac{\alpha_{ea}}{2}\vartheta_e(k)\widetilde{\Theta}_c^T(k)\left(\frac{\partial\sigma_e(e(k+1))}{\partial e(k+1)}\right)^T g(k)R_e^{-T}\right)^T \times \\ \left(\alpha_{ea}\vartheta_e(k)\vartheta_e^T(k)\widetilde{\Theta}_A(k) + \alpha_{ea}\vartheta_e(k)\varepsilon_{eAC}^T + \frac{\alpha_{ea}}{2}\vartheta_e(k)\widetilde{\Theta}_c^T(k)\left(\frac{\partial\sigma_e(e(k+1))}{\partial e(k+1)}\right)^T g(k)R_e^{-T}\right)\end{array}\right\}.\end{aligned} \tag{81}$$

Next, applying the CS inequality to (81), we arrive at

$$\begin{aligned}\Delta V_{eA}(k) \leq &-\alpha_{ec}\vartheta_{eMin}^2\left(\frac{1}{2} - \alpha_{ea}3\frac{\vartheta_{eM}^4}{\vartheta_{eMin}^2}\right)\left\|\widetilde{\Theta}_A(k)\right\|^2 + \alpha_{ec}\alpha_{ea}3\vartheta_{eM}^2\varepsilon_{eACM}^2 \\ &+ \alpha_{ec}\left(\frac{3\alpha_{ea}}{2}\vartheta_{eM}^2 + 1\right)\frac{g_M^2\left\|R_e^{-1}\right\|_F^2}{2}\left\|\widetilde{\Theta}_c(k)\right\|^2\sigma_{deM}^2 + \alpha_{ec}\varepsilon_{eACM}^2.\end{aligned} \tag{82}$$

Now, combining (72) and (82) shows

$$\begin{aligned}\Delta V_{eAC}(k) \le & -\alpha_{ec}\vartheta_{eMin}^2\left(\frac{1}{2}-\alpha_{ea}3\frac{\vartheta_{eM}^4}{\vartheta_{eMin}^2}\right)\left\|\tilde{\Theta}_A(k)\right\|^2+\eta_e(\varepsilon_{ec},\varepsilon_{eA}) \\ & -X_{eMin}^2\left(1-\frac{\alpha_{ec}}{X_{eMin}^2}\left(2X_{eM}^4+\left(\frac{3\alpha_{ea}}{2}\vartheta_{eM}^2+1\right)\frac{g_M^2\left\|R_e^{-1}\right\|_F^2}{2}\sigma_{deM}^2\right)\right)\left\|\tilde{\Theta}_c(k)\right\|^2\end{aligned} \tag{83}$$

where $\eta_e(\varepsilon_{ec},\varepsilon_{eA})=2\alpha_{ec}X_{eM}^2\Psi_{cM}^2+\Psi_{cM}^2+\alpha_{ec}\varepsilon_{eACM}^2+\alpha_{ec}\alpha_{ea}3\vartheta_{eM}^2\varepsilon_{eACM}^2$. The first difference of (79) is less than zero provided the tuning parameters are selected according to $\alpha_{ea}<\vartheta_{eMin}^2/(6\vartheta_{eM}^4)$, $\alpha_{ec}<X_{eMin}^2/(2X_{eM}^4+(3\alpha_{ea}\vartheta_{eM}^2/2+1)g_M^2\left\|R_e^{-1}\right\|_F^2\sigma_{deM}^2/2)$ and the following inequalities hold:

$$\left\|\tilde{\Theta}_A(k)\right\|>\sqrt{\frac{\eta_e(\varepsilon_{ec},\varepsilon_{eA})}{\alpha_{ec}\vartheta_{eMin}^2\left(\frac{1}{2}-\alpha_{ea}3\frac{\vartheta_{eM}^4}{\vartheta_{eMin}^2}\right)}}\equiv b_A \text{ or } \left\|\tilde{\Theta}_c^T(k)\right\|>\sqrt{\frac{\eta_e(\varepsilon_{ec},\varepsilon_{eA})}{X_{eMin}^2\left(1-\frac{\alpha_{ec}}{X_{eMin}^2}\left(2X_{eM}^4+\left(\frac{3\alpha_{ea}}{2}\vartheta_{eM}^2+1\right)\frac{g_M^2\left\|R_e^{-1}\right\|_F^2}{2}\sigma_{deM}^2\right)\right)}}\equiv b_C.$$

As a result, it can be concluded using standard Lyapunov extension [1], that $\Delta V_{eAC}(k)$ is less than zero outside a compact set so that the feedback control and cost parameter estimation errors are UUB. To show $\left\|\hat{u}_e-u_e^*\right\|\le\varepsilon_{ue}$, observe

$$\hat{u}_e-u_e^*=e_a(k)+\frac{1}{2}R_e^{-1}g^T(k)\frac{\partial\sigma_e(k+1)}{\partial e(k+1)}\tilde{\Theta}_c(k)+\frac{1}{2}R_e^{-1}g^T(k)\frac{\partial\varepsilon_{ec}(k+1)}{\partial e(k+1)}$$

where (57) and (74) were used. Then, using (77), taking the limit as $k\to\infty$ and the upper bound of $\hat{u}_e-u_e^*$ yields

$$\left\|\hat{u}_e-u_e^*\right\|\le\left\|\tilde{\Theta}_A(k)\right\|_F\vartheta_{eM}+\varepsilon_{eAM} \tag{84}$$

or $\left\|\hat{u}_e-u_e^*\right\|\le b_A\vartheta_{eM}+\varepsilon_{eAM}\equiv\varepsilon_{ue}$. ■

C. *Nearly Optimal Control Input*

Recall that the optimal tracking control input (52) to the system (1) is comprised of an optimal feedback term and a predetermined feedforward term. In the previous section, the nearly optimal feedback control law was developed. To begin the development of the feedforward control input, define the OLA representation of (59) as

$$\hat{u}_d(k) = g(k)^{-1}(x_d(k+1) - \hat{\Theta}_d^T \phi_d) \tag{85}$$

where $\hat{\Theta}_d(k)$ is the approximation of the ideal OLA parameter matrix Θ_d and ϕ_d is a linearly independent basis vector. As in the previous cases, it is assumed that there exists a nonzero lower bound such that $\phi_{dMin} \le \|\phi_d\|$. It is observed that this condition is easily met with proper selection of the basis function $\phi_d(x_d)$ since the desired trajectory x_d is bounded. Now, using (73) and (85), the estimate of the control input (52) is written as

$$u(k) = \hat{u}_d(k) + \hat{u}_e(k), \tag{86}$$

and applying (86) to the nonlinear system (1) reveals $x(k+1) = f(k) + g(k)(\hat{u}_d(k) + \hat{u}_e(k))$ or

$$e(k+1) = f(k) - \hat{\Theta}_d^T(k)\phi_d + g(k)\hat{u}_e(k)). \tag{87}$$

Then, adding and subtracting $f_d(k)$ and $g(k)u_e(k)$ to (87) and recalling the OLA representations of $f_d(k)$ and $u_e(k)$ in (59) and (58), respectively, (87) is rewritten as

$$e(k+1) = f_e(k) + g(k)u_e(k) + \tilde{\Theta}_d^T(k)\phi_d - g(k)\tilde{\Theta}_A^T(k)\vartheta(k) + \varepsilon_{Ad} \tag{88}$$

where $\tilde{\Theta}_d(k) = \Theta_d - \hat{\Theta}_d(k)$, $\varepsilon_{Ad} = \varepsilon_d - g(k)\varepsilon_A$ and $\|\varepsilon_{Ad}\| \le \varepsilon_{AdM}$. Select the tuning law for the feedforward estimator as

$$\hat{\Theta}_d(k+1) = \hat{\Theta}_d(k) + \alpha_d \phi_d (e(k+1) - g(k)\hat{u}_e(k))^T. \tag{89}$$

As a final step, we calculate the error dynamics of $\widetilde{\Theta}_d(k)$ as

$$\widetilde{\Theta}_d(k+1) = (I - \alpha_d \phi_d \phi_d^T)\widetilde{\Theta}_d(k) - \alpha_d \phi(x_d(k))(f_e(k) + \varepsilon_d)^T. \tag{90}$$

In the following section, the stability of the proposed scheme is investigated, but first, a block diagram of the proposed near optimal tracking controller design is presented in Fig. 2 where the cost, feedforward and feedback networks have been labeled accordingly.

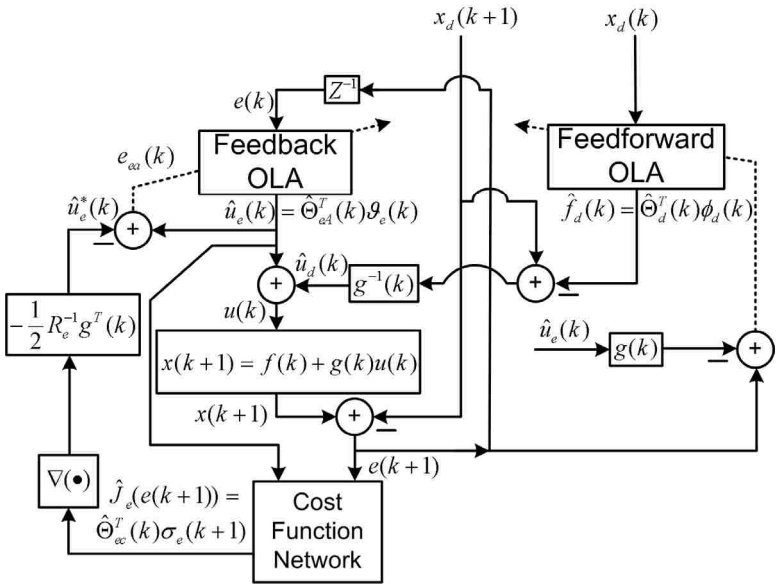


Fig. 2. Near optimal tracking control block diagram.

D. *Convergence Proof*

In this section, the convergence of tracking error (47) and the cost function, feedback control signal, and feedforward control signal OLA parameter estimation errors

is demonstrated in the following theorem while explicitly considering the OLA reconstruction errors ε_{ec}, ε_{eA}, and ε_d.

Theorem 6: (System Stability). Let $u_{e0}(k)$ be any initial admissible control for the nonlinear system (48). Let the parameter tuning for the cost OLA and feedforward OLA be provided by (63) and (75), respectively, and let the tuning law for the feedforward estimator be given by (89). Then, there exists positive constants α_{ec}, α_{ea} and Γ such that the tracking error (47) and the OLA parameter estimation errors of the cost function, feedback and feedforward terms are all *UUB* for all $k \geq k_0 + T$ with bounds given by $\|e(k)\| \leq b_e$, $\|\widetilde{\Theta}_c(k)\| \leq b'_C$, $\|\widetilde{\Theta}_d(k)\| \leq b_d$, and $\|\widetilde{\Theta}_A(k)\| \leq b'_A$ for computable positive constants b_e, b_d, b'_C. and b'_A, respectively. Further, the tracking error system (48) is regulated in a near optimal manner. That is, $\|u - u^*\| \leq \varepsilon_u$ for a small positive constant ε_u.

Proof: Consider the positive definite Lyapunov candidate

$$V_s(k) = V_{eAC}(k) + V_d(k) + V_e(k) \tag{91}$$

where $V_{eAC}(k)$ is defined in (79),

$$V_e(k) = \alpha_{ec}\alpha_d\alpha_{ea}e^T(k)e(k) + 5(2 + 4\alpha_d\phi_{dM}^2)\Gamma J_e(k) \tag{92}$$

and

$$V_d(k) = \frac{\alpha_{ea}}{\alpha_d} tr\{\widetilde{\Theta}_d^T(k)\widetilde{\Theta}_d(k)\}. \tag{93}$$

The first difference of (91) is given by $\Delta V_s(k) = \Delta V_{eAC}(k) + \Delta V_d(k) + \Delta V_e(k)$. Considering first $\Delta V_d(k) = (\alpha_{ea}/\alpha_d)(tr\{\Delta\widetilde{\Theta}_d(k)^T\Delta\widetilde{\Theta}_d(k\} - 2tr\{\widetilde{\Theta}_d^T(k)\Delta\widetilde{\Theta}_d(k)\})$, substituting (90) as well as applying the CS inequality yields

$$\begin{aligned}\Delta V_d(k) \le & -\alpha_{ea}\phi_{dMin}^2\left(1-2\alpha_d\frac{\phi_{dM}^4}{\phi_{dMin}^2}\right)\left\|\widetilde{\Theta}_d(k)\right\|^2+\alpha_{ea}\|f_e(k)\|^2(2+4\alpha_d\phi_{dM}^2)\\ & +\alpha_{ea}2\|\varepsilon_d\|^2(1+2\alpha_d\phi_{dM}^2)\end{aligned} \tag{94}$$

Next, considering $\Delta V_e(k)=\alpha_d\alpha_{ec}\alpha_{ea}e^T(k+1)e(k+1)-\alpha_d\alpha_{ec}\alpha_{ea}e^T(k)e(k)+5(2+4\alpha_d\phi_{dM}^2)\Gamma\Delta J_e(k)$ and using the error dynamics (88) and cost function (50) renders

$$\begin{aligned}\Delta V_e(k) \le & \alpha_d\alpha_{ec}\alpha_{ea}5\|f_e(k)\|^2+\alpha_d\alpha_{ec}\alpha_{ea}5g_M^2\|u_e(k)\|^2+\alpha_d\alpha_{ec}\alpha_{ea}5\phi_{dM}^2\left\|\widetilde{\Theta}_d(k)\right\|^2\\ & +\alpha_d\alpha_{ec}\alpha_{ea}5g_M^2\vartheta_M^2\left\|\widetilde{\Theta}_A(k)\right\|^2+\alpha_d\alpha_{ec}\alpha_{ea}5\|\varepsilon_{Ad}\|^2-\alpha_d\alpha_{ec}\alpha_{ea}\|e(k)\|^2\\ & -5(2+4\alpha_d\phi_{dM}^2)\Gamma(Q_e(e(k))+u_e(k)^TR_eu_e(k))\end{aligned} \tag{95}$$

Now, combining (83), (94), and (95) forming $\Delta V_s(k)$ reveals

$$\begin{aligned}\Delta V_s(k) \le & -\alpha_{ec}\vartheta_{eMin}^2\left(\frac{1}{2}-\alpha_{ea}3\frac{\vartheta_{eM}^4}{\vartheta_{eMin}^2}\right)\left\|\widetilde{\Theta}_A(k)\right\|^2+\alpha_d\alpha_{ec}\alpha_{ea}5g_M^2\vartheta_{eM}^2\left\|\widetilde{\Theta}_A(k)\right\|^2\\ & -X_{eMin}^2\left(1-\frac{\alpha_{ec}}{X_{eMin}^2}\left(2X_{eM}^4+\left(\frac{3\alpha_{ea}}{2}\vartheta_{eM}^2+1\right)\frac{g_M^2\left\|R_e^{-1}\right\|_F^2}{2}\sigma_{deM}^2\right)\right)\left\|\widetilde{\Theta}_c(k)\right\|^2\\ & -\alpha_{ea}\phi_{dMin}^2\left(1-2\alpha_d\frac{\phi_{dM}^4}{\phi_{dMin}^2}\right)\left\|\widetilde{\Theta}_d(k)\right\|^2+\alpha_d\alpha_{ec}\alpha_{ea}5\phi_{dM}^2\left\|\widetilde{\Theta}_d(k)\right\|^2\\ & +\alpha_{ea}\|f_e(k)\|^2(2+4\alpha_d\phi_{dM}^2)+\alpha_d\alpha_{ec}\alpha_{ea}5\|f_e(k)\|^2\\ & +\alpha_d\alpha_{ec}\alpha_{ea}5g_M^2\|u_e(k)\|^2-5(2+4\alpha_d\phi_{dM}^2)\Gamma\lambda_{\min}(R_e)\|u_e(k)\|^2-\alpha_d\alpha_{ec}\alpha_{ea}\|e(k)\|^2\\ & -5(2+4\alpha_d\phi_{dM}^2)\Gamma Q_e(e(k))+\eta_s(\varepsilon_{ec},\varepsilon_{eA},\varepsilon_d)\end{aligned}$$

where $\eta_s(\varepsilon_{ec},\varepsilon_{eA},\varepsilon_d)=\eta_e(\varepsilon_{ec},\varepsilon_{eA})+\alpha_{ea}2\varepsilon_{dM}^2(1+2\alpha_d\phi_{dM}^2)+\alpha_{ec}\alpha_d\alpha_{ea}5\varepsilon_{AdM}^2$. Combining like terms as well as applying the bounds of (54) results in the first difference of the Lyapunov function as

$$\Delta V_s(k) \le -\alpha_{ec}\vartheta_{eMin}^2\left(\frac{1}{2}-\frac{\alpha_{ea}}{\vartheta_{eMin}^2}\left(3\vartheta_{eM}^4+\alpha_d 5 g_M^2\vartheta_{eM}^2\right)\right)\left\|\widetilde{\Theta}_A(k)\right\|^2 \tag{96}$$
$$-X_{eMin}^2\left(1-\frac{\alpha_{ec}}{X_{eMin}^2}\left(2X_{eM}^4+\left(\frac{3\alpha_{ea}}{2}\vartheta_{eM}^2+1\right)\frac{g_M^2\left\|R_e^{-1}\right\|_F^2}{2}\sigma_{deM}^2\right)\right)\left\|\widetilde{\Theta}_c(k)\right\|^2$$
$$-\alpha_{ea}\phi_{dMin}^2\left(1-\frac{\alpha_d}{\phi_{dMin}^2}\left(2\phi_{dM}^4+\alpha_{ec}5\phi_{dM}^2\right)\right)\left\|\widetilde{\Theta}_d(k)\right\|^2-\Gamma\Xi_e\left\|Q_e(e(k))\right\|-\alpha_{ec}\alpha_d\alpha_{ea}\left\|e(k)\right\|^2$$
$$-\left(\Gamma\lambda_{\min}(R_e)\Xi_e+2\alpha_{ea}g_M^2(1+2\alpha_d\phi_{dM}^2)\right)\left\|u_e(k)\right\|^2+\eta_s(\varepsilon_{ec},\varepsilon_{eA},\varepsilon_d)$$

where $\Xi_e = 10+20\alpha_d\phi_{dM}^2-\alpha_{ea}(1+2\alpha_d\phi_{dM}^2)-5\alpha_{ec}\alpha_d\alpha_{ea}/2$. Finally, $\Delta V_s(k)<0$ provided the tuning parameters are selected according to

$$\alpha_{ea}<\vartheta_{eMin}^2/\left(6\vartheta_{eM}^4+\alpha_d 10 g_M^2\vartheta_{eM}^2\right),\quad \alpha_{ec}<X_{eMin}^2\Bigg/\left(2X_{eM}^4+\left(\frac{3\alpha_{ea}}{2}\vartheta_{eM}^2+1\right)\frac{g_M^2\left\|R_e^{-1}\right\|_F^2}{2}\sigma_{deM}^2\right), \tag{97}$$
$$\alpha_d<\phi_{dMin}^2/\left(2\phi_{dM}^4+\alpha_{ec}5\phi_{dM}^2\right)$$

and the following inequalities hold

$$\left\|e(k)\right\|>\sqrt{\frac{\eta_s(\varepsilon_{eA},\varepsilon_{ec},\varepsilon_d)}{\alpha_{ec}\alpha_d\alpha_{ea}}}\equiv b_e \text{ or}$$

$$\left\|\widetilde{\Theta}_c(k)\right\|>\sqrt{\frac{\eta_s(\varepsilon_{eA},\varepsilon_{ec},\varepsilon_d)}{X_{eMin}^2\left(1-\frac{\alpha_{ec}}{X_{eMin}^2}\left(2X_{eM}^4+\left(\frac{3\alpha_{ea}}{2}\vartheta_{eM}^2+1\right)\frac{g_M^2\left\|R_e^{-1}\right\|_F^2}{2}\sigma_{deM}^2\right)\right)}}\equiv b_C' \text{ or}$$

$$\left\|\widetilde{\Theta}_d(k)\right\|>\sqrt{\frac{\eta_s(\varepsilon_{eA},\varepsilon_{ec},\varepsilon_d)}{\alpha_{ea}\phi_{dMin}^2\left(1-\frac{\alpha_d}{\phi_{dMin}^2}\left(2\phi_{dM}^4+\alpha_{ec}5\phi_{dM}^2\right)\right)}}\equiv b_d \text{ or}$$

$$\left\|\widetilde{\Theta}_A(k)\right\|>\sqrt{\frac{\eta_s(\varepsilon_{eA},\varepsilon_{ec},\varepsilon_d)}{\alpha_{ec}\vartheta_{Min}^2\left(\frac{1}{2}-\frac{\alpha_{ea}}{\vartheta_{eMin}^2}\left(3\vartheta_{eM}^4+\alpha_d 5 g_M^2\vartheta_{eM}^2\right)\right)}}\equiv b_A'.$$

Additionally, selecting the tuning gains according (97) ensures $\Xi_e>0$. Therefore, using standard Lyapunov extension [1], it can be concluded that $\Delta V_s(k)$ is less than zero

outside of a compact set so that the tracking error (47) and the OLA parameter estimation errors of the cost function, feedback control signal, and feedforward control inputs are all *UUB*. To show $\|u-u^*\| \le \varepsilon_u$, we use (73), (85), (86) , (52), and (57) to observe

$$u-u^* = g(k)^{-1}(\widetilde{\Theta}_d^T \phi_d + \varepsilon_d) + \hat{u}_e - u_e^*. \tag{98}$$

Then, using (84) and taking the limit as $k \to \infty$ and the upper bound of (98) shows

$$\begin{aligned} \|u-u^*\| &\le g_M^I \phi_{dM} \|\widetilde{\Theta}_d(k)\|_F + \|\widetilde{\Theta}_A(k)\|_F \vartheta_{eM} + g_M^I \varepsilon_{dM} + \varepsilon_{eAM} \\ &\le g_M^I \phi_{dM} b_d + b_A' \vartheta_{eM} + g_M^I \varepsilon_{dM} + \varepsilon_{eAM} \equiv \varepsilon_u \end{aligned}. \quad \blacksquare$$

Remark 9: If the OLA approximation errors ε_{eA}, ε_{ec}, and ε_d become negligible [20], the term $\eta_s(\varepsilon_{eA}, \varepsilon_{ec}, \varepsilon_d)$ becomes zero. For this scenario, it can be shown that the tracking error and each OLA parameter estimation errors converge to zero asymptotically. That is, $u \to u^*$.

Remark 10: The results of *Theorem 6* are drawn under the assumption of an initial admissible control, $u_{e0}(k)$. This assumption is required to ensure that the initial cost function evaluated at $e(0)$ is finite. That is, $u_{e0}(k)$ ensures $J_e(e(0)) < \infty$. Further, once the OLA control input (86) is applied to the nonlinear system (1), OLA estimation errors are introduced into the closed loop system as observed in (88). Thus, the tracking error vector must be considered in the Lyapunov candidate (91), and assumptions regarding the stability of the error system cannot be made *a priori* after the OLA control input is applied.

Remark 11: In the development of the optimal regulation control problem, the OLA basis functions were not assumed to be bounded *a priori* whereas the assumption of bounded basis functions [1] was asserted during the design of the optimal tracking

control law. In the optimal regulator design in *Section III*, it was shown that by starting with an initial stabilizing control, all future control policies were also stabilizing, and thus, the OLA basis functions remained bounded. In contrast, as stated above, assumptions regarding the stability of the tracking error system cannot be made *a priori* after the OLA control input is applied. As a result, bounds are placed on the OLA basis function [1].

V. Simulation Results

To demonstrate the effectiveness of the online optimal controllers developed in this work, first the optimal regulator derived in *Section III* is considered. The optimal regulation algorithm developed in this work is first implemented on a linear system since the results can be easily verified by solving the discrete-time algebraic Riccati equation (DARE). Consider the linear system whose dynamics are given by

$$x(k+1)=\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix}=\begin{bmatrix} 0 & -0.8 \\ 0.8 & 1.8 \end{bmatrix}x(k)+\begin{bmatrix} 0 \\ -1 \end{bmatrix}u(k).$$

Using the quadratic cost function (2) with Q being the identity matrix and $R=1$, the optimal control input is found by solving the DARE and revealed to be $u^*(k)=[0.6239 \quad 1.2561]x(k)$ while the optimal cost function is found to be $J^*(k)=1.5063\mathrm{x}_1^2(\mathrm{k})+2.0098\mathrm{x}_1(\mathrm{k})\mathrm{x}_2(k) \quad +3.7879\mathrm{x}_2^2(\mathrm{k})$. The initial stabilizing policy for the algorithm was selected to be $u_0(k)=[0.5 \quad 1.4]x(k)$ while the basis functions for the critic were generated from a sixth order polynomial as

$$\{x_1^2, x_1x_2, x_2^2, x_1^4, x_1^3x_2, \ldots, x_2^6\} \tag{99}$$

while the action network basis functions were generated based on the gradient of (99). It is shown in [15] that the gradient of a linearly independent set is also linearly independent. The design parameters for the critic and action networks were selected as $a_c = 10^{-6}$ and $a_a = 0.1$ while the critic NN weights were set to zero at the beginning of the simulation. The initial weights of the action network were chosen to reflect the initial stabilizing control.

The simulation was run for 240 time steps, and the final values of the critic and actor weights are

$$\hat{\Phi}_c = [1.5071\ 2.0097\ 3.7886\ -0.0082\ -0.0015\ 0.0025\ 0.0030\ -0.0014\ 0.002($$
$$0.0000\ \ 0.0000\ \ 0.0008\ -0.0003\ 0.0009\ -0.0002]$$

and

$$\hat{\Phi}_A = [0.6208\ \ 1.2586\ \ 0.0589\ \ -0.0338\ \ 0.0095\ \ 0.0092$$
$$-0.0049\ \ 0.0074\ \ 0.0050\ \ 0.0075.\ \ -0.0054].$$

Examining the final values for the NN adaptive critic weights, it is clear that they have successfully learned the optimal values with small bounded error as the results of *Theorem 3* suggested. Additionally, the difference between the optimal control law obtained from the DARE and the optimal control learned online is shown in Fig. 3 further demonstrating the effectiveness of the online optimal control scheme. It is observed that the NN control law converges to the optimal value with small bounded error within the first 200 time steps as the theoretical conjectures of *Theorem 3* suggest. Finally, Fig. 4 illustrates the system state trajectories for the initial stabilizing control and the improved final optimal control law.

Next, a nonlinear regulation example is examined. Consider a nonlinear system defined by

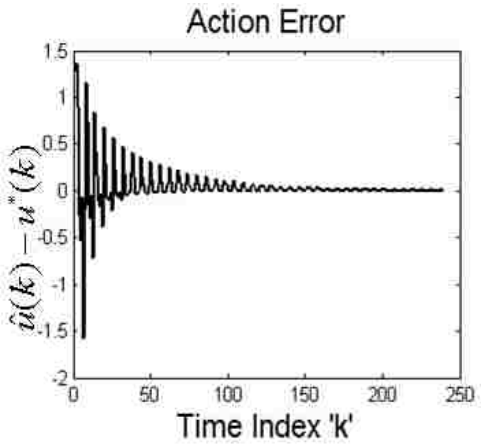


Fig. 3 Action error.

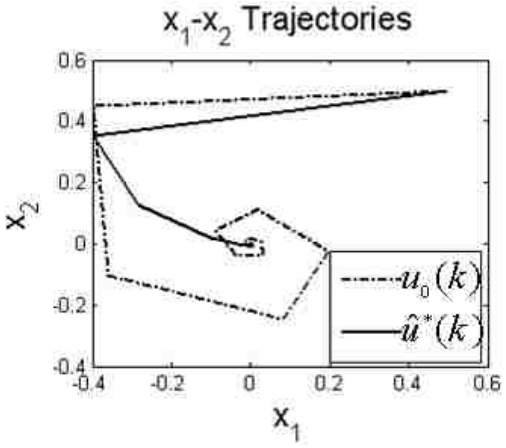


Fig. 4 State trajectories.

$$x(k+1) = \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} -x_1(k)x_2(k) \\ x_1^2(k) + 1.8x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u(k). \tag{100}$$

The design parameters, activation functions, and initial network weights were chosen similarly to the linear system example with the initial stabilizing control given by $u_0(k) = [-0.4 \quad 1.24]x(k)$. The simulation was ran for 375 time steps, and the time history of the critic and actor weights are shown in Fig. 5, and the action error (24) is shown in Fig. 6. Examining Fig. 5, it is clear that all NN weights remain bounded while Fig. 6 illustrates the action error converges to a small bounded region around the origin consistent with *Theorem 1*.

As a comparison, the SDRE algorithm [18],[23] was implemented along with the offline training algorithm presented in [7]. For the SDRE implementation, the nonlinear system (100) was parameterized according to

$$x(k+1) = \underbrace{\begin{bmatrix} 0 & -x_1(k) \\ x_1(k) & 1.8 \end{bmatrix}}_{A(x(k))} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}_{B} u(k),$$

and the discrete SDRE $P(x(k)) = A(x(k))^T [P(x(k)) - P(x(k))B^T (R + B^T P(x(k))B)^{-1} \times BP(x(k))]A(x(k)) + Q$ was solved at each time step to render a suboptimal feedback control law [23]

$$u(x(k)) = -\left((B^T P(x(k))B + R)^{-1} B^T P(x(k))A(x(k))\right)x(k)$$

with $R=I$ and $Q=I$ and where complete knowledge of the internal dynamics $A(x(k))$ is required.

For the offline training algorithm presented in [7], the training set was generated from the region $x_1 \in [-0.5,0.5]$ and $x_2 \in [-0.5,0.5]$ with a mesh size of 0.02 [7]. The control input from [7] is generated according to

$$u^*(k) = -[g^T(x)\nabla^2 J^*(x)g(x) + 2R]^{-1} g^T(x)[\nabla J^*(x) + \nabla^2 J^*(x)(f(x) - x)] \qquad (101)$$

where $\nabla J^*(x)$ and $\nabla^2 J^*(x)$ are the gradient and Hessian of the cost function, respectively. Examining (101), the optimal control law obtained via offline training requires explicit knowledge of the internal dynamics $f(x)$ [7]. The initial stabilizing control policy and the basis functions for the critic network for the offline algorithm were taken to be the same as those used to implement the online algorithm of this work.

Fig. 7 shows the state trajectories when the final optimal control policies learned online, trained offline, and using the discrete SDRE solution, respectively, are applied to the nonlinear system (100), and from the plot, it is clear that the resulting state trajectories for the online learning and offline training solutions are identical. However, the SDRE solution differs from online and offline HJB based solutions. This result illustrates that although SDRE is an attractive alternative for nonlinear optimal control, the resulting control laws are still suboptimal even when the exact dynamics are known.

Fig. 8 displays the difference between the final optimal control policy learned online and the final optimal control policy found via offline training [7]. Examining the plot, the difference between the two control policies is less than 0.015.

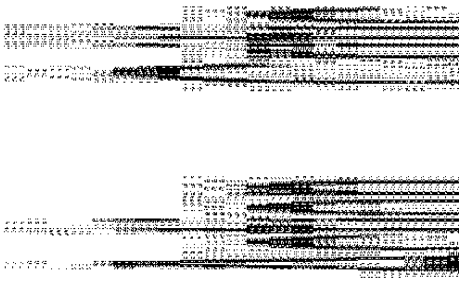


Fig. 5. Adaptive critic weights.

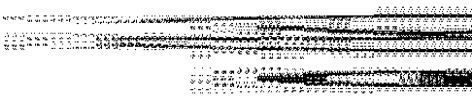


Fig. 6. Action error.

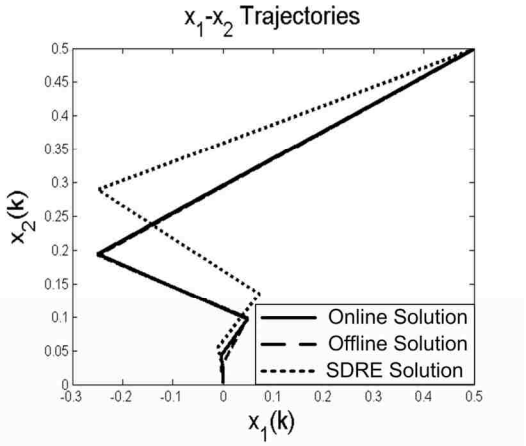


Fig. 7. State trajectories for the nonlinear example.

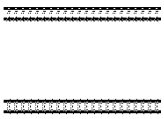


Fig. 8. Difference between optimal controls.

Next, the tracking problem is considered. To demonstrate the effectiveness of the nearly optimal nonlinear tracking controller, the algorithm developed in *Section IV* was implemented on a differentially driven nonholonomic mobile robot whose discretized nonlinear system is described by [24]

$$v(k+1)=\begin{bmatrix} v_R(k+1) \\ v_L(k+1) \end{bmatrix}=\begin{bmatrix} v_R(k) \\ v_L(k) \end{bmatrix}+Tf(v(k))+TM^{-1}\tau \tag{102}$$

where $f(v(k))=-M^{-1}(V(v(k))v(k)+F(v(k)))$, τ is the control torque, $v_R(k)$ and $v_L(k)$ are the velocities of the right and left wheels of the robot, respectively, and T is the sampling time. Further, M is the inertial matrix given by

$$\begin{bmatrix} r^2(mb^2+I)/(rb^2)+I_\omega & r^2(mb^2-I)/(rb^2) \\ r^2(mb^2-I)/(rb^2) & r^2(mb^2+I)/(rb^2)+I_\omega \end{bmatrix},$$

$V(v(k))$ is the nonlinear Coriolis forces matrix given by

$$\begin{bmatrix} 0 & -(v_R(k)-v_L(k))r^3m_cd/(4b^2) \\ (v_R(k)-v_L(k))r^3m_cd/(4b^2) & 0 \end{bmatrix}$$

and $F(v(k))$ is the nonlinear friction vector which will be modeled as

$$F(v(k))=[\alpha_1 sign(v_R(k))+\beta_1 v_R(k), \alpha_2 sign(v_L(k))+\beta_2 v_L(k)]^T$$

where α_i, β_i, $i=1,2$, are the coefficients of friction. In addition, the robot parameters considered in the simulation are the radius of the driving wheels, r, distance between the driving wheels, $2b$, distance from the driving axle to the center of mass, d, mass of the platform without wheels, m_c, mass of each wheel, m_w, robot moment of inertia about the center of mass, I_c, moment of inertial of the wheel about its axle, I_ω, and the moment of inertial of each wheel about its diameters, I_m [24]. Note that $m = m_c + 2\, m_w$. The values ofthe above parameters used in the simulation are $r=0.15m$, $b=0.5m$, $d=0.2m$,

$m_c = 30kg, m_w = 2kg$, $I_c = 15kg \cdot m^2$, $I_\omega = 0.005kg \cdot m^2$, $I_m = 0.0025kg \cdot m^2$, $\alpha_1 = 0.1$, $\alpha_2 = 0.15$, $\beta_1 = 0.2$, and $\beta_2 = 0.2$. The sampling time is taken as *T=0.01 seconds.*

The objective of the mobile robot is to track a virtual reference cart, and the desired wheel velocities are generated online according to [24]

$$v_d(k) = \begin{bmatrix} v_{d1}(k) \\ v_{d2}(k) \end{bmatrix} = \begin{bmatrix} 1/r & b/r \\ 1/r & -b/r \end{bmatrix} \begin{bmatrix} v_r \cos(e_3(k)) + k_1 e_1(k) \\ \omega_r + k_2 v_r e_2(k) + k_3 v_r \sin(e_3(k)) \end{bmatrix}$$

where v_r and ω_r are the translational and angular velocities of the virtual reference cart, $k_1, k_2,$ and k_3 are positive design constants, and e_1, e_2, and e_3 are the position tracking errors of the wheeled mobile robot as defined in [24]. For this test, the reference translational and angular velocities were taken as $v_r = 1m/s$ and $\omega_r = 0.5\sin(0.2\pi kT)\, rad/s$.

To implement the control scheme, two-layer NN's are considered consisting of one layer of randomly assigned constant weights, v_N, in the first layer and one layer of tunable weights, Θ_N, in the second layer. A compromise is made here between tuning the number of layered weights with computational complexity. It has been shown that by randomly selecting the input layer weights v_N, the activation function forms a stochastic basis, and thus the approximation property holds for all inputs in a compact set [1].

Additionally, 10 hidden layer neurons were selected for both the cost function and feedback control OLA's while 25 hidden layer neurons were selected in order to estimate the feedforward signal of the control input. The activation function of the cost function OLA was selected as hyperbolic tangent squared in order to obtain an even linearly independent basis function. Conversely, the gradient of the cost function activation

function was selected as the basis function of the feedback control signal OLA as a result of the relationship observed in (51). Finally, radial basis functions were selected as activation functions for the feedforward control estimator.

The initial stabilizing control law was selected as

$$u_{e0} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} v_R(k) - v_{d1}(k) \\ v_L(k) - v_{d2}(k) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} e_4(k) \\ e_5(k) \end{bmatrix}. \tag{103}$$

To establish a performance baseline, an initial simulation was performed using the control policy (103) while assuming knowledge of the internal system dynamics, and the cost associated with the initial control policy was found to be 5.48. Subsequently, the simulation was performed again while assuming the knowledge of the internal dynamics were not available using the feedforward estimator of *Section IV-C*, and the cost associated with this case was found to be 5.61.

Next, the OLA optimal control scheme was tested. The control gains were selected as $\alpha_{ec} = 0.1$, $\alpha_{ea} = 0.1$ and $\alpha_d = 0.09$, and all tunable NN weights were initialized to zero. The simulation was ran for 10 seconds (1000 time steps), and for the first 5 seconds, a disturbance with mean zero and variance 0.04 was added to the system in order to ensure the persistency of excitation condition holds. Recall from *Section IV* that the cost and feedback control error signals become zero when the tracking error reaches zero.

The resulting robot trajectory is shown in Fig. 9. From the trajectory, it is observed that the robot converges to the path of the virtual cart and maintains the desired course for the remainder of the test. The time histories of both the cost function and feedback control signal parameter estimates are shown in Fig 10. Examining the figure, it is clear that the parameter estimates converge to constant values and remain bounded

consistent with *Theorem 6*. It is observed that the magnitude of the cost function NN weights are on the order of 10^4; however, the proceeding discussion and comparisons will illustrate that these values are consistent with the magnitude of the actual cost function being approximated.

The cost function and feedback control errors (61) and (74), respectively, are shown in Fig. 11. Examining the plots, it is clear that both errors initially incur large values but then converge to a small bounded value near the origin. Additionally, the difference between the actual feedforward control term and the estimated feedforward term is shown in Fig. 12. Here, the estimation error is found to be small and bounded consistent with the theoretical results of *Section IV-D*.

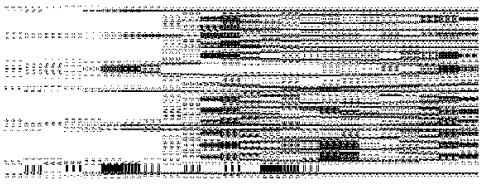


Fig. 9. Robot trajectory.

Fig. 10. OLA parameter estimates.

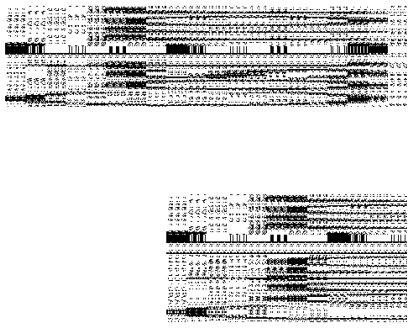


Fig. 11. OLA feedback control and cost errors.

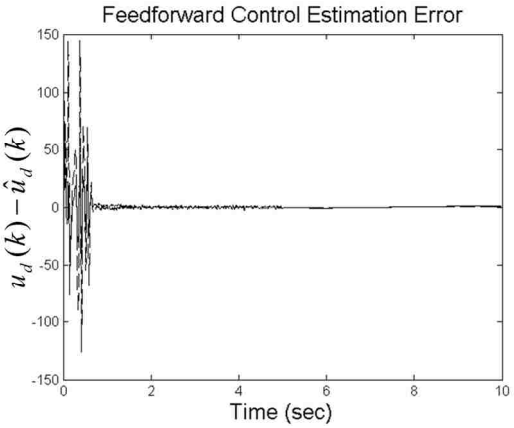


Fig. 12. OLA feedforward control term estimation error.

Next, the final OLA parameter estimates were used to re-evaluate the system performance using the improved control $\hat{u}_e^*$. The improved control was applied to the system when the internal dynamics were known and when they were unknown. These results as well as the results of the initial stabilizing control test are summarized in Table I. Comparing the costs, it is clear that the OLA-based optimal control input is an improvement over the initial control policy both when f_d and when it is not.

TABLE I. COST VALUE COMPARISONS

Control policy	Cost with f_d known	Cost with f_d unknown
u_{e0}	3.6451e5	3.3455e5
$\hat{u}_e^*$	3.0192e5	2.9168e5

VI. Conclusions

In this work, direct dynamic programming techniques were utilized to solve the Hamilton Jacobi-Bellman equation in real time for the optimal control of general affine nonlinear discrete-time systems using online approximators to address the regulation problem and the tracking control problem. The internal dynamics of the system were considered to be unknown, and a novel nearly optimal control laws were developed using OLA's. Given an initial admissible control policy, OLA's were utilized to learn the cost function and nearly optimal feedback control signal for both the regulation and tracking problems. For the tracking problem, an additional OLA was utilized in the design of a desired feedforward portion of the control input to render a stable system. All OLA parameters were tuned online using novel update laws, and Lyapunov techniques were used to demonstrate the stability of the proposed optimal control schemes. Simulation results were also provided to verify the theoretical conjectures.

References

[1] S. Jagannathan, *Neural network control of nonlinear discrete-time systems*, CRC Press, April 2006.

[2] F. L. Lewis and V. L. Syrmos, *Optimal Control*, 2nd ed. Hoboken, NJ: Wiley, 1995.

[3] A. Isidori and A. Astolfi, "Disturbance attenuations and H_∞-control via measuremnt feedback in nonlinear systems, " *IEEE Trans. Autom. Control,* vol. 37, no. 9, pp. 1283-1293, 1992.

[4] S. Keerthi and E. Gilbert, "An existence theorem for discrete-time infinite-horizon optimal control problems," *IEEE Trans. Autom. Control,* vol. 30, no. 9, pp. 907-909, 1985.

[5] W. Lin and C. I. Brynes, "H_∞-control of discrete-time nonlinear systems,"*IEEE Trans. Autom. Control*, vol. 41, no. 4, pp. , 494–510, 1996.

[6] W. Lin and C. I. Brynes, "Discrete-time nonlinear H_∞ control with measurement feedback,"*Automatica*, vol. 31, no. 3, pp. , 419–434, 1995.

[7] Z. Chen and S. Jagannathan, "Generalized Hamilton-Jacobi-Bellman formulation based neural network control of affine nonlinear discrete-time systems, "*IEEE Trans. On Neural Networks,* vol. 10, pp. 90-106, Jan. 2008.

[8] A. Al-Tamimi, F.L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof," *IEEE Trans. On Systems, Man, and Cybernetics—Part B*, vol. 38, pp. 943-949, Aug. 2008.

[9] J. Huang, "An algorithm to solve the discrete HJI equations arising in the L2 gain optimization problem," *Int. J. Control*, vol. 72, no. 1, pp. 49–57, 1999.

[10] R. W. Beard and T. W. McLain, "Successive Galerkin approximation algorithms for nonlinear optimal and robust control," *Int. J. Control*, vol. 71, no. 5, pp.717-743, 1998.

[11] J. Si and Y. Wang, "Online learning control by association and reinforcement," *IEEE Trans. on Neural Networks,* vol. 12, pp. 264-276, March 2001.

[12] D. Prokhorov and D. Wunsch, "Adaptive Critic Designs," *IEEE Trans. on Neural Networks,*" vol. 8, pp. 997-1007, Sept. 1997.

[13] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, Eds., *Handbook of Learning and Approx. Dynamics Prog.,* Wiley-IEEE Press, 2004.

[14] W. Rudin, *Principles of Mathematical Analysis,* 3rd ed. New York: McGraw-Hill, 1976.

[15] R. Beard, G. Saridis, and J. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, Vol. 33, pp. 2159-2177, 1997.

[16] G. Toussaint, T. Basar and F. Bullo, "H^∞-optimal tracking control techniques for nonlinear underactuated systems." *Proc. of IEEE Conf. on Decision and Control,* pp. 2078 – 2083, 2000.

[17] D. Gu and H. Hu, "Receding Horizon Tracking Control of Wheeled Mobile Robots," *IEEE Trans. on Control Systems Technology,* vol. 14, pp. 743-49, July 2006.

[18] J. Shamma and J. Cloutier, "Existence of SDRE stabilizing feedback," *IEEE Trans. on Automatic Control,* vol. 48, pp. 513-517, March 2003.

[19] W. Luo, Y. C., Chu, and K. V. Ling, "Inverse optimal adaptive control for attitude tracking of spacecraft," *IEEE Trans. on Automatic Control,* vol. 50, pp 1693-1654, Nov. 2005.

[20] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. on Systems, Man, and Cybernetics—Part B,* vol. 38, pp. 937-942, Aug. 2008.

[21] P.M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamical systems via a multilayer neural network feedforward and RISE feedback control structure," *IEEE Trans. on Automatic Control,* vol. 53, pp 2180-2185, Oct. 2008.

[22] H. K. Khalil, *Nonlinear Systems*, 3rd edition, Prentice-Hall, NJ, 2002.

[23] A. S. Dutka, A. W. Ordys, M. J. Grimble, "Optimized discrete-time state dependent Riccati equation regulator," *Proc. of the American Control Conf.*, pp. 2293 – 2298, 2005.

[24] M. K. Bugeja, S. G. Fabri, and L. Camilleri, "Dual adaptive dynamic control of mobile robots using nearal networks," *IEEE Trans. on Systems, Man, and Cybernetics—Part B,* vol. 39, pp. 129-141, Feb. 2009.

6. Optimal Control of Affine Nonlinear Continuous-time Systems using an Online Approximator[1]

T. Dierks and S. Jagannathan

Abstract—*In this paper, a novel single online approximator (SOLA)-based scheme is designed to solve the optimal regulation and tracking control problems for affine nonlinear continuous-time systems with known dynamics. The SOLA-based adaptive approach is designed to learn the infinite horizon continuous-time Hamilton-Jacobi-Bellman (HJB) equation, and the corresponding optimal control input that minimizes the HJB equation is calculated forward-in-time. Subsequently, the SOLA architecture is extended to learn the Hamilton-Jacobi-Isaacs (HJI) equation commonly used in H_∞ optimal control. Novel parameter tuning algorithms are derived which not only ensures the optimal cost (HJB or HJI) function and control input are achieved, but also ensure the system states remain bounded during the online learning process. Lyapunov techniques are used to show that all signals are uniformly ultimately bounded (UUB) and that the approximated control signals approach the optimal control inputs with small bounded error. In the absence of OLA reconstruction errors, asymptotic convergence to the optimal control is demonstrated. Simulation results are included to show the effectiveness of the approach.*

Index Terms— Online nonlinear optimal control; Single network adaptive critic; Hamilton-Jacobi-Bellman; Hamilton-Jacobi-Isaacs; Tracking; Online approximators.

[1] Research Supported in part by NSF ECCS#0621924 and Intelligent Systems Center. Authors are with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, 1870 Miner Circle, Rolla, MO 65409. Contact author Email: tad5x4@mst.edu

I. INTRODUCTION

The stabilization of nonlinear continuous-time systems has been considered by many researchers [1]-[3] using methods ranging from feedback linearization [1] to the use of online approximators (OLA's) [2]-[3]. However, stability is typically the only consideration for the resulting control laws [1]-[3]. In many cases, it is desirable that the control law not only stabilizes the system, but also minimizes on a pre-defined cost function to achieve optimality. Traditionally, the optimal control of linear systems accompanied by quadratic cost functions can be attained by solving the well known Riccati equation [4]. However, the optimal control of nonlinear continuous time systems is a much more challenging task that often requires solving the nonlinear Hamilton-Jacobi-Bellman (HJB) equation or the Hamilton-Jacobi-Isaacs (HJI) equation when H_∞ optimal control is being considered.

To extend the results of linear optimal control theory to nonlinear systems, the state dependent Riccati equation (SDRE) [5] was proposed; however, the SDRE yields a sub-optimal result in most cases [5]. In general, the HJB and HJI equations are more difficult to work with than Riccati equations because they involve solving nonlinear partial differential or difference equations [4]. To avoid finding exact solutions to the infinite horizon cost (HJB or HJI) functions, inverse optimal control [6], Markov decision processes [7]-[8], and receding horizon control [9] techniques have been applied for nonlinear systems. Alternatively, neural networks (NN's) and dynamic programming techniques [10]-[11] have been used to investigate both the discrete and continuous time nonlinear optimal regulation problems while attempting to solve the HJB or HJI equations [12]-[14]. However, in each case the optimal solutions are obtained offline and in an iterative manner, and the NN reconstruction errors are considered to be negligible. In addition to NN's, Taylor series

expansions and Galerkin approximation techniques have also been used to estimate the solution to the HJI equation [15]-[17].

On the other hand, the optimal tracking control problem has been considered in recent literature through linearization of the tracking error equations [18], model predictive control with a receding horizon [19], inverse optimal control [20], directly calculating the infinite horizon HJB equation via offline scheme [21], and online learning-based technique [22]. In [18], the authors consider the H_∞ optimal tracking control by linearizing the error equations about the origin yielding a locally optimal control law. To overcome linearization, the authors in [21] consider the HJB equation and employ similar techniques as [13] to find an offline solution to the optimal tracking control problem.

In contrast, several online approximator-based controller designs were presented in [10] and [22]-[25] to overcome the iterative offline training methodology and are often referred to as adaptive critic designs (ACD). The central theme of several works in [10] is that the optimal control law and HJB function are approximated by online parametric structures, such as NN's and forward-in-time. Although the techniques [10] are verified via numerical simulations, the approximation errors are not considered and mathematical proofs of convergence are not offered. Recently, several online methods to solve the continuous and discrete time HJB and HJI equations were presented in [23]-[25]. In [23] and [24], online policy iterations based on adaptive control and Q-learning [26] are developed to solve the continuous HJB and discrete HJI problems, respectively. Although, full knowledge of the system dynamics is not required, the methods [23]-[24] are only applicable to linear systems.

For affine nonlinear continuous-time systems, two policy iteration schemes using NN's have been introduced in [25] for optimal control. In each scheme, two NN's, one

referred as critic for approximating the cost function and the second NN to approximate the optimal control signal and referred to as action NN, are considered to approximate the cost (HJB) function and the corresponding optimal control policy, respectively. In the first scheme, discrete-time adaptation of the actor and critic structures is undertaken by training the cost approximator in discrete time intervals while the second algorithm tunes both the cost function network and the control policy approximator simultaneously in continuous time. In addition, proof of convergence for both algorithms is demonstrated using Lyapunov methods.

In our previous work [22], a novel approach to the optimal regulation and tracking of nonlinear discrete-time affine systems was undertaken to solve the discrete-time HJB equation online and forward-in-time. Using an initial stabilizing control, an OLA was tuned online to learn the HJB equation while a second OLA was utilized that minimizes the cost (HJB) function based on the information provided by the first OLA. Lyapunov methods were used to rigorously demonstrate that the approximated control signals approached the optimal control inputs with small bounded error. Also, in the absence of disturbances and OLA reconstruction errors, an optimal control was demonstrated. On the other hand, a single network adaptive critic (SNAC) NN-based optimal control scheme was introduced for discrete time systems in [28]. However, the SNAC was trained offline, and proof of convergence for the NN implementation has not been shown in contrast with [22].

By contrast, in this work, affine nonlinear continuous-time systems are considered in the development of a novel single online approximator-based (SOLA) unified framework to learn both the HJB and HJI functions online and forward-in-time for the optimal regulation and tracking control problems in contrast with [22] and [25] where two OLA's are utilized. First, using a single online approximator, the HJB equation is approximated online and

forward in time while the optimal control input is calculated using the parameters of the approximator. Next, for the HJI problem, the SOLA design is extended to learn the HJI equation as well as the minimizing control input and maximizing disturbance term. Novel online parameter tuning laws for the SOLA are derived that not only ensures the optimal cost (HJB or HJI) function, control inputs, and disturbance are achieved, but also ensure the system states remain bounded during the online learning process. Lyapunov theory is utilized to demonstrate the stability of the system while explicitly considering the approximation errors resulting from the use of the OLA in contrast to the other works [12]-[13], [10], and [21]. Further, the theoretical results in this work show that an initial stabilizing control is not required in contrast to [22] and [25] where an initial stabilizing control is necessary for stability. In the absence of the reconstruction errors, asymptotic stability is demonstrated while achieving optimal control. The OLA's considered in this work are NN's although any nonlinear approximator such as radial basis functions, splines, polynomials, and linear in the tunable parameter (LIP) adaptive control technique can be utilized.

The near optimal control laws proposed in this work are obtained without linearizing the equations about the origin [18] and are accomplished using the infinite horizon cost function in contrast with [9]. Additionally, the proposed approach is solved online and forward-in-time using full knowledge of the system dynamics without the need of an initial stabilizing control while using a SOLA in contrast with [22] and [25] which requires an initial stabilizing control as well as two OLA's. In addition, to extend frameworks in [22] and [25] to learn the HJI equation, a third approximator appears to be required whereas the HJI problem solved in this work using only a single network. The assumption on the

requirement of the dynamics of the continuous-time system can be relaxed by using an additional OLA [27] which will be relegated as part of future work.

To date, the authors are not aware of a continuous-time SOLA framework that 1) learns cost function and optimal control input online in the continuous time domain; 2) explicitly considers OLA approximation errors; 3) provides an explicit proof of convergence of the OLA parameters and stability of the system states; and 4) can be extended to solve the HJI optimal control. This work will address these issues.

This paper is organized as follows. First, background information for the continuous time nonlinear optimal HJB and HJI regulation problems are presented in Section II. In Section III, the nearly optimal HJB regulation control law is derived, and the stability is verified using Lyapunov theory. Subsequently, the SOLA framework is extended to learn the HJI function. The nearly optimal tracking control law is developed in Section IV, and Section V illustrates the effectiveness of the proposed regulation and tracking schemes via numerical simulations. Section VI provides concluding remarks.

II. Nonlinear Optimal Control in Continuous Time

A. Hamilton-Jacobi-Bellman Equation

Consider the continuous nonlinear affine system in the absence of disturbances described by

$$\dot{x} = f(x) + g(x)u_1 \tag{1}$$

where $x \in \Re^n$, $f(x) \in \Re^n$, $g(x) \in \Re^{nxm}$ is bounded satisfying $g_{\min} \trianglelefteq \| g(x) \|_F \leq g_{\max}$ where the Frobenius norm is applied, and $u_1 \in \Re^m$ is the control input. Without loss of generality, assume that the system is observable and controllable, smooth and drift free, with $x = 0$ a

unique equilibrium point on a compact set $\Omega \in \Re^n$ with $f(0) = 0$. Under these conditions, the optimal control input for the nonlinear system (1) can be calculated [4]. Additionally, the dynamics $f(x)$ and $g(x)$ are assumed to be known throughout the development of this work.

The infinite horizon HJB cost function for (1) is given by

$$V(x(t)) = \int_t^\infty r(x(\tau), u_1(\tau))d\tau \tag{2}$$

where $r(x(t), u_1(t)) = Q(x) + u_1{}^T R u_1$, $Q(x) > 0$ is the positive define penalty on the states, and $R \in \Re^{mxm}$ is a positive definite matrix. Selecting the state penalty $Q(x)$ to be positive definite ensures that variations in any direction of the state x affects the cost $V(x(t))$ which can be linked to the observability condition [4]. Moving on, the control input u_1 is required to be selected such that the cost function (2) is finite; or u_1 must be admissible [22].

Next, we define the Hamiltonian for the cost function (2) with an associated admissible control input u_1 to be [4]

$$H(x,u) = r(x,u) + V_x^T(x)(f(x) + g(x)u_1) \tag{3}$$

where $V_x(x)$ is the gradient of the $V(x)$ with respect to x. It is well known that the optimal control input $u_1{}^*(x)$ that minimizes the cost function (2) also minimizes the Hamiltonian (3); therefore, the optimal control is found by solving the stationary condition $\partial H(x,u_1)/\partial u_1 = 0$ and revealed to be [4]

$$u_1{}^*(x) = -\frac{1}{2} R^{-1} g(x)^T V_x^*(x). \tag{4}$$

Substituting the optimal control (4) into the Hamiltonian (3) while observing $H(x, u_1{}^*, V_x^*) = 0$ reveals the HJB equation and the necessary and sufficient condition for optimal control to be [4]

$$0 = Q(x) + V_x^{*T}(x) f(x) - \frac{1}{4} V_x^{*T}(x) g(x) R^{-1} g(x)^T V_x^*(x) \tag{5}$$

with $V^*(0) = 0$. For linear systems, equation (5) yields the standard algebraic Riccati equation (ARE) [4].

Before proceeding, the following technical lemma is required.

Lemma 1. Given the nonlinear system (1) with associated cost function (2) and optimal control (4), let $J_1(x)$ be a continuously differentiable, radially unbounded Lyapunov candidate such that $\dot{J}_1(x) = J_{1x}^T(x)\dot{x} = J_{1x}^T(x)(f(x) + g(x)u_1^*) < 0$ with $J_{1x}(x)$ being the partial derivate of $J_1(x)$. Moreover, let $\overline{Q}(x) \in \Re^{nxn}$ be a positive definite matrix satisfying $\|\overline{Q}(x)\| = 0$ only if $\|x\| = 0$ and $\overline{Q}_{\min} \le \|\overline{Q}(x)\| \le \overline{Q}_{\max}$ for $\chi_{\min} \le \|x\| \le \chi_{\max}$ for positive constants $\overline{Q}_{\min}, \overline{Q}_{\max}, \chi_{\min}$ and $\chi_{\max}$. In addition, let $\overline{Q}(x)$ satisfy $\lim_{x\to\infty} \overline{Q}(x) = \infty$ as well as

$$V_x^{*T} \overline{Q}(x) J_{1x} = r(x, u^*) = Q(x) + u_1^{*T} R u_1^*. \tag{6}$$

Then, the following relation holds

$$J_{1x}^T (f(x) + g(x)u_1^*) = -J_{1x}^T \overline{Q}(x) J_{1x}. \tag{7}$$

Proof: When the optimal control (4) is applied to the nonlinear system (1), the cost function (2) becomes a Lyapunov function rendering

$$\dot{V}^*(x) = V_x^{*T}(x)\dot{x} = V_x^{*T}(x)(f(x) + g(x)u_1^*) = -Q(x) - u_1^{*T} R u_1^* \tag{8}$$

from (5). After manipulation and substitution of (6), equation (8) is rewritten as

$$\begin{aligned} (f(x) + g(x)u_1^*) &= -(V_x^* V_x^{*T})^{-1} V_x^* (Q(x) + u_1^{*T} R u_1^*) \\ &= -(V_x^* V_x^{*T})^{-1} V_x^* V_x^{*T} \overline{Q}(x) J_{1x} \\ &= -\overline{Q}(x) J_{1x} \end{aligned} \tag{9}$$

Now, multiply both sides of (9) by J_{1x}^T yields the desired relationship in (7). ■

In [25], the closed loop dynamics $f(x)+g(x)u_1^*$ were required to satisfy a Lipschitz condition such that $\|f(x)+g(x)u_1^*\| \le K$ for a constant K. In contrast, the optimal closed loop dynamics are assumed to be upper bounded by a function of the system states in this work such that

$$\|f(x)+g(x)u_1^*\| \le \delta(x). \tag{10}$$

The generalized bound $\delta(x)$ is taken as $\delta(x) \equiv \sqrt[4]{K^*\|J_{1x}\|}$ in this work where $\|J_{1x}\|$ can be selected to satisfy general bounds and K^* is a constant. For example, if $\delta(x) = K_1\|x\|$, then it can be shown that selecting $J_1(x) = (x^T x)^{(5/2)}/5$ with $J_{1x}(x) = (x^T x)^{(3/2)} x^T$ satisfies the bound. The assumption of a time varying upper bound in (10) is a less stringent assumption than the constant upper bound required in [25].

B. Hamilton-Jacobi-Isaacs Equation

Consider the nonlinear system (1) now in the presence of disturbances and rewritten as

$$\begin{aligned} \dot{x} &= f(x)+g(x)u_2+k(x)d \\ z &= \sqrt{r(x,u_2)} \end{aligned} \tag{11}$$

where x, $f(x)$, and $g(x)$, are defined as in (1), u_2 is the control input, $k(x) \in \Re^{nxw}$ is bounded according to $\|k(x)\|_F \le k_M$, $d \in \Re^w$ is the disturbance, z is a penalty output, and $r(x,u_2)$ similarly to $r(x,u_1)$. Assumptions regarding the equilibrium point, controllability, and observability of system (11) are taken to be the same as those made for the nonlinear system (1) while the bounds on the optimal closed loop dynamics are taken similarly as (10).

The H_∞ optimal control problem aims to not only minimize a cost function but also attenuate a worst-case disturbance [29]. Thus, the H_∞ optimal control problem is often

referred to as a two-player differential game where one player (u_2) tries to minimize the cost function while the other (d) tries to maximize it. In [30], dissipativity [31] was employed to convert the H_∞ optimal control problem into an L_2-gain optimal control problem which requires solving the HJI equation. Therefore, the cost function for the HJI problem is defined as [14]

$$V_d(x(t)) = \int_t^\infty \left(r(x(\tau), u_2(\tau)) - \gamma^2 d(\tau)^T P d(\tau)\right) d\tau \tag{12}$$

where $P \in \Re^{wxw}$ is a constant positive definite matrix, $\gamma > 0$ is a constant, and where u_2 is required to be admissible.

The Hamiltonian for the HJI problem is written as [14]

$$H_d(x, u_2, d) = r(x, u_2) - \gamma^2 d^T P d + V_{dx}^T(x)(f(x) + g(x)u_2 + k(x)d) \tag{13}$$

where $V_{dx}(x)$ is the gradient of the $V_d(x)$ with respect to x. Then, applying the stationary conditions $\partial H_d(x, u_2, d)/\partial u_2 = 0$ and $\partial H_d(x, u_2, d)/\partial d = 0$ reveals the optimal control and disturbance to be

$$\left.\begin{aligned} u_2 *(x) &= -\frac{1}{2} R^{-1} g(x)^T V_{dx}^*(x) \\ d *(x) &= \frac{1}{2\gamma^2} P^{-1} k(x)^T V_{dx}^*(x) \end{aligned}\right\}. \tag{14}$$

Now, substituting (14) into (13) reveals the HJI equation as

$$0 = Q(x) + V_{dx}^{*T}(x) f(x) - \frac{1}{4} V_{dx}^{*T}(x) g(x) R^{-1} g(x)^T V_{dx}^*(x) + \frac{1}{4\gamma^2} V_{dx}^{*T}(x) k(x) P^{-1} k(x)^T V_{dx}^*(x) \tag{15}$$

with $V_d^*(0) = 0$. For linear systems, equation (15) yields the game algebraic Riccati equation (GARE) [24]. Before proceeding, the following technical results are required.

Definition 1: L_2-Gain [30]. The nonlinear system (11) is said to have an L_2-gain less than γ if

$$\int_0^T z^2(\tau)d\tau = \int_0^T r(x(\tau),u_2(\tau))d\tau \le \gamma^2 \int_0^T d^T(\tau)Pd(\tau)d\tau \tag{16}$$

for all $T \ge 0$.

Lemma 2 [30],[32]: If the nonlinear system (11) with $d = 0$ is asymptotically stable and in addition has an L_2-gain less than γ, and if the cost function (12) is smooth, then the closed loop dynamics

$$\dot{x} = f(x) + \frac{1}{2}\left(\frac{1}{\gamma^2}k(x)P^{-1}k(x)^T - g(x)R^{-1}g(x)^T\right)V_{dx}(x) \tag{17}$$

are asymptotically stable.

In the next section, a SOLA-based optimal control scheme will be introduced.

III. Single Online Approximator-based Optimal Control Scheme

Traditionally, adaptive critic based methodologies generate the optimal control using two OLAs [13], [22], [25]. In this work, the adaptive critic is realized using only one OLA and in an online fashion. First, the SOLA-based scheme will be designed to learn the HJB cost function (2) and then extended to include the HJI function (4) for generating optimal control inputs (12) and (14) respectively.

A. SOLA to Learn the HJB Function

To begin the development, we rewrite the cost function (2) using an OLA representation as

$$V(x) = \Theta^T\phi(x) + \varepsilon(x) \tag{18}$$

where $\Theta \in \Re^L$ is the constant target OLA vector, $\phi(x) : \Re^n \to \Re^L$ is a linearly independent basis vector which satisfies $\phi(0) = 0$, and $\varepsilon(x)$ is the OLA reconstruction error. The target OLA vector and reconstruction errors are assumed to be upper bounded according to

$\|\Theta\| \le \Theta_M$ and $\|\varepsilon(x)\| \le \varepsilon_M$, respectively [3]. In addition, it will be assumed that the gradient of the OLA reconstruction error with respect to x is upper bounded according to $\|\partial\varepsilon(x)/\partial x\| = \|\nabla_x \varepsilon(x)\| \le \varepsilon'_M$ [33]. The gradient of the OLA cost function (18) is written as

$$\frac{\partial V(x)}{\partial x} = V_x(x) = \nabla_x^T \phi(x)\Theta + \nabla_x \varepsilon(x). \tag{19}$$

Now, using (19), the optimal control (4) and HJB function (5) are rewritten as

$$u_1^*(x) = -\frac{1}{2} R^{-1} g(x)^T \nabla_x^T \phi(x)\Theta - \frac{1}{2} R^{-1} g(x)^T \nabla_x \varepsilon(x) \tag{20}$$

and

$$H^*(x,\Theta) = Q(x) + \Theta^T \nabla_x \phi(x) f(x) - \frac{1}{4}\Theta^T \nabla_x \phi(x) D \nabla_x^T \phi(x)\Theta + \varepsilon_{HJB} = 0 \tag{21}$$

where $D = g(x)R^{-1}g(x)^T > 0$ is bounded such that $D_{\min} \le \|D\| \le D_{\max}$ for known constants $D_{\min}$ and $D_{\max}$, and

$$\begin{aligned}\varepsilon_{HJB} &= \nabla_x \varepsilon^T (f(x) - \frac{1}{2} g(x) R^{-1} g(x)^T (\nabla_x^T \phi(x)\Theta + \nabla_x \varepsilon)) + \frac{1}{4}\nabla_x \varepsilon^T g(x) R^{-1} g(x)^T \nabla_x \varepsilon \\ &= \nabla_x \varepsilon^T (f(x) + g(x)u^*) + \frac{1}{4}\nabla_x \varepsilon^T D \nabla_x \varepsilon\end{aligned}$$

is the residual error due to the OLA reconstruction error. Asserting the bounds for the optimal closed loop dynamics (10) along with the boundedness of $g(x)$ and $\nabla_x \varepsilon$, the residual error ε_{HJB} is bounded above on a compact set according to $|\varepsilon_{HJB}| \le \varepsilon'_M \delta(x) + \varepsilon'^2_M D_{\max}$. In addition, it has been shown [25] that by increasing the dimension of the basis vector $\phi(x)$, the OLA reconstruction error decreases.

Moving on, the OLA estimate of (18) is now written as

$$\hat{V}(x) = \hat{\Theta}^T \phi(x) \tag{22}$$

where $\hat{\Theta}$ is the OLA estimate of the target parameter vector Θ. Similarly, the estimate of the optimal control (20) is written in terms of $\hat{\Theta}$ as

$$\hat{u}_1(x) = -\frac{1}{2} R^{-1} g(x)^T \nabla_x^T \phi(x) \hat{\Theta}. \tag{23}$$

In the development of this work, it will be shown that an initial stabilizing control is not required to implement the proposed SOLA-based scheme in contrast to [22] and [25] which require initial control policies to be stabilizing. Moreover, Lyapunov theory will show that the estimated optimal control input (23) approaches the real optimal control input (4) with small bounded error. As a result, the proposed online scheme is not required to provide an initially stabilizing control input whereas the proposed OLA parameter tuning law described next ensures that the system states remain bounded and that (23) will become admissible.

Now, using (22) and (23), the approximate Hamiltonian can be written as

$$\hat{H}(x, \hat{\Theta}) = Q(x) + \hat{\Theta}^T \nabla_x \phi(x) f(x) - \frac{1}{4} \hat{\Theta}^T \nabla_x \phi(x) D \nabla_x^T \phi(x) \hat{\Theta}. \tag{24}$$

Remark 1: Observing the definition of the OLA approximation of the cost function (22) and the Hamiltonian function (24), it is evident that both become zero when $\|x\| = 0$. Thus, once the system states have converged to zero, the cost function approximation can no longer be updated. This can be viewed as a persistency of excitation (PE) requirement for the inputs to the cost function OLA [25], [22]. That is, the system states must be persistently exiting long enough for the OLA to learn the optimal cost function.

Remark 2: The control of unknown continuous-time nonlinear systems has been undertaken by many researchers using OLA's [3] where objectives often include regulating the system states or tracking errors to zero while ensuring the OLA parameter estimates

remain bounded. In contrast, the objective of the proposed online optimal control scheme is to drive the OLA parameter estimates toward their ideal values while ensuring the system states or tracking errors remain bounded. Once the OLA parameter estimates have converged to their ideal values and the Hamiltonian (24) has converged to a small neighborhood around the origin, the requirement of $\|x\| > 0$ can be removed and the system states are allowed to converge zero.

Recalling the HJB equation shown in (5), the OLA estimate $\hat{\Theta}$ should be tuned to minimize $\hat{H}(x,\hat{\Theta})$. However, tuning to minimize $\hat{H}(x,\hat{\Theta})$ alone does not ensure the stability of the nonlinear system (1) during the OLA learning process. Therefore, the proposed OLA tuning algorithm is designed to minimize (24) while considering the stability of (1) and written as

$$\begin{aligned}\dot{\hat{\Theta}} = -\alpha_1 \frac{\hat{\sigma}}{(\hat{\sigma}^T\hat{\sigma}+1)^2}\left(Q(x)+\hat{\Theta}^T\nabla_x\phi(x)f(x)-\frac{1}{4}\hat{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\hat{\Theta}\right)\\ +\Sigma(x,\hat{u}_1)\frac{\alpha_2}{2}\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}(x)\end{aligned} \tag{25}$$

where $\hat{\sigma} = \nabla_x\phi f(x) - \nabla_x\phi(x)D\nabla_x^T\phi(x)\hat{\Theta}/2$, $\alpha_1 > 0$ and $\alpha_2 > 0$ are design constants, $J_{1x}(x)$ is described in *Lemma 1*, and the operator $\Sigma(x,\hat{u}_1)$ is given by

$$\Sigma(x,\hat{u}_1) = \begin{cases} 0 & \text{if } J_{1x}^{\mathrm{T}}(x)\dot{x} = J_{1x}^{\mathrm{T}}(x)(f(x)-g(x)R^{-1}g(x)^T\nabla_x^T\phi(x)\hat{\Theta}/2) < 0 \\ 1 & \text{otherwise}\end{cases}. \tag{26}$$

The first term in (25) is the portion of the tuning law which seeks to minimize (24) and was derived using a normalized gradient descent scheme with the auxiliary HJB error defined as

$$E_{HJB} = \frac{1}{2}\hat{H}(x,\hat{\Theta})^2. \tag{27}$$

Meanwhile, the second term in the OLA tuning law (25) is included to ensure the system states remain bounded while the SOLA scheme learns the optimal cost function. The form of the operator shown in (26) was selected based on the Lyapunov's sufficient condition for stability (i.e. if $J_1(x) > 0$ and $\dot{J}_1(x) = J_{1x}^T(x)\dot{x} < 0$, then the states x are stable). From the definition of the operator in (26), the second term in (25) is removed when the nonlinear system (1) exhibits stable behavior, and learning the HJB cost function becomes the primary objective of the OLA update (25). In contrast, when the system (1) exhibits signs of instability (i.e. $J_{1x}^T(x)\dot{x} \geq 0$), the second term of (25) is activated and tunes the OLA parameter estimates until the nonlinear system (1) exhibits stable behavior. This approach will be shown to render guaranteed performance in the following Lyapunov proof. In addition, the numerical examples presented in *Section V* will illustrate that system stability is lost and the OLA fails to learn the cost function if the second term in (25) is removed while including the second term renders satisfactory performance.

Remark 3: The first portion of the OLA tuning law $\dot{\hat{\Theta}}$ in (25) utilizes $(\hat{\sigma}^T\hat{\sigma}+1)^2$ instead of the traditional $(\hat{\sigma}^T\hat{\sigma}+1)$ used for normalization. This modification was also utilized in [25] for the critic update. However, the update (25) is different from the critic update proposed in [25] since two networks were utilized in [25] whereas only one network is used in this work.

Remark 4: For the case of $\Sigma(x,\hat{u}_1) = 1$, the update (25) is observed to have equilibrium points at $\hat{H}(x,\hat{\Theta}) = 0$ and $J_{1x}(x) = 0$. Thus, (25) is updated in order to minimize (27) as well as to drive the system states to zero (since $J_{1x}(x) = 0$ only when $x = 0$). However, the tuning law (25) cannot be implemented with $\Sigma(x,\hat{u}_1) = 1$ for all time because it

would prevent the optimal cost function from being learned since the OLA would continue to update even after $\hat{H}(x,\hat{\Theta})=0$. In addition, if stability were the only objective of the controller design, the tuning law (25) could be rewritten as $\dot{\hat{\Theta}}=\alpha_2\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}(x)/2$, and the stability of the system states and parameter vector $\hat{\Theta}$ could be shown using Lyapunov theory. For this case, the control law (23) would stabilize (1) but not in an optimal manner.

Moving on, we now form the dynamics of the OLA parameter estimation error $\tilde{\Theta}=\Theta-\hat{\Theta}$. Observing $Q(x)=-\Theta^T\nabla_x\phi(x)f(x)+\Theta^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\Theta/4-\varepsilon_{HJB}$ from (21), the approximate HJB equation (24) can be rewritten in terms of $\tilde{\Theta}$ as

$$\hat{H}(x,\hat{\Theta})=-\tilde{\Theta}^T\nabla_x\phi(x)f(x)+\frac{1}{2}\tilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\Theta-\frac{\tilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\tilde{\Theta}}{4}-\varepsilon_{HJB}. \quad (28)$$

Next, observing $\dot{\tilde{\Theta}}=-\dot{\hat{\Theta}}$ and $\hat{\sigma}=\nabla_x\phi(x)(\dot{x}_1^*+D\nabla_x\varepsilon/2)+\nabla_x\phi(x)D\nabla_x^T\phi(x)\tilde{\Theta}/2$ where $\dot{x}_1^*=f(x)+g(x)u_1^*$, the error dynamics of (25) are written as

$$\dot{\tilde{\Theta}}=-\frac{\alpha_1}{\rho^2}\left(\nabla_x\phi(x)\left(\dot{x}_1^*+\frac{D\nabla_x\varepsilon}{2}\right)+\frac{\nabla_x\phi(x)D\nabla_x^T\phi(x)\tilde{\Theta}}{2}\right)\left(\tilde{\Theta}^T\nabla_x\phi(x)\left(\dot{x}_1^*+\frac{D\nabla_x\varepsilon}{2}\right)+\frac{1}{4}\tilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\tilde{\Theta}+\varepsilon_{HJB}\right)$$
$$-\Sigma(x,\hat{u}_1)\frac{\alpha_2}{2}\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}(x) \quad (29)$$

where $\rho=(\hat{\sigma}^T\hat{\sigma}+1)$. Next, the stability of the SOLA-based adaptive scheme for optimal control is examined along with the stability of the nonlinear system (1). First, the following definition is required.

Definition 2 [3]: An equilibrium point x_e is said to be *uniformly ultimately bounded (UUB)* if there exists a compact set $S\subset\Re^n$ so that for all $x_0\in S$ there exists a bound B and a time $T(B,x_o)$ such that $\|x(t)-x_e\|\le B$ for all $t\ge t_0+T$.

Theorem 1: (SOLA-based scheme convergence to the HJB function and System Stability). Given the nonlinear system (1) with the target HJB equation (5), let the tuning law for the SOLA be given by (25). Then, there exists computable positive constants b_{Jx} and b_{Θ} such that the OLA approximation error $\widetilde{\Theta}$ and $\|J_{1x}(x)\|$ are *UUB* for all $t \geq t_0 + T$ with ultimate bounds given $\|J_{1x}(x)\| \leq b_{Jx}$ and $\left\|\widetilde{\Theta}\right\| \leq b_{\Theta}$. Further, under OLA reconstruction errors, $\left\|V^* - \hat{V}\right\| \leq \varepsilon_{r1}$ and $\left\|u_1^* - \hat{u}_1\right\| \leq \varepsilon_{r2}$ for small positive constants ε_{r1} and ε_{r2}, respectively.

Proof: Consider the following positive definite Lyapunov candidate

$$J_{HJB} = \alpha_2 J_1(x) + \frac{1}{2}\widetilde{\Theta}^T\widetilde{\Theta} \tag{30}$$

whose first derivate with respect to time is given by

$$\dot{J}_{HJB} = \alpha_2 J_{1x}^T(x)\dot{x} + \widetilde{\Theta}^T\dot{\widetilde{\Theta}} \tag{31}$$

where $J_1(x)$ and $J_{1x}(x)$ are given in *Lemma 1*. To begin, observe that if $\|x\| = 0$, then $J_{HJB}(x) = \widetilde{\Theta}^T\widetilde{\Theta}/2$ with $\dot{J}_{HJB}(x) = 0$, and the parameter estimation error $\left\|\widetilde{\Theta}\right\|$ remains constant and bounded [3]. On the other hand, to successfully accomplish the online learned objective, the system states are required to satisfy $\|x\| > 0$ as described in *Remark 1*. Therefore, the remainder of this proof considers the case of $\|x\| > 0$ (i.e. online learning is being performed). Then, substituting the nonlinear dynamics (1) with control input (23) applied along with the OLA estimation error dynamics (29) into (31) reveals

$$\dot{J}_{HJB} = \alpha_2 J_{1x}^T(x)\left(f(x) - \frac{1}{2}g(x)R^{-1}g(x)^T\nabla_x^T\phi(x)\hat{\Theta}\right) - \frac{\alpha_1}{\rho^2}\left(\widetilde{\Theta}^T\nabla_x\phi(x)\left(\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right)\right)^2$$
$$-\frac{\alpha_1}{8\rho^2}\left(\widetilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\widetilde{\Theta}\right)^2 - \frac{3\alpha_1}{4\rho^2}\widetilde{\Theta}^T\nabla_x\phi(x)\left(\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right)\widetilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\widetilde{\Theta}$$
$$-\frac{\alpha_1}{\rho^2}\widetilde{\Theta}^T\nabla_x\phi(x)\left(\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right)\varepsilon_{HJB} - \frac{\alpha_1}{2\rho^2}\widetilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\widetilde{\Theta}\varepsilon_{HJB}$$
$$-\Sigma(x,\hat{u}_1)\frac{\alpha_2}{2}\widetilde{\Theta}^T\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}^T(x).$$

Next, completing the squares with respect to $\widetilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\widetilde{\Theta}$ and $\widetilde{\Theta}^T\nabla_x\phi(x)\dot{x}_1^* + D\nabla_x\varepsilon/2$ and taking the upper bound yields

$$\dot{J}_{HJB} \le \alpha_2 J_{1x}^T(x)\left(f(x) - \frac{1}{2}g(x)R^{-1}g(x)^T\nabla_x^T\phi(x)\hat{\Theta}\right) + \frac{4\alpha_1}{\rho^2}\left(\widetilde{\Theta}^T\nabla_x\phi(x)\left(\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right)\right)^2$$
$$-\frac{\alpha_1}{32\rho^2}\left(\widetilde{\Theta}^T\nabla_x\phi(x)D\nabla_x^T\phi(x)\widetilde{\Theta}\right)^2 + \frac{3\alpha_1}{2\rho^2}\varepsilon_{HJB}^2 - \Sigma(x,\hat{u}_1)\frac{\alpha_2}{2}\widetilde{\Theta}^T\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}^T(x)$$
$$\le \alpha_2 J_{1x}^T(x)\left(f(x) - \frac{1}{2}g(x)R^{-1}g(x)^T\nabla_x^T\phi(x)\hat{\Theta}\right) - \Sigma(x,\hat{u}_1)\frac{\alpha_2}{2}\widetilde{\Theta}^T\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}^T(x)$$
$$-\frac{\alpha_1}{32\rho^2}\left\|\widetilde{\Theta}^T\nabla_x\phi(x)\right\|^4 D_{\min}^2 + \frac{4\alpha_1}{\rho^2}\left\|\widetilde{\Theta}^T\nabla_x\phi(x)\right\|^2\left\|\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right\|^2 + \frac{3\alpha_1}{2\rho^2}\varepsilon_{HJB}^2.$$

Now, completing the square with respect $\left\|\widetilde{\Theta}^T\nabla_x\phi(x)\right\|^2$ renders

$$\dot{J}_{HJB} \le \alpha_2 J_{1x}^T(x)\left(f(x) - \frac{1}{2}g(x)R^{-1}g(x)^T\nabla_x^T\phi(x)\hat{\Theta}\right) - \Sigma(x,\hat{u}_1)\frac{\alpha_2}{2}\widetilde{\Theta}^T\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}^T(x)$$
$$-\frac{\alpha_1}{64\rho^2}\left\|\widetilde{\Theta}^T\nabla_x\phi(x)\right\|^4 D_{\min}^2 - \frac{\alpha_1 D_{\min}^2}{\rho^2}\left(\frac{\left\|\widetilde{\Theta}^T\nabla_x\phi(x)\right\|^2}{8} - \frac{16}{D_{\min}^2}\left\|\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right\|^2\right)^2$$
$$+\frac{\alpha_1 256}{\rho^2 D_{\min}^2}\left\|\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right\|^4 + \frac{3\alpha_1}{2\rho^2}\varepsilon_{HJB}^2$$
$$\le \alpha_2 J_{1x}^T(x)\left(f(x) - \frac{1}{2}g(x)R^{-1}g(x)^T\nabla_x^T\phi(x)\hat{\Theta}\right) - \Sigma(x,\hat{u}_1)\frac{\alpha_2}{2}\widetilde{\Theta}^T\nabla_x\phi(x)g(x)R^{-1}g(x)^T J_{1x}^T(x)$$
$$-\frac{\alpha_1}{64\rho^2}\left\|\widetilde{\Theta}^T\nabla_x\phi(x)\right\|^4 D_{\min}^2 + \frac{\alpha_1 256}{\rho^2 D_{\min}^2}\left\|\dot{x}_1^* + \frac{D\nabla_x\varepsilon}{2}\right\|^4 + \frac{3\alpha_1}{2\rho^2}\varepsilon_{HJB}^2.$$

Next, observing the bound in (10) and applying the Cauchy-Schwarz inequality, $\dot{J}_{HJB}$ is upper bounded according to

$$\dot{J}_{HJB} \le \alpha_2 J_{1x}^T(x)\left(f(x) - \frac{1}{2} g(x) R^{-1} g(x)^T \nabla_x^T \phi(x) \hat{\Theta} \right) - \Sigma(x, \hat{u}_1) \frac{\alpha_2}{2} \tilde{\Theta}^T \nabla_x \phi(x) g(x) R^{-1} g(x)^T J_{1x}^T(x) - \frac{\alpha_1}{\rho^2} \left\| \tilde{\Theta} \right\|^4 \beta_1 + \frac{\alpha_1}{\rho^2} \eta(\varepsilon) + \frac{\alpha_1}{\rho^2} \beta_2 \delta^4(x) \tag{32}$$

with $\beta_1 = \nabla \phi_{\min}^4 D_{\min}^2 / 64$, $\beta_2 = 1024 / D_{\min}^2 + 3/2$, and $\eta(\varepsilon) = 64 / D_{\min}^2 + 3(\varepsilon_M'^4 + \varepsilon_M'^4 D_{\max}^2)/2$, and $0 < \nabla \phi_{\min} \le \|\nabla \phi(x)\|$ is ensured by $\|x\| > 0$ for a constant $\nabla \phi_{\min}$.

Now, the cases of $\Sigma(x, \hat{u}_1) = 0$ and $\Sigma(x, \hat{u}_1) = 1$ will be considered. First, for $\Sigma(x, \hat{u}_1) = 0$, the first term in (32) is less than zero by the definition of the operator in (26). Recalling $\delta(x) \equiv \sqrt[4]{K^* \|J_{1x}\|}$ and observing $\left\| 1/\rho^2 \right\| \le 1$, (32) is rewritten as

$$\dot{J}_{HJB} \le -\left(\alpha_2 \dot{x}_{\min} - \alpha_1 \beta_2 K^*\right) \|J_{1x}(x)\| - \frac{\alpha_1}{\rho^2} \left\| \tilde{\Theta} \right\|^4 \beta_1 + \frac{\alpha_1}{\rho^2} \eta(\varepsilon), \tag{33}$$

and (33) is less than zero provided $\alpha_2 / \alpha_1 > \beta_2 K^* / \dot{x}_{\min}$ and the following inequalities hold

$$\|J_{1x}(x)\| > \frac{\alpha_1 \eta(\varepsilon)}{\left(\alpha_2 \dot{x}_{\min} - \alpha_1 \beta_2 K^*\right)} \equiv b_{Jx0} \qquad \text{or} \qquad \left\| \tilde{\Theta} \right\| > \sqrt[4]{\frac{\eta(\varepsilon)}{\beta_1}} \equiv b_{\Theta 0}. \tag{34}$$

Note that $\|x\| > 0$ and the operator (26) ensure the existence of a constant $\dot{x}_{\min}$ satisfying $0 < \dot{x}_{\min} < \|\dot{x}\|$. According to standard Lyapunov extensions [3], the inequalities above guarantee that $\dot{J}_{HJB}$ is less than zero outside of a compact set. Thus, $\|J_{1x}(x)\|$ as well as the OLA parameter estimation error $\left\| \tilde{\Theta} \right\|$ remain bounded for the case $\Sigma(x, \hat{u}_1) = 0$. Recalling the Lyapunov candidate $J_1(x)$ is radially unbounded and continuously differentiable (*Lemma 1)*, the boundedness of $\|J_{1x}(x)\|$ implies the boundedness of the system states, $\|x\|$. Next, we consider the case of $\Sigma(x, \hat{u}_1) = 1$ which implies the OLA based input (23) may not stabilizing. To begin, add and subtract $\alpha_2 J_{1x}^T(x) D(\nabla_x^T \phi(x) \Theta + \nabla_x \varepsilon)/2$ to (32) to get

$$\dot{J}_{HJB} \le \alpha_2 J_{1x}^T(x)\left(f(x) - \frac{1}{2}D(\nabla_x^T \phi(x)\Theta + \nabla_x \varepsilon)\right) + \frac{\alpha_2}{2} J_{1x}^T(x) D \nabla_x \varepsilon - \frac{\alpha_1}{\rho^2}\left\|\widetilde{\Theta}^T\right\|^4 \beta_1 + \frac{\alpha_1}{\rho^2}\eta(\varepsilon) + \frac{\alpha_1}{\rho^2}\beta_2 \delta^4(x)$$
$$= \alpha_2 J_{1x}^T(x)(f(x) + g(x)u^*) + \frac{\alpha_2}{2} J_{1x}^T(x) D \nabla_x \varepsilon - \frac{\alpha_1}{\rho^2}\left\|\widetilde{\Theta}\right\|^4 \beta_1 + \frac{\alpha_1}{\rho^2}\eta(\varepsilon) + \frac{\alpha_1}{\rho^2}\beta_2 K^* \left\|J_{1x}\right\|.$$

Next, using *Lemma 2* and recalling the boundedness of D, $\dot{J}_{HJB}$ is rewritten as

$$\dot{J}_{HJB} \le -\alpha_2 \overline{Q}_{\min}\left\|J_{1x}(x)\right\|^2 + \alpha_2\left(\frac{D_{\max}\varepsilon'_M}{2} + \frac{\alpha_1 \beta_2 K^*}{\alpha_2 \rho^2}\right)\left\|J_{1x}(x)\right\| - \frac{\alpha_1}{\rho^2}\left\|\widetilde{\Theta}\right\|^4 \beta_1 + \frac{\alpha_1}{\rho^2}\eta(\varepsilon)$$

where $\overline{Q}_{Min} > 0$ satisfies $\overline{Q}_{Min} \le \|Q(x)\|$ and is ensured by the condition $\|x\| > 0$. As a final step, complete the square with respect to $\left\|J_{1x}(x)\right\|^2$ to reveal

$$\dot{J}_{HJB} \le -\frac{\alpha_2 \overline{Q}_{\min}}{2}\left\|J_{1x}(x)\right\|^2 - \frac{\alpha_1}{\rho^2}\left\|\widetilde{\Theta}\right\|^4 \beta_1 + \frac{\alpha_1}{\rho^2}\eta(\varepsilon) + \frac{\alpha_2 D_{\max}^2 \varepsilon'^2_M}{4\overline{Q}_{\min}} + \frac{\alpha_1^2 \beta_2^2 K^{*2}}{\alpha_2 \rho^4 \overline{Q}_{\min}}, \tag{35}$$

and $\dot{J}_{HJB} < 0$ provided both of the following inequalities hold

$$\left\|J_{1x}(x)\right\| > \sqrt{\frac{D_{\max}^2 \varepsilon'^2_M}{2\overline{Q}_{\min}^2}} \equiv b'_{Jx1} \quad \text{and} \quad \left\|\widetilde{\Theta}\right\| > \sqrt[4]{\frac{\eta(\varepsilon)}{\beta_1} + \frac{\alpha_1 \beta_2^2 K^{*2}}{\beta_1 \alpha_2 \overline{Q}_{\min}}} \equiv b_{\Theta 1}. \tag{36}$$

According to standard Lyapunov extensions [3], the inequalities in (36) guarantee that $\dot{J}_{HJB}$ is less than zero outside of a compact set. Thus, $\left\|J_{1x}(x)\right\|$ as well as the OLA parameter estimation error estimation error $\left\|\widetilde{\Theta}\right\|$ remain bounded for the case $\Sigma(x, \hat{u}_1) = 1$. Recalling the Lyapunov candidate $J_1(x)$ is a radially unbounded and continuously differentiable (*Lemma 1)*, the boundedness of $\left\|J_{1x}(x)\right\|$ implies the boundedness of the system states, $\|x\|$. The overall bounds for the cases $\Sigma(x, \hat{u}_1) = 0$ and $\Sigma(x, \hat{u}_1) = 1$ are then given by $\|J_{1x}(x)\| \le b_{Jx}$ and $\left\|\widetilde{\Theta}\right\| \le b_\Theta$ for computable positive constants $b_{Jx} = \max(b_{Jx0}, b_{Jx1})$ and $b_\Theta = \max(b_{\Theta 0}, b_{\Theta 1})$. Note that b_{Jx0} and $b_{\Theta 1}$ in (34) and (36), respectively, can be reduced through appropriate selection of α_1 and α_2.

To complete the proof, subtract (22) from (18) and (23) from (20) to reveal

$$V^*(x)-\hat{V}(x)=\tilde{\Theta}^T\phi(x)+\varepsilon(x)$$
$$u_1^*(x)-\hat{u}_1(x)=-\frac{1}{2}R^{-1}g(x)^T\nabla_x^T\phi(x)\tilde{\Theta}-\frac{1}{2}R^{-1}g(x)^T\nabla_x\varepsilon(x)$$

Next, observing that the boundedness of the system states ensures the existence of positive constants ϕ_M and ϕ'_M such that $\|\phi\|\le\phi_M$ and $\|\nabla_x\phi\|\le\phi'_M$, respectively, and taking norm and the limit as $t\to\infty$ when $\Sigma(x,\hat{u}_1)=0$ reveals

$$\begin{aligned}\left\|V^*(x)-\hat{V}(x)\right\| &\le\left\|\tilde{\Theta}\right\|\|\phi(x)\|+\varepsilon_M\le b_\Theta\phi_M+\varepsilon_M\equiv\varepsilon_{r1}\\ \|u_1^*(x)-\hat{u}_1(x)\| &\le\frac{1}{2}\lambda_{\max}(R^{-1})g_M\left\|\nabla_x^T\phi(x)\right\|\left\|\tilde{\Theta}\right\|+\frac{1}{2}\lambda_{\max}(R^{-1})g_M\varepsilon'_M\\ &\le\frac{1}{2}\lambda_{\max}(R^{-1})g_Mb_\Theta\phi'_M+\frac{1}{2}\lambda_{\max}(R^{-1})g_M\varepsilon'_M\equiv\varepsilon_{r2}.\end{aligned}$$

■

Remark 5: For the case of $\Sigma(x,\hat{u}_1)=0$, the bounds in (34) reveal that $\dot{J}_{HJB}<0$ for $\left\|\tilde{\Theta}\right\|>b_{\Theta0}$ or $\|J_{x1}(x)\|>b_{x0}$. However, recalling the requirement of $\|x\|>0$ described in *Remark 1*, the system states are required to be bounded away from zero in order to learn the HJB equation. Thus, the condition of $\|J_{x1}(x)\|>b_{x0}$ can be satisfied through the condition $\|x\|>0$, $\left\|\tilde{\Theta}\right\|$ can become arbitrarily small, and $\dot{J}_{HJB}<0$ is still satisfied. In contrast, for the case of $\Sigma(x,\hat{u}_1)=1$, the bounds of (36) reveal that $\dot{J}_{HJB}<0$ provided $\|J_{x1}(x)\|>b_{Jx1}$ and $\left\|\tilde{\Theta}\right\|>b_{\Theta1}$. Similarly to the case of $\Sigma(x,\hat{u}_1)=0$, the inequality $\|J_{x1}(x)\|>b_{Jx1}$ can be satisfied through the requirement of $\|x\|>0$. In addition, the inequality $\left\|\tilde{\Theta}\right\|>b_{\Theta1}$ is not surprising since $\Sigma(x,\hat{u}_1)=1$ implies $\hat{\Theta}$ has not provided a stabilizing control input for the nonlinear system (1), and it is known the target OLA parameter Θ provides a stabilizing control input for (1). Thus, when $\hat{\Theta}$ does not provide a stabilizing control, one would expect $\left\|\tilde{\Theta}\right\|$ to be

bounded away from zero. This relationship is also depicted in Fig. 1. The actual convergence region is a subset of the regions determined by (34) and (36), respectively, [3].

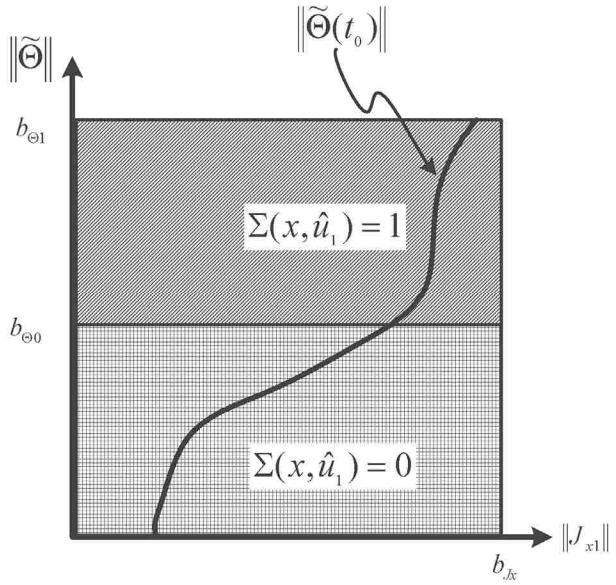


Fig. 1. Graphical representation of the convergence region.

Remark 6: The results of *Theorem 1* indicate that the system states and OLA parameter estimation errors are *UUB* even when $\hat{\Theta}$ does not provide a stabilizing control input. This result implies that an initial stabilizing control is not required for implementation of the proposed SOLA design. Further, *Theorem 1* illustrates that the estimated control input (23) approaches the target optimal control input (4) with small bounded error as $t \to \infty$. As a result, the OLA tuning law (25) ensures that the system states remain bounded and that (23) will become admissible during the online learning if it is not initially stabilizing. The simulation results in *Section V* also support this claim.

Next, the stability of the SOLA-based optimal control scheme is examined when there are no OLA reconstruction errors as would be the case when standard adaptive control techniques [2] are utilized. In other words, when a NN is replaced with a standard linear in the unknown parameter (LIP) adaptive control, the parameter estimation errors and the states are globally asymptotically stable according to Corollary 1.

Corollary 1: (Ideal SOLA-based Optimal Control Scheme Convergence). Let the hypothesis of *Theorem 1* hold in the absence of OLA reconstruction errors. Then, the OLA approximation error $\widetilde{\Theta}$ and system states x are globally asymptotically stable (GAS) and $\hat{V} \to V^*$ and $\hat{u}_1 \to u_1^*$.

Proof: Consider the Lyapunov candidate (30) whose first derivative is found using similar methods as those described in (32)-(33) with $\varepsilon = \varepsilon_{HJB} = 0$ and $\nabla_x \varepsilon = 0$. For the case of $\Sigma(x, \hat{u}_1) = 0$, the first derivative of (30) is written similarly to (33) with $\eta(\varepsilon) = 0$. Therefore, $\dot{J}_{HJB}$ is less than zero provided $\alpha_2 / \alpha_1 > \beta_2 K^* / \dot{x}_{\min}$, and $\left\| J_{1x}(x) \right\|$ as well as the OLA parameter estimation error $\left\| \widetilde{\Theta} \right\|$ converge to zero asymptotically.

Next, when $\Sigma(x, \hat{u}_1) = 1$, $\dot{J}_{HJB}$ is upper bounded similarly to (35) with $\eta(\varepsilon) = \varepsilon'_M = 0$, and $\dot{J}_{HJB} < 0$ provided the following inequalities hold

$$\left\| J_{1x}(x) \right\| > \sqrt{\frac{\alpha_1 \beta_2^2 K^{*2}}{\beta_1 \alpha_2 \overline{Q}_{\min}}} \quad \text{or} \quad \left\| \widetilde{\Theta} \right\| > \sqrt[4]{\frac{\alpha_1 \beta_2^2 K^{*2}}{\beta_1 \alpha_2 \overline{Q}_{\min}}}.$$

Note that the above bounds can be made arbitrarily small through proper selection of α_1 and α_2. As in *Theorem 1,* it is not surprising that $\left\| \widetilde{\Theta} \right\| > 0$ for $\Sigma(x, \hat{u}_1) = 1$ since the case of $\Sigma(x, \hat{u}_1) = 1$ implies $\hat{\Theta}$ has not stabilized (1). Therefore, when $\Sigma(x, \hat{u}_1) = 1$, the secondary tuning algorithm in (25) is activated until $J_{1x}^{\mathrm{T}}(x)\dot{x} < 0$. Then, once the system is stabilized,

the case of $\Sigma(x,\hat{u}_1)=0$ applies and asymptotic convergence of the parameter estimation error is observed. Since (30) is radially unbounded, the result holds globally. To complete the proof, observe that when $\varepsilon=0$ and $\nabla_x\varepsilon=0$, the cost function and control error bounds from *Theorem 1*, ε_{r1} and ε_{r2}, respectively, also become zero. Thus, using similar methods as those used in *Theorem 1* shows that $\hat{V}\to V^*$ and $\hat{u}_1\to u_1^*$ when $\Sigma(x,\hat{u}_1)=0$.

In the following section, we extend the SOLA-based design scheme to include the HJI used in H_∞ optimal control.

B. SOLA-based Scheme to Learn the HJI Function

To begin the HJI SOLA-based scheme development, we assume that the cost function (12) can be represented as

$$V_d(x)=\Phi^T\vartheta(x)+\varepsilon_d(x) \tag{37}$$

where $\Phi\in\Re^{Ld}$ is the target OLA vector, $\vartheta(x):\Re^n\to\Re^{Ld}$ is a linearly independent basis vector, and $\varepsilon_d(x)$ is the OLA reconstruction error while the target OLA vector and OLA reconstruction error are considered to be upper bounded according to $\|\Phi_d\|\le\Phi_{dM}$ and $\|\varepsilon_d(x)\|\le\varepsilon_{dM}$, respectively [3]. In addition, it will be assumed that the gradient of the OLA reconstruction error with respect to x is upper bounded according to $\|\partial\varepsilon_d(x)/\partial x\|=\|\nabla_x\varepsilon_d(x)\|\le\varepsilon'_{dM}$ [33].

The gradient of the HJI SOLA-based cost function (37) can be written similarly to the gradient of the HJB cost function shown in (19). Now, using the gradient of the HJI SOLA cost function, the optimal control and disturbance (14) and HJI function (15) are rewritten as

$$\left.\begin{aligned} u_2 *(x) &= -\frac{1}{2} R^{-1} g(x)^T (\nabla_x^T \vartheta(x)\Phi + \nabla_x \varepsilon_d(x)) \\ d *(x) &= \frac{1}{2\gamma^2} P^{-1} k(x)^T (\nabla_x^T \vartheta(x)\Phi + \nabla_x \varepsilon_d(x)) \end{aligned}\right\} \tag{38}$$

and

$$H_d^*(x,\Phi) = Q(x) + \Phi^T \nabla_x \vartheta(x) f(x) - \frac{1}{4}\Phi^T \nabla_x \vartheta(x) D_d \nabla_x^T \vartheta(x)\Phi + \varepsilon_{HJI} = 0 \tag{39}$$

where $D_d = g(x)R^{-1}g(x)^T - k(x)P^{-1}k(x)^T/\gamma^2$ and

$$\varepsilon_{HJI} = \nabla_x \varepsilon_d^T (f(x) + g(x)u* + k(x)d*) + \frac{1}{4}\nabla_x \varepsilon_d^T D_d \nabla_x \varepsilon_d(x)$$

is the HJI residual error due to the OLA reconstruction error. Additionally, it is required that γ be selected such that $D_d > 0$. It is assumed that the optimal closed loop dynamics for the HJI problem satisfy a bound defined similar to (10). As a result, the residual error ε_{HJI} is upper bounded on a compact set similarly to ε_{HJB}.

Moving on, the HJI SOLA-based estimate of (37) is now written as

$$\hat{V}_d(x) = \hat{\Phi}^T \vartheta(x) \tag{40}$$

where $\hat{\Phi}$ is the OLA estimate of the target parameter vector Φ. Similarly, the estimate of the optimal control and worst case disturbance (38) is written in terms of $\hat{\Phi}$ as

$$\left.\begin{aligned} \hat{u}_2(x) &= -\frac{1}{2} R^{-1} g(x)^T \nabla_x^T \vartheta(x)\hat{\Phi} \\ \hat{d}(x) &= \frac{1}{2\gamma^2} P^{-1} k(x)^T \nabla_x^T \vartheta(x)\hat{\Phi} \end{aligned}\right\}. \tag{41}$$

Similarly to $\hat{u}_1(x)$ and the HJB equation, $\hat{u}_2(x(t_0))$ is not required to be initially stabilizing for the HJI problem. Now, using (40) and (41), the approximate HJI Hamiltonian can be written as

$$\hat{H}_d(x,\hat{\Phi}) = Q(x) + \hat{\Phi}^T \nabla_x \phi(x) f(x) - \frac{1}{4}\hat{\Phi}^T \nabla_x \vartheta(x) D_d \nabla_x^T \vartheta(x)\hat{\Phi}. \tag{42}$$

Similarly to the HJB SOLA-based optimal controller design, the tuning law for the HJI-SOLA seeks to minimize the approximate Hamiltonian (42) while ensure the nonlinear system (11) remains stable. The portion of the tuning law which seeks to minimize (42) is derived from normalized gradient descent using the auxiliary HJI error $E_{HJI} = \hat{H}_d(x,\hat{\Phi})^2/2$ while the stabilizing portion of the tuning law is derived from Lyapunov theory. Observing the similarities of (42) and (24), the tuning law for the SOLA to solve the HJI problem is found to be

$$\begin{aligned}\dot{\hat{\Phi}} = &-\frac{\alpha_3\hat{\sigma}_2}{(\hat{\sigma}_2^T\hat{\sigma}_2+1)^2}\left(Q(x)+\hat{\Phi}^T\nabla_x\vartheta(x)f(x)-\frac{1}{4}\hat{\Phi}^T\nabla_x\vartheta(x)D_d\nabla_x^T\vartheta(x)\hat{\Phi}\right)\\ &+\frac{\Sigma_2(x,\hat{u}_2,\hat{d})\alpha_4}{2}\nabla_x\vartheta(x)D_d x\end{aligned} \tag{43}$$

where $\hat{\sigma}_2 = \nabla_x\vartheta f(x) - \hat{\Phi}^T\nabla_x\vartheta(x)D_d\nabla_x^T\vartheta(x)\hat{\Phi}/2$, α_3 and α_4 are positive a design constants, and where

$$\Sigma_2(x,\hat{u}_2,\hat{d}) = \begin{cases}0 & \text{if } J_{x2}^T(x)(f(x)+g(x)\hat{u}+k(x)\hat{d})<0\\ 1 & \text{otherwise}\end{cases} \tag{44}$$

where $J_{x2}(x)$ is the partial derivative of a continuously differentiable, radially unbounded Lyapunov candidate $J_2(x)$ for the nonlinear system (11) which satisfies similar properties as those described in *Lemma 1* for $J_1(x)$.

The region for $\Sigma_2(x,\hat{u}_2,\hat{d})=0$ was determined based on *Lemma 2*. Noting the similarities between (43) and (25), the OLA estimation error dynamics $\dot{\tilde{\Phi}} = \dot{\Phi} - \dot{\hat{\Phi}} = -\dot{\hat{\Phi}}$ can be written as

$$\begin{aligned}\dot{\tilde{\Phi}} = &-\frac{\alpha_3}{\rho_2^2}\left(\nabla_x\vartheta(x)\left(\dot{x}_2^* + \frac{D_d\nabla_x\varepsilon_d}{2}\right)+\frac{\nabla_x\vartheta(x)D_d\nabla_x^T\vartheta(x)\tilde{\Phi}}{2}\right)\left(\tilde{\Phi}^T\nabla_x\vartheta(x)\left(\dot{x}_2^* + \frac{D_d\nabla_x\varepsilon_d}{2}\right)+\frac{\tilde{\Phi}^T\nabla_x\vartheta(x)D_d\nabla_x^T\vartheta(x)\tilde{\Phi}}{4}+\varepsilon_{HJI}\right)\\ &-\Sigma_2(x,\hat{u}_2,\hat{d})\frac{\alpha_4}{2}\nabla_x\vartheta(x)D_d J_{x2}.\end{aligned} \tag{45}$$

where $\dot{x}_2^* = f(x) + g(x)u_2^* + k(x)d^*$ and $\rho_2 = \hat{\vartheta}^T\hat{\vartheta} + 1$. Now the following theorem can be stated.

Theorem 2: (SOLA-based Control Scheme Convergence to the HJI function and System Stability). Given the nonlinear system (11) with the target HJI equation (15), let the tuning law for the OLA be given by (43). Then, there exists computable positive constants B_{Jx} and b_Φ such that the OLA approximation error $\tilde{\Phi}$ and $\|J_{x2}(x)\|$ are *UUB* for all $t \geq t_0 + T$ with ultimate bounds given $\|J_{x2}(x)\| \leq B_{Jx}$ and $\|\tilde{\Phi}\| \leq b_\Phi$. Further, in the presence of OLA reconstruction errors, $\|\hat{V}_d - V_d{}^*\| \leq \varepsilon_{r3}$, $\|\hat{u}_2 - u_2{}^*\| \leq \varepsilon_{r4}$, and $\|\hat{d} - d^*\| \leq \varepsilon_{r5}$ for small positive constants ε_{r3}, ε_{r4} and ε_{r5} .

Proof: Consider the positive definite Lyapunov candidate

$$J_{HJI} = \frac{\alpha_4}{2} J_2(x) + \frac{1}{2}\tilde{\Phi}^T\tilde{\Phi} \tag{46}$$

whose first derivative is given by $\dot{J}_{HJI} = \alpha_4 J_{x2}^T(x)\dot{x} + \tilde{\Phi}^T\dot{\tilde{\Phi}}$.

Noting the similarities between the OLA error dynamics (45) and (29), proof of *Theorem 2* is shown using identical steps used to prove *Theorem 1.* In addition, the global asymptotic convergence of the OLA estimation errors and system states can be demonstrated for the HJI problem just as was shown in *Corollary 1* for the HJB optimal control problem. ■

A block diagram of the SOLA-based design is now presented in Fig. 1 where the HJI design is shown. The block diagram becomes the HJB optimal control design by taking $k(x) = 0$ and after appropriate modifications to reflect $\phi(x), \hat{\Theta}, \hat{V}(x)$, and $\hat{u}_1$, respectively.

We have just shown how a SOLA framework can be designed to solve the HJB and HJI optimal control problems. In the following section, the SOLA-based design will be

extended to include the tracking problem by effectively converting the tracking problem into a regulation problem.

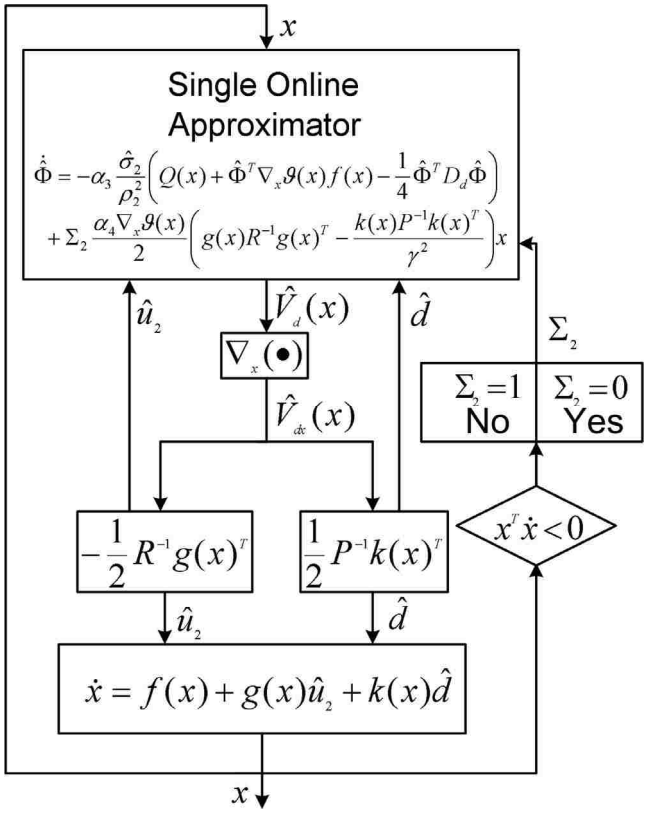


Fig. 2. SOLA for HJI regulation design.

IV. Continuous Time SOLA-based Scheme for Near Optimal Tracking

The optimal tracking control problem will be considered as an extension of the contributions presented in the previous section. The following optimal tracking development will consider the nonlinear system (11) and the HJI optimal tracking problem; however, the resulting theoretical results are easily extended to solve the HJB optimal tracking problem by taking $k(x) = 0$ in (11). Additionally, in this section it is assumed that there exists a matrix

$g(x)^I \in \Re^{mxn}$ such that $g(x)g(x)^I = I \in \Re^{nxn}$ where I is the identity matrix. Note that when $n = m$, $g(x)^I = g(x)^{-1}$.

The objective for the infinite time optimal tracking problem is to design the optimal control u_2 * to ensure that the nonlinear system (11) tracks a desired trajectory $x_d(t)$ in an optimal manner in the presence of the worst case disturbance d *. To achieve our objective, the cost function (12) must be modified accordingly to ensure it remains finite. To begin the development, define the desired trajectory to be [18]

$$\dot{x}_d = f(x_d) + g(x)u_d(x_d) \tag{47}$$

where $f(x_d)$ is the internal dynamics of the nonlinear system (11) rewritten in terms of the desired state x_d, $g(x)$ is the same input transformation matrix in (11), and $u_d(x_d)$ is the control input to the desired system. Next, define the state tracking error as

$$e = x - x_d, \tag{48}$$

and using (11) and (47), the tracking error dynamics of (48) are

$$\dot{e} = f(x) + g(x)u + k(x)d - \dot{x}_d = f_e(e) + g(k)u_e + k(x)d \tag{49}$$

where $f_e(e) = f(x) - f(x_d)$ and

$$u_e = u - u_d. \tag{50}$$

In order to control (49) in an optimal manner, it is required to select the control policy u_e that minimizes the infinite horizon HJI cost function [29]

$$V_T(e(t)) = \int_t^\infty \left(r_e(e(\tau), u_e(\tau)) - \gamma_e^2 d(\tau)^T P_e d(\tau)\right) d\tau \tag{51}$$

where $P_e \in \Re^{wxw}$ is a constant positive definite matrix, $\gamma_e > 0$ is a constant, $r_e(e(\tau), u_e(\tau))$ is defined similarly to $r(x(\tau), u_1(\tau))$ with $x(\tau)$ and $u_1(\tau)$ replaced with $e(\tau)$ and $u_e(\tau)$, respectively,

and where u_e is required to be admissible. The Hamiltonian for the HJI tracking problem is now written as

$$H_T(e,u_e,d) = r_e(e,u_e) - \gamma_e^2 d^T P_e d + V_{Te}^T(e)(f_e(e) + g(x)u_e + k(x)d) \tag{52}$$

where $V_{Te}(e)$ is the gradient of the $V_T(e)$ with respect to e.

Now applying the stationary conditions $\partial H_T(e,u_e,d)/\partial u_e = 0$ and $\partial H_T(e,u_e,d)/\partial d = 0$ reveals the optimal control and disturbance for the tracking problem to be

$$\left.\begin{aligned} u_e*(x) &= -\frac{1}{2}R^{-1}g(x)^T V_{Te}^*(e) \\ d*(x) &= \frac{1}{2\gamma_e^2}P^{-1}k(x)^T V_{Te}^*(e) \end{aligned}\right\}. \tag{53}$$

Now, substituting (53) into (52) reveals the HJI equation for the tracking problem as

$$0 = Q_e(e) + V_{Te}^{*T}(e)f_e(e) - \frac{1}{4}V_{Te}^{*T}(e)g(x)R_e^{-1}g(x)^T V_{Te}^*(e) + \frac{1}{4\gamma_e^2}V_{Te}^{*T}(e)k(x)P_e^{-1}k(x)^T V_{Te}^*(e) \tag{54}$$

with $V_T^*(0) = 0$ where $Q_e(e)$ and R_e are defined similarly to $Q(x)$ and R presented in *Section III*, respectively. Next, we observe that the optimal control input in (53) can be rewritten as

$$u^* = u_d - \frac{1}{2}R^{-1}g(x)^T V_{Te}^*(e). \tag{55}$$

Note that the expression for the desired control input u_d is obtained from (47) and written as

$$u_d(x_d) = g(x)^I(\dot{x}_d - f(x_d)). \tag{56}$$

It is observed that the optimal control input (55) consists of a predetermined feedforward term, u_d, and an optimal feedback term that is a function of the gradient of the optimal cost function (51). Thus, to implement the optimal control (55), the SOLA based

control scheme designed in *Section III-B* is utilized to learn the optimal feedback tracking control term after appropriate modifications to reflect (51)-(53).

Further, the *Theorem 2* results are applicable to the HJI optimal tracking control problem since the cost function (51) effectively converts the tracking control problem into a regulation problem [22],[29]. Moreover, by taking $k(x) = 0$, the previous development becomes the HJB optimal tracking problem, and the theoretical results of *Section III-A* and *Theorem 1* derived for the HJB problem are utilized to learn the optimal feedback tracking control term after appropriate modifications to reflect (51)-(30).

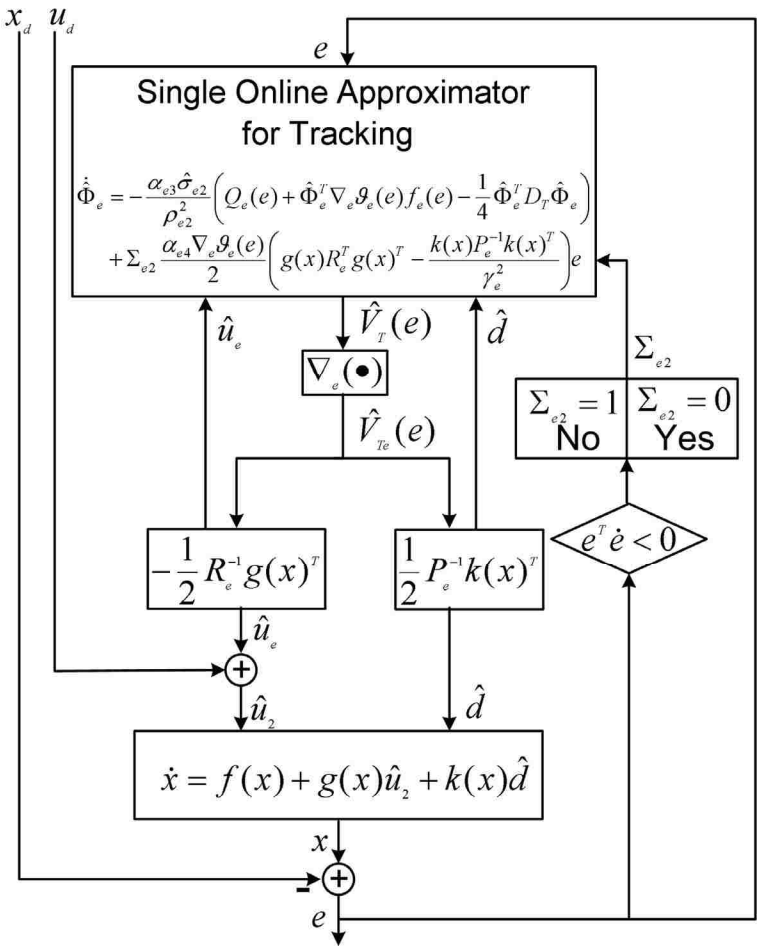


Fig. 3. SOLA design for HJI optimal tracking.

The block diagram of the SOLA-based design for HJI tracking is presented in Fig. 3. The diagram become the HJB optimal control designs by taking $k(x) = 0$ and after appropriate modifications to reflect $\hat{V}_e(e)$, and $\hat{u}_{1e}(e)$, respectively.

V. Simulation Results

To demonstrate the effectiveness of the SOLA-based designs of this work, several examples are now offered. First, three optimal regulator designs are presented for a linear system and two nonlinear systems. Then, the optimal tracker is implemented for the optimal formation control of nonholonomic mobile robots. To implement the online SOLA-based designs, a linear in the parameter (LIP) NN is utilized as the OLA. In addition, in each example, $J_1(y) = J_2(y) = y^T y / 2$ so that $J_{1y}(y) = J_{2y}(y) = y$ in (26) and (44), respectively. For regulation, $y = x$ while $y = e$ for tracking.

A. Linear HJB Example

Consider the linear system given by

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & -2 \\ 1 & -4 \end{bmatrix} x + \begin{bmatrix} 1 \\ -3 \end{bmatrix} u$$

accompanied by the HJB cost function (2) with $R = 1$ and $Q(x) = x^T x$. For linear systems with quadratic cost functions, the optimal control is found by solving the ARE, and the optimal cost function is found to be $V_1^* = W_{c1}^* x_1 x_2 + W_{c2}^* x_1^2 + W_{c3}^* x_2^2$ where $W_{c1}^* = -0.1162$, $W_{c2}^* = 0.3199$, and $W_{c3}^* = 0.1292$, respectively [34].

The basis vector for the SOLA-based implementation was selected as $\phi(x) = [x_1 x_2 \quad x_1^2 \quad x_2^2]^T$ while the tuning parameters were selected as $\alpha_1 = 200$ and $\alpha_2 = 0.01$. The initial conditions of the system states were taken as $x(0) = [2 \quad -2]^T$ while all NN weights

were initialized to zero. That is, no initial stabilizing control was utilized for implementation of this online design.

Figure 4 illustrates the time history of the OLA weights since it is approximating the cost function. Examining the trajectories, it is observed that the NN weights begin from zero and converge to constant values as the results of *Theorem 1* predicted. The final values of the OLA weights were $\hat{W}_{c1} = -0.1300$, $\hat{W}_{c2} = 0.3269$, and $\hat{W}_{c3} = 0.1193$ which illustrates the convergence of the approximated cost function to the optimal cost function with small bounded error which is again consistent with the theoretical results derived in this work. Since the SOLA-based design uses only one OLA, convergence of the critic weights ensures the convergence of the control input to the optimal control.

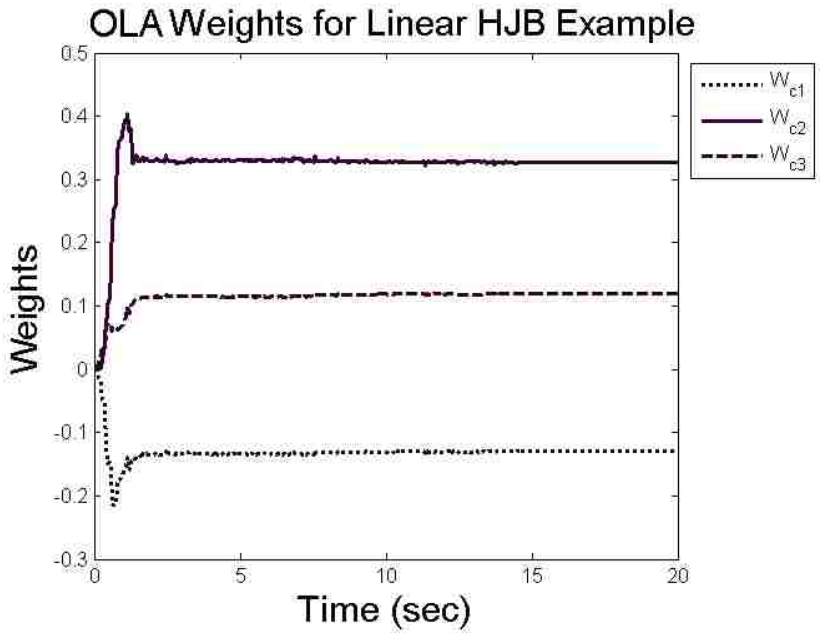


Fig. 4. OLA weights for example 1.

The time history of the system states is shown in Fig. 5. To satisfy the PE condition discussed in *Remark 1* in *Section III*, probing noise was added to the nonlinear dynamics (1). After 15 seconds, this signal was removed and the system states were allowed to converge to zero.

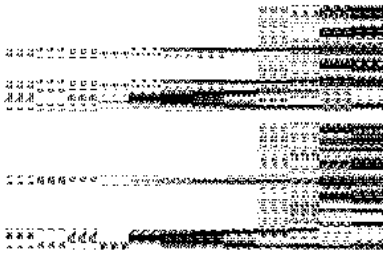


Fig. 5. System states for example 1.

B. Nonlinear HJB Example

Consider the nonlinear system in the form of (1) with $x = [x_1 \;\; x_2]^T$,

$$f(x) = \begin{bmatrix} x_2 \\ -x_1\left(\frac{\pi}{2} + \tan^{-1}(5x_1)\right) - \frac{5x_1^2}{2(1+25x_1^2)} + 4x_2 \end{bmatrix}, \text{ and } g(x) = [0 \;\; 3]^T.$$

Using the HJB cost function (2) with $Q(x) = x_2^2$ and $R = 1$, the optimal cost function is given by $V_2^* = W_{c4}^* x_1^2 + W_{c5}^* x_1^2 \tan^{-1}(5x_1) + W_{c6}^* x_2^2$ with $W_{c4}^* = \pi/2$, $W_{c5}^* = 1$ and $W_{c6}^* = 1$ [35]. The basis vector for the SOLA-based scheme implementation was selected as $\phi(x) = [x_1 \;\; x_2 \;\; x_1 x_2 \;\; x_1^2 \;\; x_2^2 \;\; x_1^2 \tan^{-1}(5x_1) \;\; x_1^3]^T$ while the tuning parameters were selected as $\alpha_1 = 200$ and $\alpha_2 = 0.01$. The initial conditions of the system states were taken as $x(0) = [4 \; -4]^T$ while all NN weights were initialized to zero. That is, no initial stabilizing control was utilized for implementation of this online design for the nonlinear system.

Figure 6 depicts the evolution of the OLA weights during the online learning. Starting from zero, the weights of the online OLA are tuned to learn the optimal cost function, and the final values of the OLA weights are found to be $\hat{W}_{c4} = 1.5838$, $\hat{W}_{c5} = 1$,

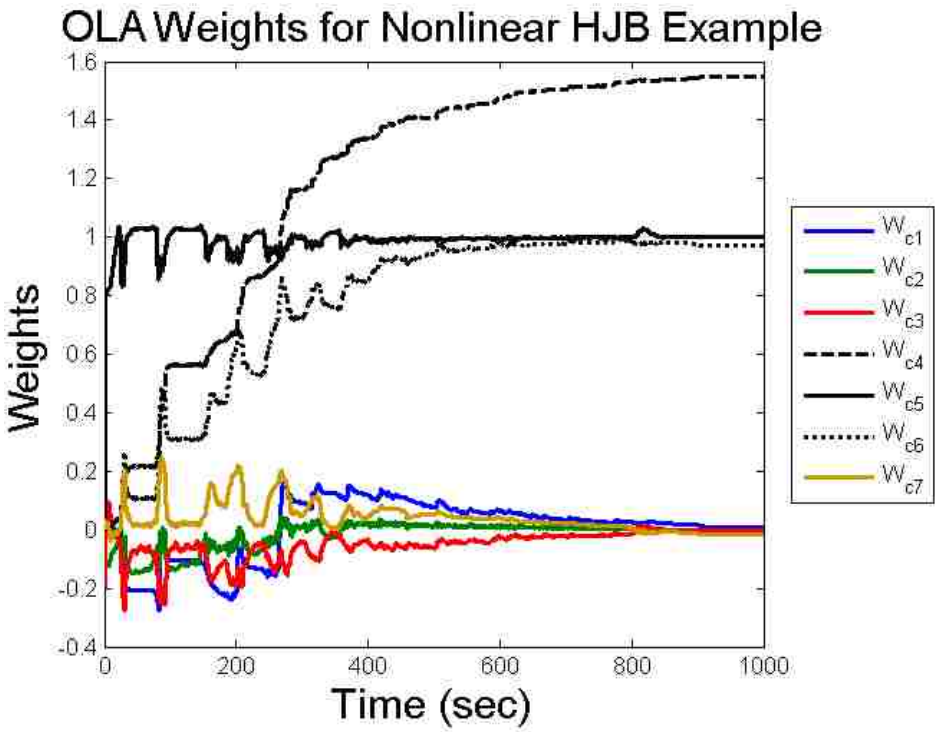


Fig. 6. OLA weights for example 2.

and $\hat{W}_{c6} = 1$, with $[\hat{W}_{c1}\ \hat{W}_{c2}\ \hat{W}_{c3}\ \hat{W}_{c7}] = [-0.0213\ \ -0.0038\ \ 0.0076\ \ 0.008]$. Again, the online SOLA design was observed to converge to the actual optimal cost function with small bounded error as the theoretical results suggested.

The system states are shown in Fig. 7, and probing noise is added similarly to the linear case to ensure the PE condition is satisfied. After 850 seconds, the PE condition was no longer required and was thus removed. To illustrate the importance of the secondary term in the tuning law in (25), the online OLA design is attempted with $\Sigma(x, \hat{u}) = 0$. That is, the learning algorithm only seeks to minimize the auxiliary HJB residual (27) and does not consider system stability. Fig. 8 shows the results of not considering the nonlinear system's

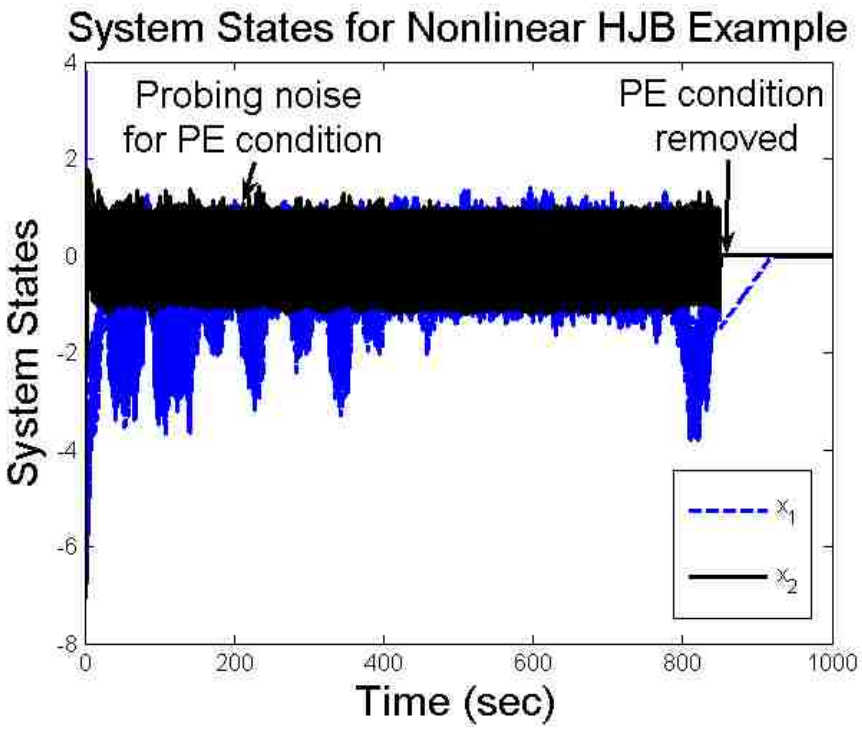


Fig. 7. System states for example 2.

stability while learning the optimal HJB function. From this figure, it is clear that the system state quickly escape to infinity, and the SOLA-based controller fails to learn the HJB function. Thus, the importance of the secondary term in (25) which ensures the stability of the system is revealed.

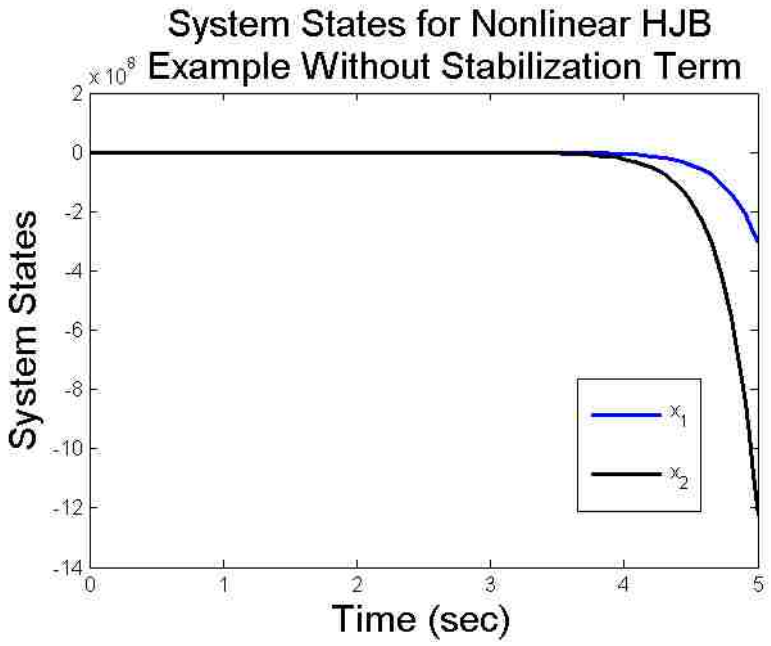


Fig. 8. Divergence of the system states when the stabilizing OLA update is removed ($\Sigma(x, \hat{u}_1) = 0$) for example 2.

C. *Nonlinear HJI Example*

Next, consider the nonlinear system in the form of (11) with $x = [x_1 \ x_2]^T$,

$$f(x) = \begin{bmatrix} -\frac{29x_1 + 87x_1x_2^2}{8} - \frac{2x_2 + 3x_2x_1^2}{4} \\ -\frac{x_1 + 3x_1x_2^2}{4} \end{bmatrix}, \ g(x) = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}, \text{ and } k(x) = [0.5 \ \ 1]^T.$$

The above dynamics were derived using the converse optimal control method [36] using the HJI cost function (12) with $R = I$, $P = \gamma = 1$, and

$$Q(x) = 2\left((2x_1 + 6x_1x_2^2)^2 + (4x_2 + 6x_1^2x_2)^2\right).$$

The optimal cost function was then found to be

$$V_3^*(x) = W_{c4}^* x_1^2 + W_{c5}^* x_2^2 + W_{c6}^* x_1^2 x_2^2$$

with $W_{c4}^* = 1$, $W_{c5}^* = 2$ and $W_{c6}^* = 3$. The basis vector for the HJI SOLA implementation was selected as $\vartheta(x) = [x_1 \;\; x_2 \;\; x_1x_2 \;\; x_1^2 \;\; x_2^2 \;\; x_1^2x_2^2 \;\; x_1^3]^T$ while the tuning parameters were selected as $\alpha_3 = 200$ and $\alpha_4 = 0.01$. The initial conditions of the system states were taken as $x(0) = [4 \;\; -4]^T$ while all NN weights were initialized to zero. That is, no initial stabilizing control was utilized for implementation of this online design for the nonlinear system.

Figure 9 shows the OLA weights during the online learning. Starting from zero, the weights of the online OLA are tuned to learn the optimal HJ function, and the final values of the OLA weights are found to be $\hat{W}_{c4} = 1.0007$, $\hat{W}_{c5} = 2.0003$, and $\hat{W}_{c6} = 2.9944$, with $[\hat{W}_{c1} \;\; \hat{W}_{c2} \;\; \hat{W}_{c3} \;\; \hat{W}_{c7}] = [0.0001 \;\; 0.0000 \;\; 0.0006 \;\; 0.004]$. As in the previous examples, the online SOLA design was observed to converge to the actual optimal cost function with small bounded error as the *Theorem 2* ensured. The convergence of the OLA parameters or critic NN weights ensures the convergence of the control input and approximated worst case disturbance to the optimal control and disturbance, respectively.

The system states are shown in Fig. 10, and probing noise is added similarly to the linear case to ensure the PE condition is satisfied. After 4750 seconds, the PE condition was no longer required as convergence of the OLA weights was observed. To reiterate the importance of the secondary term in the tuning law in (43), the online SOLA design is

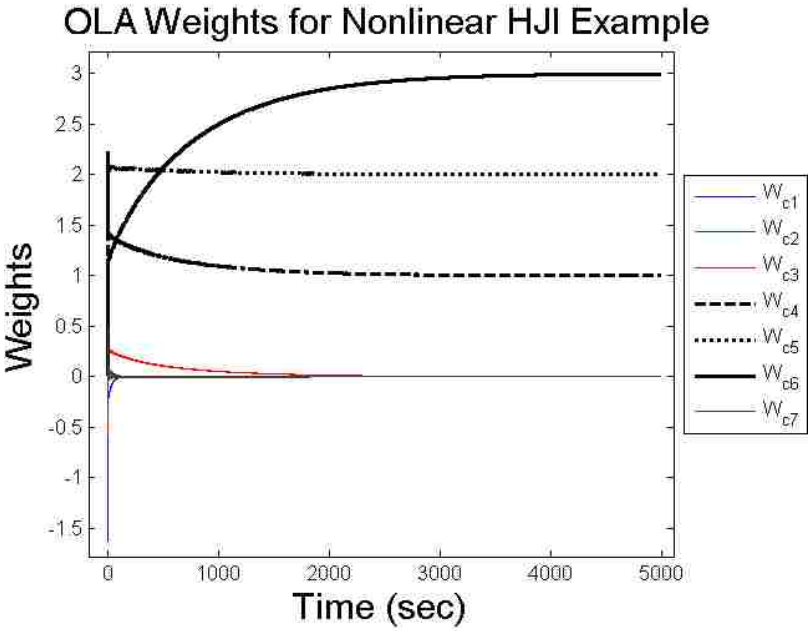


Fig. 9. OLA weights for example 3.

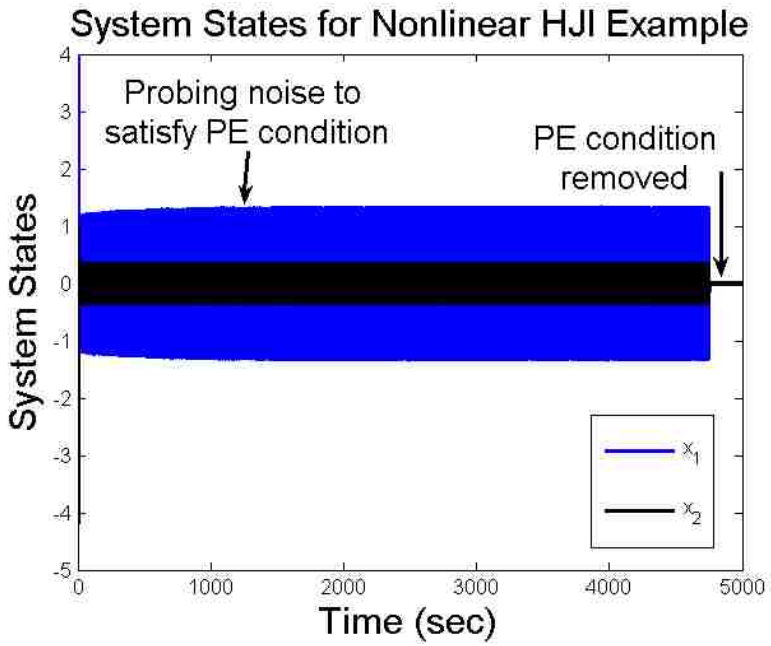


Fig. 10. System states for example 3.

attempted with $\Sigma_2(x,\hat{u}_2,\hat{d})=0$. That is, the learning algorithm only seeks to minimize the auxiliary HJI residual and does not consider system stability. Fig. 11 shows the results of not considering the nonlinear system's stability while learning the optimal HJI function. Similar to the observations of Fig. 8, the state trajectories shown in Fig. 11 show that the system states diverge resulting in a failed HJI learning session.

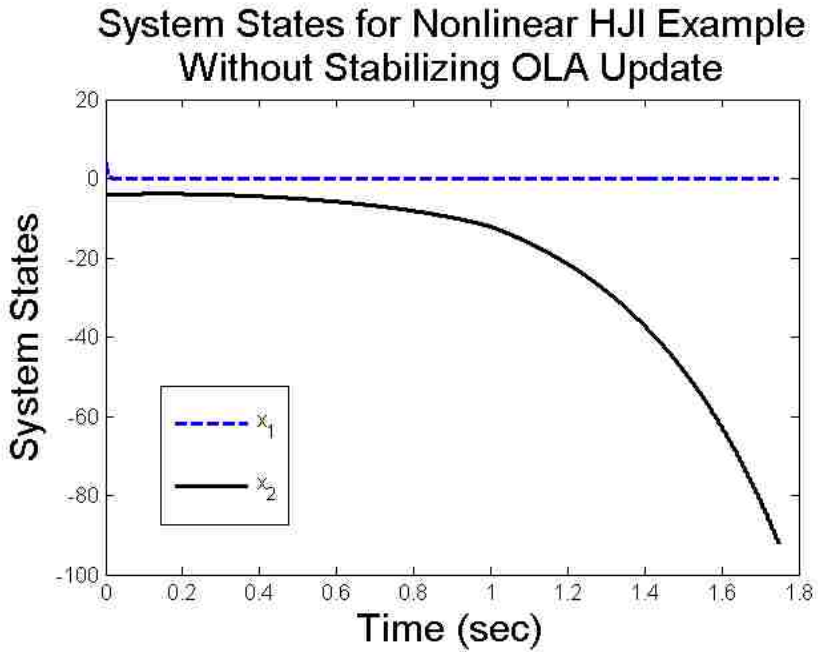


Fig. 11. Divergence of the system states when the stabilizing OLA update is removed ($\Sigma_2(x,\hat{u}_2,\hat{d})=0$) for example 3.

D. Optimal Tracking Control of Nonholonomic Mobile Robot Formations

To demonstrate the effectiveness of the proposed optimal tracker, the HJB equation is solved online for leader-follower based formation control of nonholonomic mobile robots. First, a brief overview of nonholonomic mobile robots [37] and leader-follower based

formation control [38] is provided. The dynamics of a nonholonomic mobile robot are written as [37]

$$\dot{\bar{v}} = f(\bar{v}) + g\tau\,, \tag{57}$$

where $f(\bar{v}) = -\overline{M}^{-1}\overline{V}_m(q,\dot{q})\bar{v} - \overline{M}^{-1}\overline{F}(\bar{v})$, $g = \overline{M}^{-1}\overline{B}$, and $\bar{v} = [v \;\; \omega]^T$ is the velocity vector with v being the translational velocity and ω the robot angular velocity. In addition, $\overline{M}$ is a constant positive definite inertia matrix, $\overline{V}_m$ is the bounded centripetal and coriolis matrix, $\overline{F}$ is the friction vector, $\overline{B}$ is a constant, nonsingular input transformation matrix, and τ is the control torque vector [37]. For complete details on the robot dynamic equation above, see [37].

The objective of separation-bearing leader-follower formation control [38] is for each robot to maintain a desired separation distance and bearing angle with respect to a designated leader as shown in Fig. 12 where the leader is denoted with a subscript *'i'* and the follower is denoted by the subscript *'j'*. In our previous work [38], an auxiliary velocity control input, $v_{jc}(t)$, was found to ensure that $\lim_{t\to\infty}(L_{ijd} - L_{ij}) = 0$ and $\lim_{t\to\infty}(\Psi_{ijd} - \Psi_{ij}) = 0$ where L_{ij} and Ψ_{ij} are the measured separation and bearing of the follower *j* with respect to leader *i* while L_{ijd} and Ψ_{ijd} represent the desired distance and bearing. Then, the control torque τ was calculated to ensure $\bar{v} \to v_{jc}(t)$. In [38], system stability was the only design criterion.

In contrast, using the optimal control framework proposed this work, the control torque τ is now re-designed to ensure the control velocity $v_{jc}(t)$ of our previous work [37] is tracked in an optimal manner. A formation of identical nonholonomic mobile robots is considered where the leader's trajectory is the desired formation trajectory as shown in Fig.

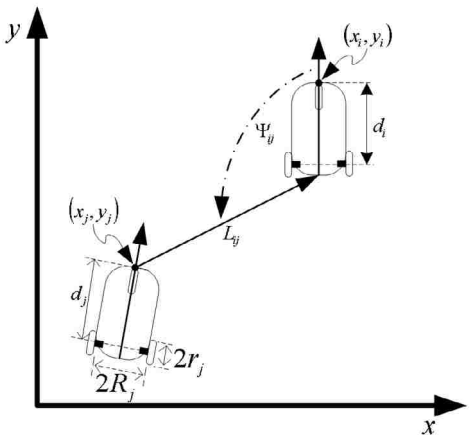


Fig. 12. Leader-follower formation control.

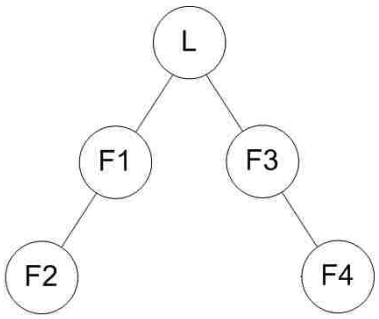


Fig. 13. Desired formation.

13 where $L, F1, F2, F3,$ and $F4$ denote the *leader, follower 1*, *follower 2, follower 3,* and *follower 4,* respectively. Additionally, the robot parameters used in the simulation are as described in [37] where the robots have the same physical dimensions including masses and moments of inertial, but different coefficients of friction.

For implementation of the OLA based SOLA tracker, the HJB equation written in the form similar to (59) is considered with $Q_e(e) = e^T Q_e e$, $Q_e = 50$, $R_e = I$ and where I is the identity matrix. The tuning parameters were selected as $\alpha_1 = 200$ and $\alpha_2 = 0.01$. As in the previous examples, all tunable weights were initialized to zero. That is, no initial stabilizing control was utilized for implementation of the online tracker design. For this example, we specify the gradient of the activation function instead of the activation function itself since the gradient of activation function is required for the SOLA implementation and not the actual activation function. This type of dynamic programming is often referred to as *dual heuristic dynamic programming* [10] where the gradient of the cost function is approximated instead of the actual cost function. In the previous examples, the optimal cost functions were

observed to contain terms that were also present in the internal dynamics ($f(x)$) of the nonlinear systems. Therefore, the gradient of the activation function is selected based on the terms one would expect to find in $f_e(e)$. That is, the gradient of the activation function is selected to be

$$\nabla_e \vartheta_e = \begin{bmatrix} \nabla \vartheta_1 & 0 \\ 0 & \nabla \vartheta_2 \end{bmatrix}$$

where

$$\nabla \vartheta_1 = \nabla \vartheta_2 =$$
$$[e_1 \;\; e_2 \;\; e_1 e_2 \;\; (\mathrm{sgn}(v) - \mathrm{sgn}(v_{c1})) \;\; (\mathrm{sgn}(\omega) - \mathrm{sgn}(v_{c2})) \;\; (\mathrm{sgn}(\omega) - \mathrm{sgn}(v_{c2})) \times (\mathrm{sgn}(v) - \mathrm{sgn}(v_{c1}))$$
$$e_1(\mathrm{sgn}(\omega) - \mathrm{sgn}(v_{c2})) \;\; e_1(\mathrm{sgn}(v) - \mathrm{sgn}(v_{c1})) \;\; e_2(\mathrm{sgn}(v) - \mathrm{sgn}(v_{c1})) \;\; e_2(\mathrm{sgn}(\omega) - \mathrm{sgn}(v_{c2}))]^T .$$

During the online learning, the virtual reference cart for the formation leader traveled a constant translational velocity, $v_{ir} = 1\,m/s$ while the reference angular velocity was selected as $\omega_{ir} = 0.1\sin(0.25t)$. The results of the SOLA algorithm for the leader and its followers are shown in Figs. 14 through 19. First, Figs. 14 through 16 display the SOLA weights for the leader, follower 1, and follower 3, respectively. In each case, the weights are observed to converge to constant values as the theoretical results of *Section IV* predicted. The SOLA weights for followers 2 and 4 were observed to be similar to those of followers 1 and 3, respectively. The slight differences in the SOLA weights for each robot are linked to differences in the coefficients of friction for each robot and the use of the *sign*(•) function in the gradient of the activation functions. In addition, the dominant weights in Fig. 14-16 are observed to correspond to the e_1 and e_2 terms in $\nabla_e \vartheta_e$.

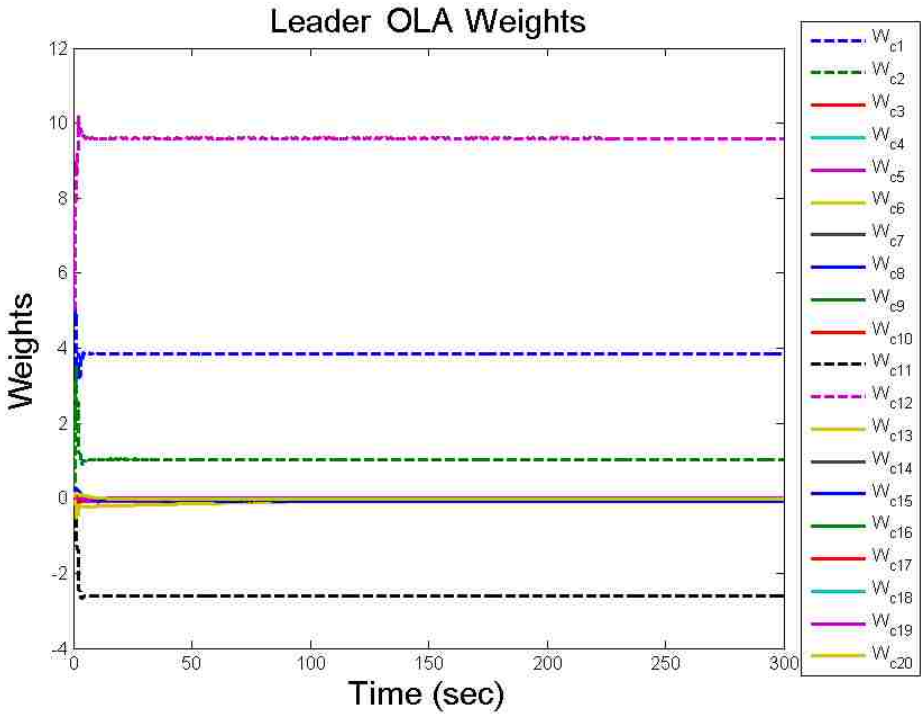


Fig. 14. OLA weights for the formation leader.

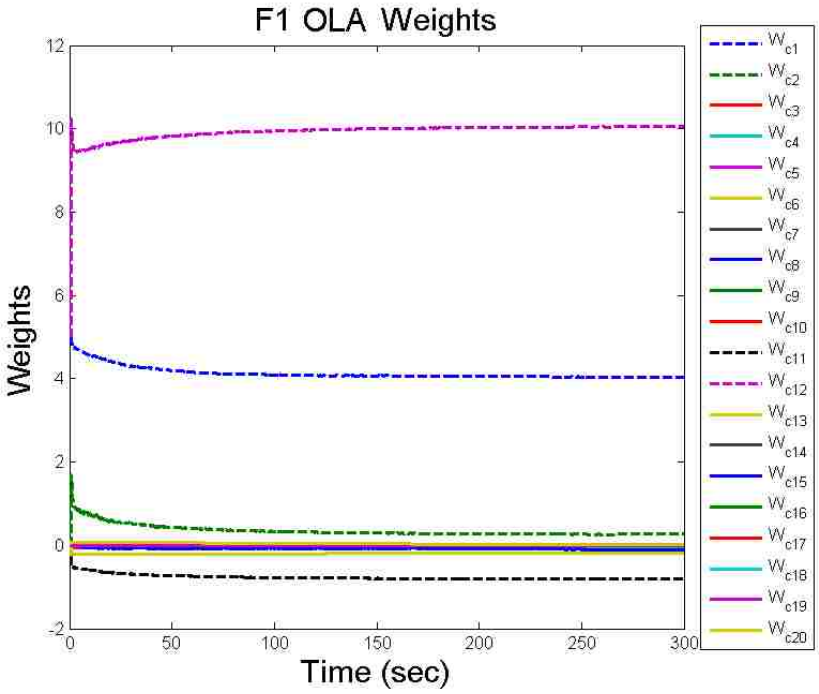


Fig. 15. OLA weights for follower 1.

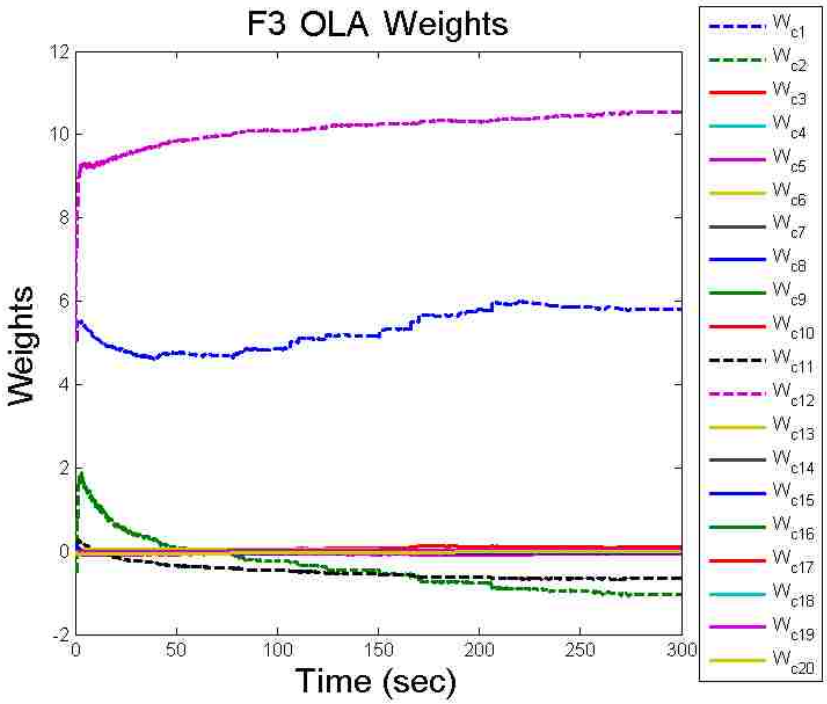


Fig. 16. OLA weights for follower 3.

Figures 17 and 18 depict the velocity tracking errors for the leader and follower 1. For the leader tracker errors in Fig. 17, probing noise is introduced to satisfy the PE condition and subsequently removed after 225 seconds. Similarly, the tracking errors of follower 1 are shown in Fig. 18 where probing noise was removed after 275 seconds. In addition, the effect of the leader's probing noise signal on the followers is also observed in Fig. 18 when the PE condition is removed for the leader at 225 seconds illustrating the effects of the formation dynamics on the follower robots. Similar results were observed for the other followers although not shown. In all cases, the velocity tracking errors converged to a small bounded region around the origin after the probing noise was removed as the theory suggested.

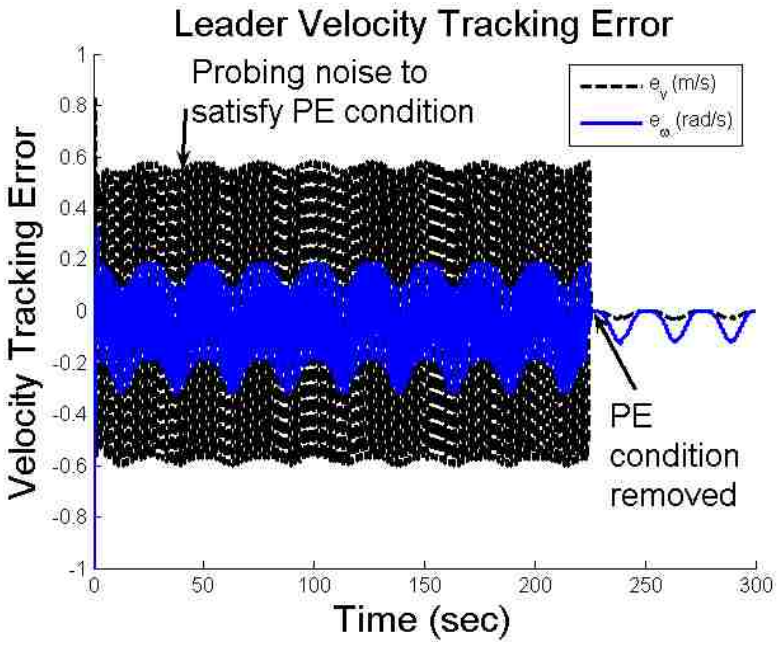


Fig. 17. Velocity tracking errors for the formation leader.

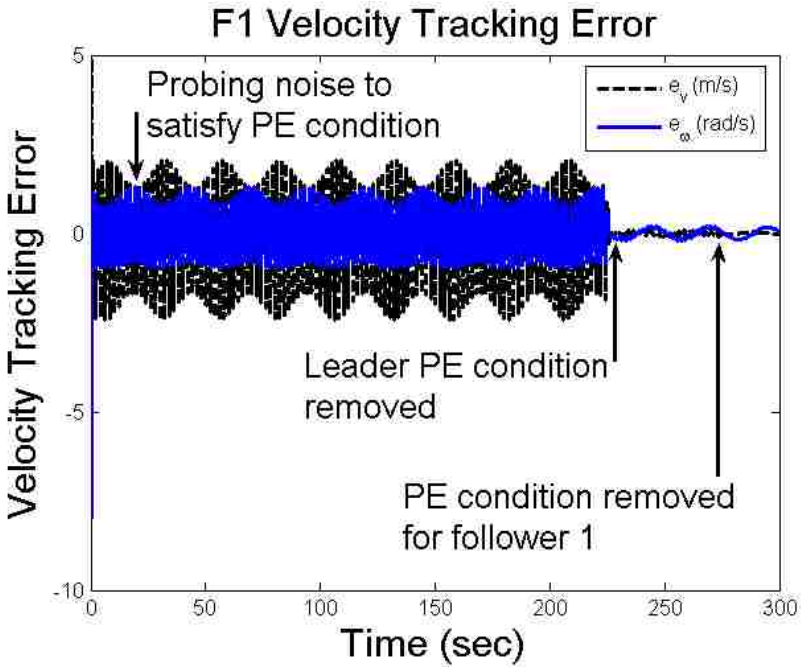


Fig. 18. Velocity tracking errors for follower 1.

Next, Fig. 19 show the Hamiltonian approximation for the formation leader, and examining the figure, it is observed that as the approximate Hamiltonian converges to a small bounded region containing the origin as the leader's OLA parameters converge to constant values. This illustrates that the OLA weights are indeed minimizing the Hamiltonian even when the velocity tracking errors are not zero.

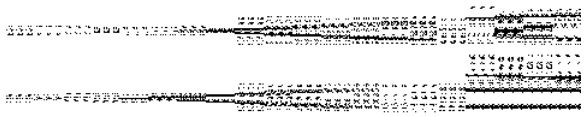


Fig. 19. Approximate Hamiltonian for the formation leader.

In contrast to the previous examples for regulation, converse optimal control techniques do not provide insight as to what the real optimal cost function and control input should be. Therefore, a comparison in terms of cost will be used to evaluate the performance of the proposed optimal tracker with the cost defined by

$$V_T(e(t)) = \int_t^{\infty} (r_e(e(\tau), u_e(\tau)))d\tau$$

where $r_e(e(\tau), u_e(\tau))$ is defined similarly to $r(x(\tau), u_1(\tau))$ with $x(\tau)$ and $u_1(\tau)$ replaced with $e(\tau)$ and $u_e(\tau)$, respectively, and where u_e is required to be admissible.

As a baseline test, the control input (55) is applied to the robot systems (57) when the feedback control signal is not optimal. That is,

$$\tau = u_d - g^{-1}(K e + f_e(e)) \tag{58}$$

where u_d is given by (56) with $x_d = v_c$, $f_e(e)$ is defined as in (49), $K > 0$ is a constant design matrix, and $e = [e_1 \;\; e_2]^T$ is the robot velocity tracking error. Substituting (58) into the robot dynamics (57) reveals the closed loop robot velocity tracking error dynamics to be

$$\dot{e} = -K e .$$

It can be shown that the control input (58) guarantees the velocity tracking error to converge to zero exponentially. For the comparison, K was selected as $K = diag\{10,10\}$.

For the comparisons, the virtual reference cart for the formation leader traveled a constant translational velocity, $v_{ir} = 1\, m/s$ while the reference angular velocity was selected as $\omega_{ir} = 0.1\sin(0.25t)$. The formation trajectories when the control input (58) is applied for the leader and its followers is shown in Fig. 20 where the robots start in the bottom left corner of the figure and travel towards the top right corner. Similar robot trajectories were observed when the learned SOLA control law was applied.

Next, the cost associated with the non-optimal feedback control input of (58) and the cost associated with the learned SOLA control input was calculated and compared for the leader and its followers. Figure 21 shows the resulting costs for the formation leader while Fig. 22 shows the costs for follower 1. In each case, the costs associated with the SOLA control inputs which were learned online were less than the costs associated with the non-

optimal control input (58) illustrating the effectiveness of the online optimal controller. Similar trends were observed for the other followers although they are not shown.

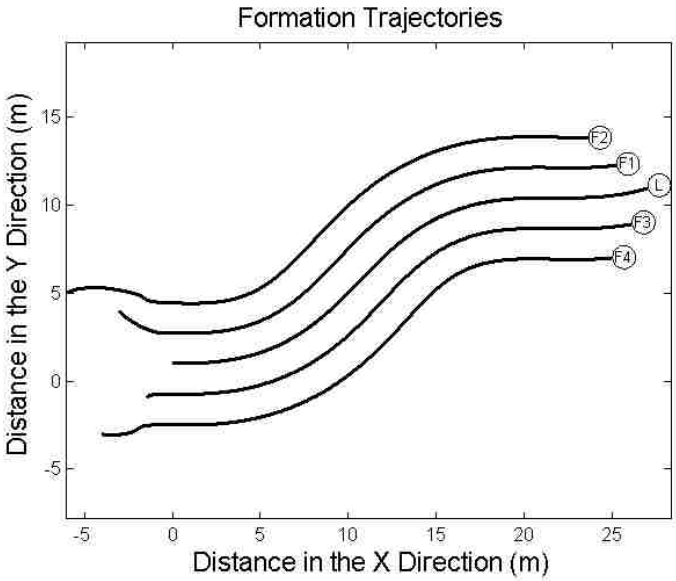


Fig. 20. Formation trajectories.

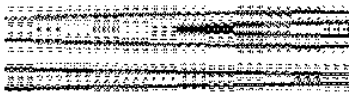


Fig. 21. Costs for the formation leader.

Fig. 22. Costs for follower 1.

VI. Conclusions

In this work, a single OLA was utilized to design a single network adaptive critic to solve both the Hamilton Jacobi-Bellman and Hamilton Jacobi-Isaacs equations in real time for the optimal control of general affine nonlinear continuous-time systems. In the presence of known dynamics, the optimal regulation and tracking control problems were undertaken. The SOLA based design was utilized to learn the cost function and nearly optimal feedback control signal for the HJB optimal control problem and the cost function, nearly optimal feedback control signal, and optimal disturbance of the HJI optimal control problem. All OLA parameters were tuned online using novel update laws, and Lyapunov techniques were used to demonstrate the stability of the proposed optimal control schemes. Simulation results were also provided to verify the theoretical conjectures.

REFERENCES

[1] H. K. Khalil, *Nonlinear Systems*, 3rd edition, Prentice-Hall, Upper Saddle River, NJ, 2002.

[2] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[3] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems,* Taylor and Francis, Philadelphia, PA, 1999.

[4] F. L. Lewis and V. L. Syrmos, *Optimal Control*, 2nd ed. Hoboken, NJ: Wiley, 1995.

[5] J. Shamma and J. Cloutier, "Existence of SDRE stabilizing feedback," *IEEE Trans. on Automatic Control,* vol. 48, pp. 513-517, March 2003.

[6] M. Krstic and Z. H. Li, "Inverse optimal design of input-to-state stabilizing nonlinear controllers," *IEEE Trans. on Automatic Control*, vol. 43, no. 3, pp. 336-350, 1998.

[7] H. S. Chang and S. I. Marcus, "Two-person zero-sum Markov games: receding horizon approach," *IEEE Trans. on Automatic Control*, vol. 48, no. 11, pp. 1951-1961, 2003.

[8] D. P. Bertsekas*, Dynamic Programming and Optimal Control*, 2nd ed. Belmonth, MA: Athena Scientific, 2000.

[9] D. Q. Mayne and H. Michalska, "Receding horizon control of nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 35, no. 7, pp. 814-824, 1990.

[10] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, Eds., *Handbook of Learning and Approx. Dynamics Prog.,* Wiley-IEEE Press, 2004.

[11] D. P. Bertsekas and J. Tsitsiklis*, Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.

[12] Z. Chen and S. Jagannathan, "Generalized Hamilton-Jacobi-Bellman formulation based neural network control of affine nonlinear discrete-time systems," *IEEE Trans. On Neural Networks,* vol. 10, pp. 90-106, Jan. 2008.

[13] A. Al-Tamimi, F.L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof," *IEEE Trans. On Systems, Man, and Cybernetics—Part B*, vol. 38, pp. 943-949, Aug. 2008.

[14] M. Abu-Khalaf, F. L. Lewis, and Jie Huang, "Neurodynamic programming and zero-sum games for constrained control systems," *IEEE Trans. on Neural Networks,* vol. 19, no. 7, pp. 1243-1252, 2008.

[15] J. Huang and C. F. Lin, "Numerical approach to computing nonlinear H_∞ control laws," *Journal of Guidance control and Dynamics*, vol. 18, no. 5, pp. 989-994, 1995.

[16] J. Huang, "An algorithm to solve the discrete HJI equation arising in the *L*2 gain optimization problem," *Int. J. Control*, vol. 72, no. 1, pp. 49- 57, 1999.

[17] R. W. Beard and T. W. McLain, "Successive Galerkin approximation algorithms for nonlinear optimal and robust control," *Int. J. Control*, vol. 71, no. 5, pp.717-743, 1998.

[18] G. Toussaint, T. Basar and F. Bullo, "H^∞-optimal tracking control techniques for nonlinear underactuated systems." *Proc. of IEEE Conf. on Decision and Control,* pp. 2078 – 2083, 2000.

[19] L. Magni, G. De Nicolao, and R. Scattolini, "Output feedback and tracking of nonlinear systems with model predictive control," *Automatica*, vol. 37, no. 10, pp. 1601-1607, 2001.

[20] W. Luo, Y. C., Chu, and K. V. Ling, "Inverse optimal adaptive control for attitude tracking of spacecraft," *IEEE Trans. on Automatic Control,* vol. 50, pp 1693-1654, Nov. 2005.

[21] H. Zhang, Q. Wei, and Y. Luo, "A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm," *IEEE Trans. on Systems, Man, and Cybernetics—Part B,* vol. 38, pp. 937-942, Aug. 2008.

[22] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear discrete-time systems," *Proc. of the Mediterranean Conference on Control and Automation*, pp. 1390 – 1395, June 2009.

[23] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica,* vol. 45, no. 2, pp. 477-484, 2009.

[24] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to h-infinity control," *Automatica,* vol. 43, pp 473-481, 2007.

[25] D. Vrabie, K. Vamvoudakis, and F. Lewis, "Adaptive optimal controllers based on generalized policy iteration in a continuous-time framework," *Proc. of the IEEE Mediterranean Conf. on Control and Automation,* pp. 1402-1409, June 2009.

[26] C. H. Watkins, *Learning from delayed rewards.* PhD Dissertation, University of Cambridge, England, 1989.

[27] T. Dierks, B. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Networks*, vol. 22, pp. 851-860, 2009.

[28] R. Padhi and S. N. Balakrishnan, "Optimal management of beaver population using a reduced-order distributed parameter model and single network adaptive critics," *IEEE Trans. on Control Systems Tech.,* vol. 15, no. 4, pp. 628-640, July 2006.

[29] T. Basar and G.L. Olsder, *Dynamic Noncooperative Game Theory*, New York, Academic, 1982.

[30] Van Der Schaft, A. J., "L_2-gain analysis of nonlinear systems and nonlinear state feedback H_∞ control," *IEEE Trans. Automatic Control*, vol. 37 no. 6, pp. 770-784, 1992.

[31] J. C. Willems, "Dissipative dynamical systems part I: general theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no.1, pp. 321-351, 1972.

[32] H. Knoblock, A. Isidori, and D. Flockerzi, *Topics in Control Theory,* Boston, MA, Springer-Verlag, 1993.

[33] P.M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, "Asymptotic tracking for uncertain dynamical systems via a multilayer neural network feedforward and RISE feedback control structure," *IEEE Trans. on Automatic Control,* vol. 53, pp 2180-2185, Oct. 2008.

[34] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-time adaptive critics," *IEEE Trans. on Neural Networks,* vol. 18, no. 3, pp. 631-647, May 2007.

[35] J. Curtis and R. Beard, "Successive collocation: an approximation to optimal nonlinear control," *Proc. IEEE American Control Conf.*, pp. 3481-3485, 2001.

[36] J. Doyle, J. Primbs, B. Shapiro, and V. Nevistic, "Nonlinear games: examples and counter examples," *Proc. IEEE Conf. on Decision and Control,* pp. 3915-3920, Dec. 1996.

[37] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. on Neural Networks*, vol. 8, pp. 589-600, July 1998.

[38] T. Dierks and S. Jagannathan, "Neural network control of mobile robot formations using RISE feedback," *IEEE Transactions on Systems, Man, and Cybernetics—Part B,* vol. 39, pp. 332-346, April 2009.

SECTION

2. CONCLUSIONS AND FUTURE WORK

In this dissertation, the control of nonholonomic mobile robot formations and UAV formations was undertaken while addressing many of the common assumptions and simplifications of existing approaches. The dynamics of the individual agents and the formation were explicitly considered and compensated using NN's and online weight tuning. The tracking performance of the overall formation was guaranteed by compensating the formation dynamics either explicitly through communication or implicitly via decentralized control schemes. For the quadrotor UAV, the derived formation control laws were independent of a specific operating point and without the use of small angle approximations. In addition, the infinite horizon HJB equation was solved online and forward-in-time, for both discrete-time and continuous-time systems, while the infinite horizon HJI equation was solved online for continuous time systems to achieve near optimal control.

In the first paper, the asymptotically stable NN tracking controller for leader-follower based formation control considers the dynamics of the leader and the followers using the backstepping technique with RISE feedback. The benefit of the feedback control scheme is that asymptotic stability of the formation is guaranteed even when the dynamics of the followers and their leader are unknown since the NN learns them all online, and the RISE ensures robustness in the presence of unmodeled dynamics and disturbances provided they are upper bounded by known functions. The numerical simulations also illustrated the strength of asymptotic stability over a *uniformly ultimately bounded (UUB)* controller of our previous work. Although superior tracking over

existing controllers was demonstrated, the RISE feedback technique is observed to have several drawbacks. First, the RISE feedback scheme requires that the upper bounds on the disturbances and unmodeled dynamics to be well known. In addition, the RISE feedback relies on the integral of a high gain term, which was observed to have negative effects on the transient response of the formation. In practical applications of the NN/RISE control scheme, better overall performance may be observed by initially applying the control law with the RISE feedback portion of the controller disabled. The NN controller would drive the tracking errors into the compact set guaranteed by the *UUB* stability result, and then the RISE feedback portion of the control input could be enabled to regain the asymptotic steady-state tracking performance observed in paper 1.

A NN output feedback tracking controller for leader-follower based formation control was presented in the second paper. Each robot had many challenging uncertainties to overcome including limited communication, immeasurable velocity vectors, unknown dynamics, and bounded disturbances. These challenges were overcome by using a novel NN observer and controller and enabled the leader-follower formation control objective to be completed without the need of the separation principle. The impact of the leader's states on the control laws of the followers was also illustrated in the simulation results since the formation tracking errors were not observed to converge until convergence of the followers' observer estimates of their leaders' velocity vectors was achieved.

In addition, the first two papers consider the stability of the formation in the presence of obstacles. The obstacle avoidance control laws were shown to be effective in both a static and dynamic obstacle environment. Further, by treating robots in the formation as obstacles, collisions within the formation were guaranteed not to occur. The

proposed obstacle avoidance scheme is observed to have potential limitations. Since the scheme only considers the closest obstruction, it is possible that in a highly cluttered environment there may be more than one obstacle within the robot's safety zone; one of which could potentially be another robot in the formation. In this case, the follower may exhibit an oscillatory behavior between multiple obstructions located within the safety zone which is not ideal; however, the goal of the obstacle avoidance scheme is still achieved in that collisions are avoided.

The control of a quadrotor UAV was considered in paper three where a NN output feedback control law was developed and the separation principle was relaxed. Despite being underactuated, adaptive backstepping techniques were utilized to control all six DOF in the presence of unmodeled dynamics and bounded disturbances. The dynamics of the UAV were not required to be known since the neural networks learned the complete UAV dynamics online. Numerical simulations confirmed that the proposed nonlinear NN controller outperformed a conventional linear controller which used state-feedback. In the comparison, large control gains were required by the linear controller to achieve the same tracking performance observed when the proposed NN output feedback controller was applied. Further, the use of large control gains in the linear controller led to significant noise amplification while the proposed controller did not rely on noisy velocity measurements by using output feedback. A drawback of the proposed scheme is that three NN's were required for implementation.

Subsequently, the fourth paper proposed a framework for quadrotor UAV leader-follower formation control by converting the formation control problem into a tracking control problem. The state feedback scheme did not require explicit knowledge of the UAV or formation dynamics since NN's learned the complete UAV and formation

dynamics online while in the presence of bounded disturbances. The importance of considering the formation dynamics was illustrated in the simulation results by ignoring the formation dynamics in the followers' controller design, and in the experiment, the formation was observed to exhibit poor tracking when the dynamics were ignored. In contrast, acceptable steady-state tracking was observed when the proposed controller was applied. A potential draw back of the proposed NN scheme is the transient response observed in the simulations. Although brief, several large spikes were observed in the follower UAV velocity tracking error signals which is undesirable.

In paper five, the Hamilton Jacobi-Bellman equation was solved in real time for the optimal regulation and tracking control of affine nonlinear discrete-time systems using online approximators. Knowledge of the system's internal dynamics was not required, and novel nearly optimal control laws were developed using OLA's to address the regulation problem and the tracking control problems. All OLA's were tuned online in contrast to offline methods which exist in the literature, and convergence to the optimal control signal was rigorously demonstrated while explicitly considering OLA reconstruction errors which is also not typical of most current approaches. Although the simulation results illustrated the effectiveness of the proposed approach, a drawback of the scheme is the need of an initial stabilizing control and the fact that the system states must to be persistently exiting (PE) while the OLA's learn the optimal HJB function and optimal control signal. That is, we cannot simply apply the proposed scheme to a nonlinear system and expect the optimal control to be learned by the time the system states have reached zero. To satisfy the PE condition, system noise was added to the nonlinear system dynamics. Thus, the proposed optimal control scheme is still being

trained albeit in an online fashion and without full knowledge of the system dynamics. Offline training traditionally requires full knowledge of the system dynamics.

Finally, paper six addressed the optimal control of affine nonlinear systems in continuous time. In contrast to paper five, the approach solved the optimal control problem online using a single OLA (SOLA) in continuous-time, and the SOLA was shown to solve both the HJB and HJI equations in real time in the presence of known dynamics. The SOLA-based design was utilized to solve the optimal regulation and tracking control problems, and all OLA parameters were tuned online using novel update laws. Simulation results illustrated that by using a secondary tuning law, an initial stabilizing control policy was not required to ensure the HJB or HJI functions were successfully learned. In fact, it was shown that by removing the secondary tuning law, system stability was lost and the OLA's failed to learn the HJB and HJI cost functions. A drawback of the proposed SOLA-design is the need for full knowledge of the system dynamics and the need for the PE condition on the system states. In addition, the choice of the probing noise signal added to the nonlinear system dynamics to satisfy the PE condition was found to have an impact on the learning ability of the SOLA-based optimal control scheme, and the best overall performance of the SOLA-based adaptive approach was observed by satisfying the PE condition using square waves.

Future applications of the RISE feedback scheme should focus on extending the method to include output feedback control. In addition, robust adaptive control methods could be used to relax the requirement on known upper bounds on the uncertainties and disturbances. Also, a more comprehensive obstacle avoidance scheme for leader-follower formation control could be considered in future work. This work would focus on alleviating the observed limitations of the current obstacle avoidance scheme so that

multiple objects and more complex environments can be navigated while completing the leader-follower formation control objective.

In the context of optimal control, future work should include relaxing the requirement of a known input coefficient matrix for the discrete-time optimal control development. In contrast, efforts in the continuous-time optimal control framework should include relaxing the requirement of known internal dynamics and subsequently the requirement of a known input coefficient matrix. In addition, the design of the feedforward term in the optimal tracker could be redesigned to include optimality for steady state and transient performance tracking. Finally, the optimal control using nonlinear approximators should be extended to other classes of nonlinear systems such as strict feedback and others.

VITA

Travis Alan Dierks was born December 22, 1982 in Chester, Illinois. He earned the Bachelor of Science and Master of Science degrees in Electrical Engineering from the Missouri University of Science and Technology (formerly the University of Missouri-Rolla) in 2005 and 2007, respectively. While completing his doctoral degree, Travis was a GAANN fellow and a Chancellor's fellow. He is a member of IEEE – Institute of Electrical and Electronics Engineers and INNS – International Neural Network Society. Travis received the degree of Doctor of Philosophy in December 2009.