NEURO ADAPTIVE CONTROL FOR AEROSPACE AND DISTRIBUTED SYSTEMS

by

ABHIJIT DAS

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

AUGUST 2010

Copyright © by ABHIJIT DAS 2010

All Rights Reserved

To my sister *Anindita*

who makes this day possible

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Frank Lewis for constantly motivating and encouraging me, and also for his invaluable advice during the course of my doctoral studies. A also thank my co-supervising professor Dr. Kamesh Subbarao for his help and advice during the course. I wish to thank my academic advisors Dr. Dan Popa, Dr. Harry Stephanou, and Dr. Kai Yeung for their interest in my research and for taking time to serve in my dissertation committee.

I like to thank National Science Foundation and Air Force Office of Scientific Research for funding my research.

Finally, I would like to express my deep gratitude to my sister *Anindita* who encouraged and inspired me to pursue doctorate study in USA. I am extremely fortunate to be so blessed. I am also extremely grateful to my father and wife *Thea* for their sacrifice, encouragement and patience. I also thank several of my friends who have helped me throughout my career.

July 1, 2010

ABSTRACT

NEURO ADAPTIVE CONTROL FOR AEROSPACE AND DISTRIBUTED SYSTEMS

ABHIJIT DAS, Ph.D.

The University of Texas at Arlington, 2010

Supervising Professors: Frank L. Lewis, Kamesh Subbarao

Nonlinear and adaptive control is generally considered one of the most effective techniques for stabilizing complex nonlinear systems, where linear control techniques may fail completely. Thousands of research papers are published on either theory or applications of nonlinear and adaptive control. But often one obvious question arises how to implement these techniques in real life model? The best answer that one can think of is to develop simple nonlinear control laws which are easy to implement. Moreover for controlling multi-agent systems, it is often required to distribute the control laws based on limited information available among the agents. This research provides some of these issues in the following way. *a*) *Autopilot design for Aerospace systems:* this research developes adaptive backstepping and dynamic inversion methods with internal dynamics stabilization for the quadrotor. Quadrotor helicopter models usually show two main characteristics. First, strong coupling among the system states and second, underactuation where many states are to be controlled with few control inputs. Due to these unique characteristics, the design of stabilizing control inputs is always challenging for quadrotor models. To confront these problems, first, a dynamic inversion technique with zero dynamics stabilization loop is introduced to a practical quadrotor model, second, an

adaptive-backstepping technique is developed to a lagrangian quadrotor model. The stabilizing control laws for both of these techniques are developed using on Lyapunov based method; and *b*) *Coordination of multi-agent systems:* coordination among multiple agents is generally done based on balanced or bi-directed communication graph models. If the agents are nonlinear and passive then for a balanced graph model synchronization is possible. But, for other than balanced and bi-directed graph models, it is difficult to synchronize nonlinear systems. Moreover, the performance of synchronization is normally dependent on the second largest eigenvalue of the laplacian matrix of the network graph. This research shows how to implement distributed nonlinear and adaptive controllers using pinning techniques for generalized directed communication graph models. The dynamics of the agent at each node are non-identical and unknown. A Lyapunov based technique is adopted to show the tracking performance when the tracker dynamics are also considered unknown. It is also shown using pinning adaptive control that the synchronization speed no longer depends on Fiedler eigenvalue of the graph laplacian matrix. The research also develops duality properties of linear controllers and observers for general cooperative di-graph systems. The choice of gains for controller and observer using riccati equations ensures stable synchronization on general di-graphs. Finally the research is extended to decentralized control of HVAC systems using pinning control methodology.

July 1, 2010

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF FIGURES xii

LIST OF TABLES xiv

Chapter Page

1. INTRODUCTION 1

1.1 Introduction 1

1.2 Approaches to Nonlinear and Adaptive Control 1

1.3 Motivation 3

1.4 Background 4

1.5 Contribution 10

1.6 Outline of Thesis 13

1.7 Publications 16

2. DYNAMIC INVERSION WITH ZERO-DYNAMICS STABILIZATION FOR QUADROTOR CONTROL 19

2.1 Introduction 19

2.2 Quadrotor Dynamics 21

2.3 Dynamic Inversion Inner Control Loop 26

2.3.1 FBLC linear controller 29

2.4 Desired Tracking Loop and Stabilization of the Internal Dynamics 31

2.5 Controller Structure and Stability Analysis 33

2.6 Simulation Results 35

2.6.1 Reference Trajectory Generation 35

2.6.2 Simulation Result Without Disturbances 37

2.6.2.1 From Initial Position at $(0, 0, 0)$ to Final Position at $(20, 5, 10)$ 37

2.6.2.2 From Initial Position at $(0, 5, 10)$ to Final Position at $(20, 5, 10)$ 38

2.6.3 Simulation Result with Unmodeled Input Disturbances 41

2.6.3.1 From Initial Position at $(0, 0, 0)$ to Final Position at $(20, 5, 10)$ 41

2.7 Conclusion 42

3. BACKSTEPPING APPROACH FOR CONTROLLING A QUADROTOR USING LAGRANGE FORM DYNAMICS 45

3.1 Introduction 45

3.2 Quadrotor Dynamics 48

3.3 Backstepping Control Design 51

3.4 Solution and Feasibility of $\hat{F}_d$ 58

3.5 Stability Analysis and NN Tuning 59

3.6 Simulation Result 63

3.6.1 Rotorcraft model 63

3.6.2 Neural Network Parameters 64

3.6.3 Effectiveness in Tracking a Desired (guided) Trajectory 64

3.6.4 Motion Decoupling in Response to Position Commands 65

3.6.5 Comparison Study: Adaptive Backstepping Proposed in this Research vs.The controller Adopted from [1] 67

3.7 Conclusion 71

4. DISTRIBUTED ADAPTIVE CONTROL FOR SYNCHRONIZATION OF UNKNOWN NONLINEAR NETWORKED SYSTEMS 73

4.1 Introduction 73

4.2 Synchronization Control Formulation 75

4.2.1 Synchronization Control Problem 76
4.2.2 Synchronization Error Dynamics 77
4.2.3 Synchronization Control Design 79
4.3 Lyapunov Design for Cooperative Systems: Distributed NN Tuning Protocols 81
4.4 Discussion 88
4.4.1 Relation of Error Bounds to Graph Structural Properties 91
4.5 Simulation Results 92
4.5.1 Nonlinear node dynamics and disturbances 93
4.5.2 Synchronization of Second-Order Dynamics 95
4.6 Conclusion 97
5. DISTRIBUTED ADAPTIVE CONTROL FOR SECOND ORDER SYSTEMS WITH UNKNOWN NONLINEARITY 99
5.1 Introduction 99
5.2 Synchronization Control Formulation 102
5.2.1 Cooperative Tracking Problem for Synchronization of Multiagent Systems 102
5.2.2 Synchronization Tracking Error 104
5.2.3 Synchronization Control Design and Error Dynamics 105
5.3 Lyapunov Design for Cooperative Adaptive Tracking Control 108
5.3.1 Discussion 116
5.4 Simulation Result 117
5.5 Conclusion 122
6. COOPERATIVE SYSTEMS STATE FEEDBACK, OBSERVERS, DUALITY, AND OPTIMAL DESIGN FOR SYNCHRONIZATION 123
6.1 Introduction 123
6.2 State Feedback Design for Cooperative Systems on Graphs 126

6.2.1 Graph Background and Notation 127
6.2.2 State Feedback for Synchronization of Networked Systems 127
6.2.3 Region of Consensus 132
6.2.4 Bounded Consensus Region for Arbitrarily Chosen Stabilizing SVFB Gain 133
6.2.5 Unbounded Consensus Region for Optimal SVFB Gain 134
6.3 Observer Design for Cooperative Systems on Graphs 135
6.4 Duality and Dynamic Output Feedback Tracker Design for Cooperative Systems 140
6.4.1 Control/Observer Duality on Graphs 140
6.5 Dynamic Tracker for Synchronization Using Output Feedback 142
6.6 Local vs. Global Variables and Selection of Pinned Nodes 144
6.7 Conclusion 148
7. DECENTRALIZED CONTROL OF HVAC SYSTEM 149
7.1 Introduction 149
7.2 Dynamic Modeling of the HVAC System- The Model Graph 151
7.3 Control of the HVAC System - The Control Graph 155
7.4 Calculating the Airflow at Nodes: Pipe Network Analysis 158
7.4.1 Pipe Network Geometry 158
7.4.2 Analysis of Distribution Mains 159
7.5 Lyapunov Proof for HVAC Temperature Consensus 163
7.6 Conclusion 166
8. CONCLUSION AND FUTURE WORK 167
8.1 Contribution 167
8.2 Future work 169
REFERENCES 170

BIOGRAPHICAL STATEMENT 183

LIST OF FIGURES

Figure		Page
2.1	Model of a quadrotor	22
2.2	Control block diagram	33
2.3	Example trajectory simulation for different final positions	37
2.4	Position tracking - simultaneous command in x, y and z	38
2.5	Resultant angular positions, errors and velocities	39
2.6	Input commands for Case 1	39
2.7	Position tracking (x command only)	40
2.8	Angular variations and rates (x command tracking)	40
2.9	Input commands for x - command tracking	41
2.10	Simultaneous command in x, y and z + Input disturbances	42
2.11	Angular variations, errors and velocities (with input disturbances)	43
2.12	Force and torque input variations (with input disturbances)	43
3.1	Control configuration	53
3.2	Three position commands simultaneously	65
3.3	Resultant angular positions and errors	66
3.4	Input commands for Case I	66
3.5	Plots of position and position tracking errors for command only	67
3.6	Resultant angular positions and errors	68
3.7	Input commands for variation in x (Case II)	68
3.8	Case III: Comparison result: position response	69
3.9	Case III: Comparison result: attitude response	70

3.10 Case III: Comparison result: input commands 71

4.1 5 node SC digraph with one leader node 92

4.2 Consensus tracking and input for all nodes 93

4.3 Consensus errors and NN weight estimation errors with distributed adaptive control 94

4.4 NN weight dynamics 95

4.5 State synchronization errors δ_i and NN estimation errors 96

4.6 Synchronized motion phase plane plot 97

5.1 Five-node SC digraph with one leader node 117

5.2 Tracking performance (position and velocity) and control input 119

5.3 Disagreement vector and NN estimation error vector 119

5.4 Actual and estimated $f(x)$ 120

5.5 NN weight dynamics 121

5.6 Phase plane plot for all the nodes for second order consensus 121

6.1 Bounded consensus region for arbitrarily chosen stabilizing SVFB gain . . 134

6.2 Unbounded consensus region for optimal SVFB gain 135

7.1 Distributed HVAC system showing air diffuser vents, control valves, and thermostats 152

7.2 Uniform mesh imposed on spatial coordinates (x, y) for heat equation numerical solution 153

7.3 A typical pipe network for HVAC system in a room 159

7.4 A typical pumping distribution main 161

LIST OF TABLES

Table Page

4.1 Comparison of $f(x)$ and $\hat{f}(x)$ 94

CHAPTER 1

INTRODUCTION

1.1 Introduction

This introductory chapter discusses motivation, background and contribution of the dissertation. The list of publications which resulted from this research is given in Section 1.7.

1.2 Approaches to Nonlinear and Adaptive Control

Nonlinear and adaptive control is a tool that is used to handle nonlinear systems with or without unknown dynamics and model uncertaities. As most practical systems are nonlinear in nature it is often proved to be crucial to use nonlinear adaptive control techniques to obtain a stable closed loop system which is sometime very hard to achieve with known linear control techniques. Some of well-known nonlinear techniques include feedback linearization and dynamic inversion, backstepping, sliding mode control. The inclusion of NN (Neural Net) in the closed loop control is a typical example of an adaptive control. An overview of nonlinear and adaptive control techniques is given in the seminal paper [2] which has inspired many of our own ideas. Feedback state linearization and dynamic inversion (similar to input-output linearization) is one of the popular methods among engineers to tackle plan nonlinearities. The idea is to transform the nonlinear system into a linear system through the choice of appropriate input. The transformed system can be represented in normal form via coordinate transformation. As long as the transformation is diffiomorphism, the transformed system will remain smooth in the new coordinate if the system in original coordinate is smooth. Once

the system is linearized, then all linear control strategies become applicable easily. If the state space system is square then dynamic inversion can be used if input-to-output system is invertible.

Backstepping is another effective and popular technique which is similar to feedback linearization but in contrast to feedback linearization, backstepping does not need any cancellation of nonlinearities which might be useful for stability. The backstepping is easy to implement for nonlinear systems which is already in cascaded form. For higher order systems, the recursive application of backstepping leads to a strict feedback system which is easy to stabilize. But the techniques like feedback linearization and backstepping are only effective and useful if plant dynamics is known. In practical scenario the plant model typically comes with input disturbances, unknown dynamics and model uncertainties. A robust control approach like sliding mode control could be effective to confront these problems. With uncertainties in the plant model, it is often required to choose multiple control structure which allows the closed loop system to slide along a single favorable trajectory. The control signal may be discontinuous and the trajectory only exists for all control signals across the trajectory. The main benefit is that one does not need to know whole system dynamics to design either control signal or the trajectory. Sliding mode control, for its typical nature, is also known as variable structure control. To maintain the system slide across the favorable trajectory, the control signal often results in chattering.

If designed properly, a NN (neural net) can effectively estimate the unknown dynamics and disturbances and thereby use of chattering sliding mode control can be avoided. The design of NN can be static or dynamic which is also known as adaptive in nature. The most popular one is adaptive because it gradually becomes more efficient in estimating exact nonlinearities hidden in the plant model. The backbone of the NN consists of artificial neurons which has similar working principle as actual neurons in

human body. The number of neurons associated with the NN design as well as their adaptive weight tuning laws is the key to develop a successful adaptive control strategy. In general, selecting a large number of neurons requires huge processing power that often becomes very hard to implement. The complexity of the overall closed loop control architecture increases in the price of estimation accuracy. But a well-designed implementable adaptive control can be very effective for controlling complex nonlinear plants such as manned and unmanned aerospace vehicles.

1.3 Motivation

Most previous research that develops nonlinear and adaptive control methods for nonlinear and multiagent systems include:

- Adaptive backstepping for state space model of quadrotor.
- Dynamic inversion techniques for nonlinear quadrotor model without stabilizing internal dynamics.
- Linear protocols for simple integrator dynamics multiagent systems.
- Nonlinear adaptive control for nonlinear multiagent systems with undirected graph topology where all the agent dynamics are same.

This work shows the following innovative and improved aspects:

> Adaptive backstepping for quadrotor helicopters with Lagrangian dynamics where the states appear in bilinear fashion.

> A new approach to dynamic inversion which provides stability of internal dynamics of the quadrotor helicopter.

> Nonlinear adaptive control techniques for multiagent systems where agent dynamics are nonlinear and unknown. The network topology is considered to be a strongly connected di-graph which is more generalized version of undirected graph topology. Moreover at each node the agent dynamics need not to be same.

- Extension of above for second-order systems which makes these methods applicable to more general situations such as formation control.
- A duality property for controller and observer design is shown for generalized cooperative di-graphs under limited information exchange. This provides new results.
- Application of pinning control for di-graph is developed for Heating, Ventilating and Air Conditioning (HVAC) systems.

1.4 Background

Nowadays, unmanned rotorcraft are designed to operate with great agility and rapid maneuvering, and are capable of work in degraded environments such as wind conditions etc. The control of these rotorcraft is a subject of research especially in applications such as rescue, surveillance, inspection, mapping etc. For these applications the ability of the rotorcraft to maneuver sharply and hover precisely is important [3]. Rotorcraft control in these applications often requires holding a particular trimmed state, generally hover, as well as making changes of velocity and acceleration in a desired way [4]. Similar to aircraft control, rotorcraft control too involves controlling the pitch, yaw, and roll motion. But the main difference is that, due to the unique body structure of rotorcraft (as well as the rotor dynamics and other rotating elements) the pitch, yaw and roll dynamics are strongly coupled. Therefore, it is difficult to design decoupled control laws of sound structure that stabilize the faster and slower dynamics simultaneously. On the contrary, for a fixed wing aircraft it is relatively easy to design decoupled standard control laws with intuitively comprehensible structure and guaranteed performance [5]. There are many different approaches available for rotorcraft control such as [6, 7, 8, 9, 10] etc. Popular methods include input-output linearization and backstepping.

The 6-DOF airframe dynamics of a typical quadrotor involve the typical translational and rotational dynamical equations as in [4, 1]. The dynamics of a quadrotor is essentially a simplified form of helicopter dynamics that exhibits the basic problems including underactuation, strong coupling, multi-input/multi-output, and unknown nonlinearities. The quadrotor is classified as a rotorcraft where lift is derived from the four rotors. Most often they are classified as helicopters as its movements are characterized by the resultant force and moments of the four rotors. Therefore the control algorithms designed for a quadrotor could be applied to a helicopter with relatively straightforward modifications. Most of the papers [7, 8, 10] etc deal with either input-output linearization for decoupling pitch yaw roll or backstepping to deal with the underactuation problem. The problem of coupling in the yaw- pitch-roll of a helicopter, as well as the problem of coupled dynamics-kinematic underactuated system, can be solved by backstepping [11, 12, 13]. Dynamic inversion [5] is effective in the control of both linear and nonlinear systems and involves an inner inversion loop (similar to feedback linearization) which results in tracking if the residual or internal dynamics is stable. Typical usage requires the selection of the output control variables so that the internal dynamics is guaranteed to be stable. This implies that the tracking control cannot always be guaranteed for the original outputs of interest.

The application of dynamic inversion to UAV's and other flying vehicles such as missiles, fighter aircrafts etc is proposed in several research works such as [14, 15, 16] etc. It is also shown that the inclusion of dynamic neural networks for estimating the dynamic inversion errors can improve the controller stability and tracking performance. Some other papers such as [17, 18, 19, 20, 21] etc discuss the application of dynamic inversion to nonlinear systems to tackle model and parametric uncertainties using neural nets. It is also shown that a reconfigurable control law can be designed for fighter aircraft using neural net and dynamic inversion. Sometimes the inverse transformations

required in dynamic inversion or feedback linearization are computed by neural network to reduce the inversion error by online learning.

Specifically, [1, 22, 23] considered the Lagrangian model of a quadrotor and derived what was in effect a backstepping controller with 3 loops: heading/ motion, motion, and motion. Nested saturation control was used in each loop. Exact compensation was assumed in each loop. The control law was tested on a practical quadrotor model. References [24] compared the performance of a proportional-integral-derivative (PID) controller with a LQ control law on a micro vertical take-off and landing (VTOL) test bench. These papers presented a good accurate dynamic model, and discussed the mechanical structure and sensors of the quadrotor. Reference [25] discussed the robust trajectory tracking of a Lagrangian quadrotor model using backstepping control and proved the stability for any given but bounded trajectory. A nonlinear type robust control law was used in [10, 26], with [26] demonstrating that the robust hover controller can handle large parameter uncertainty with respect to their nominal values.

A popular method for handling unknown nonlinearities is to introduce neural networks, tuned online using adaptive control techniques. Reference [27] used an adaptive dynamic programming (ADP) based controller. Reference [16] used dynamic inversion with neural networks to account for unknown dynamics and dynamic inversion errors for a helicopter model. Reference [28] used an innovative method known as pseudo-control hedging (PCH) with a single neural network employed for both inner and outer loops to account for unknown dynamics and bounded control inputs. Reference [29] used backstepping with neural networks for an unmanned air vehicle to confront modeling errors and unmodeled dynamics. It was based on a nonlinear state space model of an aircraft. Reference [30] presented a Lagrangian dynamic model of X4-flyer with motor dynamics and gyroscopic effects included. References [9, 7] used a feedback linearization approach. Reference [6] presented different control architectures

such as feedback linearization, backstepping based on visual feedback as the primary sensor. Reference [31] discussed the dynamic modeling of a unmanned aerial vehicle (UAV) Dragan flyer III and a loop shaping flight controller as a position control law. Reference [32] presented a quasioptimal trajectory planner with a LQR path following controller. Reference [33] developed a simulation tool for an indoor flight-capable quadrotor based on MEMS sensors and rangefinders.

Most work in quadrotor control that uses backstepping design or other nonlinear control formulations like feedback linearization are based on state-variable formulations. This results in control laws that are complex and often require the evaluation of lie derivatives. Reference [34] does perform backstepping on the Euler-Lagrange dynamics, but backstepping is performed on a virtual input that is a bilinear product involving thrust input. This results in complications and extra states are added.

Coordination and consensus of distributed groups of agents is inspired by naturally occurring phenomena such as flocking in birds, swarms in insects, circadian rhythms in nature, synchronization and phase transitions in physical and chemical systems, and the laws of thermodynamics [35]. Early work has been done in the control systems community by [36, 37, 38, 39, 40], which by now are well known. Consensus has been studied for systems on communication graphs with fixed or varying topologies and communication delays. The average consensus problem has garnered much interest. Synchronization to time-varying trajectories has been studied based on physical or natural systems by [41, 42, 43, 44]. Synchronization of nonlinear passive dynamical systems has been studied by [45]. Consensus using nonlinear protocols has been considered there and in [35].

Convergence of consensus to a virtual leader or header node has been studied in [39, 40, 41, 42, 43, 44, 45, 46].Dynamic consensus for tracking of time-varying signals has been presented in [47].Recently, the pinning control has been introduced for synchro-

nization control of coupled complex dynamical systems [48, 49, 50, 51].Pinning control is a powerful technique that allows controlled synchronization of interconnected dynamical systems by adding a control or leader node that is connected (pinned) into a small percentage of nodes in the network. These pinned or controlled nodes view the control node simply as another neighbor, and consider the control node's state value in computing their local protocols. Analysis shows that all nodes converge to the state of the control node, which may be time-varying. Analysis has been done using Lyapunov techniques by assuming either a Jacobian linearization of the nonlinear node dynamics or a Lipschitz condition. A related idea is soft control [52] where a shill node moves through the network, and is perceived by existing nodes simply as another neighbor for purposes of computation of their own averaging protocols. Proper placement and motion of the shill agent results in consensus to the state of the shill.

Consensus and collective motion of distributed agents has been analyzed using the theory of graphs and/or Markov processes. Recent publications allow analysis using traditional control theory notions including matrix analysis, Lyapunov theory, etc., upon the introduction of certain key definitions including irreducibility, M-matrices, Frobenius form, special Lyapunov forms, etc. Notable are the books [53] and [54]. Such techniques allow one to bring in the machinery of matrix analysis [55]. Instrumental in this analysis are the techniques employed in [56].

Distributed multiagent systems with unknown nonlinear dynamics and disturbances were studied in [57] where distributed adaptive controllers were designed to achieve robust consensus. That treatment assumed undirected graphs and solved the consensus problem, that is, the nodes reach a steady-state consensus that depends on the initial conditions. Expressions for the consensus value were not given.

The higher order consensus is also being studied at the same time. The study of second order as well as higher order consensus is required to implement the syn-

chronization in most real life situations like coordination among UAVs etc. [58] studied the case of higher order consensus for chain integrator system. The derivation proposed in the paper is mainly for 3rd order system which is claimed true for more general case. The necessary and sufficient condition given in the paper is dependent on the multiplicity of the zero eigenvalues of graph laplacian matrix. Some other papers including [59, 60, 61] discuss the higher order consensus with time delays for integrator dynamics. [62] proposes the second order consensus with identical linear systems for all agents and the formulation is also dependent on eigenvalues of graph laplacian matrix which are sometime difficult to analyze. [57] shows an extension of adaptive backstepping techniques for second order system connected in undirected manner. The formulation is again involving a Lyapunov candidate with graph laplacian which is not considered as general case for any graph. A wide range of studies can be found in the literature regarding indoor air distributions.[63] demonstrates a numerical method to control air distribution inside a office building. It is shown that the efficiency of HVAC system decreases along with the increase in number of occupants in the building. The paper also investigates the characteristics of air to be distributed inside the building for quality thermal consensus among the occupants. Other related papers includes [64, 65, 66, 67, 68]. These papers investigate the method of air distribution from different inlet and outlet positions. Natural convection from heated room surfaces is studied in [65]. The HVAC with digital control system using computer program is studied by [69, 70, 71] demonstrate the use of decentralized control for thermal consensus in HVAC system.

1.5 Contribution

This thesis makes two major contributions in two different fields of study.

1. First, in the field of aerospace control, it develops new adaptive and nonlinear autopilots for the quadrotor helicopter. These new approaches not only show improved performance in comparison with past works but also deliver key aspects of implementation in practical models.
2. Second, in the field of cooperative graph and multiagent systems, it develops nonlinear adaptive control for synchronization of heterogeneous multiagent systems where the nonlinear dynamic models of the agents are unknown. It also develops the duality properties in cooperative graph networks of agents with linear dynamics

This work presents, in a continuous-time framework, new formulations of nonlinear and adaptive control techniques for aerospace systems as well as for generalized heterogeneous cooperative graph topology.

The contributions of this thesis are the following:

1. A dynamic inversion technique with internal dynamics stabilization for highly nonlinear underactuated aerospace systems such as quadrotor helicopters.
2. An adaptive backstepping approach for quadrotor helicopters with Lagrangian dynamics where the system states appear in bilinear fashion.
3. Nonlinear adaptive control for synchronization of first order unknown nonlinear multiagent systems.
4. Nonlinear adaptive control for synchronization of second order unknown nonlinear systems. This can be extended easily to higher order systems.
5. Duality properties for controller and observer design in cooperative graph.

The first contribution illustrates a new approach for dynamic inversion technique [72]. For a quadrotor, one can identify the two well known inherent rotorcraft charac-

teristics; underactuation and strong coupling in pitch-yaw-roll. To confront these problems and design a station-keeping and tracking controller, dynamic inversion is used in this research. Typical applications of dynamic inversion require the selection of the output control variables to render the internal dynamics stable. This means that in many cases, perfect tracking cannot be guaranteed for the actual desired outputs. Instead, here the internal dynamics of the feedback linearized system is stabilized using a robust control term. Unlike standard dynamic inversion, the linear controller gains are chosen uniquely to satisfy the tracking performance. Stability and tracking performance are guaranteed using a Lyapunov-type proof. Simulation with a typical nonlinear quadrotor dynamic model is performed to show the effectiveness of the designed control law in the presence of input disturbances.

The second contribution illustrates the adaptive backstepping techniques for quad - rotor helicopters [73]. Control design for the quadrotor is more tractable yet reveals corresponding approaches for helicopter and UAV control design. In this research, a backstepping approach is used for quadrotor controller design. In contrast to most other approaches, we apply backstepping on the Lagrangian form of the dynamics, not the state space form. This is complicated by the fact that the Lagrangian form for the position dynamics is bilinear in the controls. We confront this problem by using an inverse kinematics solution akin to that used in robotics. In addition, two neural nets are introduced to estimate the aerodynamic components, one for aerodynamic forces and one for aerodynamic moments. The result is a controller of intuitively appealing structure having an outer kinematics loop for position control and an inner dynamics loop for attitude control. The control approach described in this research is robust since it explicitly deals with unmodeled state-dependent disturbances and forces without needing any prior knowledge of the same. A simulation study validates the results obtained in the research.

The third contribution illustrates the application of nonlinear adaptive control technique for first order multiagent systems [74]. Traditional research on cooperative networked systems has focused on analysis of given protocols for linear systems, mainly the integrator or double integrator dynamics. Pinning control has been developed for nonlinear systems, generally using either a Jacobian linearization method or a Lipschitz assumption. In this research we present a design method for adaptive controllers for distributed systems having non-identical unknown nonlinear dynamics, and for a target dynamics to be tracked that is also nonlinear and unknown. The development is for general digraph communication structures. A Lyapunov technique is presented for designing a robust synchronization control protocol. The proper selection of the Lyapunov function is the key to ensuring that the resulting control laws thus found are implementable in a distributed fashion. Lyapunov functions are defined in terms of a local neighborhood tracking synchronization error and the Frobenius norm. The resulting protocol consists of a linear protocol and a nonlinear control term with adaptive update law at each node. Singular value analysis is used. It is shown that the singular values of certain key matrices are intimately related to structural properties of the graph.

The fourth contribution extends the above mentioned methodologies for higher order multiagent systems [75]. This research extends the result for generalized strongly connected network structure with unknown agent and tracker dynamics of order two. A lyapunov based proof shows the ultimate boundedness of the tracking error. The proposed theory can easily extended to higher order dynamics.

The fifth contribution develops a new theory in the field of cooperative graph and linear multi agent systems [76]. This research studies synchronization using reduced state information where the nodes are only allowed to pass measured output information to their neighbors. The node states are estimated using cooperative dy-

namical observers at each node. It is shown how to design cooperative state variable state feedbacks and observers at each node that are dual to each other, in a sense that extends the classical system theory notion of duality to dynamical systems on graphs. It is shown that unbounded consensus regions that guarantee synchronization on arbitrary strongly connected digraphs can be guaranteed by using OPTIMAL CONTROL AND OBSERVER design methods at each node based on Riccati equations. A duality theorem for systems on graphs is presented and it is shown how to design cooperative dynamical regulators that use output feedback for tracking the state of a control or leader node. Finally, it will be shown that the pinned or controlled nodes in a network should be selected as nodes of large out-degree (highest *social standing*) that are far apart from each other in the communication graph.

1.6 Outline of Thesis

Chapter 2 illustrates a new approach for dynamic inversion for rotorcraft UAV. For a quadrotor, one can identify the two well known inherent rotorcraft characteristics; underactuation and strong coupling in pitch-yaw-roll. To confront these problems and design a station-keeping and tracking controller, dynamic inversion is used in this research. Typical applications of dynamic inversion require the selection of the output control variables to render the internal dynamics stable. This means that in many cases, perfect tracking cannot be guaranteed for the actual desired outputs. Instead, here the internal dynamics of the feedback linearized system is stabilized using a robust control term. Unlike standard dynamic inversion, the linear controller gains are chosen uniquely to satisfy the tracking performance. Stability and tracking performance are guaranteed using a Lyapunov-type proof. Simulation with a typical nonlinear quadrotor dynamic model is performed to show the effectiveness of the designed control law in the presence of input disturbances.

Chapter 3 develops adaptive backstepping techniques for quadrotor helicopters. In this research, a backstepping approach is used for quadrotor controller design. In contrast to most other approaches, we apply backstepping on the Lagrangian form of the dynamics, not the state space form. This is complicated by the fact that the Lagrangian form for the position dynamics is bilinear in the controls. We confront this problem by using an inverse kinematics solution akin to that used in robotics. In addition, two neural nets are introduced to estimate the aerodynamic components, one for aerodynamic forces and one for aerodynamic moments. The result is a controller of intuitively appealing structure having an outer kinematics loop for position control and an inner dynamics loop for attitude control. The control approach described in this research is robust since it explicitly deals with unmodeled state-dependent disturbances and forces without needing any prior knowledge of the same. A simulation study validates the results obtained in the research.

Chapter 4 illustrates the application of nonlinear adaptive control technique for first order multiagent systems. Traditional research on cooperative networked systems has focused on analysis of given protocols for linear systems, mainly the integrator or double integrator dynamics. Pinning control has been developed for nonlinear systems, generally using either a Jacobian linearization method or a Lipschitz assumption. In this research we present a design method for adaptive controllers for distributed systems having non-identical unknown nonlinear dynamics, and for a target dynamics to be tracked that is also nonlinear and unknown. The development is for general digraph communication structures. A Lyapunov technique is presented for designing a robust synchronization control protocol. The proper selection of the Lyapunov function is the key to ensuring that the resulting control laws thus found are implementable in a distributed fashion. Lyapunov functions are defined in terms of a local neighborhood tracking synchronization error and the Frobenius norm. The resulting protocol consists

of a linear protocol and a nonlinear control term with adaptive update law at each node. Singular value analysis is used. It is shown that the singular values of certain key matrices are intimately related to structural properties of the graph.

Chapter 5 extends the above mentioned methodologies for higher order multiagent systems. This research extends the result for generalized strongly connected network structure with unknown agent and tracker dynamics of order two. A lyapunov based proof shows the ultimate boundedness of the tracking error. The proposed theory can easily extended to higher order dynamics.

Chapter 6 develops a new theory in the field of cooperative graph and linear multi agent systems. This research studies synchronization using reduced state information where the nodes are only allowed to pass measured output information to their neighbors. The node states are estimated using cooperative dynamical observers at each node. It is shown how to design cooperative state variable state feedbacks and observers at each node that are dual to each other, in a sense that extends the classical system theory notion of duality to dynamical systems on graphs. It is shown that unbounded consensus regions that guarantee synchronization on arbitrary strongly connected digraphs can be guaranteed by using OPTIMAL CONTROL AND OBSERVER design methods at each node based on Riccati equations. A duality theorem for systems on graphs is presented and it is shown how to design cooperative dynamical regulators that use output feedback for tracking the state of a control or leader node. Finally, it will be shown that the pinned or controlled nodes in a network should be selected as nodes of large outdegree (highest *social standing*) that are far apart from each other in the communication graph.

In Chapter 7 decentralized control for HVAC system is discussed. The model of HVAC system is derived from the basic poison heat equations and air flow dynamics in pipe network. The mathematical modeling is also include selection of neighborhood

for each node associated for control purpose. Then the pinning control topology is introduced. It is shown from Lyapunov perspective that the thermal synchronization is possible among the nodes inside the building by choosing a suitable control input.

Chapter 8 presents Conclusion and future work.

1.7 Publications

Book Chapters

1. A. Das, F. Lewis, `Distributed Adaptive Control for Networked Multi Robot Systems`, appeared as book chapter in Multi Robot Systems, Trends and Development, INTECH, 2010.
2. A. Das, F. Lewis and S. Subbarao, `Back-stepping Control of Quadrotor, A Higher OrderNeural Network Approach`, published as a book chapter in Artificial Higher Order Neural Networks for Computer Science and Engineering: Trends for Emerging Applications, 2010.
3. A. Das, F. Lewis and S. Subbarao, `Quadrotor Control by Dynamic Inversion with Zero Dyanmics Stabilization`, published as book chapter in Control & Computer Vision in Robotics & Automation, Narosa Publisher, New Delhi, India.

Journal Papers

1. A. Das, F. Lewis, `Distributed Adaptive Control for Consensus of Unknown Second Order Nonlinear Systems`, International Journal of Robust and Nonlinear Control, Volume 21, Issue 3, pp 1493-1610, 26 August, 2010, DOI: 10.1002/rnc.1647.

2. F. Lewis, A. Das, `Cooperative System State Feedback, Observers and Duality Optimal Design for Synchronization`, IEEE Transaction of Automatic Control, Volume 56 Issue 8, pp 1948 - 1952, 07 April 2011.
3. A. Das, F. Lewis, `Distributed Adaptive Control for Synchronization of Unknown Nonlinear Networked Systems`, Automatica, Volume 46, Issue 12, December 2010, Pages 2014-2021.
4. A. Das, F. Lewis and S. Subbarao, `Backstepping Approach for Controlling a Quadrotor using Neural Network`, Special Issue of *Journal of Intellegent and Robotics Systems*, vol 56, Issue 1, pp 127, Springer, April 2009.
5. A. Das, S. Subbarao and F. Lewis, `Dynamic Inversion with Zero Dynamics Stabilization for Quadrotor Control`, *IET Control Theory & Applications*, 2009, vol. 3, Iss. 3, pp. 303-314.

Conference Papers

1. F. Lewis, A. Das, `Distributed State Feedback, Observer and Duality for Multiagent Systems`, proceedings on Decision and Control (CDC), pp 7265-7271, Atlanta, GA, 2010.
2. A. Das, F. Lewis, `Lyapunov Design for Synchronization of Unknown Nonlinear Networked Systems`, invited paper for 8th IEEE International Conference on Control & Automation, Xiamen, China, June 9-11, 2010.
3. Chaoyong Li, Zhihua Qu, Abhijit Das, Frank L. Lewis, `Cooperative Control With Improvable Network Connectivity`, Proceedings of American Control Conference, pp 87-92, Baltimore, MD, July 29, 2010.
4. A. Das, S. Subbarao and F. Lewis, `Dynamic Inversion of Quadrotor with Zero-Dynamic Stabilization`, Proceedings of the 2008 IEEE Multicon-

ference on Systems and Control, pp 1189-1194, September 3-5, 2008, San Antonio, Texas.

5. A. Das, F. Lewis and S. Subbarao, `Dynamic Neural Network based Robust Backstepping Control approach for Quadrotors`, Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit, 18-21 August, 2008, Honolulu, Hawaii.

CHAPTER 2

DYNAMIC INVERSION WITH ZERO-DYNAMICS STABILIZATION FOR QUADROTOR CONTROL

2.1 Introduction

Nowadays, unmanned rotorcraft are designed to operate with great agility and rapid maneuvering, and are capable of work in degraded environments such as wind conditions etc. The control of these rotorcraft is a subject of research especially in applications such as rescue, surveillance, inspection, mapping etc. For these applications the ability of the rotorcraft to maneuver sharply and hover precisely is important [3]. Rotorcraft control in these applications often requires holding a particular trimmed state, generally hover, as well as making changes of velocity and acceleration in a desired way [4]. Similar to aircraft control, rotorcraft control too involves controlling the pitch, yaw, and roll motion. But the main difference is that, due to the unique body structure of rotorcraft (as well as the rotor dynamics and other rotating elements) the pitch, yaw and roll dynamics are strongly coupled. Therefore, it is difficult to design decoupled control laws of sound structure that stabilize the faster and slower dynamics simultaneously. On the contrary, for a fixed wing aircraft it is relatively easy to design decoupled standard control laws with intuitively comprehensible structure and guaranteed performance [5]. There are many different approaches available for rotorcraft control such as [6, 7, 8, 9, 10] etc. Popular methods include input-output linearization and backstepping.

The 6-DOF airframe dynamics of a typical quadrotor involves the typical translational and rotational dynamical equations as in [4], [1]. The dynamics of a quadrotor is

essentially a simplified form of helicopter dynamics that exhibits the basic problems including underactuation, strong coupling, multi-input/multi-output, and unknown nonlinearities. The quadrotor is classified as a rotorcraft where lift is derived from the four rotors. Most often they are classified as helicopters as its movements are characterized by the resultant force and moments of the four rotors. Therefore the control algorithms designed for a quadrotor could be applied to a helicopter with relatively straightforward modifications. Most of the papers [7, 8, 10] etc deal with either input-output linearization for decoupling pitch yaw roll or backstepping to deal with the underactuation problem. The problem of coupling in the yaw- pitch-roll of a helicopter, as well as the problem of coupled dynamics-kinematic underactuated system, can be solved by backstepping [11, 77, 13]. Dynamic inversion [5] is effective in the control of both linear and nonlinear systems and involves an inner inversion loop (similar to feedback linearization) which results in tracking if the residual or internal dynamics is stable. Typical usage requires the selection of the output control variables so that the internal dynamics is guaranteed to be stable. This implies that the tracking control cannot always be guaranteed for the original outputs of interest.

The application of dynamic inversion on UAV's and other flying vehicles such as missiles, fighter aircrafts etc are proposed in several research works such as [14, 15, 16] etc. It is also shown that the inclusion of dynamic neural network for estimating the dynamic inversion errors can improve the controller stability and tracking performance. Some other papers such as [17, 18, 19, 20, 21] etc discuss the application of dynamic inversion on nonlinear systems to tackle the model and parametric uncertainties using neural nets. It is also shown that a reconfigurable control law can be designed for fighter aircrafts using neural net and dynamic inversion. Sometimes the inverse transformations required in dynamic inversion or feedback linearization are computed by neural network to reduce the inversion error by online learning.

In this Chapter we apply dynamic inversion to tackle the coupling in quadrotor dynamics which is in fact an underactuated system [72]. Dynamic inversion is applied to the inner loop, which yields internal dynamics that are not necessarily stable. Instead of redesigning the output control variables to guarantee stability of the internal dynamics, we use a robust control approach to stabilize the internal dynamics. This yields a two-loop structured tracking controller with a dynamic inversion inner loop and an internal dynamics stabilization outer loop. Section 2.2 of this Chapter discusses the basic quadrotor dynamics which is used for control law formulation. Section 2.3 shows dynamic inversion of a nonlinear state-space model of a quadrotor. Sections 2.5 discuss the robust control method to stabilize the internal dynamics. In the final Section 2.6, simulation results are shown to validate the control law discussed in this chapter.

2.2 Quadrotor Dynamics

Fig. 2.1 shows a basic model of an unmanned quadrotor. The quadrotor has some basic advantage over the conventional helicopter. Given that the front and the rear motors rotate counter-clockwise while the other two rotate clockwise, gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight. This four-rotor rotorcraft does not have a swashplate [22]. In fact it does not need any blade pitch control. The collective input (or throttle input) is the sum of the thrusts of each motor (see Fig. 2.1). Pitch movement is obtained by increasing (reducing) the speed of the rear motor while reducing (increasing) the speed of the front motor. The roll movement is obtained similarly using the lateral motors. The yaw movement is obtained by increasing (decreasing) the speed of the front and rear motors while decreasing (increasing) the speed of the lateral motors [24].

In this section we will describe the basic state-space model of the quadrotor. The dynamics of the four rotors are relatively much faster than the main system and

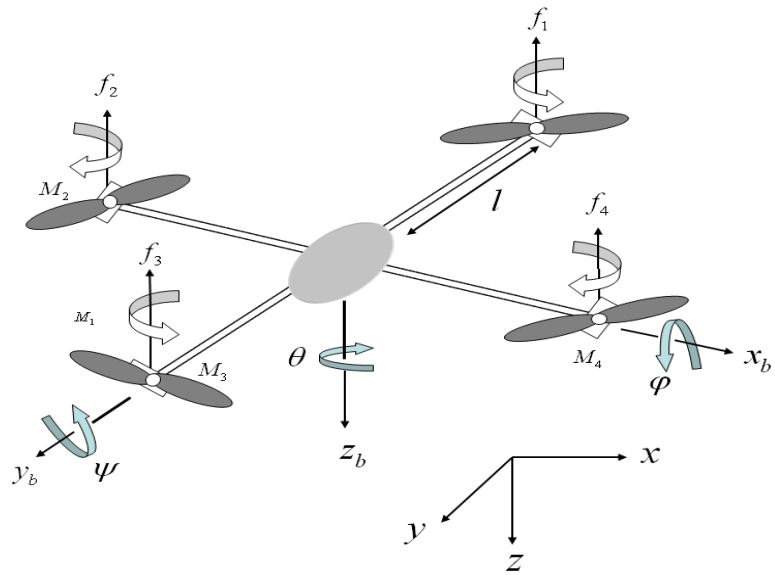

Figure 2.1. Model of a quadrotor.

thus neglected in our case. The generalized coordinates of the rotorcraft are $q = (x, y, z, \psi, \theta, \varphi)$, where (x, y, z) represents the relative position of the center of mass of the quadrotor with respect to an inertial frame $\Im$, and (ψ, θ, φ) are the three *Euler* angles representing the orientation of the rotorcraft, namely yaw-pitch-roll of the vehicle.

Let us assume that the transitional and rotational coordinates are in the form $\xi = (x, y, z)^T \in R^3$ and $\eta = (\psi, \theta, \varphi) \in R^3$. Now the total transitional kinetic energy of the rotorcraft will be $T_{trans} = \frac{m}{2}\dot{\xi}^T\dot{\xi}$ where m is the mass of the quadrotor. The rotational kinetic energy is described as $T_{rot} = \frac{1}{2}\dot{\eta}^T J\dot{\eta}$, where matrix $J = J(\eta)$ is the auxiliary matrix expressed in terms of the generalized coordinates η . The potential energy in the system can be characterized by the gravitational potential, described as $U = mgz$. Defining the Lagrangian $L = T_{trans} + T_{rot} - U$, where $T_{trans} = (m/2)\dot{\xi}^T\dot{\xi}$ is the translational kinetic energy, $T_{rot} = (1/2)\omega^T I\omega$ is the rotational kinetic energy with

ω as angular speed, $U = mgz$ is the potential energy, z is the quadrotor altitude, I is the body inertia matrix, and g is the acceleration due to gravity.

Then the full quadrotor dynamics is obtained as a function of the external generalized forces $F = (F_\xi, \tau)$ from

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = F \tag{2.1}$$

The principal control inputs are defined as follows. Define

$$F_R = \begin{pmatrix} 0 \\ 0 \\ u \end{pmatrix} \tag{2.2}$$

where the main thrust is

$$u = f_1 + f_2 + f_3 + f_4 \tag{2.3}$$

and f_i' s are described as $f_i = k_i\omega_i^2$, where k_i are positive constants and ω_i are the angular speed of the motor i . Then F_ξ can be written as

$$F_\xi = RF_R \tag{2.4}$$

where R is the transformation matrix representing the orientation of the rotorcraft as

$$R = \begin{pmatrix} c_\theta c_\psi & s_\psi s_\theta & -s_\theta \\ c_\psi s_\theta s_\varphi - s_\psi c_\varphi & s_\psi s_\theta s_\varphi + c_\psi c_\varphi & c_\theta s_\varphi \\ c_\psi s_\theta c_\varphi + s_\psi s_\varphi & s_\psi s_\theta c_\varphi - c_\psi s_\varphi & c_\theta c_\varphi \end{pmatrix} \tag{2.5}$$

The generalized torques for the η variables are

$$\tau = \begin{pmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\varphi \end{pmatrix} \tag{2.6}$$

where

$$\tau_\psi = \sum_{i=1}^{4} \tau_{M_i} = c(f_1 - f_2 + f_3 - f_4) \tag{2.7}$$

$$\tau_\theta = (f_2 - f_4)l \tag{2.8}$$

$$\tau_\varphi = (f_3 - f_1)l \tag{2.9}$$

Thus the control distribution from the four actuator motors of the quadrotor is given by

$$\begin{pmatrix} u \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ -l & 0 & l & 0 \\ 0 & l & 0 & -l \\ c & -c & c & -c \end{pmatrix}}_{C} \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}}_{f} \tag{2.10}$$

where l is the distance from the motors to the center of gravity, τ_{M_i} is the torque produced by motor M_i , c and is a constant known as force-to-moment scaling factor. So, if a required thrust and torque vector are given, one may solve for the rotor force using (2.10).

The final dynamic model of the quadrotor looks as

$$m\ddot{\xi} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = F_R \tag{2.11}$$

$$J(\eta)\ddot{\eta} + \frac{d}{dt}\{J(\eta)\}\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial\eta}(\dot{\eta}^T J(\eta)\dot{\eta}) = \tau \tag{2.12}$$

$$J(\eta)\ddot{\eta} + \frac{d}{dt}\{J(\eta)\}\dot{\eta} - \bar{C}(\eta,\dot{\eta}) = \tau \tag{2.13}$$

$$J(\eta)\ddot{\eta} + C(\eta,\dot{\eta}) = \tau \tag{2.14}$$

where, $F_R = u \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\varphi \\ \cos\theta\cos\varphi \end{pmatrix}$, auxiliary Matrix $\{J(\eta)\} = J = T_\eta^T I T_\eta$ with $T_\eta = \begin{pmatrix} -\sin\theta & 0 & 1 \\ \cos\theta\sin\psi & \cos\psi & 0 \\ \cos\theta\cos\psi & -\sin\psi & 0 \end{pmatrix}$.

Now finally the dynamic model of the quadrotor in terms of position (x, y, z) and rotation (φ, θ, ψ) is written as,

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \frac{1}{m} \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\varphi \\ \cos\theta\cos\varphi \end{pmatrix} u \tag{2.15}$$

$$\begin{pmatrix} \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = f(\varphi, \theta, \psi) + g(\varphi, \theta, \psi)\tau \tag{2.16}$$

where, $f(\varphi, \theta, \psi) = \begin{pmatrix} \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_p}{I_x}\dot{\theta}\Omega \\ \dot{\varphi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_p}{I_y}\dot{\varphi}\Omega \\ \dot{\varphi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) \end{pmatrix}$, $g(\varphi, \theta, \psi) = \begin{pmatrix} \frac{l}{I_x} & 0 & 0 \\ 0 & \frac{l}{I_y} & 0 \\ 0 & 0 & \frac{l}{I_z} \end{pmatrix}$, $u \in R^1$ and $\tau = \begin{bmatrix} \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \in R^3$ are the control inputs, $I_{x,y,z}$ are body inertia, J_p is propeller/rotor inertia and $\Omega = \omega_2 + \omega_4 - \omega_1 - \omega_3$. Thus, the system is the form of an underactuated system with six outputs and four inputs.

2.3 Dynamic Inversion Inner Control Loop

Dynamic inversion [5] is an approach where a feedback linearization loop is applied to the tracking outputs of interest. The residual dynamics, not directly controlled, is known as the internal dynamics. If the internal dynamics are stable, dynamic inversion is successful. Typical usage requires the selection of the output control variables so that the internal dynamics is guaranteed to be stable. This means that tracking cannot always be guaranteed for the original outputs of interest.

In this research we apply dynamic inversion to the system given by (2.15) and (2.16) to achieve station-keeping tracking control for the position outputs (x, y, z, ψ). Initially we select the convenient output vector $y_1 = (z, \varphi, \theta, \psi)$ which makes the dynamic inverse easy to find. Dynamic inversion now yields effectively an inner control loop that feedback linearizes the system from the control $u_h = (u, \tau_\varphi, \tau_\theta, \tau_\psi)$ to the output $y_1 = (z, \varphi, \theta, \psi)$.

Note however that y_1 is not the desired system output. Moreover, dynamic inversion generates a specific internal dynamics, as detailed below, which may not always be stable. Therefore, a second outer loop is designed to generate the required values for $y_1 = (z, \varphi, \theta, \psi)$ in terms of the values of the desired tracking output (x, y, z, ψ). A robustifying term is then added to stabilize the internal dynamics. An overall Lyapunov proof guarantees stability and performance. The following background is required. Consider a nonlinear system of the form

$$\dot{q} = f\left(q, u_q\right) \tag{2.17}$$

where $u_q \in R^m$ is the control input and $q \in R^n$ is state vector. The technique of designing the control input u using dynamic inversion involves two steps. First, one finds a state transformation $z = z(q)$ and an input transformation $u_q = u_q(q, v)$ so that

the nonlinear system dynamics is transformed into an equivalent linear time invariant dynamics of the form

$$\dot{z} = az + bv \tag{2.18}$$

where $a \in R^{n\times n}, b \in R^{n\times m}$ are constant matrices with v is known as new input to the linear system. Secondly one can design easily from the linear control theory approach such as pole placement etc. To obtain the desired linear equations (2.18), one has to differentiate outputs until input vector u appears. That is, for a set of output $y = h(q)$ with $h \in R^m$ if

$$y^r = M(q) + E(q)u_q \tag{2.19}$$

then the desired input which cancels the nonlinearities in the system is obtained as

$$u_q = E(q)^{-1}\left(-M(q) + v\right) \tag{2.20}$$

where$E(q) \in R^{m\times m}, M(q) \in R^{m\times 1}$. This control is known as dynamic inverse. r is known as the relative degree of the system and generally $r < n$; if $r = n$ then full state feedback linearization is achieved if $E(q)$is invertible. Note that for multi-input multi-output system, if number of outputs is not equal to the number of inputs (underactuated system), then $E(q)$becomes non-square and is difficult to obtain a feasible linearizing input u_q.

The system (2.15) and (2.16) is an underactuated system if we consider the states $(x, y, z, \varphi, \theta, \psi)$ as outputs and $u_h = (u, \tau_\varphi, \tau_\theta, \tau_\psi)$ as inputs. To overcome these difficulties we consider four outputs $y_1 = (z, \varphi, \theta, \psi)$ which are used for feedback linearization.

Differentiating the output vector twice with respect to the time we get from (2.15) and (2.16) that,

$$\ddot{y}_1 = \begin{bmatrix} \ddot{z} \\ \ddot{\varphi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = M_h + E_h u_h \tag{2.21}$$

where, $M_h = \begin{bmatrix} -g \\ \dot{\theta}\dot{\psi}\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_p}{I_x}\dot{\theta}\Omega \\ \dot{\varphi}\dot{\psi}\left(\frac{I_z - I_x}{I_y}\right) + \frac{J_p}{I_y}\dot{\varphi}\Omega \\ \dot{\varphi}\dot{\theta}\left(\frac{I_x - I_y}{I_z}\right) \end{bmatrix}$, $E_h = \begin{bmatrix} -(1/m)\cos\theta\cos\varphi & 0 & 0 & 0 \\ 0 & \frac{l}{I_x} & 0 & 0 \\ 0 & 0 & \frac{l}{I_y} & 0 \\ 0 & 0 & 0 & \frac{l}{I_z} \end{bmatrix}$, and $u_h = \begin{bmatrix} u \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$. It is seen that for invertibility of E_h, $0 \leqslant \theta, \varphi < 90°$. The equation (2.21) is equivalent to (2.19); the relative degree of the system is calculated as 8 where as the order of the system is 12. So, the remaining dynamics which does not come out in the process of feedback linearization is known as internal dynamics. To guarantee the stability of the whole system, it is mandatory to guarantee the stability of the remaining 4th order dynamics. In the next Section 2.4 we will discuss how to control the internal dynamics using a backstepping approach. Now using (2.20) we can write the desired input to the system

$$u_h = {E_h}^{-1}\left(-M_h + v_h\right) \tag{2.22}$$

which yields

$$\ddot{y}_1 = v_h \tag{2.23}$$

where, $v_h = \begin{pmatrix} v_z & v_\varphi & v_\theta & v_\psi \end{pmatrix}^T$. This system is decoupled and linear. The auxiliary input v_h is designed as described below.

2.3.1 FBLC linear controller

Assuming the desired output to the system is $y_d = \begin{pmatrix} z_d & \varphi_d & \theta_d & \psi_d \end{pmatrix}^T$, the linear controller is designed in the following way

$$v_h = \begin{bmatrix} v_z \\ v_\varphi \\ v_\theta \\ v_\psi \end{bmatrix} = \begin{bmatrix} \ddot{z}_d - K_{1_z}(\dot{z} - \dot{z}_d) - K_{2_z}(z - z_d) \\ \ddot{\varphi}_d - K_{1_\varphi}(\dot{\varphi} - \dot{\varphi}_d) - K_{2_\varphi}(\varphi - \varphi_d) + v_{r_\varphi} \\ \ddot{\theta}_d - K_{1_\theta}(\dot{\theta} - \dot{\theta}_d) - K_{2_\theta}(\theta - \theta_d) + v_{r_\theta} \\ \ddot{\psi}_d - K_{1_\psi}(\dot{\psi} - \dot{\psi}_d) - K_{2_\psi}(\psi - \psi_d) \end{bmatrix} \tag{2.24}$$

where, $K_{1_\varphi}, K_{2_\varphi}, \ldots.$ etc. are positive constants so that the poles of the error dynamics arising from (2.23) and (2.24) are in the left half of the $s-$plane. For hovering control, z_d and ψ_d are chosen depending upon the designer choice. We note that, v_{r_φ} and v_{r_θ} are the robustifying terms. Now, the other reference inputs such as (φ_d, θ_d) are derived such that the internal dynamics are rendered stable as shown below. Let us define the state error $e_1 = \begin{pmatrix} z_d - z & \varphi_d - \varphi & \theta_d - \theta & \psi_d - \psi \end{pmatrix}^T$ and a sliding error as

$$r_1 = \dot{e}_1 + \Lambda_1 e_1 \tag{2.25}$$

where, Λ_1 is a diagonal positive definite design parameter matrix. Common usage is to select Λ_1 diagonal with positive entries. Then, (25) is a stable system so that e_1 is bounded as long as the controller guarantees that the filtered error r_1 is bounded. In fact it is easy to show [78] that one has

$$\|e_1\| \leqslant \frac{\|r_1\|}{\sigma_{\min}(\Lambda_1)}, \|\dot{e}_1\| \leqslant \|r_1\| \tag{2.26}$$

Note that $\dot{e}_1 + \Lambda_1 e_1 = 0$ defines a stable sliding mode surface. The function of the controller to be designed is to force the system onto this surface by making r_1 small. The parameter Λ_1 is selected for a desired sliding mode response

$$e_1(t) = {e_1}^{-\Lambda_1 t} e_1(0) \tag{2.27}$$

We now focus on designing a controller to keep $\|r_1\|$ small. From (2.25),

$$\dot{r}_1 = \ddot{e}_1 + \Lambda_1 \dot{e}_1 \tag{2.28}$$

Adding an integrator to the linear controller given in (2.24), and now we can rewrite the same as

$$v_h = \ddot{y}_{h_d} + K_1 \dot{e}_1 + K_2 e_1 + K_3 \int_0^t r_1 dt + V_r \tag{2.29}$$

where, $y_{h_d} = \left[\ddot{z}_d, \ddot{\varphi}_d, \ddot{\theta}_d, \ddot{\psi}_d\right]^T$, $K_i = diag(K_{i_z}, K_{i_\varphi}, K_{i_\theta}, K_{i_\psi}) > 0$, $i = 1, 2, 3, 4$ and $V_r = \left[0, v_{r_\varphi}, v_{r_\theta}, 0\right]^T$.

Now using equation (2.22) and (2.23) we can rewrite the equation (2.21) in the form of error dynamics as

$$\ddot{e}_1 + K_1 \dot{e}_1 + K_2 e_1 + K_3 \int_0^t r_1 dt + V_r = 0 \tag{2.30}$$

Thus (2.28) Becomes

$$\dot{r}_1 = -K_1 \dot{e}_1 - K_2 e_1 - K_3 \int_0^t r_1 dt + \Lambda_1 \dot{e}_1 - V_r \tag{2.31}$$

If we choose $K_1 = (\Lambda_1 + R), K_2 = \Lambda_1 R$, then (2.31) will be look like

$$\dot{r}_1 = -R r_1 - K_3 \int_0^t r_1 dt - V_r \tag{2.32}$$

2.4 Desired Tracking Loop and Stabilization of the Internal Dynamics

The internal dynamics [13] for the feedback linearized system given by

$$\ddot{x} = -\frac{u}{m}\sin\theta \tag{2.33}$$

$$\ddot{y} = \frac{u}{m}\cos\theta\sin\varphi \tag{2.34}$$

For the stability of the whole system as well as for the tracking purposes, x, y should be bounded and controlled in a desired way. Note that the height z is controlled by the linear controller (2.24).

To stabilize the zero dynamics, we select some desired θ_d and φ_d such that (x, y) is bounded. Then that (θ_d, φ_d) is given to (2.24) as a reference. Therefore, let us redefine the equation (2.33) as follows

$$\ddot{x} = -\frac{u}{m}\theta_d - \frac{u}{m}(-\tilde{\theta}) \tag{2.35}$$

$$\ddot{y} = \frac{u}{m}\varphi_d - \frac{u}{m}\tilde{\varphi} \tag{2.36}$$

where,

$$\theta_d = -\frac{m}{u}\left\{\ddot{x}_d + C_{11}\left(\dot{x}_d - \dot{x}\right) + C_{12}\left(x_d - x\right) + C_{13}\right\} \tag{2.37}$$

$$\varphi_d = \frac{m}{u}\left\{\ddot{y}_d + C_{21}\left(\dot{y}_d - \dot{y}\right) + C_{22}\left(y_d - y\right)\right\} \tag{2.38}$$

$$\tilde{\theta} = \theta_d - \sin\theta \tag{2.39}$$

$$\tilde{\varphi} = \varphi_d - \cos\theta\sin\varphi \tag{2.40}$$

and

$$\tilde{A} = \begin{bmatrix} -\tilde{\theta} \\ \tilde{\varphi} \end{bmatrix} \tag{2.41}$$

with $C_{ij} > 0, i, j = 1, 2$.

Define the state error

$$e_2 = \begin{pmatrix} x_d - x & y_d - y \end{pmatrix}^T \tag{2.42}$$

and the sliding mode error for the internal dynamics as

$$r_2 = \dot{e}_2 + \Lambda_2 e_2 \tag{2.43}$$

where, Λ_2 is a diagonal positive definite design parameter matrix with similar characteristic of Λ_1. Then designing a controller to keep $\|r_2\|$ small will guarantee that $\|e_2\|$ and $\|\dot{e}_2\|$ are small. Differentiating r_2 we get

$$\dot{r}_2 = \ddot{e}_2 + \Lambda_2 \dot{e}_2 \tag{2.44}$$

Combining the equations (2.37)-(2.44)

$$\ddot{e}_2 + C_1 \dot{e}_2 + C_2 e_2 - U_m \tilde{A} + C_3 \int_0^t r_2 dt = 0 \tag{2.45}$$

where $U_m = \frac{u}{m}$. Therefore

$$\dot{r}_2 = -C_1 \dot{e}_2 - C_2 e_2 - C_3 \int_0^t r_2 dt + U_m \tilde{A} + \Lambda_2 \dot{e}_2 \tag{2.46}$$

Let

$$C_1 = \Lambda_2 + S_0 \tag{2.47}$$

$$C_2 = \Lambda_2 S_0 \tag{2.48}$$

Therefore

$$\dot{r}_2 = -\left(\Lambda_2 + S_0\right) \dot{e}_2 - \Lambda_2 S_0 e_2 - C_3 \int_0^t r_2 dt + U_m \tilde{A} + \Lambda_2 \dot{e}_2 \tag{2.49}$$

$$\dot{r}_2 = -S_0 r_2 - C_3 \int_0^t r_2 dt + U_m \tilde{A} \tag{2.50}$$

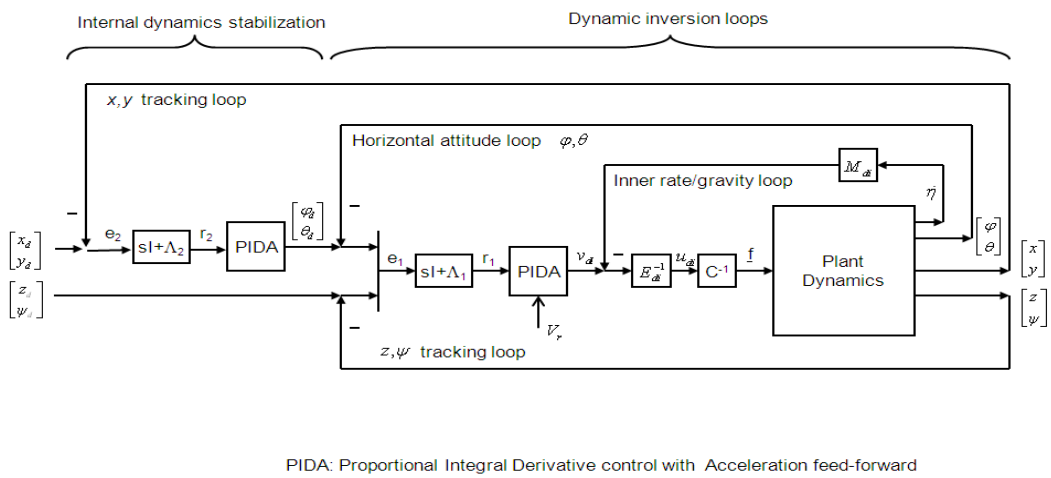

Figure 2.2. Control block diagram.

2.5 Controller Structure and Stability Analysis

The overall control system has two loops and is depicted in Fig. 2.2. The following theorem details the performance of the controller.

Definition 2.5.1. *The equilibrium point x_e is said to be uniformly ultimately bounded (UUB) if there exist a compact set $S \subset R^n$ so that for all $x_0 \in S$ there exist a bound B and a time $T(B, x_0)$ such that $\|x(t) - x_e(t)\| \leqslant B \forall t \geqslant t_0 + T$.*

Theorem 2.5.2. *Given the system as described in* (2.15) *and* (2.16) *with a control law shown in Fig. 2.2. and given by* (2.22),(2.29), (2.37) *and* (2.38) *. The robustifying term is $V_r = \left[0, v_{r_\varphi}, v_{r_\theta}, 0\right]^T = P_s^T r_1 \left(r_1^T P_s P_s^T r_1\right)^{-1} r_2^T Q_1 U_m \tilde{A}$ with P_s and Q_1 defined in the proof and $U_m = \frac{u}{m}$. Then, the tracking errors r_1 and r_2 and thereby e_1 and e_2 are UUB with practical bounds* (2.59), (2.61) *and can be made arbitrarily small with a suitable choice of gain parameters. According to the definition given by* (2.25) *of r_1 and* (2.43) *of r_2, this guarantees that e_1 and e_2 are UUB since*

$$\begin{aligned} \|e_1\| &\leqslant \frac{\|r_1\|}{\sigma_{\min}(\Lambda_1)} \leqslant \frac{b_{r_1}}{\sigma_{\min}(\Lambda_1)} \qquad b_{r_1} > 0 \\ \|e_2\| &\leqslant \frac{\|r_2\|}{\sigma_{\min}(\Lambda_2)} \leqslant \frac{b_{r_2}}{\sigma_{\min}(\Lambda_2)} \qquad b_{r_2} > 0 \end{aligned} \tag{2.51}$$

where $\sigma_{\min}(\Lambda_i)$ *is the minimum eigenvalue of* $\Lambda_i, i = 1, 2$.

Proof. Consider the Lyapunov function

$$L = \frac{1}{2} r_1^T P_1 r_1 + \frac{1}{2} r_2^T Q_1 r_2 + \frac{1}{2} \left[\int_0^t r_1^T dt \right] P_2 \left[\int_0^t r_1 dt \right] + \frac{1}{2} \left[\int_0^t r_2^T dt \right] Q_2 \left[\int_0^t r_2 dt \right] \tag{2.52}$$

with symmetric matrices $P_1, P_2, Q_1, Q_2 > 0$ Therefore, by differentiating L we will get the following

$$\begin{aligned} \dot{L} = -r_1^T P_1 R r_1 - r_1^T P_1 K_3 \int_0^t r_1 dt + r_1 P_2 \int_0^t r_1 dt - r_2^T Q_1 S_0 r_2 - \\ r_2^T Q_1 C_3 \int_0^t r_2 dt + r_2^T Q_2 \int_0^t r_2 dt + r_2^T Q_1 U_m \tilde{A} - r_1^T P_1 V_r \end{aligned} \tag{2.53}$$

If $P_2 = P_1 K_3$ and $Q_2 = Q_1 C_3$, then integration term vanishes.

$$\dot{L} = -r_1^T P_1 R r_1 - r_2^T Q_1 S_0 r_2 + r_2^T Q_1 U_m \tilde{A} - r_1^T P_1 V_r \tag{2.54}$$

Lets define that q_{ij} and p_{ij} denotes the element of the matrices Q_1 and P_1 respectively. Now it is possible to find appropriate v_{r_φ} and v_{r_θ} to construct a suitable V_r for canceling the term $r_2^T Q_1 U_m \tilde{A}$. Define, $P_s = \begin{bmatrix} 0 & p_{22} & 0 & 0 \\ 0 & 0 & p_{33} & 0 \end{bmatrix}^T$ and $V_r = \left[0, v_{r_\varphi}, v_{r_\theta}, 0\right]^T = P_s^T r_1 \left(r_1^T P_s P_s^T r_1\right)^{-1} r_2^T Q_1 U_m \tilde{A}$ then the following cases will arise: assume that ε is a very small positive number near zero. Case 1: When $\left\| r_1^T P_s P_s^T r_1 \right\| > \varepsilon$ then

$$\dot{L} = -r_1^T P_1 R r_1 - r_2^T Q_1 S_0 r_2 < 0 \tag{2.55}$$

Case 2: When $\left\| r_1^T P_s P_s^T r_1 \right\| \leqslant \varepsilon$ then

$$\dot{L} = -r_1^T P_1 R r_1 - r_2^T Q_1 S_0 r_2 - \left(\frac{r_1^T P_s P_s^T r_1}{\varepsilon} - 1 \right) r_2^T Q_1 U_m \tilde{A} \tag{2.56}$$

or

$$\dot{L} \leqslant -\|r_1\|^2 P_{R_{\min}} - \|r_2\|^2 Q_{S_{\min}} - \frac{\|r_1\|^2 \|r_2\| A_{PQ_{\min}}}{\varepsilon} + \|r_2\| A_{QU_{\max}} \tag{2.57}$$

where, $P_{R_{\min}}, Q_{S_{\min}}, A_{PQ_{\min}}$ are the minimum singular values of P_1R, Q_1S_0 and $P_SP_S^TQ_1$ - $U_m\tilde{A}$ respectively. Similarly, $A_{QU_{\max}}$ is the maximum singular value of $Q_1U_m\tilde{A}$.

$$\dot{L} \leqslant -\|r_1\|^2 P_{R_{\min}} - \|r_2\|^2 Q_{S_{\min}} - \|r_2\| \left(\frac{\|r_1\|^2 A_{PQ_{\min}}}{\varepsilon} - A_{QU_{\max}} \right) \tag{2.58}$$

So, $\dot{L} \leqslant 0$ as long as

$$\|r_1\| > \sqrt{\frac{\varepsilon A_{QU_{\max}}}{A_{PQ_{\min}}}} = b_{r_1} \tag{2.59}$$

Again from (2.57) we can rewrite the following

$$\dot{L} \leqslant -\|r_1\|^2 P_{R_{\min}} - \frac{\|r_1\|^2 \|r_2\| A_{PQ_{\min}}}{\varepsilon} - \|r_2\| \left(\|r_2\| Q_{S_{\min}} - A_{QU_{\max}} \right) \tag{2.60}$$

So, $\dot{L} \leqslant 0$ as long as

$$\|r_2\| > \frac{A_{QU_{\max}}}{Q_{S_{\min}}} = b_{r_2} \tag{2.61}$$

Thus, $\dot{L}$ is negative outside a compact set. Selecting the gains P_1, R, Q_1, S_0 ensures that the compact set defined by $\|r_1\| \leqslant b_{r_1}, \|r_2\| \leqslant b_{r_2}$ is bounded. This ensures UUB for r_1 and r_2 stated in equations (2.59) and (2.61) respectively. Also, from equation (2.26), it trivially follows that, $\|e_1\| \leqslant (\sigma_{\min}(\Lambda_1))^{-1} b_{r_1}$ and $\|e_2\| \leqslant (\sigma_{\min}(\Lambda_2))^{-1} b_{r_2}$. $\square$

2.6 Simulation Results

Simulation for a typical quadrotor is performed using the following parameters (SI unit):$M_1 = \begin{bmatrix} 1\text{ Kg} & 0 & 0 \\ 0 & 1\text{ Kg} & 0 \\ 0 & 0 & 1\text{ Kg} \end{bmatrix}$, $I = \begin{bmatrix} 5\text{ Kgm}^2 & 0 & 0 \\ 0 & 5\text{ Kgm}^2 & 0 \\ 0 & 0 & 15\text{ Kgm}^2 \end{bmatrix}$, $J_p = 2\text{ Kgm}^2$, $g = 9.81$ m/s^2.

2.6.1 *Reference Trajectory Generation*

As outlined in [79, 80], a reference trajectory a reference trajectory is derived that minimizes the jerk (rate of change of acceleration) over the time horizon. The trajectory

ensures that the velocities and accelerations at the end point are zero while meeting the position tracking objective. The following summarizes this approach:

$$\dot{x}_d(t) = a_{1_x} + 2a_{2_x}t + 3a_{3_x}t^2 + 4a_{4_x}t^3 + 5a_{5_x}t^4 \tag{2.62}$$

Differentiating again,

$$\ddot{x}_d(t) = 2a_{2_x} + 6a_{3_x}t + 12a_{4_x}t^2 + 20a_{5_x}t^3 \tag{2.63}$$

As we indicated before that initial and final velocities and accelerations are zero; so from (2.62) and we can conclude the following:

$$\begin{bmatrix} d_x \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & t_f & t_f^2 \\ 3 & 4t_f & 5t_f^2 \\ 6 & 12t_f & 20t_f^2 \end{bmatrix} \begin{bmatrix} a_{3_x} \\ a_{4_x} \\ a_{5_x} \end{bmatrix} \tag{2.64}$$

where, $d_x = \left(x_{d_f} - x_{d_0}\right)/t_f^3$. Now, solving for coefficients

$$\begin{bmatrix} a_{3_x} \\ a_{4_x} \\ a_{5_x} \end{bmatrix} = \begin{bmatrix} 1 & t_f & t_f^2 \\ 3 & 4t_f & 5t_f^2 \\ 6 & 12t_f & 20t_f^2 \end{bmatrix}^{-1} \begin{bmatrix} d_x \\ 0 \\ 0 \end{bmatrix} \tag{2.65}$$

Thus the desired trajectory for the x direction is given by

$$x_d(t) = x_{d_0} + a_{3_x}t^3 + a_{4_x}t^4 + a_{5_x}t^5 \tag{2.66}$$

Similarly, the reference trajectories for the y and z directions are given by (2.67) and (2.68) respectively.

$$y_d(t) = y_{d_0} + a_{3_y}t^3 + a_{4_y}t^4 + a_{5_y}t^5 \tag{2.67}$$

$$z_d(t) = z_{d_0} + a_{3_z}t^3 + a_{4_z}t^4 + a_{5_z}t^5 \tag{2.68}$$

The advantage of this trajectory generation method lies in the fact that more de-

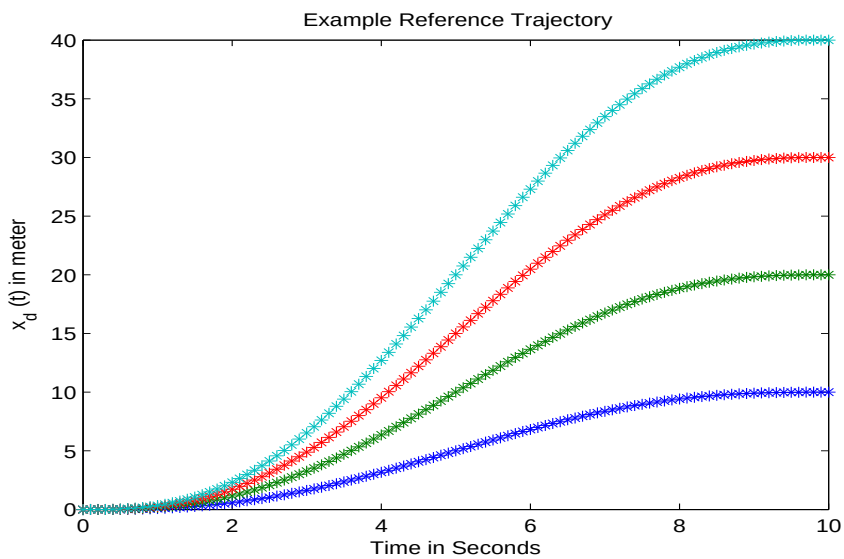

Figure 2.3. Example trajectory simulation for different final positions.

manding changes in position can be accommodated by varying the final time. That is acceleration/torque ratio can be controlled smoothly as per requirement and constraints explicitly enforced on the achievable accelerations. For example, let us assume at $t = 0$, $x_{d_0} = 0$ and at $t = 10$s, $x_{d_f} = 10$. Therefore $d_x = 0.01$ and the trajectory is given by (2.69) and shown in Fig. 2.3 for various desired final positions.

$$x_d(t) = 0.1t^3 - 0.015t^4 + 0.0006t^5 \tag{2.69}$$

2.6.2 Simulation Result Without Disturbances

2.6.2.1 From Initial Position at $(0, 0, 0)$ to Final Position at $(20, 5, 10)$

Fig. 2.4 describes the controlled motion of the quadrotor from its initial position $(0, 0, 0)$ to final position $(20, 5, 10)$ for a given time (20 seconds). The actual trajectories $x(t), y(t), z(t)$ match exactly their desired values $x_d(t), y_d(t), z_d(t)$ respectively nearly exactly. The errors along the three axes are also shown in the same figure. It can be seen that the tracking is almost perfect as well as the tracking errors are significantly

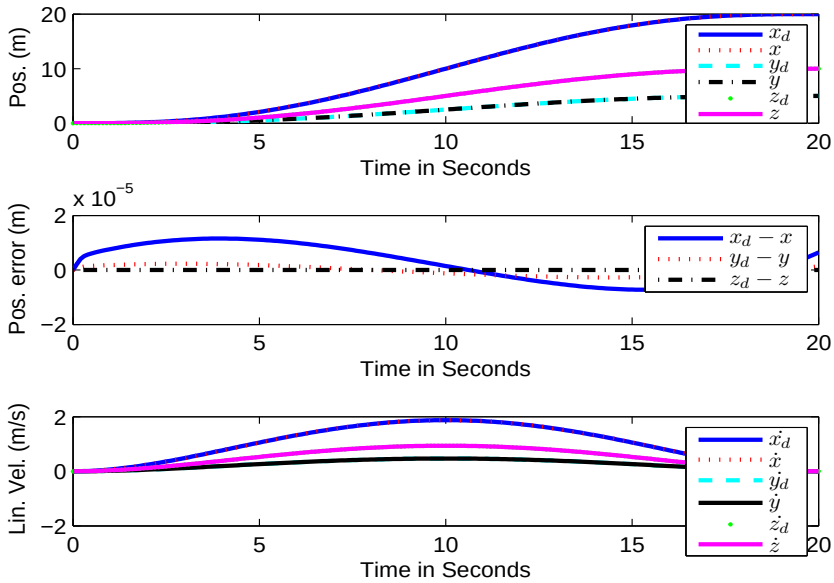

Figure 2.4. Position tracking - simultaneous command in x, y and z.

small. Linear velocities of the quadrotor are shown in the same figure also. Fig. 2.5 describes the attitude of the quadrotor φ, θ along with their demands φ_d, θ_d and attitude errors in radian. Again the angles match their command values nearly perfectly. Angular velocities are also shown in the same figure. Fig. 2.6 describes the control input requirement which is very much realizable. Note that as described before the control requirement for yaw angle is $\tau_\psi = 0$ and it is seen from Fig. 2.6.

2.6.2.2 From Initial Position at $(0, 5, 10)$ to Final Position at $(20, 5, 10)$

Fig. 2.7 and Fig. 2.8 illustrates the decoupling phenomenon of the control law. Fig. 2.7 shows that $x(t)$ follows the command $x_d(t)$ nearly perfect unlike $y(t)$ and $z(t)$ are held their initial values. Fig. 2.8 shows that the change in x does not make any influence on φ. The corresponding control inputs are also shown in Fig. 2.9 and due to the full decoupling effect it is seen that τ_φ is almost zero.

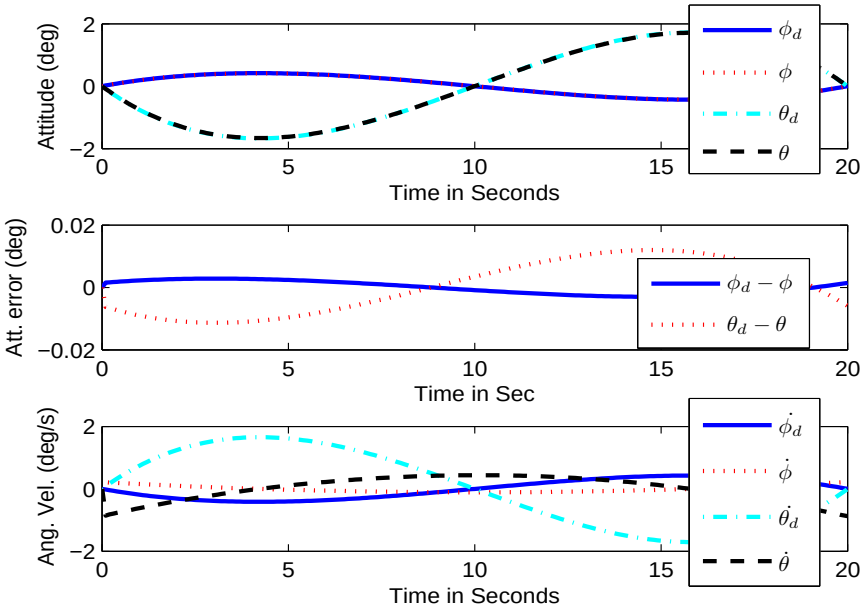

Figure 2.5. Resultant angular positions, errors and velocities.

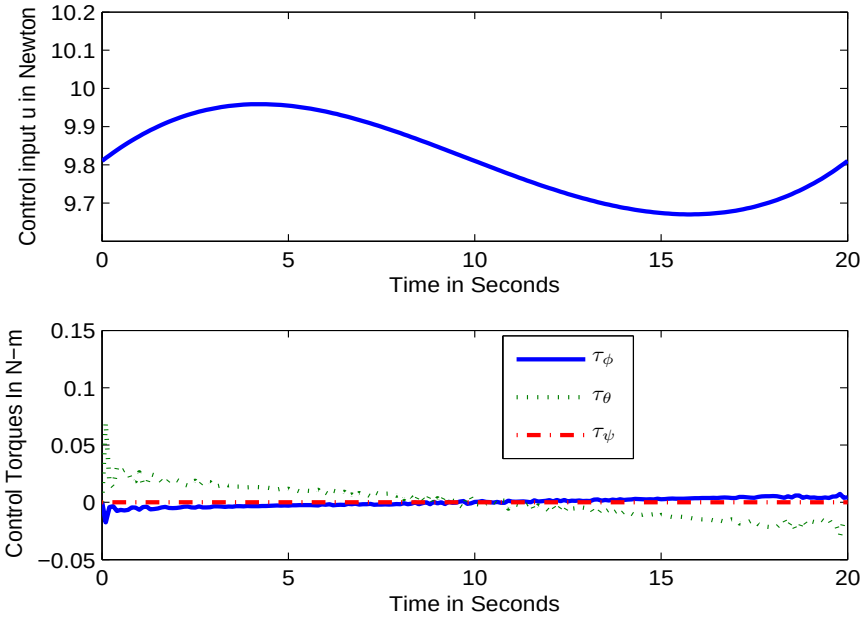

Figure 2.6. Input commands for Case 1.

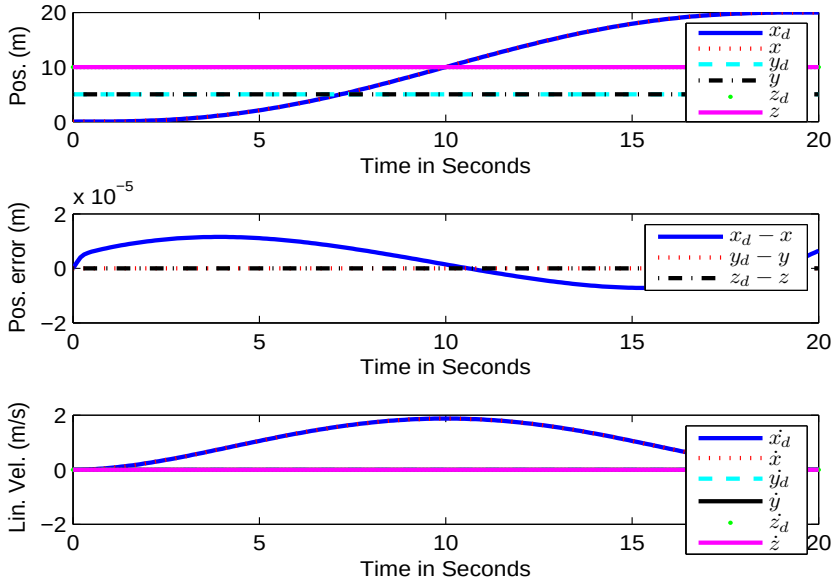

Figure 2.7. Position tracking (x command only).

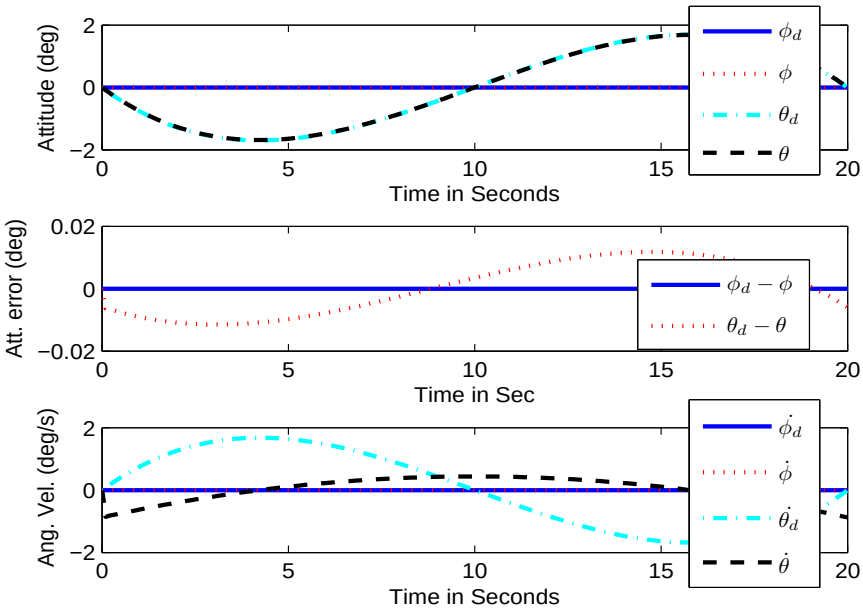

Figure 2.8. Angular variations and rates (x command tracking).

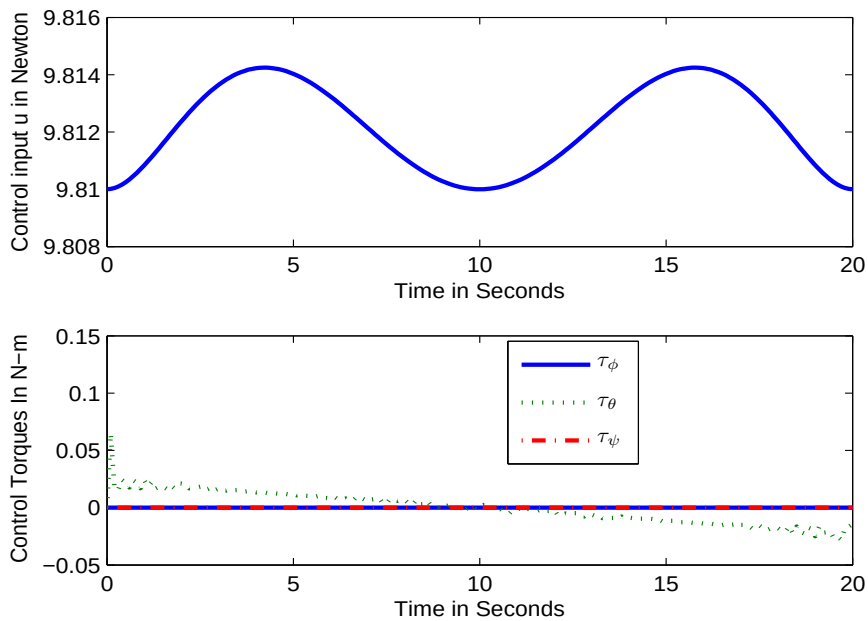

Figure 2.9. Input commands for x - command tracking.

The similar type of simulations are performed for y and z directional motions separately and similar plots are obtained showing excellent tracking.

2.6.3 Simulation Result with Unmodeled Input Disturbances

The simulation is performed to verify its robustness properties against unmodeled input disturbances. For this case we simulate the dynamics with high frequency disturbance $\sin(5t)$ (10% of maximum magnitude of force) for force channel and $0.01\sin(5t)$ (15% of maximum angular acceleration) for torque channel.

2.6.3.1 From Initial Position at $(0, 0, 0)$ to Final Position at $(20, 5, 10)$

Fig. 2.10 and Fig. 2.11 describe the motion of the quadrotor from its initial position $(0, 0, 0)$ to final position $(20, 5, 10)$ for a given time (20 seconds) with input disturbances. It can be seen from Fig. 2.10 that the quadrotor can track the desired

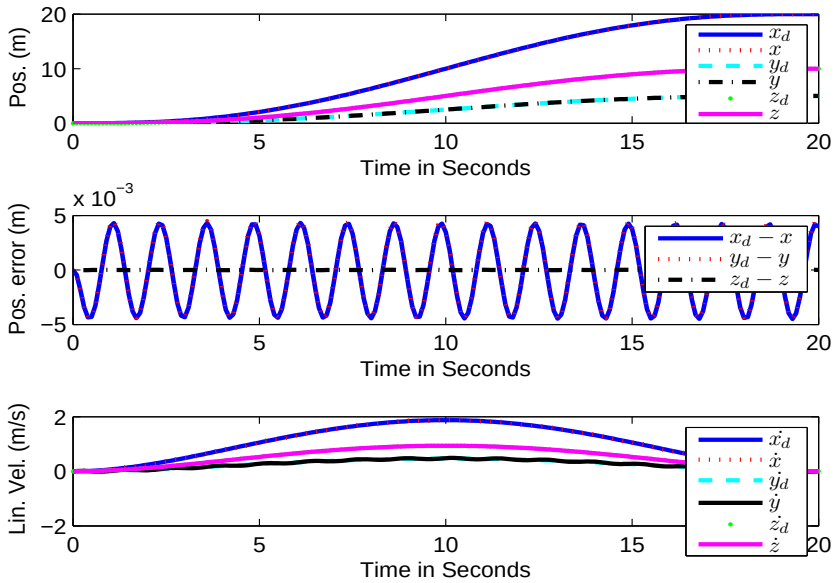

Figure 2.10. Simultaneous command in x, y and z + Input disturbances.

position effectively without any effect of high input disturbances. From Fig. 2.10 and Fig. 2.11, it is also seen that the position errors are bounded and small. Fig. 2.12 shows the bounded variation of control inputs in presence of disturbance. Similar tracking performance is obtained for other commanded motion.

2.7 Conclusion

A two loop approach using input-output linearization to design nonlinear controller for a quadrotor dynamics is discussed in this research. Using this approach, an intuitively structured controller was derived that has an outer proportional derivative loop and an inner feedback linearizing control loop. The dynamics of a quadrotor are a simplified form of helicopter dynamics that exhibits the basic problems including underactuation, strong coupling, multi-input/multi-output. The derived controller is capable of dealing with such problems simultaneously and satisfactorily. As the quadrotor

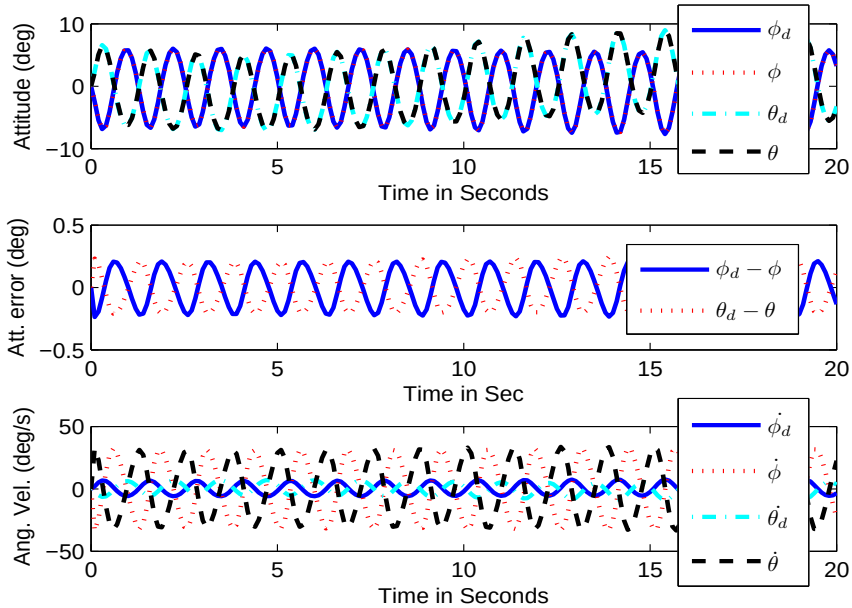

Figure 2.11. Angular variations, errors and velocities (with input disturbances).

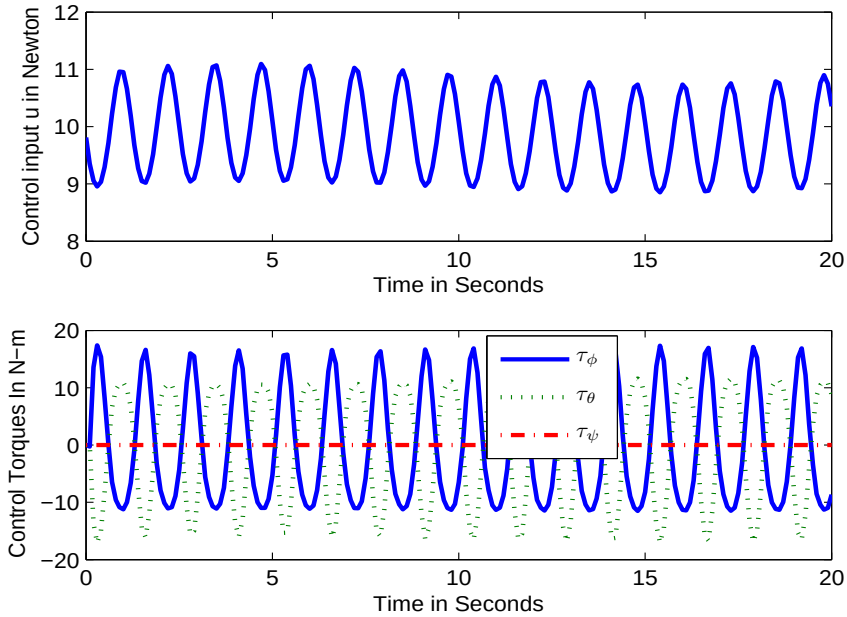

Figure 2.12. Force and torque input variations (with input disturbances).

model discuss in this research is similar to a full scale unmanned helicopter model, the same control configuration derived for quadrotor is also applicable for a helicopter model. The simulation results with and without input disturbances are shown above. Some aspects are still remained untouched. The controller shows high sensitivity to state disturbances, which may be in the interest of future research.

CHAPTER 3

BACKSTEPPING APPROACH FOR CONTROLLING A QUADROTOR USING LAGRANGE FORM DYNAMICS

3.1 Introduction

Nowadays helicopters are designed to operate with high agility and rapid maneuvering, and are capable of work in degraded environments including wind gusts etc. Helicopter control often requires holding at a particular trimmed state, generally hover, as well as making changes of velocity and acceleration in a desired way [3]. The control of unmanned rotorcraft is becoming more and more important due to their usefulness in rescue, surveillance, inspection, mapping etc. For these applications the ability of the rotorcraft to maneuver sharply and hover precisely is important.

Like fixed-wing aircraft control, rotorcraft control is involved in controlling attitude roll, pitch, and yaw and position x, y, z either separately or in a coupled way. But the main difference is that, due to the unique body structure of a rotorcraft, as well as the rotor dynamics, the rotorcraft attitude dynamics and position dynamics are strongly coupled. Therefore, it is very difficult to design a decoupled control law of good structure that stabilizes the faster and slower dynamics simultaneously. On the contrary, for a fixed-wing aircraft it is known how to design decoupled standard controllers [5] with intuitively comprehensible structure and performance. Flight controllers of good structure are needed for robustness, as well as to give some intuitive feel for the functioning of autopilots, Stability Augmentation System (SAS), and Control Augmentation System (CAS).

The dynamics of a quadrotor [1, 24, 8, 10, 81] are a simplified form of helicopter dynamics that exhibit the basic problems of rotorcraft including underactuation, strong coupling, multi-input/multi-output design, and unknown nonlinearities. In the quadrotor the movement is caused by the resultant forces and moments of four independent rotors. Control design for a quadrotor is quite similar to a helicopter but simplified, so that the quadrotor serves as a suitable, more tractable, case study for rotorcraft controls design. Therefore, numerous research papers have dealt with control of the quadrotor, with many approaches available in that literature that use different control techniques. Popular methods include linearization and linear quadratic regulator (LQR) or H_∞ state-space design, model-predictive control [14] feedback linearization, and backstepping [11].

Specifically, [1, 23, 22] considered the Lagrangian model of a quadrotor and derived what was in effect a backstepping controller with 3 loops: heading/ motion, motion, and motion. Nested saturation control was used in each loop. Exact compensation was assumed in each loop. The control law was tested on a practical quadrotor model. References [24] compared the performance of a proportional-integral-derivative (PID) controller with a LQ control law on a micro vertical take-off and landing (VTOL) test bench. These papers presented a good accurate dynamic model, and discussed the mechanical structure and sensors of the quadrotor. Reference [25] discussed the robust trajectory tracking of a Lagrangian quadrotor model using backstepping control and proved the stability for any given but bounded trajectory. A nonlinear H_∞ type robust control law was used in [10, 26], , with [26] demonstrating that the robust hover controller can handle large parameter uncertainty with respect to their nominal values. A popular method for handling unknown nonlinearities is to introduce neural networks, tuned online using adaptive control techniques. Reference [27] used an adaptive dynamic programming (ADP) based controller. Reference [16] used dynamic

inversion with neural networks to account for unknown dynamics and dynamic inversion errors for a helicopter model. Reference [28] used an innovative method known as pseudo-control hedging (PCH) with a single neural network employed for both inner and outer loops to account for unknown dynamics and bounded control inputs. Reference [29] used backstepping with neural networks for an unmanned air vehicle to confront modeling errors and unmodeled dynamics. It was based on a nonlinear state space model of an aircraft. Reference [30] presented a Lagrangian dynamic model of X4-flyer with motor dynamics and gyroscopic effects included. References [9, 7] used a feedback linearization approach. Reference [6] presented different control architectures such as feedback linearization, backstepping based on visual feedback as the primary sensor. Reference [31] discussed the dynamic modeling of a unmanned aerial vehicle (UAV) Dragan flyer III and a H_∞ loop shaping flight controller as a position control law. Reference [32] presented a quasioptimal trajectory planner with a LQR path following controller. Reference [33] developed a simulation tool for an indoor flight-capable quadrotor based on MEMS sensors and rangefinders.

Most work in quadrotor control that uses backstepping design or other nonlinear control formulations like feedback linearization are based on state-variable formulations. This results in control laws that are complex and often require the evaluation of lie derivatives. Reference [25] does perform backstepping on the Euler-Lagrange dynamics, but backstepping is performed on a virtual input that is a bilinear product involving thrust input. This results in complications and extra states are added, which we avoid.

The contribution of this Chapter is to apply backstepping to the coupled Lagrangian form (not the state-variable form) of the dynamics [73]; this yields a controller of appealing structure having an attitude control inner loop and a position control outer loop. It is not straightforward to apply backstepping to the Lagrangian dynamics since

the position subsystem is bilinear in the virtual control (see [25]). This is confronted herein using an inverse kinematics solution such as is used in robotics [82]. This allows us to simply take the attitudes directly as the virtual inputs. To combat unknown nonlinearities we use neural networks [83, 84] to estimate unknown nonlinear terms and aerodynamic forces and moments. The resulting controller has a structure that is intuitively appealing, as in fixed-wing aircraft SAS and CAS. Moreover, rigorous stability proofs provided herein give increased confidence in this structure. Simulation results are shown to validate the control law discussed in this research.

3.2 Quadrotor Dynamics

In this Section we will describe the dynamical model of the quadrotor (see Fig. 2.1). The quadrotor has some basic advantages over the conventional helicopter [22] in terms of simplicity of dynamics and control design. Given that the front and the rear motors rotate counter-clockwise while the other two rotate clockwise, gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight. This four-rotor rotorcraft does not have a swashplate. In fact it does not need any blade pitch control. The collective input (or throttle input) is the sum of the thrusts of the individual motors (see Fig. 2.1). This results in simplified dynamics. Pitch movement is obtained by increasing (resp. reducing) the speed of the rear motor while reducing (resp. increasing) the speed of the front motor. The roll movement is obtained similarly using the lateral motors. The yaw movement is obtained by increasing (resp. decreasing) the speed of the front and rear motors while decreasing (resp. increasing) the speed of the lateral motors [1]. The dynamics of the four rotors are relatively much faster than the main system and thus neglected.

The dynamic model of the rotorcraft can be written [23] as the position subsystem.

$$m\ddot{\xi} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = F_d \tag{3.1}$$

$$J(\eta)\,\ddot{\eta} + \frac{d}{dt}\{J(\eta)\}\,\dot{\eta} - \frac{1}{2}\frac{\partial}{\partial\eta}(\dot{\eta}^T J(\eta)\,\dot{\eta}) = \tau \tag{3.2}$$

or, by appropriate definition of variables,

$$J(\eta)\,\ddot{\eta} + C(\eta,\dot{\eta})\,\dot{\eta} = \tau \tag{3.3}$$

The auxiliary (effective) inertia matrix is defined as,

$$J(\eta) = J = T_\eta^T \Sigma T_\eta \tag{3.4}$$

with $T_\eta = \begin{pmatrix} -\sin\theta & 0 & 1 \\ \cos\theta\sin\psi & \cos\psi & 0 \\ \cos\theta\cos\psi & -\sin\psi & 0 \end{pmatrix}$ a rotation transformation and Σ the (constant) quadrotor inertia matrix.

The virtual input (e.g. an input that we cannot directly control) into the position subsystem (1) is the force vector $F_d = u \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\varphi \\ \cos\theta\cos\varphi \end{pmatrix}$. The force exerted by the ith rotor is denoted as f_i and the control inputs are defined as the total thrust $u = \sum_{i=1}^{4} f_i$,

and the torques $\tau = \begin{pmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\varphi \end{pmatrix}$. The control inputs $\left(u, \tau^T\right)^T$ are given in terms of the forces $(f_i = k_{\omega_i}\omega_i^2, i = 1...4)$ of the four rotors of the quadrotor by the relation

$$\begin{pmatrix} u \\ \tau_\varphi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ l & 0 & -l & 0 \\ 0 & -l & 0 & l \\ c & -c & c & -c \end{pmatrix}}_{M} \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}}_{f} = Mf \tag{3.5}$$

where l is the distance from the motors to the center of gravity, and c is a constant known as force-to-moment scaling factor. Thus, if a required thrust and torque vector are given, one may solve for the rotor forces via matrix inversion using (3.5). Therefore, we show in the subsequent development how to design the control inputs $\left(u, \tau^T\right)^T$.

Vector $C(\eta, \dot{\eta})\,\dot{\eta}$ is the coriolis/centripetal term. From (3.2), the matrix $C(\eta, \dot{\eta})$ can be represented, assuming the inertia matrix Σ for a typical quadrotor is symmetric, as

$$\begin{aligned} C(\eta, \dot{\eta}) &= \frac{d}{dt}J - \frac{1}{2}\frac{\partial}{\partial \eta}\left(\dot{\eta}^T J(\eta)\right) \\ &= 2T_\eta^T \Sigma \dot{T}_\eta - \frac{1}{2}\frac{\partial}{\partial \eta}\left(\dot{\eta}^T J(\eta)\right) \end{aligned} \tag{3.6}$$

Note that the representation of matrix $C(\eta, \dot{\eta})$ is not unique [82]. However, using the specific representation (3.6), it can be proved that $\left(\frac{d}{dt}J(\eta) - 2C(\eta, \dot{\eta})\right)$ is skew-symmetric. This is needed later in Lyapunov proofs referred to the dynamics (3.3). Note that for a quadrotor the inertia matrix Σ is constant but the effective inertia $J = J(\eta)$ is not a constant.

In summary, the coupled Lagrangian form of the dynamics is given as

$$m\ddot{\xi} = u \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\varphi \\ \cos\theta\cos\varphi \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \tag{3.7}$$

$$J\ddot{\eta} = -C\left(\eta, \dot{\eta}\right)\dot{\eta} + \tau \tag{3.8}$$

where total thrust $u \in R$ and torque $\tau \in R^3$ are the control inputs. This system is a coupled Lagrangian form underactuated system with six outputs and four inputs.

3.3 Backstepping Control Design

Backstepping control design is effective for a class of underactuated systems [11, 85]. Many robot and aircraft control problems are included in the class of underactuated systems [82]. Backstepping for the quadrotor has normally been applied to the state-variable form, which introduces unnecessary complications (e.g. lie derivatives). Note that the state of the quadrotor in (3.7),(3.8) is $(\xi, \dot{\xi}, \eta, \dot{\eta})$, which has 12 components. In this research we use backstepping control design directly on the Lagrangian quadrotor dynamics. This is complicated by the fact that (3.7) is bilinear in the controls due to the first term on the right-hand side (i.e. F_d). Taking this entire bilinear term as a virtual control leads to complications (see [25]).

The quadrotor dynamics have a simplified form due to its physical construction and the decoupled control effects inherent in the four motor design. However, for aggressive maneuverability, we would like to include aerodynamic forces and moments in the quadrotor equations. These are unknown forces containing nonlinear terms in the states that appear in any object that moves through the air. Note that the quadrotor has an equivalent area cross section in each of the x, y and z directions. These areas are difficult to determine, are state dependent, and can usually be found only through wind

tunnel tests. The aerodynamic forces are of two types: lift force due to dynamic pressure and drag force due to viscosity effects. They can be separated into components in each of the x, y, z directions. Likewise, the aerodynamic torques have three components as torques around the three axes.

Therefore, rewrite the quadrotor dynamics (3.7),(3.8) to include unmodeled state dependent disturbances that include aerodynamic forces and moments A_x, A_y, A_z, A_η.

This yields the Lagrangian form

$$\begin{aligned} m\ddot{x} &= -u\sin\theta + A_x \\ m\ddot{y} &= u\cos\theta\sin\varphi + A_y \\ m\ddot{z} &= u\cos\theta\cos\varphi - mg + A_z \end{aligned} \tag{3.9}$$

$$J\begin{pmatrix} \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\varphi} \end{pmatrix} = \begin{pmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\varphi \end{pmatrix} - C\left(\eta, \dot{\eta}\right)\dot{\eta} + A_\eta \tag{3.10}$$

Now we will use backstepping to determine a control structure to make the position $\xi = (x, y, z)^T$ of the quadrotor follow a desired smooth (e.g. at least twice differentiable) trajectory $(x_d(t), y_d(t), z_d(t))$. The end result is shown in Fig. 3.1. To achieve this purpose, the first step of backstepping focuses on the position dynamics (3.9). Define the position tracking error $e_1(t)$ of the subsystem given by (3.9) and the sliding mode error $r_1(t)$ as [83]

$$e_1 = (x_d - x, y_d - y, z_d - z)^T = \xi_d - \xi \tag{3.11}$$

$$r_1 = \dot{e}_1 + \Lambda_1 e_1 \tag{3.12}$$

where, Λ_1 is a diagonal positive definite design parameter matrix. Common usage is to select Λ_1 diagonal with positive entries. Then, (3.12) is a stable system so that e_1

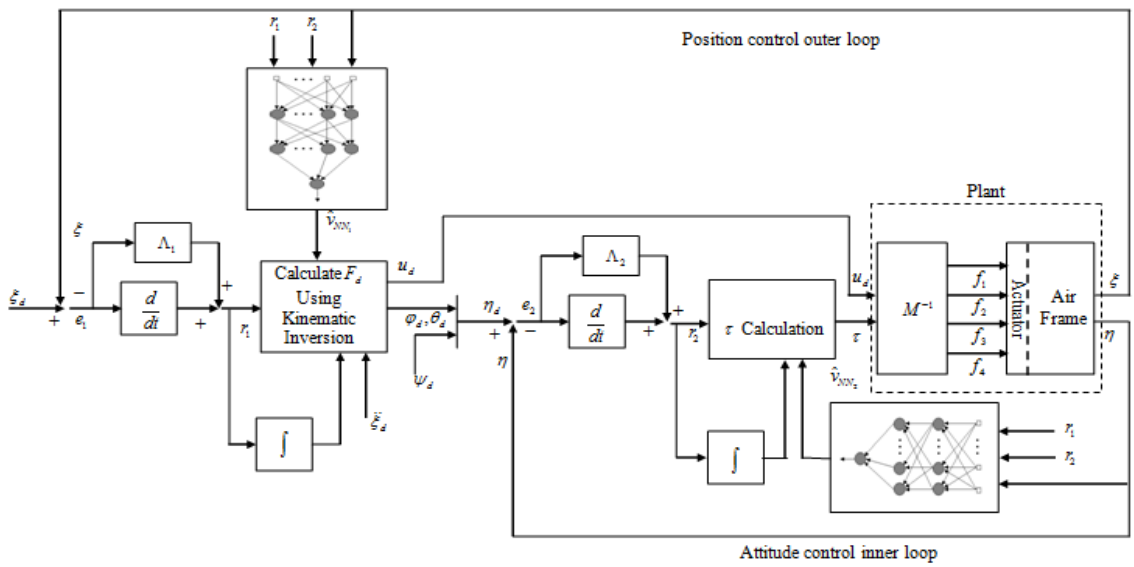

Figure 3.1. Control configuration.

is bounded as long as the controller guarantees that the filtered error r_1 is bounded. In fact it is easy to show [83] that one has

$$\|e_1\| \leqslant \frac{\|r_1\|}{\sigma_{\min}(\Lambda_1)}, \|\dot{e}_1\| \leqslant \|r_1\| \tag{3.13}$$

Note that $\dot{e}_1 + \Lambda_1 e_1 = 0$ defines a stable sliding mode surface. The function of the controller to be designed is to force the system onto this surface by making r_1 small. The parameter Λ_1 is selected for a desired sliding mode response

$$e_1(t) = {e_1}^{-\Lambda_1 t} e_1(0) \tag{3.14}$$

We now focus on designing a controller to keep $\|r_1\|$ small. Then the error dynamics becomes

$$m\dot{r}_1 = m\ddot{e}_1 + m\Lambda_1\dot{e}_1 \tag{3.15}$$

$$\Rightarrow m\dot{r}_1 = m\begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \\ \ddot{z}_d \end{bmatrix} - m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + m\Lambda_1\left(r_1 - \Lambda_1 e_1\right) \tag{3.16}$$

$$m\dot{r}_1 = m\ddot{\xi}_d - \begin{pmatrix} -u\sin\theta \\ u\cos\theta\sin\varphi \\ u\cos\theta\cos\varphi \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} - \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} + m\Lambda_1(r_1 - \Lambda_1 e_1) \quad (3.17)$$

or,

$$m\dot{r}_1 = m\ddot{\xi}_d + m\Lambda_1 r_1 - m\Lambda_1^2 e_1 - F_d - m_g - A_{xyz} \quad (3.18)$$

where, $F_d = \begin{pmatrix} -u\sin\theta \\ u\cos\theta\sin\varphi \\ u\cos\theta\cos\varphi \end{pmatrix}$, $A_{xyz} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix}$ and $m_g = \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix}$.

The ideal force (virtual force) which stabilizes the tracking error dynamics (3.18) is

$$\hat{F}_d = m\ddot{\xi}_d - m\Lambda_1^2 e_1 - m_g - \hat{A}_{xyz} + K_{r_1} r_1 + K_{i_1} \int_0^t r_1 dt' \quad (3.19)$$

where, $\hat{A}_{xyz}$ is an approximation (to be detailed later) of the aerodynamic function A_{xyz} and control gain matrices $K_{r_1} > 0, K_{i_1} > 0$ are diagonal. Using (3.19) in (3.18) yields the closed-loop error dynamics

$$m\dot{r}_1 = -\left(K_{r_1} - m\Lambda_1\right) r_1 - \tilde{A}_{xyz} - K_{i_1} \int_0^t r_1 dt' \quad (3.20)$$

where,$\tilde{A}_{xyz} = A_{xyz} - \hat{A}_{xyz}$ is the approximate error. We shall show in subsequent proofs the beneficial properties of this error dynamics system. Now it is necessary to perform step 2 of the backstepping procedure, which selects the control inputs, u and torque τ , to generate the desired force (3.19). The simplicity of the Lagrangian form must be preserved in an effective backstepping procedure. This means one must determine the desired values of u and of (ψ, θ, φ) for the second backstepping step that involves the attitude dynamics (3.10). Note that yaw ψ does not appear in (3.9), so its desired value ψ_d can be selected as a reference input. Call the remaining desired values

φ_d, θ_d and u_d . To generate the ideal force $\hat{F}_d$ in (3.19) would require the use of φ_d, θ_d and u_d given by the solution to

$$\begin{pmatrix} -u_d \sin\theta_d \\ u_d \cos\theta_d \sin\varphi_d \\ u_d \cos\theta_d \cos\varphi_d \end{pmatrix} = \hat{F}_d \tag{3.21}$$

Thus, given $\hat{F}_d$ as computed from (3.19), one must solve for φ_d, θ_d and u_d . This is straightforward using robot inverse kinematics notions. The procedure is given in Section 3.4.

Now a NN is used to approximate A_{xyz} which is unknown. It can be shown from [86] that A_{xyz} can be approximated using a NN as

$$v_{NN_1} = -A_{xyz} \equiv W_1^T \mu_1 (\chi_1) + \varepsilon_1 \tag{3.22}$$

where W_1 is an unknown ideal NN output weight matrix and $\chi_1 = (\xi, r_1, \eta, \dot{\eta})^T$ is the input of the NN. The NN activation function vector $\mu_1 (\cdot)$ must be selected as a basis set for the unknown function, which is required for suitable NN approximation. The NN activation function $\mu_1(\cdot)$ should be chosen to capture any known nonlinearity present in the aerodynamics. The NN estimation error ε_1 is bounded by $\|\varepsilon_1\| < \varepsilon_{N_1}$ on a compact set, with a known bound ε_{N_1} .

Note that one could use a two-layer neural net for approximation. This is nonlinear in the weight parameters (NLIP), and introduces extra complications in the proof, which are nonetheless tractable. This technology has been described in [83, 78]. In order to focus on the contributions of this research without extra complications, we use the linear-in-the-parameter NN here. Therefore, an approximation to the unknown aerodynamic forces is taken as

$$\hat{v}_{NN_1} = -\hat{A}_{xyz} = \hat{W}_1^T \mu_1 (\chi_1) \tag{3.23}$$

with $\hat{W}_1$ the actual NN weights. In Section 3.5 we will discuss how to tune the NN weights $\hat{W}_1$ to provide stability and bounded sliding mode error dynamics (3.20).

The desired force input into the position dynamics is thus taken as

$$\hat{F}_d = m\ddot{\xi}_d - m\Lambda_1^2 e_1 - m_g + \hat{W}_1^T \mu_1(\chi_1) + K_{r_1} r_1 + K_{i_1} \int_0^t r_1 dt' \tag{3.24}$$

Define,

$$-\tilde{A}_{xyz} = W_1^T \mu_1(\chi_1) - \hat{W}_1^T \mu_1(\chi_1) + \varepsilon_1 \equiv \tilde{W}_1^T \mu_1(\chi_1) + \varepsilon_1 \tag{3.25}$$

Now (3.20) becomes,

$$m_1 = -(K_{r_1} - m\Lambda_1) r_1 + \tilde{W}_1^T \mu_1(\chi_1) - K_{i_1} \int_0^t r_1 dt' + \varepsilon_1 \dot{r} \tag{3.26}$$

By solving (3.21) with $\hat{F}_d$ computed using (3.24), as detailed in Section 3.4, the desired pseudo-commands φ_d, θ_d, u_d for the subsystem given by (3.9) are determined.Note that u_d is directly applied to the plant as input. Now, the second step of backstepping is performed.We select a control input τ for (3.10) to generate φ_d, θ_d.Note that ψ_d does not appear in (3.9) and so can be selected as an independent desired reference input. Define $_d = (\varphi_d, \theta_d, \psi_d)^T \eta$

Defining the tracking error $e_2(t)$ and the sliding mode error $r_2(t)$ as,

$$e_2 = (\psi_d - \psi, \theta_d - \theta, \varphi_d - \varphi)^T \tag{3.27}$$

$$r_2 = \dot{e}_2 + \Lambda_2 e_2 \tag{3.28}$$

where, Λ_2 is a diagonal positive definite design parameter matrix with similar characteristic of Λ_1 . Then designing a controller to keep $\|r_2\|$ small will guarantee that $\|e_2\|$ and $\|\dot{e}_2\|$ are small.

Then error dynamics given by

$$\begin{aligned} J\dot{r}_2 &= J\,\ddot{e}_2 + J\Lambda_2\dot{e}_2 \\ &= J\ddot{\eta}_d - \{\tau - C(\eta,\dot{\eta})\,\dot{\eta} + A_\eta\} + J\Lambda_2\,(r_2 - \Lambda_2 e_2) \end{aligned} \tag{3.29}$$

and from (3.28)

$$\dot{\eta} = -r_2 + \dot{\eta}_d + \Lambda_2 e_2 \tag{3.30}$$

Substituting $\dot{\eta}$ from (3.30) in (3.29) yields

$$J\dot{r}_2 = J\ddot{\eta}_d - \{\tau - C(\eta,\dot{\eta})\,(-r_2 + \dot{\eta}_d + \Lambda_2 e_2) + A_\eta\} + J\Lambda_2\,(r_2 - \Lambda_2 e_2) \tag{3.31}$$

$$J\dot{r}_2 = J\ddot{\eta}_d - \{\tau - C(\eta,\dot{\eta})\,(-r_2 + \dot{\eta}_d + \Lambda_2 e_2) + A_\eta\} + J\Lambda_2\,(r_2 - \Lambda_2 e_2) \tag{3.32}$$

Now, a second NN is introduced to approximate $J\ddot{\eta}_d - A_\eta + C(\eta,\dot{\eta})\,(\dot{\eta}_d + \Lambda_2 e_2) - J\Lambda_2^2 e_2$. According to [86], this can be approximated as

$$v_{NN_2} \triangleq \left\{J\ddot{\eta}_d - A_\eta + C(\eta,\dot{\eta})\,(\dot{\eta}_d + \Lambda_2 e_2) - J\Lambda_2^2 e_2\right\} \equiv W_2^T \mu_2(\chi_2) + \varepsilon_2 \tag{3.33}$$

where W_2 is the unknown ideal NN weight matrix, is an another basis activation function which is to be chosen to estimate the known nonlinearities present in the system. is estimation error which is bounded by on a compact set with the

$$\hat{v}_{NN_2} \equiv \hat{W}_2^T \mu_2(\chi_2) \tag{3.34}$$

with $\hat{W}_2$ is the actual NN weights.

Define the control input τ as

$$\tau = \hat{v}_{NN_2} + K_{r_2} r_2 + K_{i_2} \int_0^t r_2 dt' \tag{3.35}$$

with $K_{r_2} > 0, K_{i_2} > 0$ are diagonal gain matrices and also

$$\tilde{v}_{NN_2} = v_{NN_2} - \hat{v}_{NN_2} = W_2^T \mu_2(\chi_2) - \hat{W}_2^T \mu_2(\chi_2) + \varepsilon_2 = \tilde{W}_2^T \mu_2(\chi_2) + \varepsilon_2 \tag{3.36}$$

Then (3.32) reduces to the form,

$$J\dot{r}_2 = \tilde{v}_{NN_2} + J\Lambda_2 r_2 - C\left(\eta, \dot{\eta}\right) r_2 - K_{r_2} r_2 - K_{i_2} \int_0^t r_2 dt' \tag{3.37}$$

$$J\dot{r}_2 = \tilde{W}_2^T \mu_2\left(\chi_2\right) - \left\{K_{r_2} + C\left(\eta, \dot{\eta}\right) - J\Lambda_2\right\} r_2 - K_{i_2} \int_0^t r_2 dt' + \varepsilon_2 \tag{3.38}$$

with the estimation error

$$\tilde{v}_{NN_2} = v_{NN_2} - \hat{v}_{NN_2} = W_2^T \mu_2\left(\chi_2\right) - \hat{W}_2^T \mu_2\left(\chi_2\right) + \varepsilon_2 = \tilde{W}_2^T \mu_2\left(\chi_2\right) + \varepsilon_2 \tag{3.39}$$

3.4 Solution and Feasibility of $\hat{F}_d$

Here, we show how to confront the bilinearity in the control which appears in the position dynamics given by (3.9).We derive the final control law based on the assumption that $\hat{F}_d$ is feasible, that is, it is possible to find a solution for u_d , φ_d and θ_d in (3.21).In fact, given any $\hat{F}_d$, the solution of (3.21) is easy using robot inverse kinematics [82] and is given here. Inverse Kinematics is one of the popular methods in robotics used for finding the joint variables given a desired Cartesian position [87].

For the problem discussed above, one must find angular positions $(u_d, \varphi_d, \theta_d)$ for a given $\hat{F}_d$ in

$$\begin{pmatrix} -u_d \sin\theta_d \\ u_d \cos\theta_d \sin\varphi_d \\ u_d \cos\theta_d \cos\varphi_d \end{pmatrix} = \hat{F}_d(t) \in R^3 \tag{3.40}$$

Define

$$a = u_d \sin\theta_d, b = u_d \cos\theta_d \tag{3.41}$$

Then

$$\begin{pmatrix} -a \\ b\sin\varphi_d \\ b\cos\varphi_d \end{pmatrix} = \hat{F}_d(t) \equiv \begin{pmatrix} f_{x_d}(t) \\ f_{y_d}(t) \\ f_{z_d}(t) \end{pmatrix} \tag{3.42}$$

so that

$$a = -f_{x_d}(t) \tag{3.43}$$

and

$$b^2 \sin^2 \varphi_d + b^2 \cos^2 \varphi_d = f_{y_d}^2(t) + f_{z_d}^2(t) \tag{3.44}$$

$$\Rightarrow b = \sqrt{f_{y_d}^2(t) + f_{z_d}^2(t)} \tag{3.45}$$

From (3.42),

$$(u_d \sin \theta_d)^2 + (u_d \cos \theta_d)^2 = a^2 + b^2 \tag{3.46}$$

$$\Rightarrow u_d^2 = a^2 + b^2 \tag{3.47}$$

$$u_d = \sqrt{a^2 + b^2} \tag{3.48}$$

and

$$\tan \theta_d = \frac{a}{b} \tag{3.49}$$

$$\theta_d = \tan^{-1}\left(\frac{a}{b}\right) \tag{3.50}$$

Finally from (3.42)

$$\tan \varphi_d = \frac{f_{y_d}}{f_{z_d}} \tag{3.51}$$

$$\varphi_d = \tan^{-1}\left(\frac{f_{y_d}}{f_{z_d}}\right) \tag{3.52}$$

This procedure shows that (3.21) has a solution $(u_d, \theta_d, \varphi_d)$ for all $\hat{F}_d$. We assume that $f_{z_d} \neq 0$ i.e. φ and ψ are not $90°$. Then $(u_d, \varphi_d, \theta_d)$ are given respectively by (3.48), (3.50) and (3.52).

3.5 Stability Analysis and NN Tuning

In previous Section, the control law given by (3.35) is derived from desired attitudes θ_d, φ_d and where θ_d, φ_d are obtained from the solution of (3.21) and given by (3.50), (3.52) respectively. The required lift u_d is given by (3.48), and desired yaw ψ_d is

an external reference command. The control structure is shown in Fig.3.1. There, one has an outer kinematics loop with inputs the desired reference positions (x_d, y_d, z_d), and an inner dynamics loop for attitude control to generate the required angles $(\varphi_d, \theta_d, \psi_d)$. Note that, the desired command inputs are $\xi_d(t) = (x_d(t), y_d(t), z_d(t))$ as well as desired yaw $\psi_d(t)$, which are given as reference trajectory inputs.

Based on the Lyapunov theory the feasibility of the control law given by (3.35) is discussed in this section, and tuning laws are given for the NN weights to guarantee stability. The related background literature can be found in [78].

Assumption 3.5.1. *a) Desired trajectory* $q_d = \begin{pmatrix} \xi_d^T & \psi_d \end{pmatrix}^T$ *is bounded i.e.* $\|q_d\| < q_B$ *for some known bound* W_B *. b) Ideal NN weights* W_1, W_2 *are bounded, i.e.* $\left\| \begin{matrix} W_1 \\ W_2 \end{matrix} \right\| < W_B$ *for some known bound* W_B *. c) The auxiliary matrix can be upper bounded as* $J(\eta) \leqslant J_M$ *with* $J_M \in \Re$ *a known positive constant.*

Note that assumption *a)* always holds. Assumption *b)* holds on a compact set, and assumption *c)* holds since J is an inertia matrix containing sines and cosines of $\begin{bmatrix} \varphi & \theta & \psi \end{bmatrix}^T$ [82] Now follows the main result given by Theorem 3.5.2 which proves that r_1 and r_2 are Uniformly Ultimately Bounded (UUB) if the NN are properly tuned. According to the definition given by (3.12) of r_1 and (3.28) of r_2 , this guarantees that e_1 and e_2 are UUB since

$$\begin{aligned} \|e_1\| &\leqslant \|s + \Lambda_1\|^{-1} \|r_1\| \leqslant (\sigma_{\min}(\Lambda_1))^{-1} \|r_1\| \\ \|e_2\| &\leqslant \|s + \Lambda_2\|^{-1} \|r_2\| \leqslant (\sigma_{\min}(\Lambda_2))^{-1} \|r_2\| \end{aligned} \tag{3.53}$$

where $\sigma_{\min}(\Lambda_i)$ is the minimum eigenvalue of $\Lambda_i, i = 1, 2$.

Theorem 3.5.2. *Given the system as described in* (3.9) *and* (3.10) *with a control architecture shown in Fig. 3.1 and control inputs given by* (3.48) *and* (3.35)*. Under the assumptions,*

suppose that $[K_{r_1} - m\Lambda_1] > 0, [K_{r_2} - J_M\Lambda_2] > 0$ *and the NN weight tuning laws are given as*

$$\left.\begin{aligned} \dot{\hat{W}}_1 &= F_1\mu_1 r_1^T - \kappa \|r\| F_1\hat{W}_1 \\ \dot{\hat{W}}_2 &= F_2\mu_2 r_2^T - \kappa \|r\| F_2\hat{W}_2 \end{aligned}\right\} \tag{3.54}$$

with proposed desired tuning parameter matrices $F_1 = F_1^T > 0, F_2 = F_2^T > 0$. *Then the errors* $r_1(t), r_2(t)$ *and the NN weight estimation errors* $\tilde{W}_1(t), \tilde{W}_2(t)$ *are UUB with practical bounds given by* (3.63) *and* (3.65). *Moreover the tracking errors* r_1, r_2 *can be made arbitrarily small with a suitable choice of design parameters* $K_{r_1}, K_{r_2}, \Lambda_1$ *and* Λ_2 .

Proof. Define, $r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \hat{W} = \begin{pmatrix} \hat{W}_1 \\ \hat{W}_2 \end{pmatrix}, \mu_r = \begin{pmatrix} \mu_1 r_1^T \\ \mu_2 r_2^T \end{pmatrix}$ and $F = \begin{bmatrix} \underline{F_1} & \underline{0} \\ \underline{0} & \underline{F_2} \end{bmatrix} > 0, \kappa > 0$

Consider the following Lyapunov function candidate:

$$\begin{aligned} L = &\frac{1}{2}r_1^T m r_1 + \frac{1}{2}r_2^T J r_2 + \frac{1}{2}\left[\int_0^t r_1^T dt'\right] K_{i_1} \left[\int_0^t r_1 dt'\right] \\ &+\frac{1}{2}\left[\int_0^t r_2^T dt'\right] K_{i_2} \left[\int_0^t r_2 dt'\right] + \frac{1}{2}tr\left\{\tilde{W}^T F^{-1}\tilde{W}\right\} \end{aligned} \tag{3.55}$$

Then,

$$\begin{aligned} \dot{L} = -r^T &\begin{bmatrix} K_{r_1} - m\Lambda_1 & 0 \\ 0 & K_{r_2} - J\Lambda_2 \end{bmatrix} r + r^T \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} + \frac{1}{2}r_2^T \dot{J} r_2 - r_2^T C r_2 \\ &+tr\left\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} + \mu_r\right)\right\} \end{aligned} \tag{3.56}$$

Now let,

$$\dot{\tilde{W}} = -F\mu_r + \kappa \|r\| F\hat{W} \tag{3.57}$$

Continuing with (3.56) using (3.57),

$$\dot{L} \leqslant -r^T K_v r + r^T \varepsilon_v + \frac{1}{2}r^T\left(\dot{J} - 2C\right) r + \kappa \|r\| tr\left\{\tilde{W}^T\hat{W}\right\} \tag{3.58}$$

where

$$K_v = \begin{bmatrix} K_{r_1} - m\Lambda_1 & 0 \\ 0 & K_{r_2} - J_M\Lambda_2 \end{bmatrix}, \varepsilon_v = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \tag{3.59}$$

The hypothesis guarantees $K_v > 0$. Therefore

$$\dot{L} \leqslant -r^T K_v r + r^T \varepsilon_v + \frac{1}{2} r^T \left(\dot{J} - 2C\right) r + \kappa \left\| r \right\| tr \left\{ \tilde{W}^T \left(W - \tilde{W} \right) \right\} \tag{3.60}$$

where, $\left(\dot{J} - 2C\right)$ is skew symmetric [88] and thus it can be shown that $r^T \left(\dot{J} - 2C\right) r = 0$.

Since, $tr \left\{ \tilde{W}^T \left(W - \tilde{W} \right) \right\} \leqslant \left\| \tilde{W} \right\|_F \|W\|_F - \left\| \tilde{W} \right\|_F^2$ with $\|\cdot\|_F$ the Frobenius norm, there results

$$\begin{aligned} \dot{L} &\leqslant -\|K_v\|_{\min} \|r\|^2 + \kappa \left\| r \right\| \left\{ \left\| \tilde{W} \right\|_F \|W\|_F - \left\| \tilde{W} \right\|_F^2 \right\} + \|\varepsilon_v\|_{\max} \left\| r \right\| \\ &= - \left\| r \right\| \left\{ \|K_v\|_{\min} \left\| r \right\| + \kappa \left\{ \left\| \tilde{W} \right\|_F^2 - \left\| \tilde{W} \right\|_F \|W\|_F \right\} - \|\varepsilon_v\|_{\max} \right\} \end{aligned} \tag{3.61}$$

where,$\|K_v\|_{\min}$ is the minimum possible norm of for different parameter values of K_v. Similarly,$\|\varepsilon_v\|_{\max}$ is the maximum possible vector norm of ε_v. Now (3.61) can be rewritten as,

$$\dot{L} \leqslant - \left\| r \right\| \left\{ \kappa \left[\left\| \tilde{W} \right\|_F - \frac{\|W\|_F}{2} \right]^2 - \kappa \frac{\|W\|_F^2}{4} + \|K_v\|_{\min} \left\| r \right\| - \|\varepsilon_v\|_{\max} \right\} \tag{3.62}$$

If $\|W\|_F \leqslant W_B$ then, $\dot{L} \leqslant 0$ if and only if,

$$\|r\| > \frac{\kappa \frac{W_B^2}{4} + \|\varepsilon_v\|_{\max}}{\|K_v\|_{\min}} \equiv b_r \tag{3.63}$$

or from (3.61) we can write considering positive definiteness of K_v and

$$\kappa \left\| \tilde{W} \right\|_F \left\{ \left\| \tilde{W} \right\|_F - \|W\|_F \right\} - \|\varepsilon_v\|_{\max} > 0 \tag{3.64}$$

$$\left\| \tilde{W} \right\|_F > \frac{W_B}{2} + \sqrt{\left(\frac{W_B^2}{4} + \frac{\|\varepsilon_v\|_{\max}}{\kappa} \right)} \equiv b_w \tag{3.65}$$

Thus, $\dot{L}$ is negative outside a compact set. Selecting the gain K_v ensures that the compact set defined by $\|r\| \leqslant b_r$ is bounded, so that the approximation property holds throughout. This ensures UUB for $\|r\|$ and $\left\|\tilde{W}\right\|_F$ stated in (3.63) and (3.65) respectively. □

3.6 Simulation Result

In practice the NN's can be initialized with zero weights. Then, during the initialization, the PID controllers defined by (3.19) and (3.35) with NN's removed stabilize the closed loop system until the NN weights begin to learn using the tuning laws (3.54). An assumption on the initial tracking error magnitude is technically needed. For details see [83] in page 198.

3.6.1 Rotorcraft model

Simulation for a typical quadrotor is performed using the following parameters (SI units):

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \Sigma = diag(\Sigma_x, \Sigma_y, \Sigma_z) = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}; g = 9.81$$

The centripetal/coriolis components are defined by (3.6) with the auxiliary matrix

$$J(\eta) = \begin{pmatrix} \Sigma_x s_\theta^2 + \Sigma_y c_\theta^2 s_\psi^2 + \Sigma_z c_\theta^2 c_\psi^2 & (\Sigma_y - \Sigma_z) c_\theta s_\psi c_\psi & -\Sigma_x s_\theta \\ (\Sigma_y - \Sigma_z) c_\theta c_\psi s_\psi & \Sigma_y c_\psi^2 + \Sigma_z s_\psi^2 & 0 \\ -\Sigma_x s_\theta & 0 & \Sigma_x \end{pmatrix} \tag{3.66}$$

The aerodynamic components are taken as representative functions as follows

$$A_{xyz} = \begin{pmatrix} \dot{x}^2 \dot{\varphi} \\ \dot{y}^2 \dot{\theta} \\ \dot{z}^2 \dot{\psi} \end{pmatrix}, A_\eta = \begin{pmatrix} \dot{x} \dot{\varphi}^2 \\ \dot{y} \dot{\theta}^2 \\ \dot{z} \dot{\psi}^2 \end{pmatrix} \tag{3.67}$$

The physical interpretation of A_{xyz}, A_η have been elaborated in Section 3.2. The nonlinear terms introduced here are merely a representation of unmodeled state dependent disturbances used in this particular study to show the effectiveness of the controller in rejecting these unknown terms. These terms are used in the simulation but are not required to implement the control law. In actual UAV, the aerodynamic forces and moments will be more complicated, but they are nevertheless effectively estimated by the NNs.

3.6.2 Neural Network Parameters

The number of NN neurons used in the simulation is 10. This number was selected as follows. The simulation was done with 10 neurons. Then, using 8 neurons the performance deteriorated. On the other hand, a simulation using 12 neurons did not improve performance. Therefore, 10 neurons are selected. NN weight tuning gains are given by $F_1 = diag\,(5\,[1;1;1])$;$F_2 = diag\,(5\,[1;1;1])$;$\kappa_1 = 0.01; \kappa_2 = 0.01$. The NN weights were initialized at zero, so that initially the tracking loops held the vehicle stable. Once the NNs begin to learn, their outputs are added to the control signals to better compensate the unknown nonlinear force and moment terms.

Two cases are considered in the simulation to show respectively the good tracking property of the control law and its decoupling effectiveness. In the all following cases we set the yaw angle to $\psi_d = 0$.

3.6.3 Effectiveness in Tracking a Desired (guided) Trajectory

Fig. 3.2 describes the controlled motion of the quadrotor from its initial position $(0, 0, 0)$ to final position $(20, 5, 10)$ for a given time (40 seconds). The desired trajectory were generated using reference trajectory generated in the Chapter 2. The actual trajectories $x(t), y(t), z(t)$ match exactly their desired values $x_d(t), y_d(t), z_d(t)$ respec-

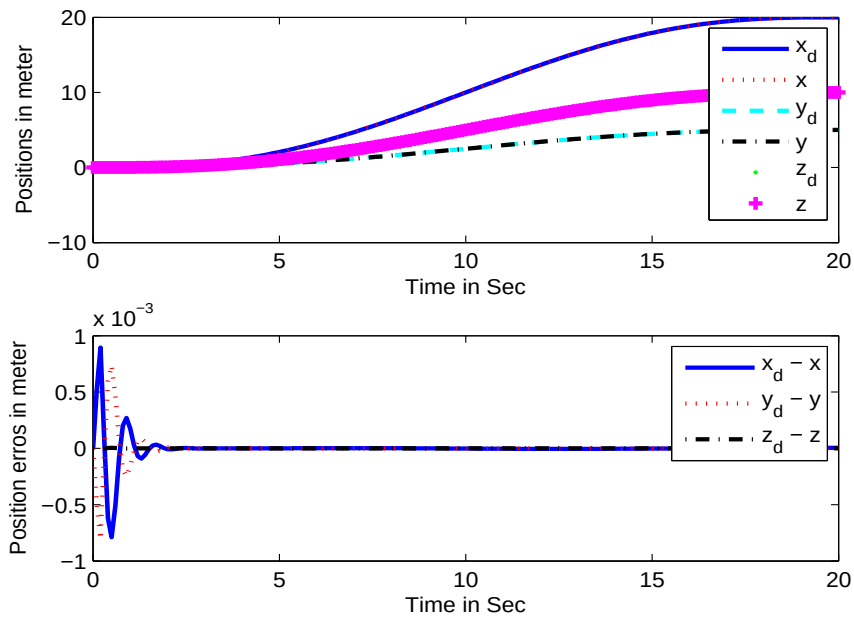

Figure 3.2. Three position commands simultaneously.

tively nearly exactly. The errors along the three axes are also shown in the same figure. It can be seen that the tracking is almost perfect as well as the tracking errors are significantly small. This is in spite of the unknown nonlinearities that are injected into the simulation dynamics, which are effectively compensated by the neural networks. Fig. 3.3 describes the attitude of the quadrotor φ, θ along with their demands φ_d, θ_d and attitude errors in radian. Again the angles match their command values nearly perfectly. Fig. 3.4 describes the control input requirement which is very much realizable. Note that as described before the control requirement for yaw angle is $\tau_\psi = 0$ and it is seen from Fig. 3.4.

3.6.4 Motion Decoupling in Response to Position Commands

The quadrotor is commanded to move from (0,5,10) to (20,5,10), which only involves a translation in the x direction. Fig 3.5-Fig. 3.6 illustrate the decoupling

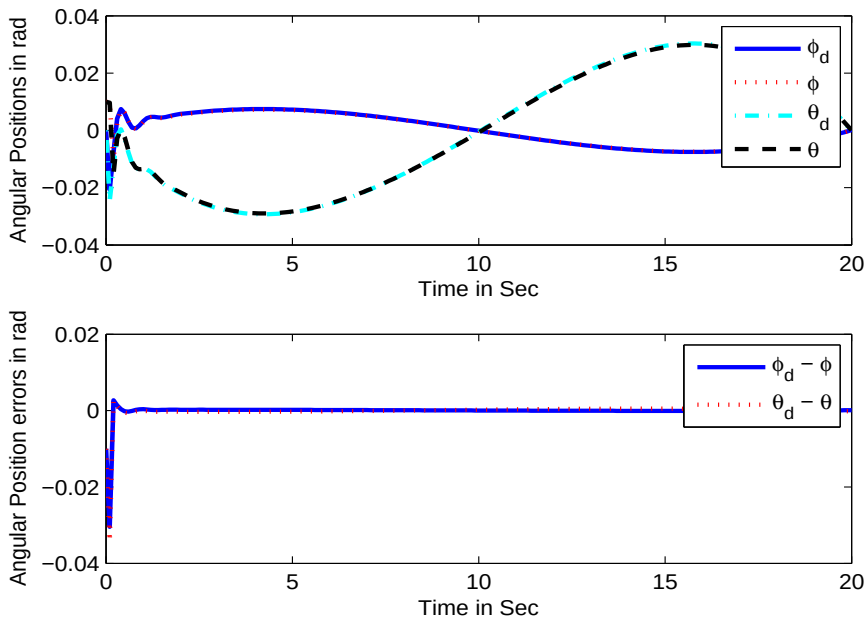

Figure 3.3. Resultant angular positions and errors .

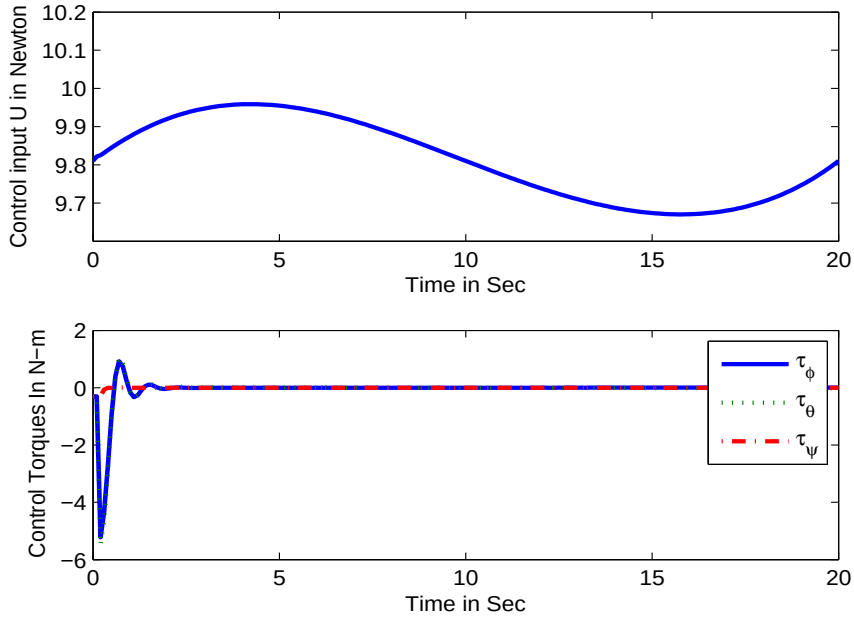

Figure 3.4. Input commands for Case I.

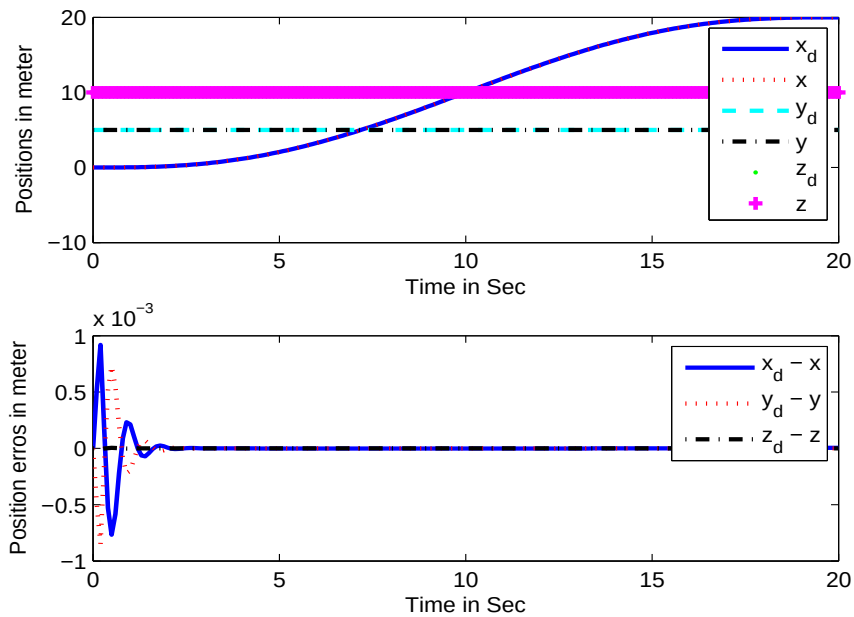

Figure 3.5. Plots of position and position tracking errors for command only.

performance of the control law. Fig 3.5 shows that $x(t)$ follows the command $x_d(t)$ nearly perfectly. The other variables $y(t)$ and $z(t)$ are held at their initial values. Fig. 3.6 shows that the change in x does not make any influence on φ. The corresponding control inputs are also shown in Fig. 3.7 and due to the full decoupling effect it is seen that τ_φ is almost zero.

Similar type of simulations performed for y and z directional motions separately yield similar plots that show excellent tracking with decoupling.

3.6.5 Comparison Study: Adaptive Backstepping Proposed in this Research vs.The controller Adopted from [1]

Reference [1] presented a three-loop design for a quadrotor controller. A first loop controlled altitude z and yaw. Assuming perfect control in that loop, a second loop stabilized lateral position y and roll to zero using a 'nested saturation control'. Assuming

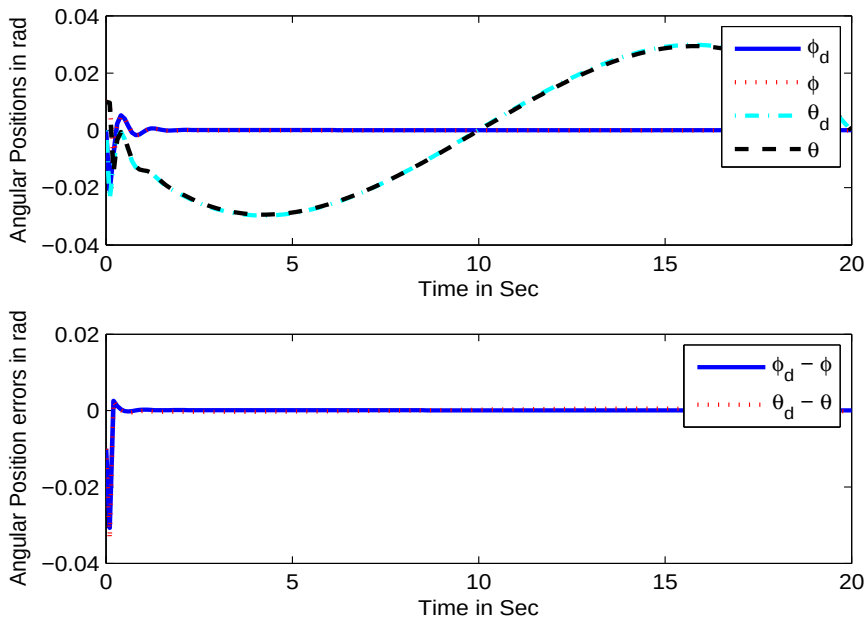

Figure 3.6. Resultant angular positions and errors .

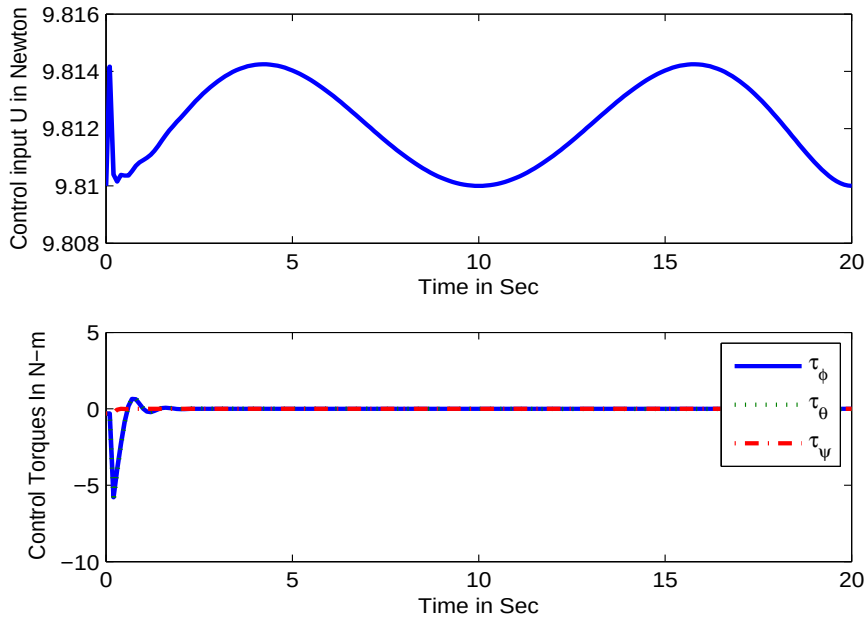

Figure 3.7. Input commands for variation in x (Case II).

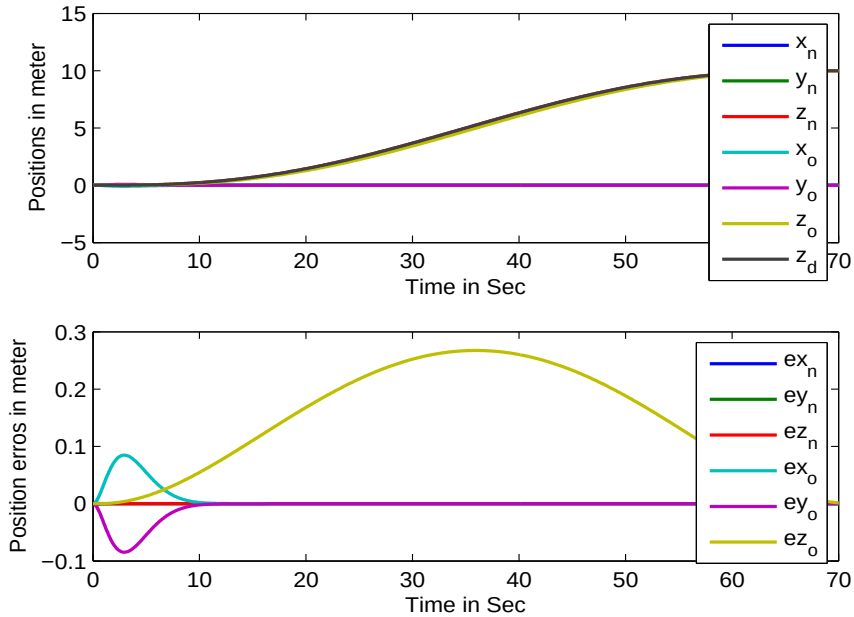

Figure 3.8. Case III: Comparison result: position response.

again perfect compensation, a third and final loop stabilized forward position x and pitch to zero. Though not developed as such in [1], this is actually a very interesting and intuitive form of backstepping control. Here we compare the performance of the controller in [1] with our controller for an altitude change command trajectory.

Fig. 3.8-Fig. 3.10 shows the comparison simulation results between the adaptive backstepping controller proposed in this research and the controller adopted from [1].

Fig. 3.8-Fig. 3.10 shows the simulation study for hovering control of a quadrotor. That is, the quadrotor is to be controlled along a guided desired trajectory z_d while x_d, y_d should remain silent. The response for the adaptive backstepping controller proposed in this research is depicted with subscript and the response of the controller described in [1] is depicted with subscript o . Firstly from Fig. 3.8 we can see that the controller proposed in this research follows the trajectory z_d more accurately than the backstepping controller in [1] (see error response $z_d - z_o$). From the error responses found in

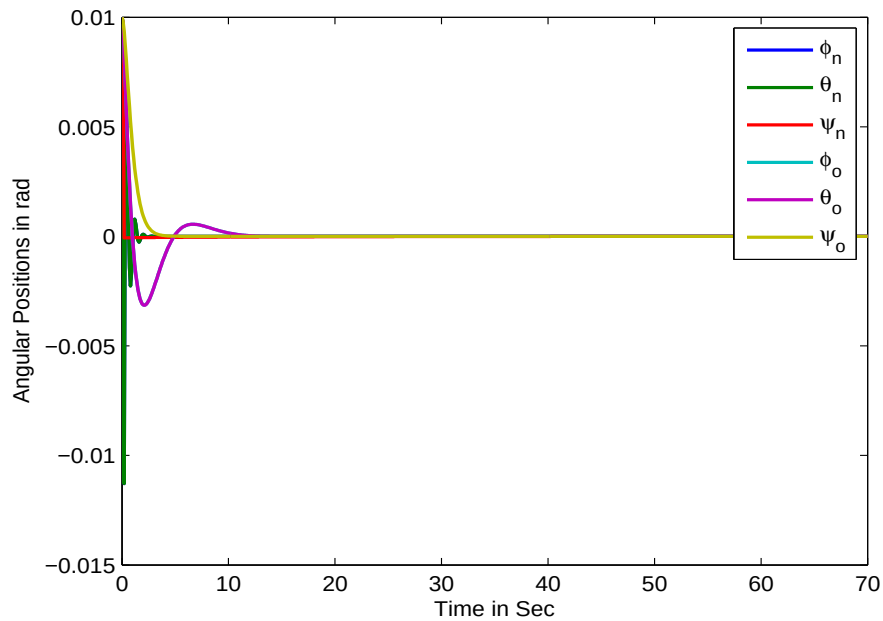

Figure 3.9. Case III: Comparison result: attitude response.

the simulation, it is seen that NN backstepping controller proposed in this research also has less position error magnitude in x and y directions than the backstepping controller in [1]. This shows that the controller proposed in this research has more decoupling capacity than the controller proposed in [1]. Note that the controller in [1] did not consider nonlinear forces and moments.

From Fig. 3.9 it is seen that the attitude stabilization occurs more quickly for the controller proposed in this research than controller proposed in [1]. The thrust input shown in Fig. 3.10 shows the chattering due to the nested saturation controller in [1], and reveals that our backstepping controller generates a smoother control. It is also seen from Fig. 20-Fig. 22, that initial variations in attitude and torque are higher for our controller than that in [1]. This is because of the transient phase of NN weights from their initial assigned values of zero to the learned final values for effective compensation

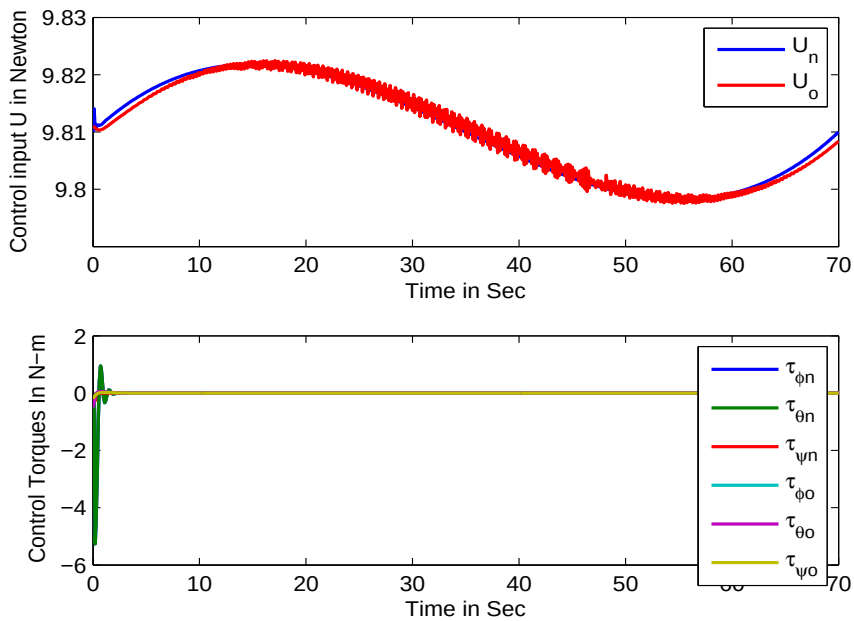

Figure 3.10. Case III: Comparison result: input commands.

of the unknown nonlinearities. From the simulation it is seen, however, that after a very short time these transients vanish.

Remark 3.6.1. *We conclude from the simulation results that NN backstepping controller gives satisfactory results for a wide range of nonlinearities. The resulting controller structure in Fig. 3.1 yields tracking with excellent decoupling between command channels. The NN learns on-line to effectively compensate for the unknown aerodynamics forces and moments as well as the nonlinear terms* $C(\eta, \dot{\eta})$ *.*

3.7 Conclusion

A backstepping approach to design nonlinear controller for a quadrotor dynamics is discussed in this research. Using this approach, an intuitively structured controller was derived that has an outer kinematics position control loop and an inner dynamic attitude control loop. The dynamics of a quadrotor are a simplified form of

helicopter dynamics that exhibits the basic problems including underactuation, strong coupling, multi-input/multi-output, and unknown nonlinearities. The force dynamics of the quadrotor is completely independent of the rotational dynamics except through the aerodynamic state dependent disturbances including aerodynamic coefficients. These unmodeled state dependent dynamics are estimated by NN and thus yielding a robust control law. The simulation results shown at the end demonstrate the validity of the control law discussed in the research.

CHAPTER 4

DISTRIBUTED ADAPTIVE CONTROL FOR SYNCHRONIZATION OF UNKNOWN NONLINEAR NETWORKED SYSTEMS

4.1 Introduction

Coordination and consensus of distributed groups of agents is inspired by naturally occurring phenomena such as flocking in birds, swarms in insects, circadian rhythms in nature, synchronization and phase transitions in physical and chemical systems, and the laws of thermodynamics [35]. Early work has been done in the control systems community by [36, 37, 38, 39, 40], which by now are well known. Consensus has been studied for systems on communication graphs with fixed or varying topologies and communication delays. The average consensus problem has garnered much interest. Synchronization to time-varying trajectories has been studied based on physical or natural systems by [41, 42, 43, 44]. Synchronization of nonlinear passive dynamical systems has been studied by [45]. Consensus using nonlinear protocols has been considered there and in [35].

Convergence of consensus to a virtual leader or header node has been studied in [39, 46]. Dynamic consensus for tracking of time-varying signals has been presented in [47]. Recently, the pinning control has been introduced for synchronization control of coupled complex dynamical systems [48, 49, 50, 51, 51]. Pinning control is a powerful technique that allows controlled synchronization of interconnected dynamical systems by adding a control or leader node that is connected (pinned) into a small percentage of nodes in the network. These pinned or controlled nodes view the control node simply as another neighbor, and consider the control node's state value in computing their local

protocols. Analysis shows that all nodes converge to the state of the control node, which may be time-varying. Analysis has been done using Lyapunov techniques by assuming either a Jacobian linearization of the nonlinear node dynamics or a Lipschitz condition. A related idea is soft control [52] where a shill node moves through the network, and is perceived by existing nodes simply as another neighbor for purposes of computation of their own averaging protocols. Proper placement and motion of the shill agent results in consensus to the state of the shill.

Consensus and collective motion of distributed agents has been analyzed using the theory of graphs and/or Markov processes. Recent publications allow analysis using traditional control theory notions including matrix analysis, Lyapunov theory, etc., upon the introduction of certain key definitions including irreducibility, M-matrices, Frobenius form, special Lyapunov forms, etc. Notable are the books [53, 54]. Such techniques allow one to bring in the machinery of matrix analysis [55]. Instrumental in this analysis are the techniques employed in [56]. Distributed multiagent systems with unknown nonlinear dynamics and disturbances were studied in [57] where distributed adaptive controllers were designed to achieve robust consensus. That treatment assumed undirected graphs and solved the consensus problem, that is, the nodes reach a steady-state consensus that depends on the initial conditions. Expressions for the consensus value were not given.

The study of control protocols on digraphs is significantly more involved than their study on undirected graphs, where the graph Laplacian potential can be taken as a Lyapunov function. In this Chapter we present a Lyapunov technique for design of protocols for robust synchronization to tracking of a leader or control node [74]. Design techniques are developed for general digraphs. The leader has unknown nonlinear dynamics, and the nodes have unknown, non-identical, nonlinear dynamics and disturbances. Suitable control protocols are derived using a Lyapunov function which is

carefully crafted to depend on a special local synchronization error which can be computed in a distributed fashion. The control laws thus derived are distributed in nature and can be implemented locally by each node. They consist of a linear protocol plus a nonlinear adaptive learning term. These protocols are robust to uncertain disturbances and dynamics, and to modeling errors in a sense to be made precise. It is shown that the synchronization error converges to a residual set. Singular value analysis is used. It is shown that the singular values of certain key matrices are intimately related to structural properties of the graph. Simulation results are given to show the effectiveness of the proposed method.

4.2 Synchronization Control Formulation

Consider a graph $G = (V, E)$ with a nonempty finite set of N nodes $V = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $E \subseteq V \times V$. We assume the graph is simple, e.g. no repeated edges and $(v_i, v_i) \notin E$, $\forall i$ no self loops. General directed graphs are considered. Denote the connectivity matrix as $A = [a_{ij}]$ with $a_{ij} > 0 \; if \; (v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note $a_{ii} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D = [d_i]$ with $d_i = \sum\limits_{j \in N_i} a_{ij}$ the weighted in-degree of node i (i.e. i^{th} row sum of A). Define the graph Laplacian matrix as $L = D - A$, which has all row sums equal to zero. Define $d_i^o = \sum\limits_j a_{ji}$, the (weighted) out-degree of node i, that is the ith column sum of A.

We assume the communication digraph is strongly connected, i.e. there is a directed path from v_i to v_j for all distinct nodes $v_i, v_j \in V$. Then A and L are irreducible

[53]. That is they are not cogredient to a lower triangular matrix, i.e., there is no permutation matrix U such that

$$L = U \begin{bmatrix} * & 0 \\ * & * \end{bmatrix} U^T \tag{4.1}$$

The results of this research can easily be extended to graphs having a spanning tree (i.e. not necessarily strongly connected) using the Frobenius form in (4.1).

4.2.1 Synchronization Control Problem

Consider node dynamics defined for the ith node as

$$\dot{x}_i = f_i(x_i) + u_i + w_i(t) \tag{4.2}$$

where $x_i(t) \in R$ is the state of node i, $u_i(t) \in R$ its the control input, and $w_i(t) \in R$ is a disturbance acting upon each node. Note that each node may have its own distinct dynamics. Standard assumptions for existence of unique solutions are made, e.g. $f_i(x_i)$ either continuously differentiable or Lipschitz. The overall graph dynamics is

$$\dot{x} = f(x) + u + w \tag{4.3}$$

where the overall (global) state vector is $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T \in R^N$, global node dynamics vector is $f(x) = \begin{bmatrix} f_1(x_1) & f_2(x_2) & \cdots & f_N(x_N) \end{bmatrix}^T \in R^N$, input $u = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}^T \in R^N$, and $w = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^T \in R^N$.

Definition 4.2.1. *The local neighborhood synchronization error for node i is defined as [50, 56]*

$$e_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) + b_i(x_0 - x_i) \tag{4.4}$$

with pinning gains $b_i \geqslant 0$, and $b_i > 0$ for at least one i. Then, $b_i \neq 0$ if and only if there exist an arc from the control node to the ith node in G. We refer to the nodes i for which $b_i \neq 0$ as the pinned or controlled nodes.

Note that (4.4) represents the information that is available to any node i for control purposes.

The state of the leader or control node is $x_0(t)$ which satisfies the (generally nonautonomous) dynamics

$$\dot{x}_0 = f(x_0, t) \tag{4.5}$$

A special case is the standard constant consensus value with $\dot{x}_0 = 0$. The drift term could represent, e.g. motion dynamics of the control node. That is, we assume that the control node can have a time-varying state.

The Synchronization control design problem confronted herein is as follows: Design control protocols for all the nodes in G to synchronize to the state of the control node, i.e. one requires $x_i(t) \to x_0(t),\ \forall i$. It is assumed that the dynamics of the control node is unknown to any of the nodes in G. It is assumed further that both the node nonlinearities $f_i(.)$ and the node disturbances $w_i(t)$ are unknown. Thus, the synchronization protocols must be robust to unmodelled dynamics and unknown disturbances.

In fact, (4.5) is a command generator. Therefore we are considering a distributed command generator tracker problem with unknown node dynamics and unknown command generator dynamics.

4.2.2 Synchronization Error Dynamics

From (4.4), the global error vector for network G is given by

$$e = -(L+B)\left(x - \underline{1}x_0\right) = -(L+B)\left(x - \underline{x}_0\right) \tag{4.6}$$

where, $e = \begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix}^T \in R^N$ and $\underline{x}_0 = \underline{1}x_0 \in R^N$. $B \in R^{N\times N}$ is a diagonal matrix with diagonal entries b_i and $\underline{1}$ is the N-vector of ones. Differentiating (4.6),

$$\dot{e} = -(L+B)\left(\dot{x} - \dot{\underline{x}}_0\right) \ = -(L+B)\left[f(x) - \underline{f}(x_0, t) + u + w(t)\right] \tag{4.7}$$

where $\underline{f}(x_0, t) = \underline{1} f(x_0(t), t) \in R^N$.

Remark 4.2.2. *If the node states are vectors $x_i(t) \in R^n$ and $x_0(t) \in R^n$, then $x, e \in R^{nN}$ and* (4.6) *becomes*

$$\dot{e} = -\left[(L+B) \otimes I_n\right] \left[f(x) - \underline{f}(x_0, t) + u + w(t) \right] \tag{4.8}$$

with $\otimes$ the Kronecker product. To avoid obscuring the essentials, throughout the research we take the node states as scalars, $x_i(t) \in R$. If they are vectors, all of the following development is easily modified by introducing the Kronecker product terms as appropriate. In fact, a simulation example is presented for the case $x_i(t) \in R^2$, namely $1 - D$ motion control for coupled inertial agents.

Remark 4.2.3. *Note that*

$$\delta = \left(x - \underline{x}_0 \right) \tag{4.9}$$

is the disagreement vector in [37]. We do not use this error herein because it is a global quantity that cannot be computed locally at each node, in contrast to the local neighborhood error (4.4)*. As such, it is suitable for analysis but not for distributed controls design using Lyapunov techniques.*

Remark 4.2.4. *We take the communication digraph as strongly connected. Therefore, if $b_i \neq 0$ for at least one i then $(L+B)$ is an irreducibly diagonally dominant M-matrix and hence nonsingular [53].*

An M -matrix is a square matrix having its off-diagonal entries nonpositive and all principal minors nonnegative. Based on Remark 4.2.4, the next result is therefore obvious from (4.6) and the Cauchy Schwartz inequality. (see also [56]).

Lemma 4.2.5. *Let the graph be strongly connected and $B \neq 0$. Then*

$$\|\delta\| \leqslant \|e\| / \underline{\sigma}(L+B) \tag{4.10}$$

with $\underline{\sigma}(L+B)$ the minimum singular value of $(L+B)$, and $e=0$ if and only if the nodes synchronize, that is

$$x(t) = \underline{x}_0(t) \tag{4.11}$$

4.2.3 Synchronization Control Design

The control problem confronted in this research is to design a control strategy so that local neighborhood error $e(t)$ is bounded to a small residual set. Then according to Lemma 4.2.5 all nodes synchronize so that $\|x_i(t) - x_0(t)\|$ is small $\forall i$. It is assumed that the dynamics $f(x_0, t)$ of the control node is unknown to any of the nodes in G. It is assumed further that both the node nonlinearities $f_i(.)$ and the node disturbances $w_i(t)$ are unknown. As such, the synchronization protocols must be robust to unmodelled dynamics and unknown disturbances.

To achieve this goal, define the input u_i for node i as

$$u_i = v_i - \hat{f}_i(x_i) \tag{4.12}$$

where, $\hat{f}_i(x_i)$ is an estimate of $f_i(x_i)$ and $v_i(t)$ is an auxiliary control signal to be designed via Lyapunov techniques in Theorem 4.3.6. This can be written in vector form for the overall network of N nodes as

$$u = v - \hat{f}(x) \tag{4.13}$$

where $v = \begin{bmatrix} v_1 & v_2 & \cdots & v_N \end{bmatrix}^T \in R^N$,

$$\hat{f}(x) = \begin{bmatrix} \hat{f}_1(x_1) & \hat{f}_2(x_2) & \cdots & \hat{f}_N(x_N) \end{bmatrix}^T \in R^N$$

. Then from (4.7) one gets

$$\dot{e} = -(L+B)\left[f(x) - \underline{f}(x_0, t) - \hat{f}(x) + v + w\right] \tag{4.14}$$

Following the techniques in [11], assume that the unknown nonlinearities in (4.2) are locally smooth and thus can be approximated on a compact set $\Omega_i \in R$ by

$$f_i\left(x_i\right) = W_i^T \varphi_i(x_i) + \varepsilon_i \tag{4.15}$$

with $\varphi_i(x_i) \in R^{\nu_i}$ a suitable basis set of ν_i functions at each node i and $W_i \in R^{\nu_i}$ a set of unknown coefficients. According to the neural network (NN) approximation literature [89], a variety of basis sets can be selected, including sigmoids, gaussians, etc. There $\varphi_i(x_i) \in R^{\nu_i}$ is known as the NN activation function vector and $W_i \in R^{\nu_i}$ as the NN weight matrix. The ideal approximating weights $W_i \in R^{\nu_i}$ in (4.15) are assumed unknown. The intention is to select only a small number ν_i of NN neurons at each node (see Simulations).

To compensate for unknown nonlinearities, each node will maintain a neural network locally to keep track of the current estimates for the nonlinearities. The idea is to use the information of the states from the neighbors of node i to evaluate the performance of the current control protocol along with the current estimates of the nonlinear functions. Therefore, select the local node's approximation $\hat{f}_i(x_i)$ as

$$\hat{f}_i\left(x_i\right) = \hat{W}_i^T \varphi_i(x_i) \tag{4.16}$$

where $\hat{W}_i \in R^{\nu_i}$ is a current estimate of the NN weights for node i, and ν_i is the number of NN neurons maintained at each node i. It will be shown in Theorem 4.3.6 how to select the estimates of the parameters $\hat{W}_i \in R^{\nu_i}$ using the local neighborhood synchronization errors (4.4).

The global node nonlinearity $f(x)$ for G is now written as

$$f(x) = W^T \varphi(x) + \varepsilon \tag{4.17}$$

where $W^T = diag\{W_i^T\}$, $\varphi(x) = \begin{bmatrix} \varphi_1{}^T(x_1) & \varphi_2{}^T(x_2) & \cdots & \varphi_N{}^T(x_N) \end{bmatrix}^T$ and $\varepsilon = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_N \end{bmatrix}^T \in R^N$. The estimate $\hat{f}(x)$ is

$$\hat{f}(x) = \hat{W}^T \varphi(x) \tag{4.18}$$

with $\hat{W}^T = diag\{\hat{W}_i^T\}$. Now, the error dynamics (4.14) takes the form $v(t)$

$$\dot{e} = -(L+B)\left[\tilde{f}(x) + v + w(t) - \underline{f}(x_0, t)\right] \tag{4.19}$$

where the parameter estimation error is $\tilde{W}_i = W_i - \hat{W}_i$ and the function estimation error is

$$\tilde{f}(x) = f(x) - \hat{f}(x) = \tilde{W}^T \varphi(x) + \varepsilon \tag{4.20}$$

with $\tilde{W}^T = W^T - \hat{W}^T$. Therefore, one obtains finally the error dynamics

$$\dot{e} = -(L+B)\left[\tilde{W}^T \varphi(x) + v + \varepsilon + w(t) - \underline{f}(x_0, t)\right] \tag{4.21}$$

with $v(t)$ an auxiliary control yet to be designed.

4.3 Lyapunov Design for Cooperative Systems: Distributed NN Tuning Protocols

Now we show how to select the auxiliary control and NN weight tuning laws such as to guarantee that all nodes synchronize to the desired control node signal, i.e., $x_i(t) \to x_0(t),\ \forall i$. It is assumed that the dynamics $f(x_0, t)$ of the control node (which could represent its motion) are unknown to any of the nodes in G. It is assumed further that the node disturbances $w_i(t)$ are unknown. The Lyapunov analysis technique approach of [77] is used, though there are some complications arising from the fact that $v(t)$ and the NN weight tuning laws must be implemented as distributed protocols. This entails a careful selection of the Lyapunov function.

The singular values of a matrix M are denoted $\sigma_i(M)$ with $\bar{\sigma}(M)$ the maximum singular value and $\underline{\sigma}(M)$ the minimum singular value. The Frobenius norm is $\|M\|_F =$

$\sqrt{tr\{M^T M\}}$ with $tr\{\cdot\}$ the trace. The Frobenius inner product of two matrices is $\langle M_1, M_2 \rangle_F = \sqrt{tr\{{M_1}^T M_2\}}$.

The following Fact gives two standard results used in neural adaptive control [77].

Fact 4.3.1. *Let the nonlinearities $f(x)$ in (4.17) be smooth on a compact set $\Omega \in R^N$. Then:*

1. *The NN estimation error $\varepsilon(x)$ is bounded by $\|\varepsilon\| \leqslant \varepsilon_M$ on Ω, with ε_M a fixed bound [89].*
2. *Weierstrass higher-order approximation theorem. Select the activation functions $\varphi(x)$ as a complete independent basis (e.g. polynomials). Then NN estimation error $\varepsilon(x)$ converges uniformly to zero on Ω as $\nu_i \to \infty,\ i = 1, N$. That is $\forall \xi > 0$ there exist $\bar{\nu}_i,\ i = 1, N$ such that $\nu_i > \bar{\nu}_i,\ \forall i$ implies $\sup_{x \in \Omega} \|\varepsilon(x)\| < \xi$ [90].*

The following standard assumptions are required. Although the bounds mentioned are assumed to exist, they are not used in the design and do not have to be known. They appear in the error bounds in the proof of Theorem 4.3.6. (Though not required, if desired, standard methods can be used to estimate these bounds including [91].)

Assumption 4.3.2.

1. *The unknown disturbance w_i is bounded for all i. Thus the overall disturbance vector w is also bounded by $\|w\| \leqslant w_M$ with w_M a fixed bound.*
2. *The unknown consensus variable dynamics $\underline{f}(x_0, t)$ is bounded so that $\left\|\underline{f}(x_0, t)\right\| \leqslant F_M,\ \forall t$.*
3. *The target trajectory is in a bounded region, e.g. $\|x_0(t)\| < X_0,\ \forall t$, with X_0 a constant bound.*
4. *Unknown ideal NN weight matrix W is bounded by $\|W\|_F \leqslant W_M$.*
5. *NN activation functions φ_i are bounded $\forall i$, so that one can write for the overall network that $\|\varphi\| \leqslant \phi_M$.*

The next definitions extend standard notions [77, 78] to synchronization for distributed systems.

Definition 4.3.3. *The global neighborhood error* $e(t) \in R^N$ *is uniformly ultimately bounde -d (UUB) if there exists a compact set* $\Omega \subset R^N$ *so that* $\forall e(t_0) \in \Omega$ *there exists a bound* B_m *and a time* $t_f(B_m, e(t_0))$*, both independent of* $t_0 \geqslant 0$*, such that* $\|e(t)\| \leqslant B_m \; \forall t \geqslant t_0 + t_f$.

Definition 4.3.4. *The control node trajectory* $x_0(t)$ *given by* (4.5) *is cooperative UUB with respect to solutions of node dynamics* (4.2) *if there exist a compact set* $\Omega \subset R$ *so that* $\forall \;\; (x_i(t_0) - x_0(t_0)) \in \Omega$ *there exist a bound* B_m *and a time* $t_f(B_m, (x_i(t_0) - x_i(t_0)))$*, both independent of* $t_0 \geqslant 0$*, such that* $\|x_i(t) - x_0(t)\| \leqslant B_m \; \forall i, \; \forall t \geqslant t_0 + t_f$.

The next key constructive result is needed. An M-matrix is a square matrix having nonpositive off-diagonal elements and all principal minors nonnegative.

Lemma 4.3.5. *[53] Let* L *be irreducible and* B *have at least one diagonal entry* $b_i > 0$*. Then* $(L + B)$ *is a nonsingular* $M-$*matrix. Define*

$$q = \begin{bmatrix} q_1 & q_2 & \cdots & q_N \end{bmatrix}^T = (L + B)^{-1} \underline{1} \tag{4.22}$$

$$P = diag\,\{p_i\} \equiv diag\,\{1/q_i\} \tag{4.23}$$

Then $P > 0$ *and the matrix* Q *defined as*

$$Q = P\,(L + B) + (L + B)^T P \tag{4.24}$$

is positive definite.

The main result of this research is given by the following theorem, which shows how to design the control protocols (4.12) and tune the NN weights such that the local neighborhood cooperative errors for all nodes are UUB, which implies consensus variable $x_0(t)$ is cooperative UUB, thereby showing synchronization and cooperative stability for the whole network G.

Theorem 4.3.6. ***Distributed Adaptive Control Protocol for Synchronization***

Consider the networked systems given by (4.2), (4.3) *under Assumption 4.3.2. Let the communication digraph be strongly connected. Select the auxiliary control signals in* (4.12) *as* $v_i(t) = ce_i(t)$ *with the neighborhood synchronization errors* $e_i(t)$ *defined in*(4.4) *so that the local node control protocols are given by*

$$u_i = ce_i - \hat{f}_i(x_i) = c\sum_{j\in N_i} a_{ij}\left(x_j - x_i\right) + cb_i\left(x_0 - x_i\right) \quad - \hat{W}_i^T\varphi_i(x_i) \tag{4.25}$$

or

$$u = ce - \hat{W}^T\varphi(x) \tag{4.26}$$

with control gains $c > 0$. *Let local node NN tuning laws be given by*

$$\dot{\hat{W}}_i = -F_i\varphi_i e_i^T p_i(d_i + b_i) - \kappa F_i\hat{W}_i \tag{4.27}$$

with $F_i = \Pi_i I_{\nu_i}$, I_{ν_i} *the* $\nu_i \times \nu_i$ *identity matrix,* $\Pi_i > 0$ *and* $\kappa > 0$ *scalar tuning gains, and* $p_i > 0$ *defined in* (4.23). *Select* $\kappa = \frac{1}{2}c\underline{\sigma}\left(Q\right)$ *and the control gain* c *so that*

$$c\underline{\sigma}(Q) > \tfrac{1}{2}\phi_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right) \tag{4.28}$$

with $P > 0$, $Q > 0$ *the matrices in Lemma 4.3.5 and* A *the graph adjacency matrix. Then there exist numbers of neurons* $\bar{\nu}_i, i = 1, N$ *such that for* $\nu_i > \bar{\nu}_i,\ \forall i$ *the overall local cooperative error vector* $e(t)$ *and the NN weight estimation errors* $\tilde{W}$ *are UUB, with practical bounds given by* (4.46) *and* (4.47) *respectively. Therefore the control node trajectory* $x_0(t)$ *is cooperative UUB and all nodes synchronize to* $x_0(t)$. *Moreover, the bounds on local consensus errors* (4.4) *can be made small by increasing the control gains* c.

Proof. **Part I**

We claim that for a fixed $\varepsilon_M > 0$, there exist numbers of neurons $\bar{\nu}_i, i = 1, N$ such that for $\nu_i > \bar{\nu}_i,\ \forall i$ the NN approximation error is bounded by $\|\varepsilon\| \leqslant \varepsilon_M$. The claim is proven in Part II of the proof. Consider now the Lyapunov function candidate

$$V = \frac{1}{2}e^T Pe + \frac{1}{2}tr\left\{\tilde{W}^T F^{-1}\tilde{W}\right\} \tag{4.29}$$

with $e(t)$ the vector of local neighborhood cooperative errors (4.4), $0 < P = P^T \in R^{N\times N}$ the diagonal matrix defined in (4.23), and F^{-1} a block diagonal matrix defined in terms of $F = diag\{F_i\}$. Then,

$$\dot{V} = e^T P \dot{e} + tr\left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\} \tag{4.30}$$

and from (4.21)

$$\dot{V} = -e^T P\left(L+B\right)\left[\tilde{W}^T\varphi(x) + ce + \varepsilon + w - \underline{f}(x_0,t)\right] + tr\left\{\tilde{W}^T F^{-1}\dot{\tilde{W}}\right\} \tag{4.31}$$

$$\begin{gathered}\dot{V} = -ce^T P\left(L+B\right)e - e^T P\left(L+B\right)\left\{\varepsilon + w - \underline{f}(x_0,t)\right\} - \\ e^T P\left(L+B\right)\tilde{W}^T\varphi(x) + tr\left\{\tilde{W}^T F^{-1}\dot{\tilde{W}}\right\}\end{gathered} \tag{4.32}$$

$$\begin{gathered}\dot{V} = -ce^T P\left(L+B\right)e - e^T P\left(L+B\right)\left\{\varepsilon + w - \underline{f}(x_0,t)\right\} + \\ tr\left\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi e^T P\left(L+B\right)\right)\right\}\end{gathered} \tag{4.33}$$

$$\begin{gathered}\dot{V} = -ce^T P\left(L+B\right)e - e^T P\left(L+B\right)\left\{\varepsilon + w - \underline{f}(x_0,t)\right\} + \\ tr\left\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi e^T P\left(D+B-A\right)\right)\right\}\end{gathered} \tag{4.34}$$

Since L is irreducible and B has at least one diagonal entry $b_i > 0$, then $(L+B)$ is a nonsingular $M-$matrix. Defining therefore Q according to (4.24) one has

$$\begin{gathered}\dot{V} = -\frac{1}{2}ce^T Qe - e^T P\left(L+B\right)\left\{\varepsilon + w - \underline{f}(x_0,t)\right\} + \\ tr\left\{\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi e^T P\left(D+B\right)\right)\right\} + tr\left\{\tilde{W}^T\varphi e^T PA\right\}\end{gathered} \tag{4.35}$$

Adopt now the NN weight tuning law (4.27) or $\dot{\tilde{W}}_i = F_i\varphi_i e_i^T p_i(d_i + b_i) + \kappa F_i\hat{W}_i$. Since P and $(D+B)$ are diagonal and $\dot{W}$ has the form in (4.18), one has

$$\begin{gathered}\dot{V} = -\frac{1}{2}ce^T Qe - e^T P\left(L+B\right)\left\{\varepsilon + w(t) - \underline{f}(x_0,t)\right\} + \\ \kappa tr\left\{\tilde{W}^T(W - \tilde{W})\right\} + tr\left\{\tilde{W}^T\varphi e^T PA\right\}\end{gathered} \tag{4.36}$$

Therefore, for fixed $\varepsilon_M > 0$

$$\begin{aligned}\dot{V} \leqslant -\frac{1}{2}c\underline{\sigma}(Q)\|e\|^2 + \|e\|\,\bar{\sigma}\,(P)\,\bar{\sigma}\,(L+B)\,(\varepsilon_M + w_M + F_M) + \\ \kappa W_M\left\|\tilde{W}\right\|_F - \kappa\left\|\tilde{W}\right\|_F^2 + \left\|\tilde{W}\right\|_F\|e\|\,\phi_M\bar{\sigma}\,(P)\,\bar{\sigma}\,(A)\end{aligned} \tag{4.37}$$

Then

$$\begin{aligned}\dot{V} \leqslant -\begin{bmatrix} \|e\| & \left\|\tilde{W}\right\|_F \end{bmatrix}\begin{bmatrix} \frac{1}{2}c\underline{\sigma}(Q) & -\frac{1}{2}\phi_M\bar{\sigma}\,(P)\,\bar{\sigma}\,(A) \\ -\frac{1}{2}\phi_M\bar{\sigma}\,(P)\,\bar{\sigma}\,(A) & \kappa \end{bmatrix}\begin{bmatrix} \|e\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix} \\ + \begin{bmatrix} B_M\bar{\sigma}\,(P)\,\bar{\sigma}\,(L+B) & \kappa W_M \end{bmatrix}\begin{bmatrix} \|e\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix}\end{aligned}$$

with

$$B_M \equiv \varepsilon_M + w_M + F_M$$

. Write this as

$$\dot{V} \leqslant -z^T R z + r^T z \tag{4.38}$$

Then $\dot{V} \leqslant 0$ if R is positive definite and

$$\|z\| > \frac{\|r\|}{\underline{\sigma}(R)} \tag{4.39}$$

According to (4.29) one has

$$\tfrac{1}{2}\underline{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{\max}}\left\|\tilde{W}\right\|_F^2 \leqslant V \leqslant \tfrac{1}{2}\bar{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{\min}}\left\|\tilde{W}\right\|_F^2 \tag{4.40}$$

$$\begin{aligned}\tfrac{1}{2}\begin{bmatrix} \|e\| & \left\|\tilde{W}\right\|_F \end{bmatrix}\begin{bmatrix} \underline{\sigma}(P) & 0 \\ 0 & \frac{1}{\Pi_{\max}} \end{bmatrix}\begin{bmatrix} \|e\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix} \leqslant V \leqslant \\ \tfrac{1}{2}\begin{bmatrix} \|e\| & \left\|\tilde{W}\right\|_F \end{bmatrix}\begin{bmatrix} \bar{\sigma}(P) & 0 \\ 0 & \frac{1}{\Pi_{\min}} \end{bmatrix}\begin{bmatrix} \|e\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix}\end{aligned} \tag{4.41}$$

with $\Pi_{\min}, \Pi_{\max}$ the minimum and maximum values of Π_i. Define variables to write $\frac{1}{2}z^T\underline{S}z \leqslant V \leqslant \frac{1}{2}z^T\bar{S}z$. Then

$$\tfrac{1}{2}\underline{\sigma}(\underline{S})\|z\|^2 \leqslant V \leqslant \tfrac{1}{2}\bar{\sigma}(\bar{S})\|z\|^2 \tag{4.42}$$

Therefore

$$V > \frac{1}{2}\frac{\bar{\sigma}(\bar{S})\|r\|^2}{\underline{\sigma}^2(R)} \tag{4.43}$$

implies (4.39).

One can write the minimum singular values of R as

$$\underline{\sigma}\left(R\right) = \frac{\left(\frac{1}{2}c\underline{\sigma}\left(Q\right)+\kappa\right) - \sqrt{\left(\frac{1}{2}c\underline{\sigma}\left(Q\right)-\kappa\right) + \frac{1}{4}\varphi_M^2\bar{\sigma}^2\left(P\right)\bar{\sigma}^2\left(A\right)}}{2} \tag{4.44}$$

To obtain a cleaner form for $\underline{\sigma}(R)$, select $\kappa = \frac{1}{2}c\underline{\sigma}\left(Q\right)$. Then

$$\underline{\sigma}\left(R\right) = \frac{c\underline{\sigma}\left(Q\right) - \frac{1}{2}\varphi_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right)}{2} \tag{4.45}$$

which is positive under condition (4.28). Therefore, $z(t)$ is UUB [77].

In view of the fact that, for any vector z, one has $\|z\|_1 \geqslant \|z\|_2 \geqslant \cdots \geqslant \|z\|_\infty$, sufficient conditions for (4.39) are

$$\|e\| > \frac{B_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(L+B\right)+\kappa W_M}{\underline{\sigma}\left(R\right)} \tag{4.46}$$

or

$$\left\|\tilde{W}\right\| > \frac{B_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(L+B\right)+\kappa W_M}{\underline{\sigma}\left(R\right)} \tag{4.47}$$

Now Lemma 4.2.5 shows that the consensus errors $\delta(t)$ are UUB. Then $x_0(t)$ is cooperative UUB. Note that increasing gain c decreases the bound in 4.46.

Part II According to (4.38) $\dot{V} \leqslant -\underline{\sigma}(R)\|z\|^2 + \|r\|\,\|z\|$ and according to (4.42)

$$\dot{V} \leqslant -\alpha V + \beta\sqrt{V} \tag{4.48}$$

with $\alpha \equiv 2\underline{\sigma}(R)/\bar{\sigma}(\bar{S}),\ \ \beta \equiv \sqrt{2}\,\|r\|\,/\sqrt{\underline{\sigma}(\underline{S})}$. Thence

$$\sqrt{V(t)} \leqslant \sqrt{V(0)}e^{-\alpha t/2} + \tfrac{\beta}{\alpha}(1-e^{-\alpha t/2}) \leqslant \sqrt{V(0)} + \tfrac{\beta}{\alpha}$$

Using (4.42) one has

$$\|e(t)\| \leqslant \|z(t)\| \leqslant \sqrt{\frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}} \sqrt{\|e(0)\|^2 + \left\|\tilde{W}(0)\right\|_F^2} + \frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})} \frac{\|r\|}{\underline{\sigma}(R)}$$

Then (4.6) shows that

$$\|x(t)\| \leqslant \frac{1}{\underline{\sigma}(L+B)} \|e(t)\| + \sqrt{N} \|x_0(t)\|$$

$$\|x(t)\| \leqslant \frac{1}{\underline{\sigma}(L+B)} \left[\sqrt{\frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}} \sqrt{\|e(0)\|^2 + \left\|\tilde{W}(0)\right\|_F^2} + \frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})} \frac{\|r\|}{\underline{\sigma}(R)} \right] + \sqrt{N} X_0 \equiv r_0 \tag{4.49}$$

where $\|r\| \leqslant B_M \bar{\sigma}(P) \bar{\sigma}(L+B) + \kappa W_M$. Therefore, the state is contained for all times $t \geqslant 0$ in a compact set $\Omega_0 = \{x(t) | \|x(t)\| \leqslant r_0\}$. According to the Weierstrass approximation theorem, given any NN approximation error bound ε_M there exist numbers of neurons $\bar{\nu}_i, i = 1, N$ such that $\nu_i > \bar{\nu}_i, \ \forall i$ implies $\sup_{x \in \Omega} \|\varepsilon(x)\| < \varepsilon_M$. □

4.4 Discussion

If either (4.46) or (4.47) holds, the Lyapunov derivative is negative and V decreases. Therefore, these provide practical bounds for the neighborhood synchronization error and the NN weight estimation error. Part I of the proof contains the new material relevant to cooperative control of distributed systems. Part II of the proof is standard in the neural adaptive control literature, see e.g. [89, 78]. Note that the set defined by (4.49) depends on the initial errors and the graph structural properties. Therefore, proper accommodation of large initial errors requires a larger number of neurons in the NN. The proof also reveals that, for a given number of neurons, the admissible initial condition set is bounded [78] It is important to select the Lyapunov function candidate V in (4.29) in terms of locally available variables, e.g. the local neighborhood synchronization error $e(t)$ in (4.4) and (4.6). This means that any local control

signals $v_i(t)$ and NN tuning laws developed in the proof are distributed and hence implementable at each node. The use of the Frobenius norm in the Lyapunov function is also instrumental, since it gives rise to Frobenius inner products in the proof that only depend on trace terms, where only the diagonal terms are important. In fact, the Frobenius norm is ideally suited for the design of distributed protocols. Equation (4.23) shows that P in (4.29) is chosen diagonal. This is important in allowing selection of the NN tuning law (4.27) that leads to expression (4.36).

The result of this Lyapunov design is the local node NN tuning law (4.27) which depends on the local cooperative error $e_i(t)$. By contrast, standard NN tuning laws [78] depend on a global tracking error, equivalent to the disagreement error vector $\delta = \left(x - \underline{x}_0\right)$ which is not available at each node. Note that (4.27) is a distributed version of the sigma-mod tuning law of *P. Ioannou*. The parameters p_i in (4.27) are computed as in Lemma 4.3.5, which requires global information of the graph. As such, they are not known locally at the nodes. However, NN tuning parameters $f_i > 0$ are arbitrary, so that $p_i f_i > 0$ can be arbitrary.

Each node maintains a local NN to provide estimates of its nonlinear dynamics. The local nature of the NNs, as reflected in the distributed tuning protocols (4.27), means that the number of neurons ν_i at each node can be selected fairly small (see the Simulations). It is not necessary to select a large centralized NN to approximate the full global vector of nonlinearities $f(x)$ in (4.14). The total number of neurons in the network is $\nu = \sum_{i=1}^{N} \nu_i$.

The unknown dynamics of the control node $f(x_0, t)$ in (4.5) (which could be, e.g., motion) are treated as a disturbance to be rejected. The proof shows that even though these dynamics are unknown, synchronization of all nodes to the generally time-varying control node state $x_0(t)$ is guaranteed, within a small error. The simulations corroborate this.

Note that the node control gains c are similar to the pinning gain parameters defined in [49]. According to (4.45) and (4.46) increasing these gains results in a smaller synchronization error.

Remark 4.4.1. *If the node states are vectors $x_i \in R^n$ and $x_0 \in R^n$, the control protocols* (4.25) *and the error bounds given by* (4.46) *or* (4.47) *will remain unaltered. The NN weight tuning protocols will then be given by*

$$\dot{\hat{W}}_i = -F_i \varphi_i e_i^T \left(p_i(d_i + b_i) \otimes I_n\right) - \kappa F_i \hat{W}_i \tag{4.50}$$

Remark 4.4.2. *It is easy to extend the Theorem 4.3.6 to the case where the digraph only contains a spanning tree using the Frobenius form [53].*

Remark 4.4.3. *Given the setup in Theorem 4.3.6, suppose the NN estimation errors are equal to zero. Then under the protocol* (4.25) *the local neighborhood errors converge to the residual set defined by*

$$\|e\| \geqslant 2B_M \frac{\bar{\sigma}(P)\bar{\sigma}(L+B)}{c\,\underline{\sigma}(Q)} \tag{4.51}$$

with $B_M = d_M + F_M$ the sum of the disturbance bound and the bound on the consensus variable (control node) dynamics (Assumption 4.3.2) and P, Q defined in Lemma 4.3.5. The synchronization error norm $\|e(t)\|$ convergences to this set exponentially as $\|e(0)\|\, e^{-\alpha t/2}$ with convergence rate

$$\alpha = c\frac{\underline{\sigma}(Q)}{\bar{\sigma}(P)} \tag{4.52}$$

The speed of synchronization can be increased by increasing the control gains c in protocol (4.25).

Proof. If the NN estimation errors are zero, then according to the proof of Theorem 4.3.6 one has

$$V = \tfrac{1}{2} e^T P e$$

$$\dot{V} \leqslant -\tfrac{1}{2} c\,\underline{\sigma}(Q)\|e\|^2 + \|e\|\,\bar{\sigma}(P)\bar{\sigma}(L+P)B_M$$

which is negative if (4.51) holds. One has for large $\|e(t)\|$, approximately

$$V = \tfrac{1}{2} e^T P e$$

$$\dot{V} \leqslant -\tfrac{1}{2} c\, \underline{\sigma}(Q) \|e\|^2$$

So that using standard techniques one has

$$\dot{V} \leqslant -\alpha V$$

with α given by (4.52). Therefore $V(t) \leqslant V(0) e^{-\alpha t}$ and the errors converge at the rate $\|e(0)\|\, e^{-\alpha t/2}$. □

According to the Remark 4.4.3, the upper bounds on the residual local neighborhood errors depend graph properties through the inequalities of Lemma 4.4.4 involving $\bar{\sigma}(L+B)$.

4.4.1 Relation of Error Bounds to Graph Structural Properties

According to the bounds (4.46) and (4.47) both the synchronization error and the NN weight estimation error increase with the maximum singular values $\bar{\sigma}(A)$, $\bar{\sigma}(L+B)$, and $\bar{\sigma}(P)$. It is desired to obtain bounds on these singular values in terms of graph properties. Recall that $d_i = \sum_{j \in N_i} a_{ij}$ is the (weighted) in-degree of node i, that is, the ith row sum of adjacency matrix A, and $d_i^o = \sum_j a_{ji}$, the (weighted) out-degree of node i, that is the ith column sum of A. It is direct to obtain upper bounds on these singular values in terms of graph properties. Recalling several results from [55] one can easily show the following.

Lemma 4.4.4. ***Bounds on Maximum Singular Values of*** A ***and*** $(L+B)$

$$\bar{\sigma}(A) \leqslant \sum_{i=1}^{N} d_i \equiv vol(G)$$

$$\bar{\sigma}(A) \leqslant \sqrt{\max_i(d_i) \times \max_i(d_i^o)}$$

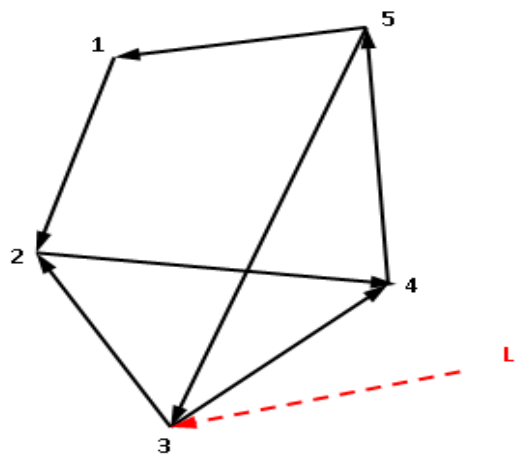

Figure 4.1. 5 node SC digraph with one leader node.

$$\bar{\sigma}(L+B) \leqslant \sum_{i-1}^{N} (b_i + d_i + d_i^o)$$

$$\bar{\sigma}(L+B) \leqslant \sqrt{\max_i (b_i + d_i + d_i^o) \times \max_i (b_i + 2d_i)}$$

These results show that the residual synchronization error is bounded above in terms of the graph complexity expressed in terms of volumes and maximum degree sums.

4.5 Simulation Results

This section will show the effectiveness of the distributed adaptive protocol of Theorem 4.3.6 on several fronts. First, it is compared to standard pinning control [49, 50]. Next, it is shown that the protocol effectively enforces synchronization, using only a few NN nodes at each node, for unknown nonlinear control node dynamics, unknown nonlinear node dynamics, and unknown disturbances at each node.

For this set of simulations, consider the 5-node strongly connected digraph structure in Fig. 4.1 with a leader node connected to node 3. The edge weights and the pinning gain in (4.4) were taken equal to 1.

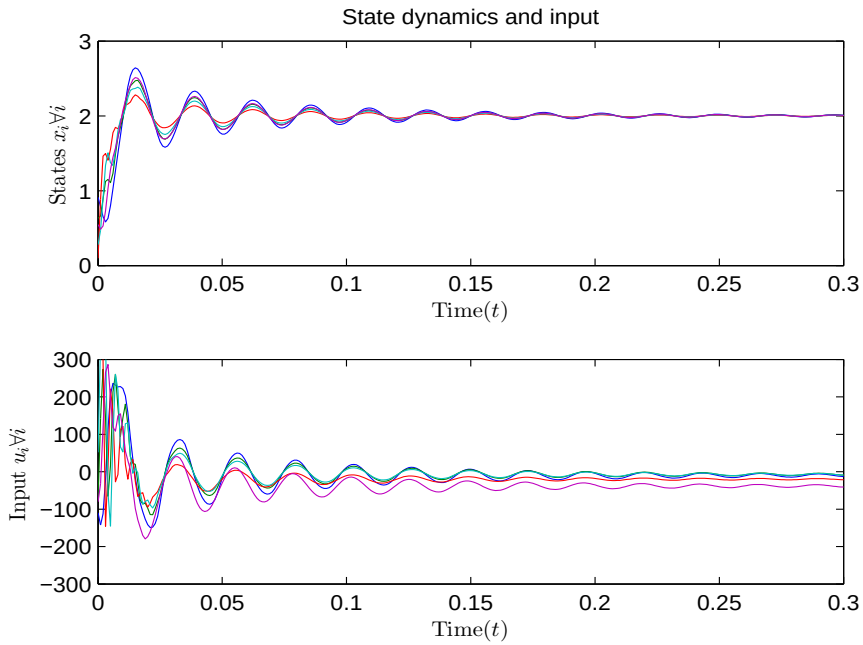

Figure 4.2. Consensus tracking and input for all nodes.

4.5.1 *Nonlinear node dynamics and disturbances*

For the graph structure shown above consider the following node dynamics

$$\begin{aligned}
\dot{x}_1 &= x_1^3 + u_1 + d_1 \\
\dot{x}_2 &= x_2^2 + u_2 + d_2 \\
\dot{x}_3 &= x_3^4 + u_3 + d_3 \\
\dot{x}_4 &= x_4 + u_4 + d_4 \\
\dot{x}_5 &= x_5^5 + u_5 + d_5
\end{aligned} \tag{4.53}$$

which has nonlinearities and disturbances at each node, all assumed unknown. The disturbance at i^{th} node is $w_i = 0.1 * \text{randn}\,(1) * \cos(t)$. (We use MATLAB symbology.) Consider now the control protocol of Theorem 4.3.6. Take the desired consensus value of $x_0 = 2$, i.e. $\dot{x}_0 = f(x_0, t) = 0$. The following parameters are used in the simulation. Control gain $c = 300$, number of neurons at each node: $\nu_i = 3$, $\kappa = 0.8$,

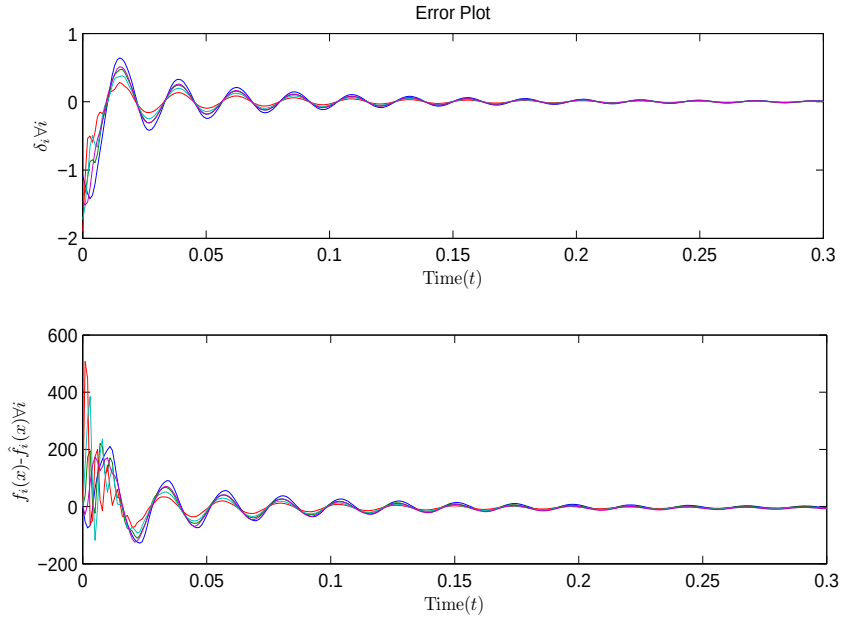

Figure 4.3. Consensus errors and NN weight estimation errors with distributed adaptive control.

$F_i = 1500$. Fig. 4.2 describes the tracking performance and control input. Fig. 4.3 plots the disagreement vector $\delta = \left(x - \underline{x}_0\right)$ and the NN estimation errors $f_i(x_i) - \hat{f}_i(x_i)$. The steady-state consensus error is approximately zero. Table 4.1 shows that at steady-state, the NNs closely estimate the nonlinearities. (That is, the nonlinearities in (4.53) evaluated at the consensus value of $x_0 = 2$. Recall there are small random disturbances $w_i(t)$ present.) The NN weight dynamics are given by Fig. 4.4.

Table 4.1. Comparison of $f(x)$ and $\hat{f}(x)$

$f(x)$	$\hat{f}(x)$
8	7.9835
16	15.8063
2	1.8051
32	31.5649

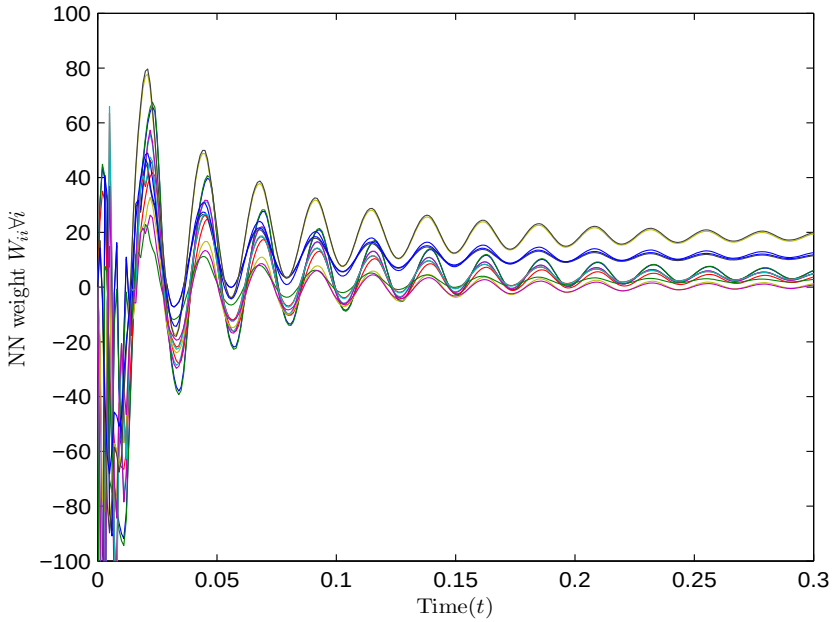

Figure 4.4. NN weight dynamics.

4.5.2 *Synchronization of Second-Order Dynamics*

Consider the node dynamics for node i given by the second-order dynamics

$$\begin{aligned} \dot{q}_{1_i} &= q_{2_i} + u_{1i} \\ \dot{q}_{2_i} &= J_i^{-1}\left[u_{2i} - B_i^r q_{2_i} - M_i g l_i \sin(q_{1_i})\right] \end{aligned} \tag{4.54}$$

where $q_i = [q_{1_i}, q_{2_i}]^T \in R^2$ is the state vector, J_i is the total inertia of the link and the motor, B_i^r is overall damping coefficient, M_i is total mass, g is gravitational acceleration and l_i is the distance from the joint axis to the link center of mass for node. J_i, B_i^r, M_i, g and l_i are considered unknown and may be different for each node. This is similar to the inertial agent dynamics $J_i\ddot{q}_i + B_i^r\dot{q}_i + M_i g l_i \sin(q_i) = u_i$, however, here we take an input into each state component, so it is not the same as second-order consensus. It is easy to treat second-order consensus using the method of this research and it will be done in another research.

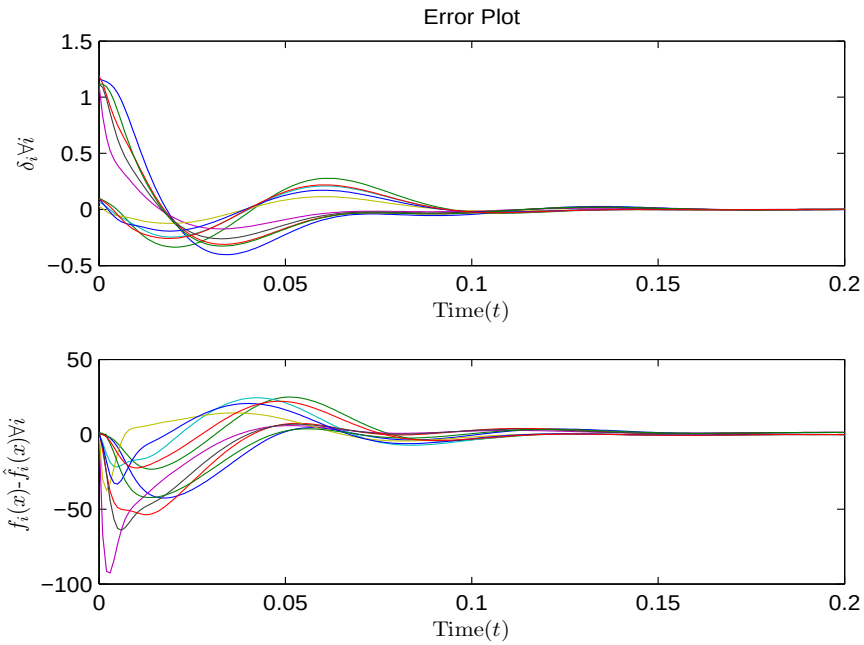

Figure 4.5. State synchronization errors δ_i and NN estimation errors.

The desired target node dynamics is taken as the inertial system

$$m_0\ddot{q}_0 + d_0\dot{q}_0 + k_o q_0 = u_0 \tag{4.55}$$

with known m_0, d_0, k_0. Select the feedback linearization input

$$u_0 = -\left[K_1\left(q_0 - \sin(\beta t)\right) + K_2\left(\dot{q}_0 - \beta\cos(\beta t)\right)\right] + d_0\dot{q}_0 + k_0 q_0 + \beta^2 m_0 \sin(\beta t) \tag{4.56}$$

for a constant $\beta > 0$. Then, the target motion $q_0(t)$ tracks the desired reference trajectory $\sin(\beta t)$.

The cooperative adaptive control law of Theorem 4.3.6 was simulated, including the Kronecker product with I_2 as in Remark 4.4.1. Fig. 4.5 verifies the fact that every node dynamics synchronizes to the target node dynamics. Fig. 4.6 is the phase plane plot for all the nodes along with the target node. It can be seen that the phase plane trajectories of all the nodes, started with different initial conditions, synchronize to the target node phase plane trajectory, finally forming a *lissajous* pattern.

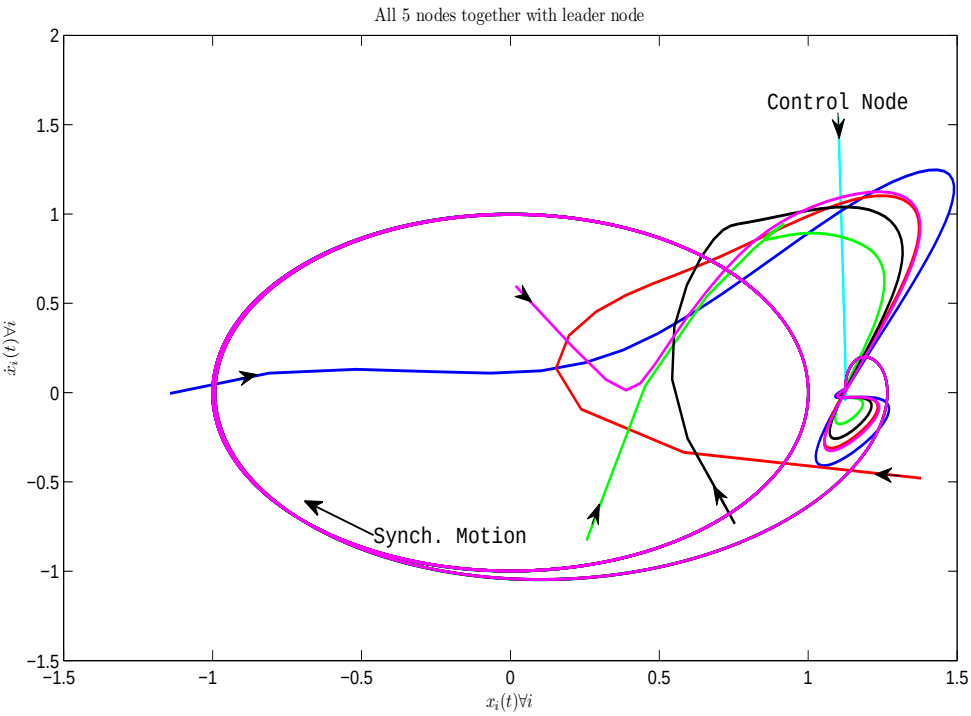

Figure 4.6. Synchronized motion phase plane plot.

4.6 Conclusion

This Chapter develops a design method for adaptive controllers for distributed systems having non-identical unknown nonlinear dynamics, and for a target dynamics to be tracked that is also nonlinear and unknown. A Lyapunov technique is presented for designing a synchronization control protocol. The proper selection of the Lyapunov function is the key to ensuring that the resulting control laws thus found are implementable in a distributed fashion. Lyapunov functions are defined in terms of a local neighborhood tracking synchronization error and the Frobenius norm. The resulting protocol consists of a linear protocol and a nonlinear control term with adaptive update law at each node. Connections are made between the convergence rate/convergence residual set and graph structural properties such as the sum of the in-degrees and out-

degrees, and the difference between the in-degrees and out-degrees. Singular value analysis is used. It is shown that the singular values of certain key matrices are intimately related to structural properties of the graph.

CHAPTER 5

DISTRIBUTED ADAPTIVE CONTROL FOR SECOND ORDER SYSTEMS WITH UNKNOWN NONLINEARITY

5.1 Introduction

Synchronization behavior among agents is found in flocking of birds, schooling of fish, and other natural systems. Synchronization among coupled oscillators was studied by [42]. Much work has extended consensus and synchronization techniques to manmade systems such as UAV to perform various tasks including surveillance, moving in formation, etc. Early work on cooperative decision and control for distributed systems includes [36]. The reader is referred to the book [38, 92, 93, 58]. Consensus has been studied for systems on communication graphs with fixed or varying topologies and communication delays. Consensus among agents having first-order integrator dynamics was studied by [37, 40, 38, 39] which proposed basic synchronizing protocols for communication topologies including balanced and strongly connected graphs.

Early work on consensus studied leaderless consensus or the cooperative regulator problem, where the consensus value reached depends on the initial conditions of the node states. On the other hand, the cooperative tracker problem seeks consensus or synchronization to the state of a control or leader node. Convergence of consensus to a virtual leader or header node was studied in [39, 46]. Dynamic consensus for tracking of time-varying signals was presented in [47]. The pinning control has been introduced for synchronization tracking control of coupled complex dynamical systems [48, 49, 50, 51]. Pinning control allows controlled synchronization of interconnected dynamical systems by adding a control or leader node that is connected (pinned) into a

small percentage of nodes in the network. Analysis has been done using Lyapunov and other techniques by assuming either a Jacobian linearization of the nonlinear node dynamics, or a Lipschitz condition, or contraction analysis. The agents are homogeneous in that they all have the same nonlinear dynamics.

The study of second-order and higher-order consensus is required to implement synchronization in most real world applications such as formation control and coordination among UAVs, where both position and velocity must be controlled. See the book by [92]. [58] studied the case of higher-order consensus for linear chained integrator systems. The detailed analysis there is performed for 3rd order systems but it extends to the higher order case. [87] studies the general second order consensus problem for double integrator systems. [56] studied second order consensus in finite time using sliding mode error and a Lyapunov analysis. Papers [59, 60, 61, 61] discuss consensus with time delays for second order integrator (Type II) systems. [62] proposed second order consensus using output feedback for agents with identical linear dynamics.

Few papers study second-order consensus for unknown nonlinear systems. Few papers study consensus for heterogeneous agents with unknown nonlinear non-identical dynamics. Leaderless synchronization with nonlinear non-identical passive systems is reported in [45], which provided a Lyapunov proof valid for balanced graph structures. Distributed multiagent systems with unknown nonlinear non-identical dynamics and disturbances were studied in [57] where distributed adaptive controllers were designed to achieve robust consensus. That treatment assumed undirected graphs and solved the leaderless consensus problem, that is, the nodes reach a steady-state consensus that depends on the initial conditions and cannot be controlled. Expressions for the consensus value were not given. Higher order consensus was proposed using a complex backstepping approach. [94] studied the cooperative tracker problem for agents on general digraphs having non-identical unknown dynamics and disturbances. A distributed

adaptive control technique was given that used pinning control to achieve synchronization to a desired command trajectory. Performance and stability were shown using a Lyapunov approach.

In this Chapter 5 we extend the idea proposed in Chapter 4 to consensus for agents having second order dynamics in Brunovsky form [75]. We confront the second order synchronization tracking problem for heterogeneous nodes with non-identical unknown nonlinear dynamics with unknown disturbances. The communication structures considered are general digraphs. Analysis of digraphs is significantly more involved than for undirected graphs. The dynamics of the leader or command node are also assumed nonlinear and unknown. A distributed adaptive control approach is taken, where cooperative adaptive controllers are designed at each node. A parametric neural network structure is introduced at each node to estimate the unknown dynamics. The choices of control protocol as well as neural net tuning laws are the key factors in stabilizing the networked multi-agent systems. A Lyapunov analysis shows how to tune the neural networks cooperatively, and guarantees the stability and performance of the networked systems. The error bounds obtained from the Lyapunov proof are dependent on control design and NN tuning parameters which can be chosen to suitably manage the tracking error and estimation error. Simulation results for networked agents with Lagrangian dynamics are provided to show the effectiveness of the proposed method. In Section 5.2 of the current Chapter is formulated the synchronization tracking control problem for second-order systems with non-identical unknown nonlinear dynamics. In Section 5.3 a Lyapunov technique is used to design cooperative adaptive controllers based on neural network approximation methods. Performance and stability guarantees are given for the networked systems. Section 5.4 presents simulation results.

5.2 Synchronization Control Formulation

Consider a graph $G = (V, E)$ with a nonempty finite set of N nodes $V = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $E \subseteq V \times V$. We assume the graph is simple, e.g. no repeated edges and $(v_i, v_i) \notin E$, $\forall i$ no self loops. General directed graphs are considered. Denote the connectivity matrix as $A = [a_{ij}]$ with $a_{ij} > 0 \; if \; (v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note $a_{ii} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D = [d_i]$ with $d_i = \sum\limits_{j \in N_i} a_{ij}$ the weighted in-degree of node i (i.e. i^{th} row sum of A). Define the graph Laplacian matrix as $L = D - A$, which has all row sums equal to zero. Define $d_i^o = \sum\limits_j a_{ji}$, the (weighted) out-degree of node i, that is the ith column sum of A.

We assume the communication digraph is strongly connected, i.e. there is a directed path from v_i to v_j for all distinct nodes $v_i, v_j \in V$. Then A and L are irreducible [53]. That is they are not cogredient to a lower triangular matrix, i.e., there is no permutation matrix U such that

$$L = U \begin{bmatrix} * & 0 \\ * & * \end{bmatrix} U^T \tag{5.1}$$

The results of this research can easily be extended to graphs having a spanning tree (i.e. not necessarily strongly connected) using the Frobenius form in (5.1)

5.2.1 *Cooperative Tracking Problem for Synchronization of Multiagent Systems*

Consider second order node dynamics defined for the ith node in *Brunovsky* form as

$$\begin{aligned} \dot{x}_i^1 &= x_i^2 \\ \dot{x}_i^2 &= f_i(x_i) + u_i + w_i \end{aligned} \tag{5.2}$$

where $x_i = [x_i^1 \;\; x_i^2]^T \in R^2$, $u_i(t) \in R$ is the control input and $w_i(t) \in R$ a disturbance acting upon each node. Note that each node may have its own distinct nonlinear dynamics. Standard assumptions for existence of unique solutions are made, e.g. $f_i(x_i)$ either continuously differentiable or *Lipschitz*. The overall graph dynamics is

$$\begin{aligned} \dot{x}^1 &= x^2 \\ \dot{x}^2 &= f(x) + u + w \end{aligned} \tag{5.3}$$

where the overall (global) state vector is $x^2 = \begin{bmatrix} x_1^2 & x_2^2 & \cdots & x_N^2 \end{bmatrix}^T \in R^N$, $x^1 = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \end{bmatrix}^T \in R^N$, $x = (x^1, x^2)^T$, $f(x) = \begin{bmatrix} f_1(x_1) & \cdots & f_N(x_N) \end{bmatrix}^T \in R^N$, input $u = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}^T \in R^N$, and $w = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^T \in R^N$.

If the states x_i^k are not scalars, this analysis carries over with the mere addition of the standard Kronecker product term [94].

Definition 5.2.1. *The local neighborhood tracking synchronization errors (position and velocity) for node i are defined as [56]*

$$e_i^1 = \sum_{j \in N_i} a_{ij} \left(x_j^1 - x_i^1\right) + b_i \left(y_0^1 - x_i^1\right) \tag{5.4}$$

and

$$e_i^2 = \sum_{j \in N_i} a_{ij} \left(x_j^2 - x_i^2\right) + b_i \left(y_0^2 - x_i^2\right) \tag{5.5}$$

with pinning gains $b_i \geqslant 0$, and $b_i > 0$ for at least one i. Then, $b_i \neq 0$ if and only if there exist an arc from the control node to the ith node in G. We refer to the nodes i for which $b_i \neq 0$ as the pinned or controlled nodes.

Note that (5.4) and (5.5) represents the information that is available to any node i for control purposes.

The state $x_0 = [x_0^1 \;\; x_0^2]^T \in R^2$ of the leader or control node satisfies the (generally nonautonomous) dynamics in Brunovsky form

$$\begin{aligned} \dot{x}_0^1 &= x_0^2 \\ \dot{x}_0^2 &= f_0(x_0, t) \end{aligned} \tag{5.6}$$

This can be regarded as a command or reference generator. A special case is the standard constant consensus value with $\dot{x}_0^1 = 0$ and x_0^2 absent. Here, we assume that the control node can have a time-varying state.

The *Synchronization tracking control problem* confronted herein is as follows: Design control protocols for all the nodes in G to synchronize to the state of the control node, i.e. one requires $x_i^k(t) \to x_0^k(t), \quad k = 1, 2, \forall i$. It is assumed that the dynamics of the control node is unknown to any of the nodes in G. It is assumed further that both the node nonlinearities $f_i(.)$ and the node disturbances $w_i(t)$ are unknown. Thus, the synchronization protocols must be robust to unmodelled dynamics and unknown disturbances.

5.2.2 Synchronization Tracking Error

Define the consensus disagreement error vector

$$\delta = \begin{bmatrix} \delta^1 & \delta^2 \end{bmatrix}^T = \begin{bmatrix} x^1 - \underline{1}x_0^1 & x^2 - \underline{1}x_0^2 \end{bmatrix}^T \tag{5.7}$$

From (5.4)-(5.5), the global error vector for network G is given by

$$e^1 = -\left(L + B\right)\left(x^1 - \underline{1}x_0^1\right) = -\left(L + B\right)\delta^1 \tag{5.8}$$

and

$$e^2 = -\left(L + B\right)\left(x^2 - \underline{1}x_0^2\right) = -\left(L + B\right)\delta^2 \tag{5.9}$$

where, $e^k = \begin{bmatrix} e_1^k & e_2^k & \cdots & e_N^k \end{bmatrix}^T \in R^N, k = 1, 2, \forall i$ and $\underline{1} \in R^N$ the vector of $1'$s.

Lemma 5.2.2. *Let the graph is strongly connected and $B \neq 0$. Then*

$$\left\|\delta^k\right\| \leqslant \left\|e^k\right\| / \underline{\sigma}\left(L+B\right),\ k=1,2 \tag{5.10}$$

with $\underline{\sigma}\left(L+B\right)$ the minimum singular value of $(L+B)$, and $e = 0$ if and only if the nodes synchronize, that is

$$x_i^k(t) = x_0^k(t),\ k=1,2,\ \forall i \tag{5.11}$$

5.2.3 Synchronization Control Design and Error Dynamics

Differentiating (5.8) and (5.9),

$$\dot{e}^1 = -\left(L+B\right)\left(x^2 - \underline{1}x_0^2\right) \tag{5.12}$$

and

$$\dot{e}^2 = -\left(L+B\right)\left(\dot{x}^2 - \underline{1}f_0(x_0,t)\right) \tag{5.13}$$

Note that $\dot{e}_1 = e_2$. Define sliding mode error for node i

$$r_i = e_i^2 + \lambda_i e_i^1 \tag{5.14}$$

or as a whole

$$r = e^2 + \Lambda e^1 \tag{5.15}$$

where $\Lambda = \text{diag}(\lambda^i) > 0$. The next result follows directly.

Lemma 5.2.3. *The velocity error is bounded according to*

$$\left\|e^2\right\| \leqslant \|r\| + \bar{\sigma}\left(\Lambda\right)\left\|e^1\right\| \tag{5.16}$$

Now differentiating r one obtains the error dynamics

$$\begin{aligned}\dot{r} &= -\left(L+B\right)\left(\dot{x}^2 - \underline{1}f_0(x_0,t)\right) - \Lambda\left(L+B\right)\left(x^2 - \underline{1}x_0^2\right)\\ &= -\left(L+B\right)\left(f\left(x\right)+u+w\right) + \left(L+B\right)\underline{1}f_0(x_0,t) + \Lambda e^2\end{aligned} \tag{5.17}$$

Following the techniques in [78, 89], assume that the unknown nonlinearities in (5.3) are smooth and thus can be approximated on a compact set $S \in R$ by

$$f_i\left(x_i\right) = W_i^T \varphi_i(x_i) + \varepsilon_i \tag{5.18}$$

with $\varphi_i(x_i) \in R^{\nu_i}$ a suitable basis set of ν_i functions at each node i and $W_i \in R^{\nu_i}$ a set of unknown coefficients. According to the Weierstrass higher-order approximation theorem [90], a polynomial basis set suffices to approximate $f_i(x_i)$ as well as its derivatives, when they exist, and moreover, the approximation error $\varepsilon_i \to 0$ uniformly as $\nu_i \to \infty$. According to the neural network (NN) approximation literature [86], a variety of basis sets can be selected, including sigmoids, gaussians, etc. There $\varphi_i(x_i) \in R^{\nu_i}$ is known as the NN activation function vector and $W_i \in R^{\nu_i}$ as the NN weight matrix. Then it is shown that ε_i is bounded on a compact set. The ideal approximating weights $W_i \in R^{\nu_i}$ in (5.18) are assumed unknown. The intention is to select only a small number ν_i of NN neurons at each node (see Simulations).

Here, to avoid distractions from the main issues being introduced, we assume a linear-in-the-parameters NN, i.e. the basis set of activation functions is fixed and only the output weights are tuned. It is straightforward to use a two-layer NN whereby the first and second-layer weights are tuned. Then one has a nonlinear-in-the-parameters NN and the basis set is automatically selected in the NN. Then, the below development can easily be modified as in [78].

To compensate for unknown nonlinearities, each node will maintain a neural network locally to keep track of the current estimates for the nonlinearities. The idea is to use the information of the states from the neighbors of node i to evaluate the performance of the current control protocol along with the current estimates of the nonlinear functions. Therefore, select the local node's approximation $\hat{f}_i(x_i)$ as

$$\hat{f}_i\left(x_i\right) = \hat{W}_i^T \varphi_i(x_i) \tag{5.19}$$

where $\hat{W}_i \in R^{\nu_i}$ is a current estimate of the NN weights for node i, and ν_i is the number of NN neurons maintained at each node i. It will be shown in Theorem 5.3.6 how to select the estimates of the parameters $\hat{W}_i \in R^{\nu_i}$ using the local neighborhood synchronization errors (4.4), (5.5). The global node nonlinearity $f(x)$ for graph G is now written as

$$f(x) = W^T\varphi(x) + \varepsilon = \begin{bmatrix} W_1^T & & & & \\ & W_2^T & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & W_N^T \end{bmatrix} \underbrace{\begin{bmatrix} \varphi_1(x_1) \\ \varphi_2(x_2) \\ \vdots \\ \vdots \\ \varphi_N(x_N) \end{bmatrix}}_{\varphi(x)} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \vdots \\ \varepsilon_N \end{bmatrix} \tag{5.20}$$

and the estimate $\hat{f}(x)$ as

$$\hat{f}(x) = \hat{W}^T\varphi(x) = \begin{bmatrix} \hat{W}_1^T & & & & \\ & \hat{W}_2^T & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \hat{W}_N^T \end{bmatrix} \underbrace{\begin{bmatrix} \varphi_1(x_1) \\ \varphi_2(x_2) \\ \vdots \\ \vdots \\ \varphi_N(x_N) \end{bmatrix}}_{\varphi(x)} \tag{5.21}$$

Consider an input

$$u_i = -\hat{f}_i(x_i) + \mu_i(x,t) \tag{5.22}$$

or

$$u = -\hat{f}(x) + \mu(x,t) \tag{5.23}$$

with $\mu_i(x,t)$ an auxiliary input for the ith node yet to be specified. Then using (5.23) the error dynamics (5.17) becomes

$$\dot{r} = -(L+B)\left(\tilde{f}(x) + \mu(x,t) + w\right) + (L+B)\,\underline{1} f_0(x_0,t) + \Lambda e^2 \tag{5.24}$$

where $\tilde{f}(x) = f(x) - \hat{f}(x) = \tilde{W}\varphi(x)$ with $\tilde{W} = \text{diag}\Big(W_1 - \hat{W}_1, \cdots\cdots, W_N - \hat{W}_N\Big)^T$.

5.3 Lyapunov Design for Cooperative Adaptive Tracking Control

It is now shown how to select the auxiliary control $\mu(t)$ and NN weight tuning laws such as to guarantee that all nodes synchronize to the desired control node signal, i.e., $x_i(t) \to x_0(t),\ \forall i$. It is assumed that the dynamics $f(x_0, t)$ of the control node (which could represent its motion) are unknown to any of the nodes in G. It is assumed further that the node nonlinearities $f_i(x_i)$ and disturbances $w_i(t)$ are unknown. The Lyapunov analysis technique approach of [78, 83] is used, though there are some complications arising from the fact that $\mu(x, t)$ and the NN weight tuning laws must be implemented as distributed protocols. This entails a careful selection of the Lyapunov function.

The maximum and minimum singular values of a matrix M are denoted $\bar{\sigma}(M)$ and $\underline{\sigma}(M)$ respectively. The Frobenius norm is $\|M\|_F = \sqrt{tr\{M^T M\}}$ with $tr\{\cdot\}$ the trace. The Frobenius inner product of two matrices is $\langle M_1, M_2\rangle_F = \sqrt{tr\{{M_1}^T M_2\}}$.

The following Fact gives two standard results used in neural adaptive control [78]

Fact 5.3.1. *Let the nonlinearities $f(x)$ in* (4.3) *be smooth on a compact set $\Omega \subset R^N$. Then:*

1. *The NN estimation error $\varepsilon(x)$ is bounded by $\|\varepsilon\| \leqslant \varepsilon_M$ on Ω, with ε_M a fixed bound [78, 86].*
2. ***Weierstrass higher-order approximation theorem**. Select the activation functions $\varphi(x)$ as a complete independent basis (e.g. polynomials). Then NN estimation error $\varepsilon(x)$ converges uniformly to zero on Ω as $\nu_i \to \infty,\ i = 1, N$. That is $\forall \xi > 0$ there exist $\bar{\nu}_i,\ i = 1, N$ such that $\nu_i > \bar{\nu}_i,\ \forall i$ implies $\sup_{x\in\Omega} \|\varepsilon(x)\| < \xi$ [90]*

The following standard assumptions are required. Although the bounds mentioned are assumed to exist, they are not used in the design and do not have to be

known. They appear in the error bounds in the proof of Theorem 5.3.6. (Though not required, if desired, standard methods can be used to estimate these bounds including [10].)

Assumption 5.3.2. *1. The unknown disturbance w_i is bounded for all i. Thus the overall disturbance vector w is also bounded by $\|w\| \leqslant w_M$ with w_M a fixed bound.*

2. Unknown ideal NN weight matrix W is bounded by $\|W\|_F \leqslant W_M$.

3. NN activation functions φ_i are bounded $\forall i$, so that one can write for the overall network that $\|\varphi\| \leqslant \phi_M$.

4. The unknown consensus variable dynamics $f_0(x_0, t)$ as well as the target output vector are bounded so that $\|f_0(x_0, t)\| \leqslant F_M, \forall t$ respectively.

5. The target trajectory is bounded so that $\|x_0^1(t)\| < X_0^1,\ \|x_0^2(t)\| < X_0^2,\ \forall t$.

Definition 5.3.3. *The time vector function $y(t)$ is said to be uniformly ultimately bounded (UUB) [78] if there exist a compact set $\Omega \subset R^N$ so that $\forall y(t_0) \in \Omega$ there exist a bound B_m and a time $t_f(B_m, y(t_0))$ such that $\|x(t) - y(t)\| \leqslant B_m\ \forall t \geqslant t_0 + t_f$.*

Definition 5.3.4. *The time vector function $x_0(t)$ is said to be cooperative uniformly ultimately bounded (CUUB) if there exist a compact set $\Omega \subset R$ so that $\forall\ (x_i(t_0) - x_0(t_0) \in \Omega$ there exist a bound B_m and a time $t_f\ (B_m, x(t_0), x_0(t_0))$ such that $\|x_i(t) - x_0(t)\| \leqslant B\ \forall i,\ \forall t \geqslant t_0 + t_f$.*

The next key constructive result is needed. An M-matrix is a square matrix having nonpositive off-diagonal elements and all principal minors nonnegative [53].

Lemma 5.3.5. *Let L be irreducible and B have at least one diagonal entry $b_i > 0$. Then $(L + B)$ is a nonsingular $M-$matrix. Define*

$$q = \begin{bmatrix} q_1 & q_2 & \cdots & q_N \end{bmatrix}^T = (L + B)^{-1}\underline{1} \tag{5.25}$$

$$P = diag\,\{p_i\} \equiv diag\,\{1/q_i\} \tag{5.26}$$

Then $P > 0$ and the matrix Q defined as

$$Q = P\left(L + B\right) + \left(L + B\right)^T P \tag{5.27}$$

is positive definite.

It is important that the matrix P is diagonal, as will be seen in the upcoming proof. The main result of the Chapter is now presented.

Theorem 5.3.6. ***Distributed Adaptive Control Protocol for Synchronization***
Consider the networked systems given by (5.3) *under the Assumption 5.3.2. Let the communication digraph be strongly connected. Select the auxiliary control signal $\mu\left(x, t\right)$ in* (5.23) *so that the local node control protocols are given by*

$$u_i = cr_i - \hat{f}_i(x_i) + \frac{\lambda_i}{d_i + b_i} e_i^2 \tag{5.28}$$

with $\lambda_i = \lambda > 0\ \forall i$, control gains $c > 0$, and $r_i(t)$ defined in (5.14)*. Then*

$$u = cr - \hat{W}^T \varphi(x) + \lambda (D + B)^{-1} e^2 \tag{5.29}$$

with Let local node NN tuning laws be given by

$$\dot{\hat{W}}_i = -F_i \varphi_i r_i^T p_i (d_i + b_i) - \kappa F_i \hat{W}_i \tag{5.30}$$

with $F_i = \Pi_i I_{\nu_i}$, I_{ν_i} the $\nu_i \times \nu_i$ identity matrix, $\Pi_i > 0$ and $\kappa > 0$ scalar tuning gains, and $p_i > 0$ defined in Lemma 5.3.5. Define

$$\lambda = \sqrt{\frac{\underline{\sigma}\left(D + B\right)}{\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right)}} \tag{5.31}$$

and select the control gain c and NN tuning gain κ so that

$$\begin{aligned} &i. \quad c = \frac{2}{\underline{\sigma}\left(Q\right)}\left(\frac{1}{\sqrt{\lambda}} + \lambda\right) > 0 \\ &ii. \quad \frac{1}{2}\varphi_m \bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right) \leqslant \kappa \leqslant \lambda - 1 \end{aligned} \tag{5.32}$$

with $P > 0, Q > 0$ define in Lemma 5.3.5 and A the graph adjacency matrix. Then there exist numbers of neurons $\bar{\nu}_i, i = 1, N$ such that for $\nu_i > \bar{\nu}_i,\ \forall i$ the overall sliding mode cooperative error vector $r(t)$, the local cooperative error vectors $e^1(t), e^2(t)$ and the NN weight estimation errors $\tilde{W}$ are UUB, with practical bounds given by (5.53)-(5.55) *respectively. Moreover the consensus variable $x_0(t) = [x_0^1, x_0^2]^T$ is cooperative UUB and all nodes synchronize such that $\|x_i^1(t) - x_0^1(t)\| \to 0$, $\|x_i^2(t) - x_0^2(t)\| \to 0$. Moreover, the bounds* (5.53)-(5.55) *can be made small by manipulating the NN and control gain parameters.*

Proof. **Part I:**

We claim that for a fixed $\varepsilon_M > 0$, there exist numbers of neurons $\bar{\nu}_i, i = 1, N$ such that for $\nu_i > \bar{\nu}_i,\ \forall i$ the NN approximation error is bounded by $\|\varepsilon\| \leqslant \varepsilon_M$. The claim is proven in Part II of the proof. Consider now the Lyapunov function candidate

$$V = \frac{1}{2} r^T P r + \frac{1}{2} \tilde{W}^T F^{-1} \tilde{W} + \frac{1}{2} \left(e^1\right)^T e^1 \tag{5.33}$$

with $P = P^T > 0$ and $F^{-1} = F^{-T} > 0$. Then

$$\dot{V} = r^T P \dot{r} + tr\left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\} + \left(e^1\right)^T \dot{e}^1 \tag{5.34}$$

Using (5.24) and (5.29)

$$\dot{r} = -\left(L + B\right)\left(\tilde{f}\left(x\right) + cr + w\right) + \left(L + B\right) \underline{1} f_0(x_0, t) + A(D + B)^{-1} \Lambda e^2 \tag{5.35}$$

Therefore

$$\begin{aligned} \dot{V} = -r^T P \left(L + B\right)\left(\tilde{W}^T \varphi\left(x\right) + \varepsilon + w + cr\right) + r^T P \left(L + B\right) \underline{1} f_0(x_0, t) + \\ r^T P A (D + B)^{-1} \Lambda e^2 + tr\left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\} + \left(e^1\right)^T e^2 \end{aligned} \tag{5.36}$$

$$\begin{aligned} \dot{V} = -r^T P \left(L + B\right) cr - r^T P \left(L + B\right) \tilde{W}^T \varphi\left(x\right) - r^T P \left(L + B\right)\left(\varepsilon + w\right) + \\ r^T P \left(L + B\right) \underline{1} f_0(x_0, t) + r^T P A (D + B)^{-1} \Lambda e^2 + tr\left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\} + \left(e^1\right)^T \left(r - \Lambda e^1\right) \end{aligned} \tag{5.37}$$

$$\dot{V} = -cr^T P\left(L+B\right) r - r^T P\left(L+B\right)\left\{\varepsilon + w - \underline{1} f_0(x_0,t)\right\} +$$
$$r^T PA(D+B)^{-1}\Lambda e^2 + tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi\left(x\right) r^T P\left(L+B\right)\right)\right] + \left(e^1\right)^T r - \left(e^1\right)^T \Lambda e^1 \tag{5.38}$$

$$\begin{aligned}\dot{V} &= -cr^T P\left(L+B\right) r - r^T P\left(L+B\right)\left\{\varepsilon + w - \underline{1} f_0(x_0,t)\right\} + \\ & r^T PA(D+B)^{-1}\Lambda\left(r - \Lambda e^1\right) + tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi\left(x\right) r^T P\left(L+B\right)\right)\right] + \\ & \left(e^1\right)^T r - \left(e^1\right)^T \Lambda e^1\end{aligned} \tag{5.39}$$

$$\begin{aligned}\dot{V} &= -cr^T P\left(L+B\right) r - r^T P\left(L+B\right)\left\{\varepsilon + w - \underline{1} f_0(x_0,t)\right\} + \\ & tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi\left(x\right) r^T P\left(D+B\right)\right)\right] + tr\left[\tilde{W}^T\varphi\left(x\right) r^T PA\right] + \\ & r^T PA(D+B)^{-1}\Lambda r - r^T PA(D+B)^{-1}\Lambda^2 e^1 + \left(e^1\right)^T r - \left(e^1\right)^T \Lambda e^1\end{aligned} \tag{5.40}$$

Since L is irreducible and B has at least one diagonal entry $b_i > 0$, then $(L+B)$ is a nonsingular $M-$matrix. Thus, according to Lemma 5.3.5, one can write

$$\begin{aligned}\dot{V} &= -\frac{1}{2}cr^T Qr - r^T P\left(L+B\right)\left\{\varepsilon + w - \underline{1} f_0(x_0,t)\right\} + \\ & tr\left[\tilde{W}^T\left(F^{-1}\dot{\tilde{W}} - \varphi\left(x\right) r^T P\left(D+B\right)\right)\right] + tr\left[\tilde{W}^T\varphi\left(x\right) r^T PA\right] + \\ & r^T PA(D+B)^{-1}\Lambda r - r^T PA(D+B)^{-1}\Lambda^2 e^1 + \left(e^1\right)^T r - \left(e^1\right)^T \Lambda e^1\end{aligned} \tag{5.41}$$

Adopt the NN weight tuning law $\dot{\tilde{W}}_i = F_i\varphi_i r_i^T p_i(d_i + b_i) + \kappa F_i \hat{W}_i$. Taking norm both sides in (5.41) one has

$$\begin{aligned}\dot{V} \leqslant & -\frac{1}{2}c\underline{\sigma}\left(Q\right)\left\|r\right\|^2 + \bar{\sigma}\left(P\right)\bar{\sigma}\left(L+B\right) B_M\left\|r\right\| + \kappa W_M\left\|\tilde{W}\right\|_F - \kappa\left\|\tilde{W}\right\|_F^2 + \\ & \phi_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right)\left\|\tilde{W}\right\|_F\left\|r\right\| + \frac{\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right)\bar{\sigma}\left(\Lambda\right)}{\underline{\sigma}\left(D+B\right)}\left\|r\right\|^2 + \\ & \left(1 + \frac{\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right)\bar{\sigma}\left(\Lambda^2\right)}{\underline{\sigma}\left(D+B\right)}\right)\left\|r\right\|\left\|e^1\right\| - \underline{\sigma}\left(\Lambda\right)\left\|e^1\right\|^2\end{aligned} \tag{5.42}$$

where $B_M = (\varepsilon_M + w_M + F_M)$. Then

$$\dot{V} \leqslant -z^T \begin{bmatrix} \underline{\sigma}\left(\Lambda\right) & \frac{1}{2}\left(1 + \frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}\left(\Lambda^2\right)}{\underline{\sigma}(D+B)}\right) & 0 \\ \frac{1}{2}\left(1 + \frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}\left(\Lambda^2\right)}{\underline{\sigma}(D+B)}\right) & \left(\frac{1}{2}c\underline{\sigma}\left(Q\right) - \frac{\bar{\sigma}(P)\bar{\sigma}(A)\bar{\sigma}(\Lambda)}{\underline{\sigma}(D+B)}\right) & \frac{1}{2}\phi_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right) \\ 0 & \frac{1}{2}\phi_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right) & \kappa \end{bmatrix} z + \begin{bmatrix} 0 & \bar{\sigma}\left(P\right)\bar{\sigma}\left(L+B\right)B_M & \kappa W_M \end{bmatrix} z \tag{5.43}$$

where $z = \begin{bmatrix} \|e^1\| & \|r\| & \left\|\tilde{W}\right\|_F \end{bmatrix}^T$.

Now write this as

$$\dot{V} \leqslant -z^T H z + h^T z \tag{5.44}$$

Clearly $\dot{V} \leqslant 0$ iff $H \geqslant 0$ and

$$\|z\| > \frac{\|h\|}{\underline{\sigma}(H)} \tag{5.45}$$

According to (5.33) this defines a level set of $V(z)$, so it is direct to show that $\dot{V} \leqslant 0$ for V large enough such that (5.45) holds [77]. To show this, according to (5.33) one has

$$\tfrac{1}{2}\underline{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{\max}}\left\|\tilde{W}\right\|_F^2 + \frac{1}{2}\left\|e^1\right\|^2 \leqslant V \leqslant \tfrac{1}{2}\bar{\sigma}(P)\|e\|^2 + \frac{1}{2\Pi_{\min}}\left\|\tilde{W}\right\|_F^2 + \frac{1}{2}\left\|e^1\right\|^2 \tag{5.46}$$

$$\frac{1}{2}\begin{bmatrix} \|e^1\| & \|r\| & \left\|\tilde{W}\right\|_F \end{bmatrix} \underbrace{\begin{bmatrix} \underline{\sigma}\left(P\right) & & \\ & \frac{1}{\Pi_{\max}} & \\ & & 1 \end{bmatrix}}_{S_1} \begin{bmatrix} \|e^1\| \\ \|r\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix} \leqslant V \leqslant \tag{5.47}$$

$$\frac{1}{2}\begin{bmatrix} \|e^1\| & \|r\| & \left\|\tilde{W}\right\|_F \end{bmatrix} \underbrace{\begin{bmatrix} \bar{\sigma}\left(P\right) & & \\ & \frac{1}{\Pi_{\min}} & \\ & & 1 \end{bmatrix}}_{S_2} \begin{bmatrix} \|e^1\| \\ \|r\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix}$$

with $\Pi_{\min}, \Pi_{\max}$ the minimum and maximum values of Π_i. Equation (5.47) is equivalent to

$$\tfrac{1}{2} z^T \underline{S} z \leqslant V \leqslant \tfrac{1}{2} z^T \bar{S} z \tag{5.48}$$

where $\underline{S} = \underline{\sigma}\left(S_1\right)$ and $\bar{S} = \bar{\sigma}\left(S_2\right)$. Then

$$\tfrac{1}{2} \underline{\sigma}(\underline{S}) \|z\|^2 \leqslant V \leqslant \tfrac{1}{2} \bar{\sigma}(\bar{S}) \|z\|^2 \tag{5.49}$$

Therefore,

$$V > \frac{1}{2} \frac{\bar{\sigma}(\bar{S}) \|h\|^2}{\underline{\sigma}^2(H)} \tag{5.50}$$

implies (5.45).

For a symmetric positive definite matrix, the singular values will be equal to its eigenvalues. Define $\Lambda = \lambda I, \lambda = \sqrt{\frac{\sigma(D+B)}{\bar{\sigma}(P)\bar{\sigma}(A)}}$, $c = \frac{2}{\sigma(Q)}\left(\frac{1}{\sqrt{\lambda}} + \lambda\right)$ and $\gamma = \frac{1}{2}\varphi_m \bar{\sigma}(P)$ $\bar{\sigma}(A)$. Then (5.43) can be written as

$$\begin{aligned} \dot{V} \leqslant - \begin{bmatrix} \|e^1\| & \|r\| & \left\|\tilde{W}\right\|_F \end{bmatrix} \underbrace{\begin{bmatrix} \lambda & 1 & 0 \\ 1 & \lambda & \gamma \\ 0 & \gamma & \kappa \end{bmatrix}}_{H} \begin{bmatrix} \|e^1\| \\ \|r\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix} + \\ \begin{bmatrix} 0 & \bar{\sigma}\left(P\right) \bar{\sigma}\left(L+B\right) B_M & \kappa W_M \end{bmatrix} \begin{bmatrix} \|e^1\| \\ \|r\| \\ \left\|\tilde{W}\right\|_F \end{bmatrix} \end{aligned} \tag{5.51}$$

The transformed H matrix is symmetric and positive definite under assumption given by (5.32). Therefore from *Gershgorin* circle's theorem

$$\underline{\sigma}\left(H\right) \geqslant \kappa - \gamma \tag{5.52}$$

with $0 < \gamma \leqslant \kappa \leqslant \lambda - 1$. Therefore $z(t)$ is UUB [77]. In view of the fact that, for any vector z, one has $\|z\|_1 \geqslant \|z\|_2 \geqslant \cdots \geqslant \|z\|_\infty$, sufficient conditions for (5.45) are:

$$\|r\| > \frac{B_M \bar{\sigma}\left(P\right) \bar{\sigma}\left(L+B\right) + \kappa W_M}{\kappa - \frac{1}{2}\varphi_m \bar{\sigma}\left(P\right) \bar{\sigma}\left(A\right)} \tag{5.53}$$

or

$$\left\|e^1\right\| > \frac{B_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(L+B\right)+\kappa W_M}{\kappa-\frac{1}{2}\varphi_m\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right)} \tag{5.54}$$

or

$$\left\|\tilde{W}\right\| > \frac{B_M\bar{\sigma}\left(P\right)\bar{\sigma}\left(L+B\right)+\kappa W_M}{\kappa-\frac{1}{2}\varphi_m\bar{\sigma}\left(P\right)\bar{\sigma}\left(A\right)} \tag{5.55}$$

Note that this shows UUB of $r(t), e_1(t), \tilde{W}(t)$. Therefore from Lemma 5.2.3, the boundedness of r and e^1 implies bounded e^2. Now Lemma 5.2.2 shows that the consensus error vector $\delta(t)$ is UUB. Then $x_0(t)$ is cooperative UUB.

Part II: See [89]

According to (5.44) $\dot{V} \leqslant -\underline{\sigma}(H)\|z\|^2 + \|h\|\,\|z\|$ and according to (5.49)

$$\dot{V} \leqslant -\alpha V + \beta\sqrt{V} \tag{5.56}$$

with $\alpha \equiv 2\underline{\sigma}(H)/\bar{\sigma}(\bar{S}), \quad \beta \equiv \sqrt{2}\,\|h\|\,/\sqrt{\underline{\sigma}(\underline{S})}$. Thence

$$\sqrt{V(t)} \leqslant \sqrt{V(0)}e^{-\alpha t/2} + \tfrac{\beta}{\alpha}(1-e^{-\alpha t/2}) \leqslant \sqrt{V(0)} + \tfrac{\beta}{\alpha} \tag{5.57}$$

Using (5.49) one has

$$\left\|e^1(t)\right\| \leqslant \|z(t)\| \leqslant \sqrt{\frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}}\sqrt{\|e(0)\|^2 + \left\|\tilde{W}(0)\right\|_F^2 + \|e^1(0)\|^2} + \frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}\frac{\|h\|}{\underline{\sigma}(H)} \equiv \rho \tag{5.58}$$

and

$$\|r(t)\| \leqslant \|z(t)\| \leqslant \sqrt{\frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}}\sqrt{\|e(0)\|^2 + \left\|\tilde{W}(0)\right\|_F^2 + \|e^1(0)\|^2} + \frac{\bar{\sigma}(\bar{S})}{\underline{\sigma}(\underline{S})}\frac{\|h\|}{\underline{\sigma}(H)} \equiv \rho \tag{5.59}$$

Then from (5.8)

$$\left\|x^1(t)\right\| \leqslant \frac{1}{\underline{\sigma}(L+B)}\left\|e^1(t)\right\| + \sqrt{N}\left\|x_0^1(t)\right\| \tag{5.60}$$

$$\left\|x^1(t)\right\| \leqslant \frac{\rho}{\underline{\sigma}(L+B)} + \sqrt{N}X_0^1 \equiv h_0^1 \tag{5.61}$$

Similarly from (5.59) and using (5.16) in Lemma 5.2.3

$$\begin{aligned} \left\|e^2(t)\right\| &\leqslant \lambda \left\|e^1(t)\right\| + \|r(t)\| \\ &\leqslant \rho\,(1+\lambda) \end{aligned} \tag{5.62}$$

This implies

$$\left\|x^2(t)\right\| \leqslant \frac{\rho\,(1+\lambda)}{\underline{\sigma}(L+B)} + \sqrt{N}X_0^2 \equiv h_0^2 \tag{5.63}$$

where $X_0^2 = \|x_0^2(t)\|$ and $\|h\| \leqslant B_M\bar{\sigma}\,(P)\,\bar{\sigma}\,(L+B) + \kappa W_M$. Therefore, the state is contained for all times $t \geqslant 0$ in a compact set $\Omega_0 = \{x(t)|\,\|x^1(t)\| < h_0^1, \|x^2(t)\| < h_0^2\}$. According to the Weierstrass approximation theorem (Fact 5.3.1), given any NN approximation error bound ε_M there exist numbers of neurons $\bar{\nu}_i, i = 1, N$ such that $\nu_i > \bar{\nu}_i,\ \forall i$ implies $\sup_{x\in\Omega}\|\varepsilon(x)\| < \varepsilon_M$. □

5.3.1 Discussion

If any one of (5.53), (5.54) or (5.55) holds, the Lyapunov derivative is negative and V decreases. Therefore, these provide practical bounds for the neighborhood synchronization error and the NN weight estimation error.

It is important to select the Lyapunov function candidate V in (5.33) as a function of locally available variables, e.g. the local sliding mode error $r(t)$ and cooperative neighborhood error $e_1(t)$ in (5.15) and (5.8) respectively. This means that any local control signals $\mu_i(x,t)$ and NN tuning laws developed in the proof are distributed and hence implementable at each node. The use of the Frobenius norm in the Lyapunov function is also instrumental, since it gives rise to Frobenius inner products in the proof that only depend on trace terms, where only the diagonal terms are important. In fact, the Frobenius norm is ideally suited for the design of distributed protocols. Finally, it is important that the matrix P of Lemma 5.3.5 is diagonal.

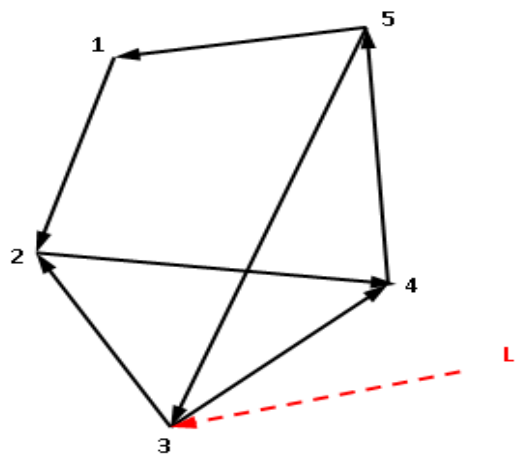

Figure 5.1. Five-node SC digraph with one leader node.

5.4 Simulation Result

For this set of simulations, consider the 5-node strongly connected digraph structure in Fig. 5.1 with a leader node connected to node 3. The edge weights and the pinning gain in (5.4) were taken equal to 1.

Consider the node dynamics for node i given by the second-order Lagrange form dynamics

$$\begin{aligned} \dot{q}_{1_i} &= q_{2_i} \\ \dot{q}_{2_i} &= J_i^{-1}\left[u_i - B_i^r q_{2_i} - M_i g l_i \sin(q_{1_i})\right] \end{aligned} \tag{5.64}$$

where $q_i = [q_{1_i}, q_{2_i}]^T \in R^2$ is the state vector, J_i is the total inertia of the link and the motor, B_i^r is overall damping coefficient, M_i is total mass, g is gravitational acceleration and l_i is the distance from the joint axis to the link center of mass for node. J_i, B_i^r, M_i, g and l_i are considered unknown and may be different for each node. The desired target node dynamics is taken as the inertial system

$$m_0\ddot{q}_0 + d_0\dot{q}_0 + k_o q_0 = u_0 \tag{5.65}$$

with known m_0, d_0, k_0. Select the feedback linearization input

$$u_0 = -\left[K_1\left(q_0 - \sin(\beta t)\right) + K_2\left(\dot{q}_0 - \beta\cos(\beta t)\right)\right] + d_0\dot{q}_0 + k_0 q_0 + \beta^2 m_0 \sin(\beta t) \quad (5.66)$$

for a constant $\beta > 0$. Then, the target motion $q_0(t)$ tracks the desired reference trajectory $\sin(\beta t)$.

The cooperative adaptive control protocols of Theorem 5.3.6 were implemented at each node. The positive definite matrix P can be chosen from $P^{-1} = (L+B)^{-1}\underline{1}$. But from in (5.30) Theorem 5.3.6, due to the presence of the arbitrary diagonal gain matrix F_i, we can choose P arbitrary for the simulation purpose without loss of generality. Sliding mode gain λ is chosen from (5.31). For this particular example we one can choose $\bar{\sigma}(P) = 0.05$. From Fig. 5.1, $\bar{\sigma}(A) = 1.73$ and $\underline{\sigma}(D+B) = 1$. Therefore $\lambda \approx 3.4$. Again, for NN activation function we use log-sigmoid of the form $\frac{1}{1+e^{-kt}}$ for positive slope parameter k. Number of neurons used at each node is 3. So, $\varphi_m \approx 1$ and $\gamma \approx 0.04$. NN gain parameter $\kappa = 1.5$ from (5.32). The pinning gain is calculated from (5.32) as 1000.

In the simulation plots we show tracking performance of positions q_{1_i} and velocities q_{2_i} for all i. At steady state q_{1_i}'s and q_{2_i}'s are synchronized following the second order single link leader trajectory given by q_0 and $\dot{q}_0$ respectively.

Fig. 5.2 shows the tracking performance of the system. One can see from the figure that positions and velocities of all the five nodes are synchronized in two different final values which are given by the final values of $[q_0, \dot{q}_0]^T$. More specifically q_{1_i}'s are synchronized with q_0 and q_{2_i}'s are synchronized with $\dot{q}_0$ at steady state. The figure also shows the inputs u_{2_i} for all agents while tracking.

Fig. 5.3 describes the position and velocity error vectors namely $(q_{1_i} - q_0, q_{2_i} - \dot{q}_0)^T$ for all i and also the NN estimation error in terms of $\left[f_i(x) - \hat{f}_i(x)\right]^T \forall i$. One can easily see that all the errors are minimized almost to zero.

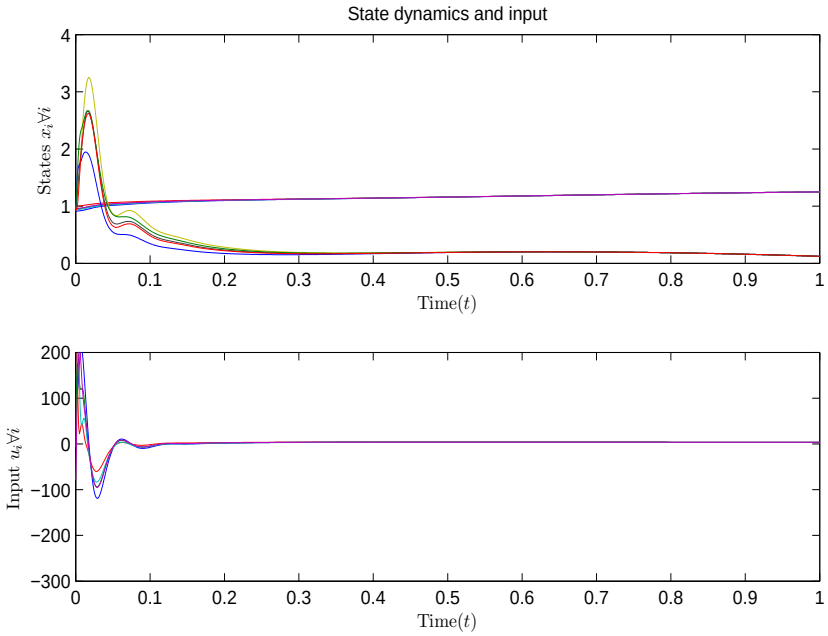

Figure 5.2. Tracking performance (position and velocity) and control input.

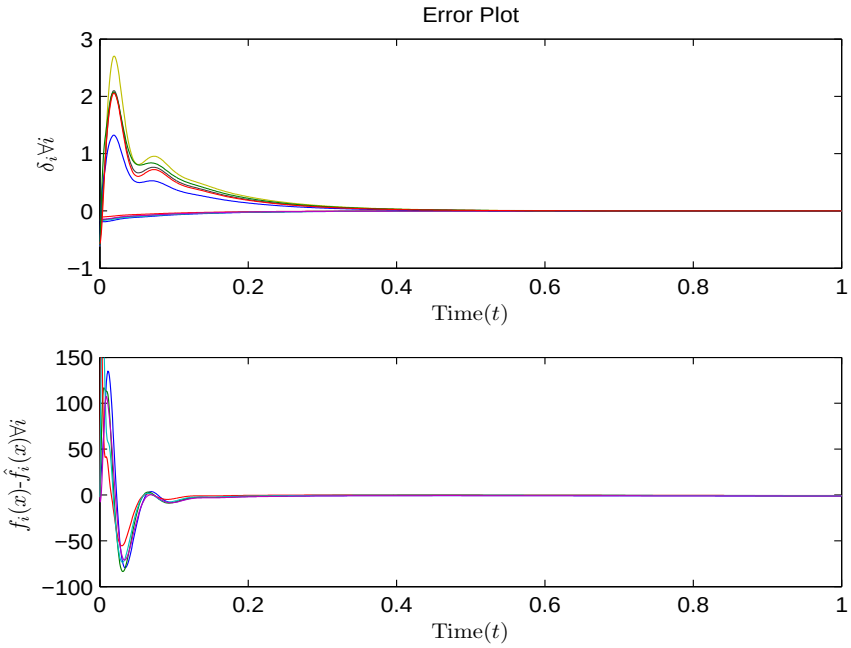

Figure 5.3. Disagreement vector and NN estimation error vector.

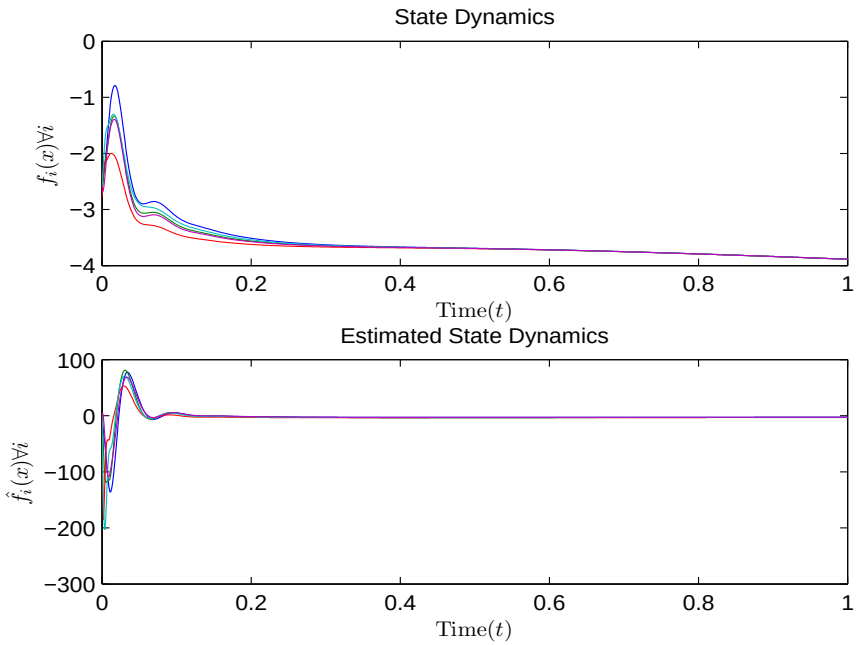

Figure 5.4. Actual and estimated $f(x)$.

Fig. 5.4 is the comparison plot for $f(x)$ and $\hat{f}(x)$ to elaborate the plot obtained in Fig. 5.3. The initial mismatch in the value is due to the initial NN weight fluctuation shown in Fig. 5.5. Fig. 5.6 is the phase plane plot of the whole system i.e. the plot of $q_{1_i}\forall i$ (along the x-axis) and $q_{2_i}\forall i$ (along the $y-$axis). At steady state the *lissajous* pattern is formed by five nodes.

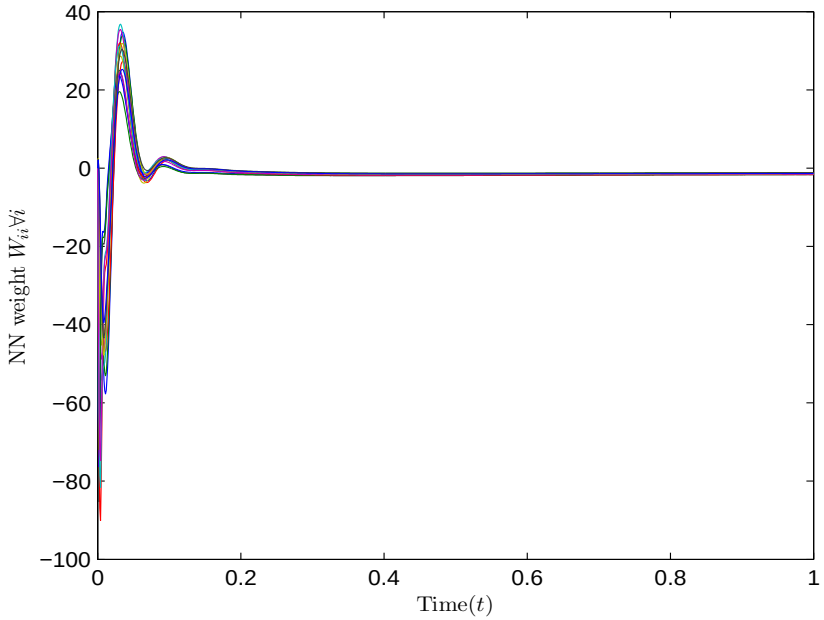

Figure 5.5. NN weight dynamics.

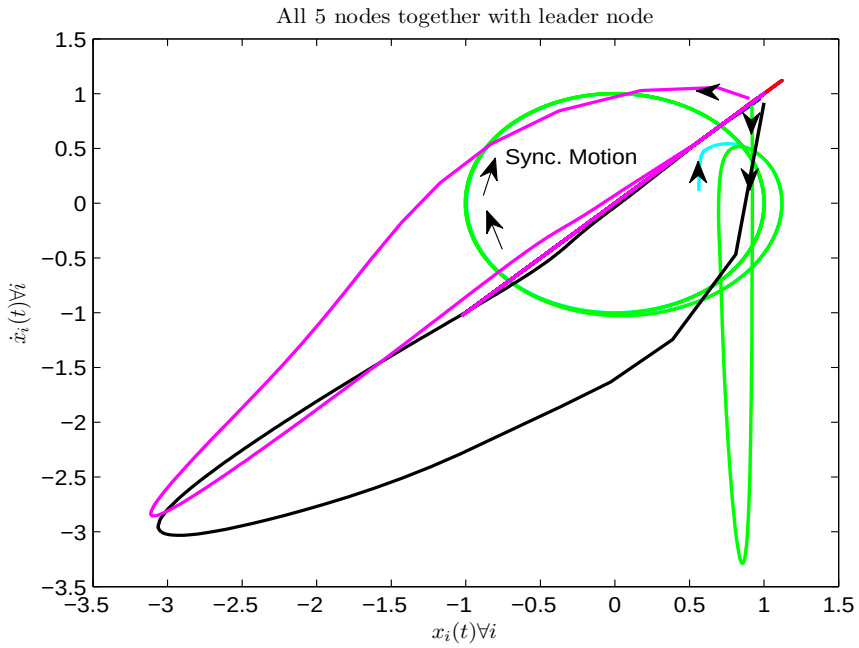

Figure 5.6. Phase plane plot for all the nodes for second order consensus.

5.5 Conclusion

This Chapter presents a method for synchronization control for multi-agent systems of order two with unknown dynamics. It illustrates the design of distributed adaptive controllers for second order nonlinear systems communicating on general strongly connected digraph network structures. The agent dynamics and command generator dynamics are considered unknown. Moreover the agent dynamics need not to be same either. It is shown that with the use of pinning control based on the exchange of cooperative neighborhood errors among the agents, one can guarantee synchronization of all the robots to the single command trajectory. A simple neural network parametric approximator is introduced at each node to estimate the unknown dynamics and disturbances. The choices of control protocol as well as neural net tuning laws are selected through a Lyapunov formulation to induce synchronization within the networked multi-robot team. A Lyapunov-based proof shows the ultimate boundedness of the tracking error. The error bounds obtained from the Lyapunov proof are dependent on the communication graph structure, and can be improved through correct pinning control design. Simulation results for 2-link robot manipulator dynamics are shown to illustrate the effectiveness of the proposed method.

CHAPTER 6

COOPERATIVE SYSTEMS STATE FEEDBACK, OBSERVERS, DUALITY, AND OPTIMAL DESIGN FOR SYNCHRONIZATION

6.1 Introduction

Coordination and consensus of distributed groups of agents is inspired by naturally occurring phenomena such as flocking in birds, swarms in insects, circadian rhythms in nature, synchronization phase transitions in physical and chemical systems, and the laws of thermodynamics [35]. Early work was done in the control systems community by [36, 37, 40, 39, 38], which by now are well known. Consensus has been studied for systems on communication graphs with fixed or varying topologies and communication delays. The average consensus problem has garnered much interest. Synchronization to time-varying trajectories has been studied based on physical or natural systems by [41, 42, 43, 44].Synchronization of nonlinear passive dynamical systems has been studied by [45]. Consensus using nonlinear protocols has been considered there, in [35], and elsewhere.

Convergence of consensus to a virtual leader or header node has been studied in [39, 46]. Recently, the pinning control has been introduced for synchronization control of coupled complex dynamical systems [50, 49, 51]. Pinning control is a powerful technique that allows controlled synchronization of interconnected dynamical systems by adding a control or leader node that is connected (pinned) into a small percentage of nodes in the network. These pinned or controlled nodes view the control node simply as another neighbor, and consider the control node's state value in computing their lo-

cal protocols. Analysis shows that, under connectivity conditions, all nodes converge to the state of the control node, which may be time-varying.

Most work on cooperative control has focused on static consensus protocols that rely on the receipt of full state variable information from one's neighbors, until recently. The focus of this research is on consensus using only output feedback (reduced state information) and a dynamic compensator. Dynamic consensus for tracking of time-varying signals using state information passed between neighbors was presented in [48]. Output feedback using a compensator was discussed in [40] which presented conditions for consensus in terms of stabilization of state-feedback systems depending on the eigenvalues of the graph Laplacian, and a Nyquist condition. It was shown that stabilizing compensators may not guarantee consensus on arbitrary communication graphs. A low gain approach to dynamic output feedback compensator design for consensus was given in [62]. Collective behavior with dynamical behavior prediction was presented in [95]. Dynamical disturbance observers were used for consensus in [96].

The idea of region of consensus was used to analyze synchronization and consensus in [97, 98, 99, 100, 101]. In [102, 103] it was shown that the synchronization region can be empty or disconnected and conditions were given for it to be unbounded. Using pinning control, methods were given to design protocols to control the topology of the synchronization region, thereby managing synchronizability properties. Consensus using output feedback and a dynamic compensator was given in [51]. This key work showed how to design compensator gains to guarantee synchronization on arbitrary strongly connected graphs using a design method based on Finsler's Lemma. A dual result to that of [40] was given for consensus in terms of stabilization of observer systems depending on the eigenvalues of the graph Laplacian. It was shown how to use

the synchronization region concept to decouple the design of stabilizing compensators from the detailed structured of the communication graph.

In [104] distributed observers were designed at each node to estimate the state of the control or leader node. State feedback protocols were used between neighbors. [105] gave a dynamic compensator for consensus, when the leader node's state is not measured, using a distributed feedback law and a distributed observer for the leader's unknown state. Riccati equation-based analysis was used. Multi-agent coordination in [106] was ensured by designing dynamic output feedback compensators using the output regulation approach.

Related to the idea of dynamic compensators for synchronization is the study of second-order consensus [56, 92, 58, 107, 87]. A lot of work has been done in this area. In fact, the node dynamics plus compensator dynamics at each node resemble the dynamics of second-order consensus at each node, though second-order consensus has almost universally used full state variable information passed between neighbors.

In this Chapter we study synchronization using reduced state information, e.g. output feedback (OPFB), where the nodes are only allowed to pass measured output information to their neighbors [76]. Thus, the node states need to be estimated for cooperative control purposes. This is achieved using cooperative dynamical observers at each node. We show how to design cooperative state variable feedbacks and observers at each node that are completely dual to each other, in a sense that extends the classical system theory notion of duality to systems on graphs. We show that unbounded consensus regions that guarantee synchronization on arbitrary strongly connected digraphs can be guaranteed by using OPTIMAL CONTROL AND OBSERVER design methods at each node based on Riccati equations. We present a duality theorem for systems on graphs and show how to design cooperative dynamical regulators that use output feedback for tracking the state of the control node.

First, in Section 6.3 of this Chapter discuss the design of cooperative state variable feedback for synchronization using a local neighborhood tracking error and pinning control. Then, we show in Section 6.4 how to design cooperative dynamical observers to estimate the node states using a local neighborhood estimation error. The observer design is dual to the SVFB design. In Section 6.5 presents a theorem for duality theory for cooperative systems on graphs. Next we design a dynamical compensator (tracker) for tracking the control node state by using only OPFB between the neighbor nodes. In Section 6.6 discuss the relation between global variables, in terms of which team performance objectives are specified, and local neighborhood variables, which are available for control purposes at the nodes. Discussion is made on how to select the pinned or controlled nodes to improve network performance. It is shown that one should pin the nodes with the highest out-degree (i.e. highest social standing), and nodes that are far apart in the original network.

6.2 State Feedback Design for Cooperative Systems on Graphs

Section 6.4 an observer is designed for coupled dynamical systems on communication graphs. To obtain the duality result in Section 6.5, the SVFB and observer structures must be properly selected. This approach refines and places into context several results in the literature that design dynamical compensators for consensus and/or synchronization, see [51] and other papers mentioned in the Introduction. In the Section 6.5, it is shown how to couple the SVFB and the observer to obtain a dynamical tracker that guarantees synchronization of coupled systems using only output feedback.

The results of this section and the next emphasize the importance of using *optimally designed* SVFB and observers to achieve guaranteed synchronization performance of cooperative systems on digraphs of arbitrary, strongly connected, structure.

6.2.1 Graph Background and Notation

Consider a graph $G = (V, E)$ with a nonempty finite set of N nodes $V = \{v_1, \cdots, v_N\}$ and a set of edges or arcs $E \subseteq V \times V$. We assume the graph is simple, e.g. no repeated edges and $(v_i, v_i) \notin E$, $\forall i$ no self loops. General directed graphs are considered. Denote the connectivity matrix as $A = [a_{ij}]$ with $a_{ij} > 0 \; if \; (v_j, v_i) \in E$ and $a_{ij} = 0$ otherwise. Note $a_{ii} = 0$. The set of neighbors of a node v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$, i.e. the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D = [d_i]$ with $d_i = \sum_{j \in N_i} a_{ij}$ the weighted in-degree of node i (i.e. i^{th} row sum of A). Define the graph Laplacian matrix as $L = D - A$, which has all row sums equal to zero. Define $d_i^o = \sum_j a_{ji}$, the (weighted) out-degree of node i, that is the ith column sum of A.

We assume the communication digraph is strongly connected, i.e. there is a directed path from v_i to v_j for all distinct nodes $v_i, v_j \in V$. Then A and L are irreducible [53]. That is they are not cogredient to a lower triangular matrix, i.e., there is no permutation matrix U such that

$$L = U \begin{bmatrix} * & 0 \\ * & * \end{bmatrix} U^T \tag{6.1}$$

The results of this research can easily be extended to graphs having a spanning tree (i.e. not necessarily strongly connected) using the Frobenius form in (6.1)

6.2.2 State Feedback for Synchronization of Networked Systems

Here we review cooperative control using full state-variable feedback (SVFB) [40, 50, 49] and give a new design result that highlights the importance of optimal design of SVFB gains for achieving synchronization of networked dynamics on arbitrary strongly connected graphs. We rely on the key concept of pinning control from the papers of R.

Chen and coworkers. Consider the N systems distributed on communication graph G_r with identical node dynamics

$$\dot{x}_i = Ax_i + Bu_i \tag{6.2}$$

where $x_i(t) \in R^n$ is the state of node i and $u_i(t) \in R^m$ its control input. Cooperative team objectives may be prescribed in terms of the local neighborhood tracking error $\varepsilon_i \in R^n$ [56]) as

$$\varepsilon_i = \sum_{j \in N_i} e_{ij}(x_j - x_i) \quad + \; g_i(x_0 - x_i) \tag{6.3}$$

The pinning gain $g_i \geqslant 0$ is nonzero for a small number of nodes i that are coupled directly to the leader or control node x_0, and $g_i > 0$ for at least one i. We refer to the nodes i for which $g_i \neq 0$ as the pinned or controlled nodes. Note that ε_i represents the information available to node i for state feedback purposes as dictated by the graph structure.

The state of the control node is $x_0(t) \in R^n$ which satisfies the dynamics

$$\dot{x}_0 = Ax_0 \tag{6.4}$$

The **Synchronization control design problem** is to design local control protocols for all the nodes in G_r to synchronize to the state of the control node, i.e. one requires $x_i(t) \to x_0(t), \; \forall i$.

Note that (6.4) is in fact a command generator [108] and we seek to design a cooperative control command generator tracker.

From (6.3), the overall error vector for network G_r is given by

$$e = -\left((L+G) \otimes I_n\right)\left(x - \underline{x}_0\right) = -\left((L+G) \otimes I_n\right)\delta \tag{6.5}$$

where $e = \begin{bmatrix} \varepsilon_1^T & \varepsilon_2^T & \cdots & \varepsilon_N^T \end{bmatrix}^T \in R^{nN}$ and $\underline{x}_0 = \underline{I}x_0 \in R^{nN}$, with $\underline{I} = \underline{1} \otimes I_n \in R^{nN \times n}$ and $\underline{1}$ the N-vector of ones. The Kronecker product is $\otimes$. $G \in R^{N \times N}$ is a

diagonal matrix with g_i. The (global) consensus or synchronization error (e.g. the disagreement vector in [37] is

$$\delta = \left(x - \underline{x}_0 \right) \in R^{nN} \tag{6.6}$$

We take the communication digraph as strongly connected. Therefore, if $g_i \neq 0$ for at least one i then $(L+G)$ is an irreducibly diagonally dominant M-matrix and hence nonsingular [53]. An M-matrix is a square matrix having its off-diagonal entries nonpositive and all principal minors nonnegative. In fact, under these connectivity assumptions all eigenvalues of $(L+G)$ have positive real parts. The next result is therefore obvious from (6.3) and the *Cauchy Schwartz* inequality and the properties of the Kronecker product [109] (See also [56]).

Lemma 6.2.1. *Let the graph be strongly connected and $G \neq 0$. Then*

$$\|\delta\| \leqslant \|e\| / \underline{\sigma}(L+G) \tag{6.7}$$

with $\underline{\sigma}(L+G)$ the minimum singular value of $(L+G)$, and $e(t) \equiv 0$ if and only if the nodes synchronize, that is

$$x(t) = \underline{I} x_0(t) \tag{6.8}$$

Select now the node control protocols as the local SVFB

$$u_i = cK\varepsilon_i = cK \left(\sum_{j \in N_i} e_{ij}(x_j - x_i) \quad + \; g_i(x_0 - x_i) \right) \tag{6.9}$$

with scalar coupling gain $c > 0$ and feedback control matrix $K \in R^{m \times n}$. These controls are implemented at each node. Then one has

$$\dot{x}_i = Ax_i + Bu_i = Ax_i + cBK \left(\sum_{j \in N_i} e_{ij}(x_j - x_i) \quad + \; g_i(x_0 - x_i) \right)$$

The overall closed-loop graph dynamics is

$$\dot{x} = \left[(I_N \otimes A) - c(L+G) \otimes BK \right] x \; + c \left[(L+G) \otimes BK \right] \underline{x}_0 \tag{6.10}$$

where the overall (global) state vector is $x = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T \in R^{nN}$. Define now the cooperative SVFB closed-loop system matrix on communication graph G_r as

$$A_c^{Gr} = [(I_N \otimes A) - c(L+G) \otimes BK], \qquad B_c^{Gr} = c\,[(L+G) \otimes BK] \tag{6.11}$$

and write

$$\dot{x} = A_c^{Gr} x \; + B_c^{Gr} \underline{x}_0 \tag{6.12}$$

A_c^{Gr} reflects the local SVFB closed-loop system matrix $(A - BK)$ as modified by distributed control on the graph structure $(L+G)$.

The global synchronization error dynamics are given as $\dot{\delta} = \dot{x} - \dot{\underline{x}}_0$ or

$$\dot{\delta} = [(I_N \otimes A) - c(L+G) \otimes BK]\,\delta = A_c^G \delta \tag{6.13}$$

The next result provides a necessary and sufficient condition for asymptotic stability of the closed-loop dynamics using SVFB.

Lemma 6.2.2. *[40] Let the graph be strongly connected with at least one pinning gain $g_i > 0$. Let $\lambda_i,\ i = 1, N$ be the eigenvalues of $(L+G)$. Then the synchronization error dynamics* (6.13) *are asymptotically stable (AS) if and only if the matrices*

$$A - c\lambda_i BK \tag{6.14}$$

are all stable.

Proof. Strong connectedness and $G \neq 0$ mean that $(L+G)$ is irreducibly diagonally dominant with all eigenvalues strictly in the right-half plane. Define a transformation M such that $J = M^{-1}(L+G)M$ is upper triangular with the eigenvalues $\lambda_i,\ i = 1, N$ of $(L+G)$ on the diagonal. Apply the state-space transformation

$$\begin{aligned} &(M^{-1} \otimes I)\,[(I_N \otimes A) - c(L+G) \otimes BK]\,(M \otimes I) \\ &\quad = [(I_N \otimes A) - cJ \otimes BK] \end{aligned}$$

and define $\xi = (M^{-1} \otimes I)\delta$. Then the transformed system is in block triangular form with diagonal blocks of the form

$$\dot{\xi}_i = (A - c\lambda_i BK)\xi_i$$

The result follows. □

It is shown in [40, 51] that an arbitrary SVFB K that stabilizes $(A - BK)$ may lead to consensus dynamics that are unstable for a prescribed graph structure. The next result shows how to select SVFB K to guarantee stability on arbitrary strongly connected graphs by OPTIMAL DESIGN and proper choice of the coupling gain c.

Theorem 6.2.3. *Suppose (A, B) is stabilizable and the graph is strongly connected with at least one pinning gain $g_i > 0$. Select design matrices $Q = Q^T > 0$, $R = R^T > 0$. Compute the SVFB gain K according to the linear quadratic regulator (LQR) control algebraic Riccati equation (CARE)*

$$0 = A^T P + PA + Q - PBR^{-1}B^T P \tag{6.15}$$

$$K = R^{-1}B^T P \tag{6.16}$$

Then the synchronization dynamics (6.13) *are asymptotically stable (AS) for all coupling gains*

$$c > \frac{1}{\underline{\lambda}(L + G)} \tag{6.17}$$

with $\underline{\lambda}(L + G) = \min_i \mathrm{Re}(\lambda_i(L + G))$.

Proof. Follows directly from the infinite gain margin robustness property of the LQR [110]. Specifically, note that LQR design renders $(A - BK)$ AS as well as $(A - kBK)$ AS for all gains $k \geqslant 1$. Under the stabilizability assumption a solution $P > 0$ to the CARE exists and the gain (6.16) renders $(A - BK)$ AS. L irreducible and at least one entry of G is positive means that $(L + G)$ is irreducibly diagonally dominant and hence

nonsingular, and its eigenvalues $\lambda_i, \ i = 1, N$ have positive real parts [53]. Hence infinite gain margin of the LQR and condition (6.17) show stability of (6.14) for λ_i real. For complex $\lambda_i = a + jb$ one has

$$(A - c(a+jb)BR^{-1}B^TP)^*P + P(A - c(a+jb)BR^{-1}B^TP) + Q +$$
$$(2ca-1)PBR^{-1}B^TP$$
$$= (A - caBR^{-1}B^TP)^TP + P(A - caBR^{-1}B^TP) + Q + (2ca-1)PBR^{-1}B^TP$$
$$= A^TP + PA + Q - PBR^{-1}B^TP$$

with $*$ the complex conjugate transpose. According to (6.15) this is equal to zero and by condition, (6.17) $(2ca-1) > 0$ so that this serves as a Lyapunov equation for $(A - c\lambda_i BK)$. □

Remark 6.2.4. *In fact, Lemma 6.2.2 and Theorem 6.2.3 show that stabilizability of* (A, B) *is necessary and sufficient for the design of stable synchronization protocols for a strongly connected graph.*

This theorem is motivated by [51] which developed a different design method based on *Finsler*'s Lemma. As noted there, the importance of the result is that it decouples the SVFB gain design from the details of the communication graph structure. The SVFB is designed using the ARE, and then the graph topology comes into the choice of the coupling gains c through condition (6.17). The theorem highlights the importance of optimal SVFB control design in guaranteeing synchronization on arbitrary strongly connected graphs.

6.2.3 Region of Consensus

The eigenvalues of $(L+G)$ may be complex. Thus, as developed in [103], according to Lemma 6.2.2 to study consensus it is necessary to analyze the stability of complex

matrix pencils [111] of the form $\sigma E + F$ with $\sigma = x + jy$. Instrumental in this is the notion of region of consensus [103, 102].

Given (A, B, K), the region of consensus is defined as the region in the complex plane of the complex parameter σ for which $(A - \sigma BK)$ is asymptotically stable. According to Lemma 6.2.2, given the eigenvalues λ_i of $(L + G)$ for a prescribed strongly connected communication graph and pinning structure, the synchronization error dynamics (6.13) are AS, on that graph if and only if there exists a coupling gain c such that $c\lambda_i,\ \forall i = 1, N$ are within the region of consensus for (A, B, K).

Theorem 6.2.3 guarantees that the region of consensus for optimal feedback gains is unbounded as $\sigma \to \infty$. The next example illustrates that optimal feedback gains designed as in Theorem 6.2.3 result in consensus on arbitrary strongly connected digraphs.

Example 6.2.5. ***Unbounded Region of Consensus for Optimal Feedback Gains*** *This example is from [51]. Let*

$$A = \begin{bmatrix} -2 & -1 \\ 2 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

6.2.4 Bounded Consensus Region for Arbitrarily Chosen Stabilizing SVFB Gain

First select the stabilizing feedback

$$K = \begin{bmatrix} 0.5 & -0.5 \end{bmatrix}$$

which gives closed-loop poles of $A - BK$ equal to $-0.25 \pm j1.199$. Compute

$$\det(sI - (A - (x + jy)BK)) = s^2 + s\,(1 - (x + jy)/2) + 3(x + jy)/2$$

The complex polynomial $p(s) = s^2 + (a + jb)s + (c + jd)$ is stable if and only if $a > 0,\ \ (abd + a^2c - d^2) > 0$. For this example that requires $x < 2$ and

$$-\left(3 - \tfrac{3x}{8}\right) y^2 + \tfrac{3x}{2}(1 - \tfrac{x}{2})^2 > 0$$

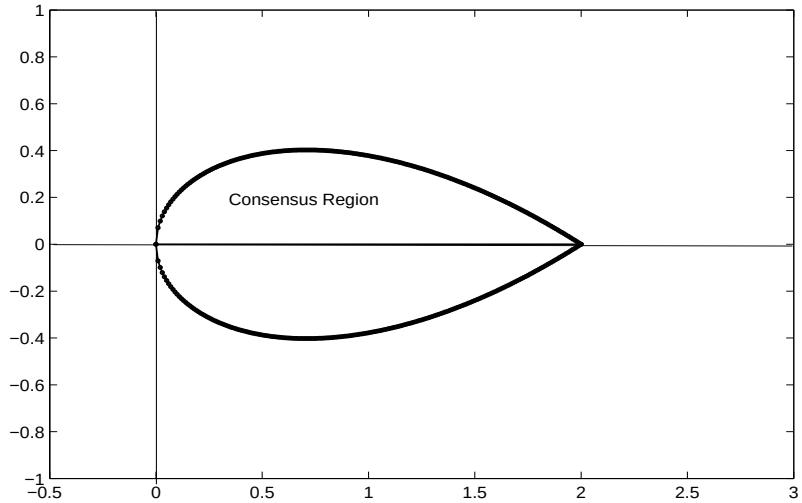

Figure 6.1. Bounded consensus region for arbitrarily chosen stabilizing SVFB gain.

This region is shown in Fig. 6.1.

The consensus region is bounded, and cannot guarantee synchronization on arbitrary graphs.

6.2.5 Unbounded Consensus Region for Optimal SVFB Gain

Now select $Q = I, R = 1$ *and perform optimal design using the CARE in Theorem 6.2.3 to obtain the optimal SVFB gain*

$$K = \begin{bmatrix} 1.544 & 1.8901 \end{bmatrix}$$

which gives closed-loop poles of $A - BK$ *equal to* $-1.272 \pm j0.7862$. *Compute* $\det(sI - (A - (x + jy)BK)) = s^2 + (a + jb)s + (c + jd)$ *and apply the conditions* $a > 0, \quad (abd + a^2c - d^2) > 0$. *The resulting consensus region is shown in Fig. 6.2.*

The consensus region is unbounded, and hence guarantees synchronization on arbitrary graphs as long as $c > 1/0.29$ *and also satisfies condition* (6.17).

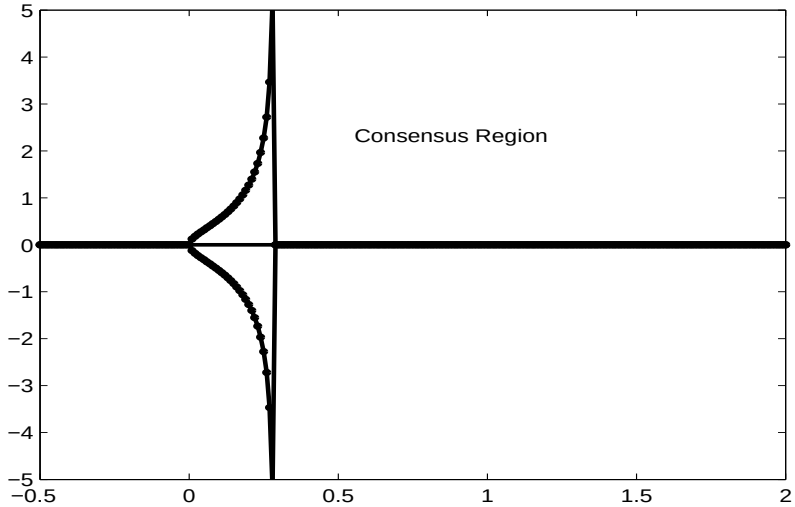

Figure 6.2. Unbounded consensus region for optimal SVFB gain.

6.3 Observer Design for Cooperative Systems on Graphs

Here we present observer design for cooperative systems. The development of this section is entirely dual to that of the previous section on SVFB. The observer structure required for duality to hold is the same as that used in [51], where however it was used as an (observer-like) dynamic compensator, and the convergence of the observer states to the node states was not guaranteed, since the (local neighborhood tracking error) feedback control structure leading to duality was not used.

Here, we give a new design result that highlights the importance of optimal design of observer gains for achieving cooperative estimation for networked dynamics on arbitrary strongly connected graphs.

In the next section we present a duality result for dynamical systems on graphs and then combine the SVFB design and the observer to design a dynamic output feedback tracker for synchronization of coupled dynamic systems. E.g. in effect, a cooperative command generator tracker [108].

Consider the N systems distributed on communication graph G_r with identical node dynamics

$$\dot{x}_i = Ax_i + Bu_i \tag{6.18}$$

$$y_i = Cx_i \tag{6.19}$$

where output $y_i(t) \in R^p$. Let $\hat{x}_i(t) \in R^n$ be an estimate for the node state $x_i(t)$ and $\hat{y}_i(t) = C\hat{x}_i(t) \in R^p$ be the consequent estimate for $y_i(t)$. Define the state estimation error and output estimation error for node i as

$$\tilde{x}_i = x_i - \hat{x}_i, \qquad \tilde{y}_i = y_i - \hat{y}_i \tag{6.20}$$

Define the local neighborhood estimation error $\varepsilon_i^o \in R^p$ as

$$\varepsilon_i^o = \sum_{j \in N_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) \quad + \; g_i(\tilde{y}_0 - \tilde{y}_i) \tag{6.21}$$

This is equivalent to

$$\varepsilon_i^o = \left[\sum_{j \in N_i} e_{ij}(y_j - y_i) \quad + \; g_i(y_0 - y_i) \right] \quad - \left[\sum_{j \in N_i} e_{ij}(\hat{y}_j - \hat{y}_i) \quad + \; g_i(\hat{y}_0 - \hat{y}_i) \quad \right] \tag{6.22}$$

which is essentially the error used in [51]. The pinning gain $g_i \geqslant 0$ is nonzero for a small number of nodes i that are coupled directly to the leader or control node x_0, and $g_i > 0$ for at least one i. Note that ε_i^o represents the information available to node i for observer design purposes as dictated by the graph structure. Using this error at each node for observer design results in cooperative estimation schemes that can be expected to outperform standard observers designed only in terms of the node estimation errors $\tilde{y}_i(t)$.

The control node $x_0(t) \in R^n$ satisfies the command generator [108] dynamics

$$\dot{x}_0 = Ax_0 \tag{6.23}$$

$$y_0 = Cx_0 \tag{6.24}$$

with output $y_0(t) \in R^p$. It is assumed that the control node knows its own state so that $\tilde{x}_0 = x_0 - \hat{x}_0 = 0, \quad \tilde{y}_0 = y_0 - \hat{y}_0 = 0$. One could design an observer for the control node as well and then include the effects of $\tilde{x}_0, \quad \tilde{y}_0$. The **Cooperative Observer design problem** is to design local observers so that all the nodes in G_r to accurately estimate the states, i.e. one requires $\tilde{x}_i(t) \to 0, \ \forall i$ or $\hat{x}_i(t) \to x_i(t), \ \forall i$.

From (6.21), the overall estimation error vector for network G_r is given by

$$e^o = -\left((L+G) \otimes I_n\right)\left(\tilde{y} - \underline{\tilde{y}}_0\right) = -\left((L+G) \otimes I_n\right)\tilde{y} = -\left((L+G) \otimes C\right)\tilde{x} \tag{6.25}$$

since $\tilde{x}_0 = 0, \quad \tilde{y}_0 = 0$. Here the overall local neighborhood error vector is $e^o = \begin{bmatrix} (\varepsilon_1^o)^T & \cdots & (\varepsilon_N^o)^T \end{bmatrix}^T \in R^{pN}$, the global estimation error vector is $\tilde{y} = \begin{bmatrix} \tilde{y}_1^T & \cdots & \tilde{y}_N^T \end{bmatrix}^T \in R^{pN}$ and $\underline{\tilde{y}}_0 = (\underline{1} \otimes I_p)\tilde{y}_0 \in R^{pN}$ with $\underline{1}$ the N-vector of ones.

If the communication digraph is strongly connected and $g_i \neq 0$ for at least one i, then $(L+G)$ is an irreducibly diagonally dominant M-matrix and hence nonsingular. The next result therefore follows directly.

Lemma 6.3.1. *Let the graph be strongly connected and $G \neq 0$. Then*

$$\|\tilde{y}\| \leqslant \left\|e^0\right\| / \underline{\sigma}(L+G) \tag{6.26}$$

with $\underline{\sigma}(L+G)$ the minimum singular value of $(L+G)$. Then, if (A, C) is observable, $\|\tilde{x}\|$ is bounded in terms of the observability gramian norm and $\|\tilde{y}\|$. Moreover, $e^0(t) \equiv 0$ if and only if $\tilde{y}_i \equiv 0, \ \forall i$. Then, if (A, C) is observable, $\tilde{x}_i \equiv 0, \ \forall i$.

Design now the local node observers as

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\varepsilon_i^o \tag{6.27}$$

with scalar coupling gain $c > 0$ and observer gain matrix $F \in R^{n \times p}$. Then one has

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\left[\sum_{j \in N_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) \quad + \ g_i(\tilde{y}_0 - \tilde{y}_i)\right] \tag{6.28}$$

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\left[\sum_{j\in N_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) \quad + \; g_i(\tilde{y}_0 - \tilde{y}_i)\right] \tag{6.29}$$

The distributed observers are implemented locally at each node in the form (6.27) or (6.28), which require that the node estimation errors $\tilde{y}_i$ are passed to neighboring nodes. The local neighborhood estimation errors can alternatively be computed according to (6.22), which requires a different local information flow between the nodes, i.e. variables $y_i, \hat{y}_i$ are passed to neighboring nodes.

The global observer dynamics is

$$\dot{\hat{x}} = \left[(I_N \otimes A) - c(L+G)\otimes FC\right]\hat{x} \; + c\left[(L+G)\otimes F\right]y + (I_N \otimes B)u \tag{6.30}$$

where $\hat{x} = \begin{bmatrix} \hat{x}_1^T & \hat{x}_2^T & \cdots & \hat{x}_N^T \end{bmatrix}^T \in R^{nN}$. Define now the cooperative observer matrix on communication graph G_r as

$$A_o^{Gr} = \left[(I_N \otimes A) - c(L+G)\otimes FC\right], \qquad F_o^{Gr} = c\left[(L+G)\otimes F\right] \tag{6.31}$$

and write

$$\dot{\hat{x}} = A_o^{Gr}\hat{x} \; + F_o^{Gr}y + (I_N \otimes B)u \tag{6.32}$$

This is similar to the standard structure for observers for state space systems. A_o^{Gr} reflects the local observer matrix $(A - FC)$ as modified by local distributed estimation on the graph structure $(L+G)$.

The estimation error dynamics are given as $\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}}$ or

$$\dot{\tilde{x}} = \left[(I_N \otimes A) - c(L+G)\otimes FC\right]\tilde{x} \; = A_o^{Gr}\tilde{x} \tag{6.33}$$

The next result is dual to Lemma 6.2.2 and provides a necessary and sufficient condition for asymptotic stability of the observer dynamics.

Lemma 6.3.2. *Let the graph be strongly connected with at least one pinning gain $g_i > 0$. Let $\lambda_i,\ i = 1, N$ be the eigenvalues of $(L + G)$. Then the observer error dynamics* (6.33) *are asymptotically stable (AS) if and only if the matrices*

$$A - c\lambda_i FC \tag{6.34}$$

are all stable.

Proof. Exactly the dual to the proof of Lemma 6.2.2. See [51]. □

It is shown in [51] that an arbitrary observer gain F that stabilizes $(A - FC)$ may lead to distributed estimation error dynamics that are unstable. Specifically, for improper choice of a stabilizing gain F, the consensus region may be bounded by a compact set or even disconnected. See Example 6.2.5 The next result shows how to select the observer gain matrix F to guarantee stability on arbitrary strongly connected digraphs by OPTIMAL DESIGN and selecting the coupling c larger than a specific threshold value that depends on the graph structure.

Theorem 6.3.3. ***Design of Observer Gain for Cooperative Estimation***
Suppose (A, C) is detectable and the graph is strongly connected with at least one pinning gain $g_i > 0$. Select design matrices $Q = Q^T > 0,\ R = R^T > 0$. Compute the observer gain F according to the observer algebraic Riccati equation (OARE)

$$0 = AP + PA^T + Q - PC^T R^{-1} CP \tag{6.35}$$

$$F = PC^T R^{-1} \tag{6.36}$$

Then the estimation error dynamics (6.33) are asymptotically stable for all coupling gains

$$c > \frac{1}{\underline{\lambda}(L + G)} \tag{6.37}$$

with $\underline{\lambda}(L + G) = \min_i \mathrm{Re}(\lambda_i(L + G))$.

Proof. Based on the robust infinite gain margin property of the Kalman filter [112], exactly dual to proof of Theorem 6.2.3. □

Remark 6.3.4. *In fact, Lemma 6.3.2 and Theorem 6.3.3 show that the detectability of* (A, C) *is necessary and sufficient for the design of stable observer protocols for a strongly connected graph.*

The article [51] developed a different design method based on Finsler's Lemma. As noted there, the importance of this result is that it decouples the observer gain design from the details of the communication graph structure. The graph topology comes into the choice of the coupling gains c through condition (6.37). The theorem highlights the importance of optimal estimation design in guaranteeing synchronization on arbitrary strongly connected graphs. Note that the OARE provides the optimal minimum mean square error, e.g. the Kalman Filter, for systems (6.18), (6.19) with zero mean white process noise having covariance Q and zero mean white measurement noise having covariance R [112].

6.4 Duality and Dynamic Output Feedback Tracker Design for Cooperative Systems

In this section are presented a control/observer duality result for coupled dynamical systems on graphs, and a methodology for the design of dynamic output feedback trackers for synchronization of cooperative systems.

6.4.1 Control/Observer Duality on Graphs

This discourse is short, yet we have decided to give it its own subsection head. Here, the crucial notion of duality in dynamical system theory is extended to dynamical systems coupled on communication graphs.

Definition 6.4.1. *The converse, transpose, or reverse of a directed graph* $G_r = (V, \mathrm{E})$ *is another directed graph* $G_r' = (V, \mathrm{E}')$ *on the same set of vertices with all of the edges reversed. That is,* $(v_j, v_i) \in \mathrm{E} \;\Rightarrow\; (v_i, v_j) \in \mathrm{E}'$. *Then, the connectivity matrix of* G_r' *is* E^T, *with* E *the connectivity matrix of* G_r.

This definition was used in [37] to extend Lyapunov proofs of consensus to balanced graphs. Note that the terminology 'dual graph' is reserved for another concept not related to the current discussion [113].

Theorem 6.4.2. ***SVFB Control/Observer Duality on Graphs***
Given dynamical nodes with input-coupling structure (A, B) *and communication coupling graph* G_r, *suppose SVFB* K *stabilizes the synchronization dynamics* (6.13). *Then observer gain* K^T *stabilizes the estimation error dynamics* (6.33) *for systems with output-coupling structure* (A^T, B^T) *and communication coupling graph* G_r'.

Proof. The synchronization error dynamics is

$$\dot{\delta} = [(I_N \otimes A) - c(L+G) \otimes BK]\,\delta \;\equiv A_c^G \delta$$

and the estimation error dynamics is

$$\dot{\tilde{x}} = [(I_N \otimes A) - c(L+G) \otimes FC]\,\tilde{x} \;\equiv A_o^{Gr} \tilde{x}$$

Standard operations on the Kronecker product [109] reveal that

$$[(I_N \otimes A) - c(L+G) \otimes BK]^T = \left[(I_N \otimes A^T) - c(L+G)^T \otimes K^T B^T\right]$$

Now the result follows from the standard dynamical system duality relation based on $(A - BK)^T = (A^T - K^T B^T) \approx (A - FC)$ [114]. □

6.5 Dynamic Tracker for Synchronization Using Output Feedback

In this section we use the above results on SVFB design and observer design to build a dynamic tracker for synchronization of cooperative systems that uses only output feedback. That is, full state measurements of the node dynamics are not available. The results herein should be taken in the light of existing results for dynamic regulator/tracker design for cooperative systems as detailed in the Introduction. The design herein capitalizes on the SVFB control/observer symmetry induced by duality, and extends the design of dynamical regulators for dynamical systems [114] to coupled systems on graphs in a rather natural way.

Consider node dynamics given by (6.18) with output measurements (6.19). The **reduced measurement synchronization control design problem** is to design local control protocols for all the nodes in G_r to synchronize to the state of the control node, i.e. one requires $x_i(t) \to x_0(t),\ \forall i$, however, only measurements of the outputs (6.19) are available. This can also be called the **output feedback (OPFB) synchronization problem**.

Given the node dynamics

$$\dot{x}_i = Ax_i + Bu_i \tag{6.38}$$

$$y_i = Cx_i \tag{6.39}$$

Design dynamic OPFB trackers as follows. Select observer dynamics at each node in terms of the local neighborhood estimation errors (6.21) as

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\varepsilon_i^o \tag{6.40}$$

$$\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\left[\sum_{j\in N_i} e_{ij}(\tilde{y}_j - \tilde{y}_i) \quad + \ g_i(\tilde{y}_0 - \tilde{y}_i)\right] \tag{6.41}$$

Select the node control protocols in terms of the estimated local neighborhood tracking errors (6.3) as

$$u_i = cK\hat{\varepsilon}_i = cK\left[\sum_{j\in N_i} e_{ij}(\hat{x}_j - \hat{x}_i) \quad + \; g_i(x_0 - \hat{x}_i)\right] \tag{6.42}$$

Assumption 6.5.1. *Assume the control node knows its own state $x_0(t)$ so that $\tilde{x}_0(t) = 0$. Assume that the control node passes its full state $x_0(t)$ to the pinned nodes.*

The assumption can be alleviated by placing an observer at the control node and passing $\hat{x}_0(t)$ instead of $x_0(t)$. That does not change the following. In [104] distributed observers are designed at each node to estimate the state of the control node using a dynamic compensator. Full state-variable information is passed between neighbors.

With this dynamical compensator structure, the closed-loop dynamics are

$$\dot{x}_i = Ax_i + cBK\left[\sum_{j\in N_i} e_{ij}(\hat{x}_j - \hat{x}_i) \quad + \; g_i(x_0 - \hat{x}_i)\right] \tag{6.43}$$

The complete global closed-loop dynamics are

$$\dot{x} = (I_N \otimes A)x - c(L+G)\otimes BK\hat{x} \; + c\left[(L+G)\otimes BK\right]\underline{x}_0 \tag{6.44}$$

$$\begin{aligned}\dot{\hat{x}} = &\left[(I_N \otimes A) - c(L+G)\otimes FC\right]\hat{x} \; + c\left[(L+G)\otimes F\right]y \\ &-c(L+G)\otimes BK\hat{x} \; + c\left[(L+G)\otimes BK\right]\underline{x}_0\end{aligned} \tag{6.45}$$

Theorem 6.5.2. ***OPFB Regulator Synchronization Performance***

Let (A, B, C) be stabilizable and detectable, the communication graph be strongly connected, and at least one pinning gain nonzero. Select the SVFB gain K and pinning gain c according to Theorem 6.2.3 and the observer gain F according to Theorem 6.3.3. Then the cooperative dynamic OPFB trackers given by (6.41), (6.42) *yield synchronization of all nodes to the control state $x_0(t)$.*

Proof. Write (6.44) as

$$\dot{x} = [(I_N \otimes A) - c(L+G) \otimes BK]\, x + c\,[(L+G) \otimes BK]\, \tilde{x} + c\,[(L+G) \otimes BK]\, \underline{x}_0 \tag{6.46}$$

or in terms of the synchronization error (6.6)

$$\dot{\delta} = [(I_N \otimes A) - c(L+G) \otimes BK]\, \delta + c\,[(L+G) \otimes BK]\, \tilde{x} = A_c^{Gr}\delta + B_c^{Gr}\tilde{x} \tag{6.47}$$

where the observer error dynamics are

$$\dot{\tilde{x}} = [(I_N \otimes A) - c(L+G) \otimes FC]\, \tilde{x} \;\; = A_o^{Gr}\tilde{x} \tag{6.48}$$

Then one has

$$\begin{bmatrix} \dot{\delta} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A_c^{Gr} & B_c^{Gr} \\ 0 & A_o^{Gr} \end{bmatrix} \begin{bmatrix} \delta \\ \tilde{x} \end{bmatrix}$$

Theorem 6.2.3 shows the stability of A_c^{Gr} while Theorem 6.3.3 shows the stability of A_o^{Gr}. □

6.6 Local vs. Global Variables and Selection of Pinned Nodes

Overall team performance specifications for cooperative dynamical systems are given in terms of global variables such as the global synchronization tracking error

$$\delta = \left(x - \underline{x}_0 \right) \in R^{nN} \tag{6.49}$$

and the global estimation errors

$$\tilde{x} = x - \hat{x} \;\in R^{nN}, \;\; \tilde{y} = y - \hat{y} \in R^{pN} \tag{6.50}$$

However, the information available at each node for control or estimation purposes is in terms of local neighborhood variables, e.g. the local neighborhood tracking error $\varepsilon_i \in R^n$ (6.3) and the local neighborhood estimation error $\varepsilon_i^o \in R^p$ (6.21).

To turn global variables into local variables for any specified communication graph structure, it is only necessary to multiply the former by $(L+G)$, with L the graph Laplacian for that graph and G the diagonal matrix of pinning gains. Then one obtains the control and estimation local neighborhood errors, respectively, equations (4.6) and (6.25), reproduced here:

$$e = -\left((L+G)\otimes I_n\right)\delta$$

$$e^o = -\left((L+G)\otimes I_n\right)\tilde{y} = -\left((L+G)\otimes C\right)\tilde{x}$$

Then, one has results like Lemma 6.2.1 and Lemma 6.3.1 which reveal that the effectiveness of local neighborhood protocols for guaranteeing global synchronization or estimation objectives depends on the minimum singular value $\underline{\sigma}(L+G)$.

Results such as Theorem 6.2.3 and Theorem 6.3.3 further show design requirements/limitations in terms of the smallest eigenvalue/smallest singular value of $(L+G)$. One advantage of pinning control as developed by R. Chen and his group [50, 49] is that one no longer has to deal with one eigenvalue of the Laplacian being at zero and the notorious *second* (Fiedler) eigenvalue of L. The nonsingularity of $(L+G)$ allows one to use the machinery of nonsingular M-matrices [53] and quite tractable max/min singular value inequalities for determining performance guarantees and limitations [94].

Design of effective protocols for cooperative systems involves selecting the pinned nodes and pinning gains to make $\underline{\sigma}(L+G)$ large. It can easily be shown that this also results in faster convergence in distributed adaptive systems [94]. Chen [50, 49] has provided insightful guidance for the selection of the pinned nodes for undirected graphs. The next results extend such notions to directed graphs. Above, $d_i = \sum_{j\in N_i} e_{ij}$, the ith row sum of E, has been defined as the weighted in-degree of node i.

Definition 6.6.1. $d_i^o = \sum_j e_{ji}$ *is the (weighted) out-degree of node* i*, that is the* i^{th} *column sum of the graph connectivity matrix* E*.*

The out degree has been called the *social standing* or *social influence* of node i by J. Baras [115, 46]. The next result relates the smallest singular value of $(L+G)$ to its determinant and the graph structure.

Lemma 6.6.2.

$$\underline{\sigma}(L+G) \geqslant \frac{|L+G|}{\sqrt{N}\left(\max_i(d_i^s)\right)^{N-1}} \tag{6.51}$$

with $d_i^S \equiv g_i + d_i + d_i^o$ the in-degree sum of node i.

Proof. See Bernstein [55]. One has

$$|L+G| = \prod_{i=1}^{N} \sigma_i(L+G) = \underline{\sigma}(L+G) \prod_{i=1}^{N-1} \sigma_i(L+G) \leqslant \underline{\sigma}(L+G)\bar{\sigma}^{N-1}(L+G)$$

with $\bar{\sigma} = \sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_N = \underline{\sigma}$ the singular values. Also

$$\bar{\sigma}(L+G) \leqslant \sqrt{N}\|L+G\|_1$$

with $\|.\|_1$ the induced 1-norm (max absolute column sum). Recalling that $L+G = D+G-E$ the result follows. □

Remark 6.6.3. *The book by Bernstein [55] is invaluable for analysis and design of dynamical systems on graphs using singular value bounds. It is shown in [110] that singular values (not Fiedler eigenvalue!) arise naturally in Lyapunov design for control of pinned graph structures.*

The bound in this lemma is not a very good one, but it shows that the design issue is to make the determinant of $(L+G)$ large. The following result shows how to accomplish this.

Theorem 6.6.4. ***Effect on Closed-Loop Network of Control Injection to Pinned Nodes*** *Select k pinned (control injection) nodes in network G_r given by the index set $(i_1,\ i_2, \cdots i_k)$,*

so that the pinning gains $g_{i_1}, g_{i_2}, \cdots, g_{i_k}$ *in the diagonal pinning matrix* G *are nonzero. Then the determinant of* $(L+G) \in R^{N\times N}$ *is given by*

$$|L+G| = |L| + \sum_{s=1}^{k} \sum_{1\leqslant j_1<j_2<\cdots<j_s\leqslant k} \left(g_{i_{j_1}} g_{i_{j_2}} \cdots g_{i_{j_s}}\right) L\left(\frac{i_{j_1}\ i_{j_2}\cdots i_{j_s}}{i_{j_1}\ i_{j_2}\cdots i_{j_s}}\right) \tag{6.52}$$

where $L\left(\frac{i_{j_1}\ i_{j_2}\cdots i_{j_s}}{i_{j_1}\ i_{j_2}\cdots i_{j_s}}\right)$ *denotes the minor of the Laplacian matrix* L *obtained by striking out the* s *rows and columns* $i_{j_1}\ i_{j_2}\cdots i_{j_s}$*; that is, the determinant of the submatrix of* L *formed by removing the specified rows and columns [116].*

Proof. The proof is by induction noting the easy determinantal identity from *Gantmacher.*

$$|G_i + L| = |L| + g_i L\left(\tfrac{i}{i}\right)$$

where G_i is a matrix having only one nonzero pinning gain g_i on the diagonal in the ith position and $L\left(\frac{i}{i}\right)$ is the minor striking out the ith row and column of L. □

The result (6.52) can be expressed as

$$|L+G| = |L| + \sum_{i=1}^{k} g_i L\left(\frac{i}{i}\right) + \sum_{\substack{i,j=1\\ i\neq j}}^{k} g_i g_j L\left(\frac{ij}{ij}\right) + \sum_{\substack{i,j,\ell=1\\ i\neq j\neq \ell}}^{k} g_i g_j g_\ell L\left(\frac{ij\ell}{ij\ell}\right) + \cdots$$

To make $|L+G|$ larger, the first term shows that one should pin into the nodes i with the largest out-degrees to make the corresponding minors large, for this increases the diagonal dominance of the minors. The second and subsequent terms reveal that one should not pin into two nodes with the same neighbors, for then the beneficial effect of crossing out the corresponding columns in increasing the diagonal dominance of the minor is lessened. This indicates that the pinned nodes should be far apart from one another in the graph.

Therefore, one should pin into widely separated nodes with the highest social influence, in the words of J. Baras.

6.7 Conclusion

We studied synchronization using reduced state information where the nodes are only allowed to pass measured output information to their neighbors. The node states were estimated using cooperative dynamical observers at each node. We showed how to design cooperative state variable state feedbacks and observers at each node that are dual to each other, in a sense that extends the classical system theory notion of duality to systems on graphs. We showed that unbounded consensus regions that guarantee synchronization on arbitrary strongly connected digraphs can be guaranteed by using OPTIMAL CONTROL AND OBSERVER design methods at each node based on Riccati equations. We presented a duality theorem for systems on graphs and showed how to design cooperative dynamical regulators that use output feedback for tracking the state of the control node. Finally, it was shown that the pinned or controlled nodes in a network should be selected as nodes of large out-degree (*social standing*) that are far apart from each other.

CHAPTER 7

DECENTRALIZED CONTROL OF HVAC SYSTEM

7.1 Introduction

The main purposes of a Heating, Ventilation, and Air-Conditioning (HVAC) system are to help maintain good indoor air quality through adequate ventilation with filtration and provide thermal comfort. A quality HVAC system has to be properly sized to provide correct air flow, and meet room-by-room calculated heating and cooling loads. It is also require that to be installed so that the static air pressure drop across the handler is within manufacturer and design specifications to have the capacity to meet the calculated loads. A good HVAC system should have balanced air flows between supply and return systems to maintain neutral pressure in the home. For a large building it is not always easy to control thermal comfort for every person inside the building. The airflow through the vent is to be properly controlled at every moment to satisfy the criterion of temperature consensus inside the building. It requires proper coordination among airflow, temperature sensor position and its accuracy, vent positions etc.

A wide range of studies can be found in the literature regarding indoor air distributions. The article [63] demonstrates a numerical method to control air distribution inside a office building. It is shown that the efficiency of HVAC system decreases along with the increase in number of occupants in the building. The paper also investigates the characteristics of air to be distributed inside the building for quality thermal consensus among the occupants. Other related papers includes [64, 65, 66, 67, 68]. These papers investigate the method of air distribution from different inlet and outlet posi-

tions. Natural convection from heated room surfaces is studied in [65]. The HVAC with digital control system using computer program is studied by [69, 70].

[71] demonstrate the use of decentralized control for thermal consensus in HVAC system. Consensus has been studied for systems on communication graphs with fixed or varying topologies and communication delays. See [37, 40, 38, 39], which proposed basic synchronizing protocols for various communication topologies. Early work on consensus studied leaderless consensus or the cooperative regulator problem, where the consensus value reached depends on the initial conditions of the node states. On the other hand, the cooperative tracker problem seeks consensus or synchronization to the state of a control or leader node. Convergence of consensus to a virtual leader or header node was studied in [39, 46]. Dynamic consensus for tracking of time-varying signals was presented in [47]. The pinning control has been introduced for synchronization tracking control of coupled complex dynamical systems [48, 49, 50],[51]. Pinning control allows controlled synchronization of interconnected dynamical systems by adding a control or leader node that is connected (pinned) into a small percentage of nodes in the network. Analysis has been done using Lyapunov and other techniques by assuming either a Jacobian linearization of the nonlinear node dynamics, or a Lipschitz condition, or contraction analysis. The agents are homogeneous in that they all have the same nonlinear dynamics.

In this Chapter we discussed the pinning control for HVAC systems. The aim of this particular research is to synchronize the room temperatures at every point with a desired set of temperature values irrespective of the number of occupants. The modeling of HVAC system as a decentralized control unit is little different than that of standard multiagent systems discussed in the previous chapters. Considering each outlet vents as a node point, a undirected graph topology can be defined around the building. As not all the vents use for control purpose and the neighborhood of a control node can

be different than that of undirected modeling graph, a directed control graph topology is defined over the modeling graph. The dynamics of the HVAC system is derived from the poison heat equations and from the flow dynamics in the pipe network installed in the building. The actual control variables are vent positions and air flow rate through the outlet vents. The control variables are appeared as bilinear fashion in the dynamics. A suitable synchronization control protocol is proposed based on the pinning control formulation. A Lyapunov method is shown to prove the overall system stability under the proposed control protocol.

7.2 Dynamic Modeling of the HVAC System- The Model Graph

This paper considers the control of a large heating, ventilation, and air conditioning (HVAC) system of the sort shown in Fig. 7.1. The HVAC system services a large living space with numerous rooms or cubicles. This is a distributed system with numerous air diffuser vents which blow air into various room spaces, and numerous control damper valves which regulate the amount of air flow into each vent subsystem. Heating and/or cooling is provided by air exiting the diffuser vents. The objective is to regulate the valve positions to control the airflow from the vents in order to yield a uniform temperature distribution through the spaces serviced by the HVAC system. This is a complex distributed control problem with numerous control inputs, numerous outputs, and interactions between multiple distributed subsystems. Various disturbances include heat sources and heat sinks distributed throughout the building space such as windows, operating machinery that generates heat, and so on.

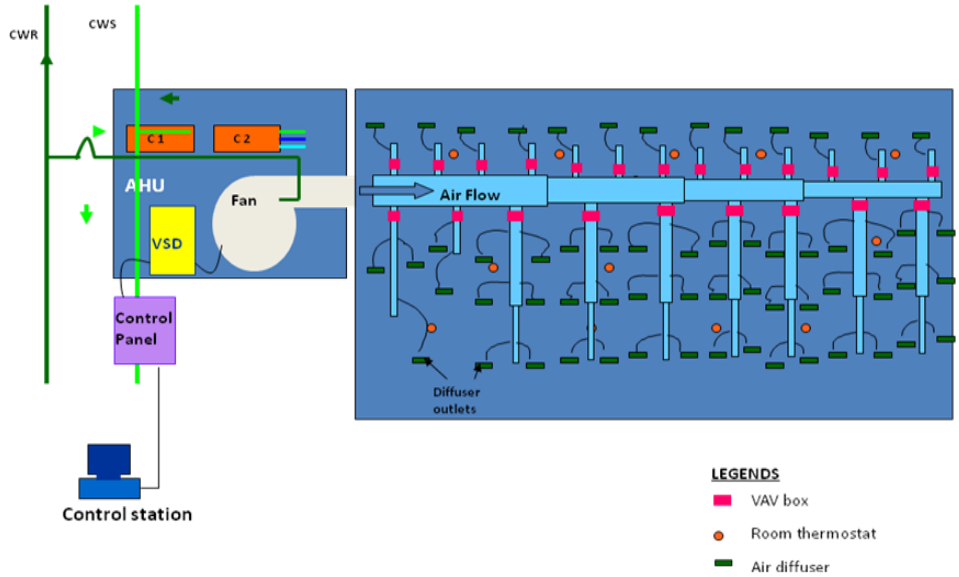

Figure 7.1. Distributed HVAC system showing air diffuser vents, control valves, and thermostats.

The heat flow through the space is modeled using the basic heat equation, which describes the distribution of heat or variation in temperature in a given region over time,

$$\frac{\partial \tau}{\partial t} = \alpha \nabla^2 \tau + Qu + d \tag{7.1}$$

with $\tau(x, y, t)$ the temperature at position (x, y) at time t, and ∇^2 the Laplace operator with $\nabla \tau$ the gradient operator. The heat generated by the air diffuser flow inlets are denoted by Q (is calculated from pipe network analysis and is discussed later) and $u(x, y, t)$ which actually denotes the valve opening and can be positive or negative (in the case of cooling). The heat disturbances (positive or negative) due to windows, heat generating machinery, etc. is denoted as $d(x, y, t)$. The heat equation in two-dimensions is written as

$$\frac{\partial \tau}{\partial t} = \alpha \left(\frac{\partial^2 \tau}{\partial x^2} + \frac{\partial^2 \tau}{\partial y^2} \right) + Qu + d \tag{7.2}$$

The heat equation is a complicated nonlinear partial differential equation. Traditional solution of the heat equation is accomplished through numerical approximation, using,

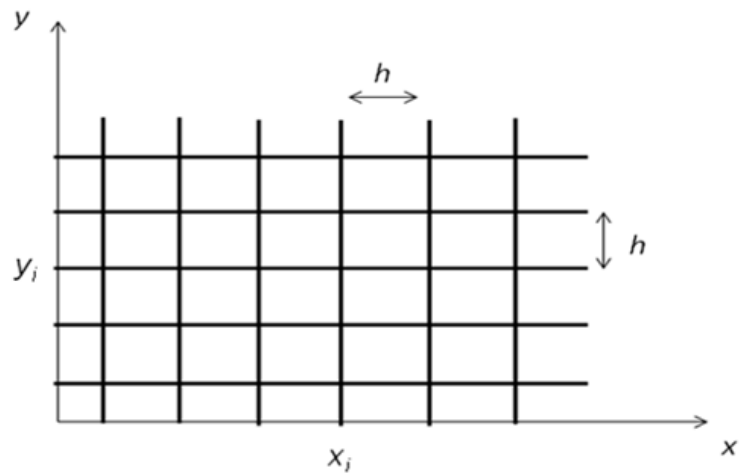

Figure 7.2. Uniform mesh imposed on spatial coordinates (x, y) for heat equation numerical solution.

e.g., the relaxation method, by imposing a mesh on the spatial coordinates (x, y). A uniform mesh of grid size h is shown in Fig. 7.2. Using the forward difference, the gradient with respect to x can be approximated as

$$\nabla_x \tau = \frac{\partial \tau}{\partial x} = \frac{1}{h}\left(\tau(x+h) - \tau(x)\right) \tag{7.3}$$

and the Laplacian with respect to x as

$$\begin{aligned} \nabla_x^2 \tau = \frac{\partial^2 \tau}{\partial x^2} &= \frac{1}{h}\left(\frac{1}{h}\left(\tau(x+h) - \tau(x)\right) - \frac{1}{h}\left(\tau(x) - \tau(x-h)\right)\right) \\ &= \frac{1}{h^2}\left(\tau(x+h) + \tau(x-h) - 2\tau(x)\right) \end{aligned} \tag{7.4}$$

Then, the $2D$ Laplacian becomes

$$\begin{aligned} \nabla^2 \tau = \frac{\partial^2 \tau}{\partial x^2} + \frac{\partial^2 \tau}{\partial y^2} &= \frac{1}{h^2}\left(\tau(x+h, y) + \tau(x-h, y) + \tau(x, y+h)\right) \\ &+ \frac{1}{h^2}\left(\tau(x, y-h) - 4\tau(x, y)\right) \end{aligned} \tag{7.5}$$

Enumerate now the points of the mesh as (x_i, y_i) and denote the temperature of mesh point or node i as $\tau_i(x_i, y_i, t)$ and the heat sources and disturbances at node i respectively as . Referring to Fig. 7.2 one has the neighbors (for computation) of node (x, y)

as $\{(x+h,y),(x-h,y),(x,y+h),(x,y-h)\}$. Define now $N_{m,i}$ as the set of nodes j that are adjacent to node i in the enumeration of the regular mesh. (We use subscripts $'m'$ to denote variables of the so-called 'model graph', which is defined shortly.) This set depends on the actual enumeration used for the nodes, e.g. row-wise, column-wise, etc. Then, one obtains a numerical approximation to the heat equation (7.2) as

$$\dot{\tau}_i = \frac{\alpha}{h^2}\left(-4\tau_i + \sum_{j\in N_{m,i}} \tau_j\right) + Q_i u_i + d_i \tag{7.6}$$

Though the set of neighbors $N_{m,i}$ of node i depends on the enumeration used for the mesh points, according to Fig. 7.2 and (7.5) each node has 4 neighbors. Define $\alpha_{ij} = 1$ if $j \in N_{m,i}$, i.e. if node j is a neighbor of node i, and zero otherwise. Then one can write

$$\dot{\tau}_i = \frac{\alpha}{h^2} \sum_{j\in N_{m,i}} \alpha_{ij}(\tau_j - \tau_i) + Q_i u_i + d_i \tag{7.7}$$

Define more generally $\alpha_{ij} > 0,\ j \in N_{m,i}$ and absorb α/h^2 into α_{ij} to write

$$\dot{\tau}_i = \sum_{j\in N_{m,i}} \alpha_{ij}(\tau_j - \tau_i) + Q_i u_i + d_i \tag{7.8}$$

Note that the matrix $A_m = [\alpha_{ij}]$ has four nonzero entries on each row. Define the row sum $d_{m,i} = \sum_{j\in N_i} \alpha_{ij}$. Define the diagonal matrix $D_m = diag\{d_{m,i}\}$. Then the Laplacian operator matrix is defined as

$$L_m = D_m - A_m \tag{7.9}$$

Note that the sum of each row of L_m is zero due to the properties of the forward difference Laplacian operator (7.5).

Definition 7.2.1. ***Model Graph*** *Define the undirected graph $G_m = (V_m, E_m)$ associated with the numerical computation mesh, where V_m is the set of mesh nodes, and E_m is the set of edges. We refer to the graph $G_m = (V_m, E_m)$ as the dynamic model graph for the heat flow within the space serviced by the HVAC system. The edges are undirected*

arcs between each node and its four neighbors in the mesh as defined in Fig. 7.2. That is $(j, i) \in E_m \; if \; \alpha_{ij} \neq 0$.

Then matrix A_m is the graph adjacency matrix of G_m, L_m is the graph Laplacian matrix, and $d_{m,i}$ is the degree of node i. Since G_m is undirected, A_m is symmetric and L_m is positive semidefinite. Note that L_m has row sum of zero. Let N_m be the total number of nodes in G_m. The set of neighbors $N_{m,i}$ of node i has cardinality of 4 for every i.

Concatenate the temperatures τ_i at the nodes i into a global state vector $\tau = \begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_{N_m} \end{bmatrix} \in R^{N_m}$, defining likewise the diagonal matrix $Q = diag\{Q_i\}$, the vector $u(t) \in R^{N_m}$ and the vector of disturbances $d(t) \in R^{N_m}$. Write the overall dynamics of all nodes as

$$\dot{\tau} = -L_m \tau + Qu + d \tag{7.10}$$

In practice, the temperatures $\tau_i(x_i, y_i, t)$ are not measured at regular mesh grid points nor are the sources and disturbances located at regular grid points. Nevertheless, (7.8), (7.10) will be taken as the heat flow model for the HVAC system, with the node points i taken as being at the (x, y) positions of the diffuser vents, disturbances, etc.

7.3 Control of the HVAC System - The Control Graph

The control objective is to regulate the control valve positions to yield a uniform prescribed temperature distribution throughout the living spaces serviced by the HVAC system. This relies on controlling the airflow of the diffuser vents to regulate the diffuser vents, which provide the heat sources $u_i(t)$ in (7.8). Here we take $u_i(t)$ as the control inputs to streamline the problem.

Define, therefore, a control or leader node which has the prescribed temperature state given according to the dynamics

$$\dot{\tau}_0 = f_0(\tau_0, t) \tag{7.11}$$

This is a command generator that generates the desired temperature profile $\tau_0(t)$ of the nodes. This may be, e.g., a function of the time of day. A special case is the standard constant consensus value problem, which has $\dot{\tau}_0 = 0$. The control objective is formalized below.

Definition 7.3.1. ***Synchronization Control Problem.*** *Select control protocols $u_i(t)$ in* (7.8), (7.10) *so that $\tau_i(t) \to \tau_0(t), \ \forall i$.*

To confront this control objective, define the local neighborhood synchronization error for node i as

$$e_i = \sum_{j \in N_i} a_{ij}(\tau_j - \tau_i) + b_i(\tau_0 - \tau_i) \tag{7.12}$$

where the error coefficients a_{ij} are defined in accordance with practical requirements and may not be the same as the modeling coefficients α_{ij} in (7.8). Moreover, the neighbor node set N_i for control purposes may not be the same as the neighbor node set for dynamic modeling $N_{m,i}$ as used in (7.8). The latter is composed of node i's nearest neighbors, which determine the heat flow dynamics in the room. On the other hand, the control neighbor node set N_i may be taken to include nodes physically located far from node i. The convergence speed of the temperature control algorithm to be developed will depend on the choice of N_i.

Definition 7.3.2. ***Control Graph.*** *Define the directed graph $G = (V, E)$ that has adjacency matrix $A = [a_{ij}]$. The edges are directed arcs between each node and its neighbors in the mesh as defined by $a_{ij} > 0$ if (j, i) is an edge of G. We refer to the graph $G = (V, E)$ as the control graph for the heat flow within the space serviced by the HVAC system. Let*

N be its total number of nodes. Then $d_i = \sum_{j \in N_i} a_{ij}$ is the in-degree of node i. Define $D = diag\{d_i\}$ and the control graph Laplacian as $L = D - A$.

Note that L has row sum of zero. The so-called pinning gains [50, 49] in (7.12) are $b_i \geqslant 0$, with $b_i > 0$ for at least one i. Then $b_i \neq 0$ if and only if there exists an arc from the control node to the i-the node in graph G. We refer to the nodes i for which $b_i \neq 0$ as the controlled or pinned nodes.

For notational convenience, we assume now that the node sets and the node enumerations of the modeling graph G_m and the control graph G are the same, that is $V = V_m$. However, the edge sets $E,\ E_m$ and the node neighborhoods $N_{m,i}$ and N_i are generally not the same.

Take the control input protocols for heat flow dynamics (7.8) as

$$u_i = ce_i + cb_i\left(\tau_0 - \tau_i\right) = c\left[\sum_{j \in N_i} a_{ij}(\tau_j - \tau_i) + b_i(\tau_0 - \tau_i)\right] + cb_i\left(\tau_0 - \tau_i\right) \tag{7.13}$$

with $c > 0$ a scalar gain. Then one has the closed-loop dynamics

$$\dot{\tau}_i = \sum_{j \in N_{m,i}} \alpha_{ij}(\tau_j - \tau_i) + cQ_i e_i + cQ_i b_i\left(\tau_0 - \tau_i\right) + d_i \tag{7.14}$$

$$\dot{\tau}_i = \sum_{j \in N_{m,i}} \alpha_{ij}(\tau_j - \tau_i) + cQ_i\left[\sum_{j \in N_i} a_{ij}(\tau_j - \tau_i) + 2b_i(\tau_0 - \tau_i)\right] + d_i \tag{7.15}$$

The overall dynamics are given by

$$\begin{aligned} \dot{\tau} &= -L_m\tau + cQe + d \\ &= -L_m\tau - cQ(L + 2B)\tau + cQ(L + B)\underline{1}\tau_0 + d \end{aligned} \tag{7.16}$$

$$\dot{\tau} = -L_m\left(\tau - \underline{1}\tau_0\right) - cQ(L + 2B)\left(\tau - \underline{1}\tau_0\right) + d \tag{7.17}$$

Define disagreement vector $\delta = \left(\tau - \underline{1}\tau_0\right)$. Now (7.17) goes into the form

$$\dot{\tau} = -\left(L_m + cQB\right)\delta - cQ(L + B)\delta + d \tag{7.18}$$

where $\underline{1} \in R^N$ is the vector of ones. One has used the fact that $L\underline{1} = 0$, that is L has row sum of zero.

This equation is not the standard form considered in cooperative control because it has two Laplacian matrices, with L_m defined by the nearest neighbor nodes in the dynamic heat model graph, and L defined by the control node neighborhoods selected. Note that L_m corresponds to an undirected graph and so is symmetric and positive definite. On the other hand, L corresponds to the control protocols, which normally generate a directed graph. The terms $-L_m\tau + d$ express the changes in node temperatures due to heat flows between the nodes and heat generating or dissipating disturbances, while terms $-c\beta(L+B)\tau + c\beta(L+B)\underline{1}\tau_0$ express the changes in node temperatures due to the control protocols (7.13).

7.4 Calculating the Airflow at Nodes: Pipe Network Analysis

For a pipe network shown in the HVAC system in Fig. 7.1, there are several methods are available to obtain the flow rate in a particular node or pipe [117]. The most common methods include loop method, node method and the element method. All these methods consider different hydraulic parameters applicable to gas or water and then solve a set of nonlinear equations mostly by iterative procedures. In this paper we use Hardy-Cross iterative procedure [117] to find out the numerical value of Q_i. For the particular example we considered in this paper and based on the model shown in Fig. 7.1, we can redraw the pipe network as shown in Fig. 7.3.

7.4.1 Pipe Network Geometry

The water distribution networks have mainly the following three types of configurations:

> Branch or Tree like configuration

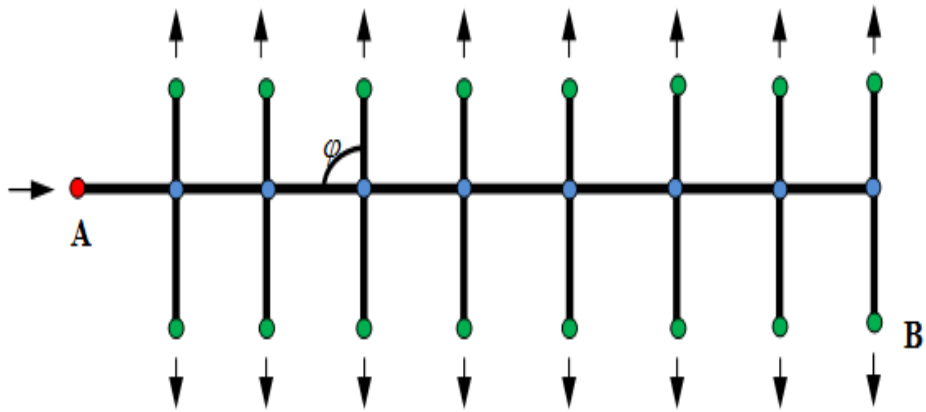

Figure 7.3. A typical pipe network for HVAC system in a room.

> Loped configuration

> Branch and looped configuration

One can clearly identify that the proposed pipe network configuration belongs to branch or tree network. A branched network or a tree network configuration is a system with no loops.

7.4.2 Analysis of Distribution Mains

A pipeline in which there are multiple withdrawals is called distribution mains. In a distribution main air flows on account of pumping with withdrawals or emissions at different nodal points. For example, in Fig. 7.3, the red node represents the nod with pump and the green nodes represent the air or gas withdrawals or emissions in the room Now we consider two distant points (including the pump node A) (A, B) in Fig. 7.3 to calculate the flow at node B. The equivalent pipe network can be seen as Fig. 7.4. The pipe flow is analyzed by using the continuity equation and the equation of motion. The continuity equation for steady flow in a circular pipe of diameter R is

$$Q = \frac{\pi}{4} R^2 V \tag{7.19}$$

where V =average velocity of flow, and Q = volumetric rate flow, called discharge. The equation of motion for steady flow is where z_A and z_B = elevations of the centerline of the pipe (from arbitrary datum), h_A and h_B pressure heads, V_A and V_B = average velocities at nodes A and B respectively, g =gravitational acceleration, and h_L^{A-B} = head loss between A and B. The head loss h_L^{A-B} is composed of two parts: h_f = head loss on account of surface resistance (also called friction loss), and h_m = head loss due to form resistance, which is the head loss on account of change in shape of the pipe line (also called minor loss). Thus

$$h_L^{A-B} = h_f + h_m \tag{7.20}$$

The term $z + h$ is called the piezometric head.

The head loss on account of surface resistance is calculated using Darcy-Weishbach equation

$$h_f = \frac{\sigma L V^2}{2gR} \tag{7.21}$$

where L =the pipe length, and σ =coefficient of surface resistance, traditionally known as friction factor. Eliminating V from (7.21) using (7.19) we get

$$h_f = \frac{8\sigma L Q^2}{\pi^2 g D^5} \tag{7.22}$$

The value of σ depends on two factors such as 1) average height of roughness projection ε, of the pipe wall, and 2) Reynolds number $\mathcal{R}$ of the flow, defined as

$$\mathcal{R} = \frac{VR}{\nu} \tag{7.23}$$

where v =kinematic velocity of fluid.

For turbulent flow ($\mathcal{R} \geqslant 4000$), Colebrook (1938) found the following implicit equation for σ:

$$\sigma = 1.325\left[\ln\left(\frac{\varepsilon}{3.7R} + \frac{2.51}{\mathcal{R}\sqrt{\sigma}}\right)\right]^{-2} \tag{7.24}$$

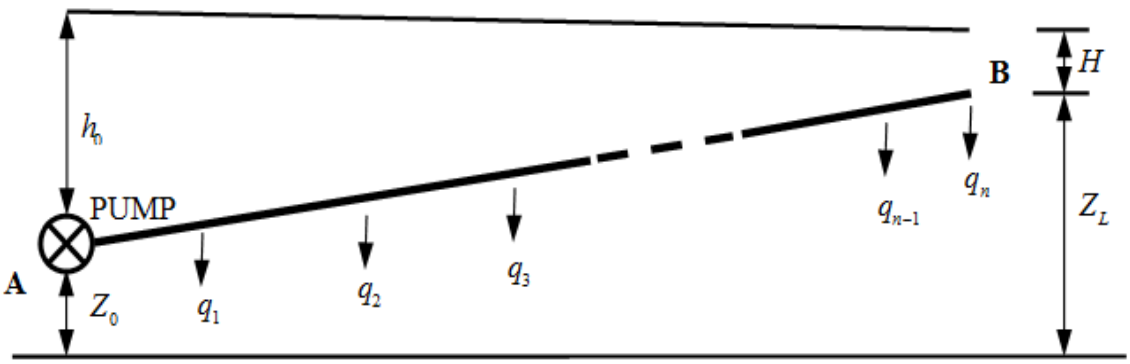

Figure 7.4. A typical pumping distribution main.

For laminar flow ($\mathcal{R} \leqslant 2000$), depends on $\mathcal{R}$ only and is given by Hegan-Poiseuille equation

$$\sigma = \frac{64}{\mathcal{R}} \tag{7.25}$$

For $2000 < \mathcal{R} < 4000$, no information is available for estimating σ. Swamee (1993) gave the following equation for σ valid for any arbitrary $\mathcal{R}$:

$$\sigma = \left\{ \left(\frac{64}{\mathcal{R}}\right)^8 + 9.5\left[\ln\left(\frac{\varepsilon}{3.7R} + \frac{5.74}{\mathcal{R}^{0.9}}\right) - \left(\frac{2500}{\mathcal{R}}\right)^6\right]^{-16} \right\}^{0.125} \tag{7.26}$$

The form-resistance losses given by h_m are due to pipe bends, elbows, valves etc. It is expressed in the following form

$$h_m = k_f \frac{V^2}{2g} \tag{7.27}$$

or its equivalent form

$$h_m = k_f \frac{8Q^2}{\pi^2 g R^4} \tag{7.28}$$

where k_f is the form-loss coefficient. For our case we are assuming that the pipe bends are sharp enough while calculating the distance between two nodes and is expressed in the following form

$$k_f = 0.442\varphi^{2.17} \tag{7.29}$$

where φ =elbow angle in radians as shown in Fig. 7.3. From Fig. 7.4, consider a distribution main with withdrawals $q_1, q_2, q_3, \cdots, q_n$ at nodal points $1, 2, 3, \cdots, n$. To find out the flow rate at each node one is required to find the nodal heads $h_1, h_2, h_3, \cdots, H$. The discharge flowing in the ith pipe link Q_i is given by

$$Q_i = \sum_{k=0}^{i} q_{n-k} \tag{7.30}$$

and the nodal head h_i is given by

$$h_i = h_0 + z_0 - z_i - \frac{8}{\pi^2 g} \sum_{k=1}^{i} \left(\frac{\sigma_k L_k}{R_k} + k_{fk} \right) \frac{Q_k^2}{R_k^4} \tag{7.31}$$

where the suffix k stands for k^{th} pipe link. The value of σ_k is given by [[117]

$$\sigma_k = 1.325 \left\{ \ln \left[\frac{\varepsilon}{3.7R} + 4.618 \left(\frac{\nu R_k}{Q_k} \right)^{0.9} \right] \right\}^{-2} \tag{7.32}$$

In typical branch network like Fig. 7.3, the pipe discharges can be obtained can be obtained by adding the nodal discharges and tracing the path from tail end to the input point until all the tail ends are covered. The nodal heads can be found by proceeding from the input point and adding the head losses in each link until a tail end (such as node B) is reached. The flow problem is defined as below:

- Airflow through pipes are known and head loss are unknown
- Airflow rate is unknown and head loss is known

In our case we assume the second condition is true i.e. the pressure head loss is known at each end of the pipe network and then by (7.22) we can calculate the flow rate Q_i (ignoring form loss or minor loss).

Remark 7.4.1. *If the pipe network is looped configuration then the flow problem is solved by Hardy-Cross method [118] or Newton-Raphson method or Linear Theory method [119].*

An M-matrix is a square matrix having nonpositive off-diagonal elements and all principal minors nonnegative.

Lemma 7.5.1. *[53] Let L be irreducible and B have at least one diagonal entry $b_i > 0$. Then $(L+B)$ is a nonsingular $M-$matrix. Define*

$$s = \begin{bmatrix} s_1 & s_2 & \cdots & s_N \end{bmatrix}^T = (L+B)^{-1}\underline{1} \tag{7.33}$$

$$P = diag\left\{p_i\right\} \equiv diag\left\{1/s_i\right\} \tag{7.34}$$

Then $P > 0$ and the matrix defined as

$$S = P\left(L+B\right) + \left(L+B\right)^T P \tag{7.35}$$

is positive definite.

Lemma 7.5.2. *Let the graph is strongly connected and $B \neq 0$. Then*

$$\left\|\delta\right\| \leqslant \left\|e\right\| / \underline{\sigma}\left(L+B\right),\ k = 1, 2 \tag{7.36}$$

with $\underline{\sigma}\left(L+B\right)$ the minimum singular value of $(L+B)$, and $e = 0$ if and only if the nodes synchronize, that is

$$\tau_i(t) = \tau_0(t), \forall i \tag{7.37}$$

Definition 7.5.3. *The time vector function $y(t)$ is said to be uniformly ultimately bounded (UUB) [78] if there exist a compact set $\Omega \subset R^N$ so that $\forall y(t_0) \in \Omega$ there exist a bound B_m and a time $t_f(B, y(t_0))$ such that $\left\|x(t) - y(t)\right\| \leqslant B\ \forall t \geqslant t_0 + t_f$.*

Definition 7.5.4. *The time vector function $\tau_0(t)$ is said to be cooperative uniformly ultimately bounded (CUUB) if there exist a compact set $\Omega \subset R$ so that $\forall\ (\tau_i(t_0) - \tau_0(t_0) \in \Omega$ there exist a bound B_m and a time $t_f\left(B_m, \tau(t_0), \tau_0(t_0)\right)$ such that $\left\|\tau_i(t) - \tau_0(t)\right\| \leqslant B_m\ \forall i,\ \forall t \geqslant t_0 + t_f$.*

The next Theorem shows the decentralized control for HVAC systems.

Theorem 7.5.5. ***Distributed Control Protocol for HVAC thermal synchronization*** *Consider the networked systems given by* (7.8) *and* (7.30)*. Select the control signal* $u(t)$ *in* (7.10) *as*

$$u_i = ce_i + cb_i\left(\tau_0 - \tau_i\right) = c\left[\sum_{j\in N_i} a_{ij}(\tau_j - \tau_i) + b_i(\tau_0 - \tau_i)\right] + cb_i\left(\tau_0 - \tau_i\right) \tag{7.38}$$

Then the cooperative neighborhood error$e(t)$ *defined by* (7.12) *is UUB, with a practical bound given by* (7.46)*. Moreover the consensus variable* $\tau_0(t)$ *is cooperative UUB and all nodes synchronize such that* $\|\tau_i(t) - \tau_0(t)\| \to 0$*. Moreover, the bound can* (7.46) *be made small by manipulating the control gain parameters.*

Proof. Consider a Lyapunov candidate

$$V_L = \frac{1}{2}e^T P e \tag{7.39}$$

Differentiating both sides we get

$$\begin{aligned} \dot{V}_L &= e^T P \dot{e} \\ &= e^T P\left(\dot{\tau} - \dot{\tau}_0\right) \end{aligned} \tag{7.40}$$

From (7.18), (7.11) and (7.40)

$$\begin{aligned} \dot{V}_L = &-e^T P\left(L+B\right)\left\{-\left(L_m + cQB\right)\delta - cQ(L+B)\delta + d\right\} + e^T P\left(L+B\right) f_0\left(\tau_0, t\right) \\ &= -e^T P\left(L+B\right)\left(L_m + cQB\right)\left(L+B\right)^{-1} e - e^T P\left(L+B\right) cQe \\ &\quad -e^T P\left(L+B\right) d + e^T P\left(L+B\right) f_0\left(\tau_0, t\right) \end{aligned} \tag{7.41}$$

Now using [55] one can write

$$\begin{aligned} \dot{V}_L \leqslant &-\sigma_{\min}\left(L_m + cQB\right) tr\left[e^T P\left(L+B\right)\left(L+B\right)^{-1} e\right] - \sigma_{\min} Q tr\left[e^T P\left(L+B\right) ce\right] \\ &+ \left\|e^T P\left(L+B\right) d\right\| + \left\|e^T P\left(L+B\right) f_0\left(\tau_0, t\right)\right\| \end{aligned} \tag{7.42}$$

where, $\sigma_{\min}(\cdot)$ denotes the minimum singular value. Now from Lemma 7.5.1, as $(L+B)$ is a irreducible diagonally dominant M-matrix, there exist a diagonal matrix $P > 0$ and a positive definite matrix S such that $S = P(L+B) + (L+B)^T P$. Also, L_m is a symmetric matrix with diagonal entries all positive. Therefore, $(L_m + cQB)$ is diagonally dominant symmetric matrix and thus $(L_m + cQB)$ is positive definite or $\sigma_{\min}(L_m + cQB) \geq 0$. Continuing from (7.42)

$$\begin{aligned} \dot{V}_L \leqslant & -\sigma_{\min}(L_m + cQB)\, tr\left[e^T P e\right] - \frac{1}{2} c\sigma_{\min} Q tr\left[e^T S e\right] \\ & + \left\| e^T P (L+B)\, d \right\| + \left\| e^T P (L+B)\, f_0(\tau_0, t) \right\| \end{aligned} \tag{7.43}$$

Now taking 2 norm in right hand side and using Equations (9.3.12, 9.6.14, 9.6.16) from [55]

$$\begin{aligned} \dot{V}_L \leqslant & -\sigma_{\min}(L_m + cQB)\, \sigma_{\min}(P) \|e\|^2 - \frac{1}{2} c\sigma_{\min}(Q)\, \sigma_{\min}(S) \|e\|^2 \\ & + \sigma_{\max}(P)\, \sigma_{\max}(L+B)\, d_M \|e\| + + \sigma_{\max}(P)\, \sigma_{\max}(L+B)\, F_M \|e\| \end{aligned} \tag{7.44}$$

$$\begin{aligned} \dot{V}_L \leqslant & -\left\{ \sigma_{\min}(L_m + cQB)\, \sigma_{\min}(P) + \frac{1}{2} c\sigma_{\min}(Q)\, \sigma_{\min}(S) \right\} \|e\|^2 \\ & + \left\{ \sigma_{\max}(P)\, \sigma_{\max}(L+B)\, d_M + \sigma_{\max}(P)\, \sigma_{\max}(L+B)\, F_M \right\} \|e\| \end{aligned} \tag{7.45}$$

where, $\sigma_{\max}(\cdot)$ denotes the maximum singular value, $\|d\|_2 \leqslant d_M$ and $\|f_0\| \leqslant F_M$. Therefore, $\dot{V}_L \leqslant 0$ if

$$\|e\| > \frac{\sigma_{\max}(P)\, \sigma_{\max}(L+B)\, d_M + \sigma_{\max}(P)\, \sigma_{\max}(L+B)\, F_M}{\sigma_{\min}(L_m + cQB)\, \sigma_{\min}(P) + \frac{1}{2} c\sigma_{\min}(Q)\, \sigma_{\min}(S)} \tag{7.46}$$

Thus $\dot{V}_L$ is negative outside a compact set. Selecting the control gain ensures that the compact set defined by (7.46) is bounded. This ensures UUB for $e(t)$. Again from Lemma 7.5.2, disagreement vector $\delta(t)$ is bounded above by $e(t)$. This ensures that at steady state the temperature at each node inside the building is matched with the desired temperature τ_0. □

7.6 Conclusion

In this Chapter decentralized control for HVAC system is discussed. The model of HVAC system is derived from the basic poison heat equations and air flow dynamics in pipe network. The mathematical modeling is also include selection of neighborhood for each node associated for control purpose. Then the pinning control topology is introduced. It is shown from Lyapunov perspective that the thermal synchronization is possible among the nodes inside the building by choosing a suitable control input.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Contribution

This work makes two major contributions. **First** in the field of aerospace control, it develops new adaptive and nonlinear autopilots for quadrotor helicopter. These new approaches not only show improved performances in comparison of the past works but also deliver the key aspects of implementation in practical model. **Second** in the field of cooperative graph and multiagent systems, it develops nonlinear adaptive control for synchronization of heterogeneous multiagent systems where nonlinear models of the agents are unknown. It also shows the duality properties in cooperative graph network which has never been done before.

This research shows adaptive backstepping and dynamic inversion with internal dynamics stabilization techniques for quadrotor. Quadrotor helicopter models usually show to main characteristics. Firstly strong coupling among the system states and secondly underactuation which more states are to be controlled with fewer control inputs. Due to these unique characteristics, the design of stabilizing control inputs are always challenging for quadrotor models. Dynamic inversion is a well known nonlinear control technique often used for stabilizing square nonlinear systems. But applying the same for an underactuated system might results unstable internal dynamics which ultimately destabilize the whole system. To avoid the problem a new approach of stabilizing internal dynamics of a quadrotor is developed so that dynamic inversion technique can easily be applied to complex nonlinear underactuated systems such as quadrotor helicopter. The overall control input is chosen from Lyapunov based proof which guarantees the

boundedness of the system states. If the quadrotor models are expressed in the form of Lagrangian equations then designing control inputs become more challenging. Such models are bilinear in states. The backstepping approach is generally not applicable for bilinear systems. This research shows how to implement nonlinear adaptive backstepping techniques for Lagrangian quadrotor models. The unmodeled dynamics along with unknown disturbances are estimated with neural nets. First attitude control loop is stabilized and then stepping back from inner attitude control loop, the outer position loop is stabilized. The problem of bilinearity is solved by kinematic inversion technique. The overall control input is chosen from Lyapunov based technique to ensure the stability of the system.

Coordination among multiple agents is generally done based on balanced or bi-directed communication graph models. If the agents are nonlinear and passive then for balanced graph model synchronization is possible. But, for other than balanced and bi-directed graph models, it is difficult to prove the same. Moreover, the performance of the synchronization is dependent on the second largest eigenvalue of the laplacian matrix of the network graph. This eigenvalue is also known as Fiedler eigenvalue. This research shows how to implement nonlinear and adaptive control along with pinning topology for generalized communication graph models. The dynamics of the agent at each node are non-identical and unknown. A Lyapunov based technique is adopted to show the tracking performance while the tracker dynamics is also considered not known. It is also shown that the synchronization speed no more depends on Fiedler eigenvalue of the graph laplacian matrix. The research also shows that the duality property of liner controller and observer exists for general cooperative graph system with linear agent models. The choice of gains for controller and observer from riccati equations ensures the optimality and generality of the design. Finally the research is extended to decentralized control of HVAC systems using pinning control methodology.

8.2 Future work

Implementation of nonlinear adaptive control in practical plants is most challenging. In spite of many breakthrough publications in the field of adaptive control techniques, the implementation of the same in a practical sense is quite rare. This is mostly because of unrealistic control command generated from adaptive law or requirement of heavy bandwidth in communication channel for multiagent systems. Use of the methods proposed in this thesis in practical plants is the most relevant future work before going further in theoretical aspect. Also, the simulation and practical implementation of decentralized control for HVAC system is also important in that perspective.

REFERENCES

[1] P. Castillo, R. Lozano, and A. Dzul, *Modelling and Control of Mini Flying Machines.* Springer-Verlag, 2005.

[2] A. J. Calise and R. T. Rysdyk, "Nonlinear adaptive flight control using neural networks," *IEEE Control Systems*, vol. 8, pp. 14–25, 1998, 6.

[3] T. J. Koo and S. Sastry, "Output tracking control design of a helicopter model based on approximate linearization," in *Proceedings of the 37th Conference on Decision and Control.* Tampa, Florida: IEEE, 1998.

[4] V. Gavrilets, B. Mettler, and E. Feron, "Dynamic model for a miniature aerobatic helicopter," *MIT-LIDS report*, vol. LIDS-P-2580, 2003.

[5] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation.* Wiley and Sons, 2003.

[6] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington DC, Virginia, June 2002.

[7] B. Bijnens, Q. P. Chu, G. M. Voorsluijs, and J. A. Mulder, "Adaptive feedback linearization flight control for a helicopter UAV," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, 2005.

[8] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.

[9] V. Mistler, A. Benallegue, and N. K. M'Sirdi, "Exact linearization and non-interacting control of a 4 rotors helicopter via dynamic feedback," in *10th IEEE Int. Workshop on Robot-Human Interactive Communication*, Paris, 2001.

[10] A. Mokhtari, A. Benallegue, and Y. Orlov, "Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle," *International Journal of Robotics and Automation*, vol. 21, pp. 39–49, 2006, 1.

[11] I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse, "Systematic design of adaptive controllers for feedback linearizable systems," *IEEE Transaction of Automatic Control*, vol. 36, pp. 1241–1253, 1991.

[12] H. K. Khalil, *Nonlinear Systems.* Prentice Hall, 1996.

[13] J. Slotine and W. Li, *Applied Nonlinear Control.* Prentice Hall, 1991.

[14] H. J. Kim, D. H. Shim, and S. Sastry, "Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles," in *Proceedings of the 37th Conference on Decision and Control*, Anchorage, AK, 2002.

[15] J. V. R. Prasad and A. J. Calise, "Adaptive nonlinear controller synthesis and flight evaluation on an unmanned helicopter," in *Proceedings of the 1999 IEEE International Conference on Control Applications*, Kohala Coast-Island of Hawaii, USA, 1999.

[16] A. J. Calise, B. S. Kim, J. Leitner, and J. V. R. Prasad, "Helicopter adaptive flight control using neural networks," in *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, FL, 1994.

[17] N. Hovakimyan, R. Rysdyk, and A. Calise, "Dynamic neural networks for output feedback control," *International Journal of Robust and Nonlinear Control*, vol. 11, pp. 23–39, 2001, 1.

[18] R. Rysdyk and A. J. Calise, "Robust nonlinear adaptive flight control for consistent handling qualities," *IEEE Transaction of Control System Technology*, vol. 13, pp. 896–910, 2005, 6.

[19] K. A. Wise, J. S. Brinker, A. J. Calise, D. F. Enns, M. R. Elgersma, and P. Voulgaris, "Direct adaptive reconfigurable flight control for a tailless advanced fighter aircraft," *International Journal of Robust and Nonlinear Control*, vol. 9, pp. 999–1012, 1999, 14.

[20] M. B. McFarland and A. J. Calise, "Nonlinear adaptive control of agile Anti-Air missiles using neural networks," *IEEE Transaction of Control System Technology*, vol. 8, pp. 749–756, 2000, 5.

[21] J. Campos, F. L. Lewis, and C. R. Selmic, "Backlash compensation in discrete time nonlinear systems using dynamic inversion by neural networks," in *Proceedings of the 2000 IEEE lntemational Conference on Robotics & Automation*, San Francisco, CA, 2000.

[22] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a Four-Rotor mini rotorcraft," *IEEE Transaction on Control System Technology*, vol. 12, pp. 510–516, 2004, 4.

[23] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft having four rotors," *IEEE Control System Magazine*, vol. 25, pp. 45–55, 2005, 6.

[24] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *International Conference on Intelligent Robots and Systems*, vol. 3. Sendal, Japan: IEEE, 2004, pp. 2451–2456.

[25] R. Mahony and T. Hamel, "Robust trajectory tracking for a scale model autonomous helicopter," *Int. J. Robust Nonlinear Control, John Wiley & Sons*, vol. 14, p. 1035âĂŞ1059, 2005.

[26] C. D. Yang and W. H. Liu, "Nonlinear h_∞ decoupling hover control of helicopter with parameter uncertainties," *Proceedings of the American Control Conference*, 2003.

[27] R. Enns and J. Si, "Helicopter flight control design using a learning control approach1," in *Proceedings of the 39th Conference on Decision and Control*, Sydney, Australia, 2000.

[28] E. Johnson and S. Kannan, "Adaptive trajectory control for autonomous helicopters," *AIAA Journal of Guidance, Control and Dynamics*, vol. 28, pp. 524–538, June 2005.

[29] J. Farrell, M. Sharma, and M. Polycarpou, "Backstepping-Based flight control with adaptive function approximation," *AIAA Journal of Guidance, Control and Dynamics*, vol. 28, 2005, 6.

[30] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modeling and configuration stabilization for an x4-flyer," in *IFAC 15th Triennial World Congress*, Barcelona, Spain, 2002.

[31] M. Cheng and M. Huzmezan, "A combined MBPC/2 DOF hinf controller for quad rotor unmanned air vehicle," in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Austin, Texas, 2003.

[32] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an autonomous controller for a quadrotor UAV," in *Proceedings of the European Control Conference*, Kos, Greece, 2007.

[33] W. Guo and J. Horn, "Modeling and simulation for the development of a quadrotor UAV capable of indoor flight," in *Modeling and Simulation Technologies Conference and Exhibit*, Keystone, Colorado, 2006.

[34] R. Mahony, T. Hamel, and A. Dzul, "Hover control via lyapunov control for an autonomous model helicopter," in *Proceedings of the 38th Conference on Decision & Control*, Phoenix, Arizona, 1999.

[35] Q. Hui and W. Haddad, "Distributed nonlinear control algorithms for network consensus," *Automatica*, vol. 44, pp. 2375–2381, 2008.

[36] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[37] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and Time-Delays," *IEEE Transaction of Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[38] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automatic Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[39] A. Jadbabaie, J. Lin, and S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[40] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[41] T. Vicsek, A. Czirok, E. B. Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," *Phys. Rev. Lett*, vol. 75, pp. 1226–1229, 1995.

[42] Y. Kuramoto, "Self-entrainment of a population of coupled non-linear oscillators," in *International Symposium on Mathematical Problems in Theoretical Physics*, ser. Lecture Notes in Physics. Springer Berlin / Heidelberg, 1975, vol. 39, pp. 420–422.

[43] S. Strogatz, "From kuramoto to crawford: exploring the onset of synchronization in populations of coupled oscillators," *Physica D*, vol. 143, pp. 1–20, 2000.

[44] N. Chopra and M. Spong, "On exponential synchronization of kuramoto oscillators," *IEEE Trans. Automatic Control*, vol. 54, no. 2, pp. 353–357, 2009.

[45] N. Chopra and M. W. Spong, "Passivity-Based control of Multi-Agent systems," in *Advances in Robot Control.* Springer Berlin Heidelberg, 2006, pp. 107–134.

[46] T. Jiang and J. Baras, "Graph algebraic interpretation of trust establishment in autonomic networks," *Preprint Wiley Journal of Networks*, 2009.

[47] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *IFAC World Congress*, 2005.

[48] J. Lu and G. Chen, "A time-varying complex dynamical network model and its controlled synchronization criteria," *IEEE Trans. Automatic Control*, vol. 50, no. 6, pp. 841–846, 2005.

[49] X. F. Wang and G. Chen, "Pinning control of scale-free dynamical networks," *Physica A*, vol. 310, no. 3, pp. 521–531, 2002.

[50] X. Li, X. Wang, and G. Chen, "Pinning a complex dynamical network to its equilibrium," *IEEE Trans. Circuits and Systems*, vol. 51, no. 10, pp. 2074–2087, 2004.

[51] Z. Li, Z. Duan, and G. Chen, "Consensus of multi-agent systems and synchronization of complex networks: a unified viewpoint," *IEEE Trans. Circuits and Systems*, 2009.

[52] J. Han, M. Li, and L. Guo, "Soft control on collective behavior of a group of autonomous agents by a shill agent," *J. System Sci. & Complexity*, vol. 19, pp. 54–62, 2006.

[53] Z. Qu, *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles.* New York: Springer-Verlag, 2009.

[54] C. Wu, *Synchronization in Complex Networks of Nonlinear Dynamical systems.* Singapore: World Scientific, 2007.

[55] D. Bernstein, *Matrix Mathematics.* NJ: Princeton Univ. Press, 2005.

[56] S. Khoo, L. Xie, and Z. Man, "Robust Finite-Time consensus tracking algorithm for multirobot systems," *IEEE Transaction on Mechatronics*, vol. 14, no. 2, pp. 219–228, 2009.

[57] Z. Hou, L. Cheng, and M. Tan, "Decentralized robust adaptive control for the multiagent system consensus problem using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 39, no. 3, pp. 636–647, June 2009.

[58] W. Ren, K. L. Moore, and Y. Chen, "High-order and model reference consensus algorithms in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, pp. 678–688, 2007.

[59] Y. Bo and F. Huajing, "Second-Order consensus in networks of agents with delayed dynamics," *Journal of Natural Sciences*, vol. 14, no. 2, pp. 158–162, 2009.

[60] W. Yang, A. L. Bertozzi, and X. Wang, "Stability of a second order consensus algorithm with time delay," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2008.

[61] H. Su and X. Wang, "Second-Order consensus of multiple agents with coupling delay," in *Proceedings of the 7th World Congress on Intelligent Control and Automation*, Chongqing, China, June 2008.

[62] J. H. Seo, H. Shima, and J. Back, "Consensus of high-order linear systems using dynamic output feedback compensator: Low gain approach," *Automatica*, vol. 45, no. 11, pp. 2659–2664, 2009.

[63] S. Baskaya and E. Eken, "Investigation of room air flow regarding effects of occupants, inlet/outlet locations, inlet velocity, and winter/summer conditions," *International communications in heat and mass transfer, Elsevier, New York*, vol. 30, no. 8, pp. 1147–1156, 2003.

[64] J. Niu and J. Kooi, "Indoor climate in rooms with cooled ceiling systems," *Building and Environment*, vol. 29, no. 3, pp. 283–290, July

1994. [Online]. Available: http://www.sciencedirect.com/science/article/B6V23-47YMWXY-4X/2/072689b1cc5f37d927d27db0a93c9e49

[65] H. B. Awbi and A. Hatton, "Mixed convection from heated room surfaces," *Energy and Buildings*, vol. 32, no. 2, pp. 153–166, July 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6V2V-40797S7-3/2/5dfcf57fd191344d11652f6553fd5140

[66] S. L. Sinha, R. C. Arora, and S. Roy, "Numerical simulation of two-dimensional room air flow with and without buoyancy," *Energy and Buildings*, vol. 32, no. 1, pp. 121–129, June 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6V2V-3YYTP50-F/2/9d7abed554c2ddfea0fe9f68a50133fc

[67] T. Yamamoto, D. S. Ensor, and L. E. Sparks, "Evaluation of ventilation performance for indoor space," *Building and Environment*, vol. 29, no. 3, pp. 291–296, July 1994. [Online]. Available: http://www.sciencedirect.com/science/article/B6V23-47YMWXY-4Y/2/883df03bc57e3bc47bcf3b7781d67574

[68] W. K. Chow and L. T. Wong, "Local air speeds measurement in mechanically ventilated spaces," *Building and Environment*, vol. 34, no. 5, pp. 553–563, Sept. 1999. [Online]. Available: http://www.sciencedirect.com/science/article/B6V23-3W9KYVP-3/2/2dc73dbeefb376efd1d4ef3636852c8d

[69] J. C. Lam and A. L. S. Chan, "CFD analysis and energy simulation of a gymnasium," *Building and Environment*, vol. 36, no. 3, pp. 351–358, Apr. 2001. [Online]. Available: http://www.sciencedirect.com/science/article/B6V23-41XV9MK-N/2/2c0f9b8db94b56a9d3fd456496908498

[70] H. Xing, A. Hatton, and H. B. Awbi, "A study of the air quality in the breathing zone in a room with displacement ventilation," *Building and Environment*, vol. 36, no. 7, pp. 809–820, Aug. 2001. [On-

line]. Available: http://www.sciencedirect.com/science/article/B6V23-438CC6R-3/2/782efc81d2c09fa2198de85576ed522f

[71] J. Kim, M. West, E. Scholte, and S. Narayanan, "Multiscale consensus for decentralized estimation and its application to building systems," in *2008 American Control Conference*, Seattle, WA, 2008, pp. 888–893. [Online]. Available: http://lagrange.mechse.illinois.edu/mwest/pubs/KiWeScNa2008/

[72] A. Das and F. Lewis, "Dynamic inversion with zero-dynamics stabilization for quadrotor control," *IET Control Theory & Applications*, vol. 3, pp. 303–314, 2009.

[73] ——, "Backstepping approach for controlling a quadrotor using neural network," *Speciall Issue: Journal of Intelligent and Robotics Systems*, vol. 56, pp. 127–151, 2009.

[74] ——, "Distributed adaptive control for synchronization of unknown nonlinear networked systems," *Automatica*, vol. 46, pp. 2014–2021, 2010.

[75] ——, "Distributed adaptive control for synchronization of second order systems with unknown nonlinearities," *International Journal of Robust and Nonlinear Control*, vol. 21, pp. 1493–1610, 2010.

[76] H. Zhang, F. Lewis, and A. Das, "Design for synchronization of cooperative systems: State feedback, observer, and output feedback," *IEEE Transaction of Automatic Control*, vol. 56, pp. 1948–1952, 2011.

[77] H. K. Khalil, *Nonlinear Systems.* New York: Macmillan, 1992.

[78] F. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems.* London: Taylor and Francis, 1999.

[79] N. Hogan, "Adaptive control of mechanical impedance by coactivation of antagonist muscles," *IEEE Transaction of Automatic Control*, vol. 29, pp. 681–690, 1984.

[80] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of Neuro Science*, vol. 5, pp. 1688–1703, 1985.

[81] A. Mokhtari, A. Benallegue, and B. Daachi, "Robust feedback linearization and gh_∞ controller for a quadrotor unmanned aerial vehicle," *Journal of Electrical Engineering*, vol. 57, 2006.

[82] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice.* CRC Press, 2004.

[83] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Networks*, vol. 7, no. 2, pp. 388–399, 1996.

[84] Y. H. Kim and F. L. Lewis, *High Level Feedback Control with Neural Networks.* World Scientific, 1998, vol. 21.

[85] R. Olfati-Saber, "Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles," Thesis (Ph.D.), Massachusetts Institute of Technology, 2001.

[86] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximations," *Neural Networks*, vol. 20, pp. 359–366, 1989.

[87] J. Zhu, Y. Tian, and J. Kuang, "On the general consensus protocol of multi-agent systems with double-integrator dynamics," *Linear Algebra and its Applications*, vol. 431, pp. 701–715, 2009.

[88] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators.* Upper Saddle River, NJ, USA: Prentice Hall, 1993.

[89] S. Ge and C. Wang, "Adaptive neural control of uncertain MIMO nonlinear systems," *IEEE Trans. Neural Networks*, vol. 15, no. 3, pp. 674–692, May 2004.

[90] M. H. Stone, "The generalized weierstrass approximation theorem," *Mathematics Magazine*, vol. 21, no. 4, pp. 167–184, 1948.

[91] "Robustness of a neural controller in the presence of additive and multiplicative external perturbations," Patras, Greece.

[92] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications.* London:Spring Verlag, 2008.

[93] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[94] A. Das and F. L. Lewis, "Distributed adaptive control for synchronization of unknown nonlinear networked systems," *Submitted to Automatica*, Dec. 2009.

[95] H. Zhang, M. Chen, G. Stan, and J. Maciejowski, "Collective behavior coordination with predictive mechanisms," *IEEE Circuits and Systems Magazine*, pp. 67–85, 2008.

[96] H. Yang, Z. Zhang, and S. Zhang, "Consensus of second-order multi-agent systems with exogenous disturbances," *Int. J. Robust and Nonlinear Control*, vol. 1, no. 6, pp. 1–12, 2002.

[97] M. Barahona and L. M. Pecora, "Synchronization in small-world systems," *Phys. Rev. Lett*, vol. 89, no. 5, p. 054101, 2002.

[98] L. Kocarev and P. Amato, "Synchronization in power-law networks," *Chaos*, vol. 15, p. 024101, 2005.

[99] A. E. Motter, C. S. Zhou, and J. Kurths, "Enhancing complex-network synchronization," *Europhys. Lett*, vol. 69, no. 3, p. 334âĂŞ340, 2005.

[100] L. M. Pecora and T. L. Carroll, "Master stability functions for synchronized coupled systems," *Phys. Rev. Lett*, vol. 80, pp. 2109âĂŞ–2112, 1998.

[101] C. S. Zhou and J. Kurths, "Dynamical weights and enhanced synchronization in adaptive complex networks," *Phys. Rev. Lett*, vol. 96, p. 164102, 2006.

[102] Z. Duan, G. Chen, and L. Huang, "Disconnected synchronized regions of complex dynamical systems," *IEEE Trans. Automatic Control*, vol. 54, no. 4, pp. 845–849, Apr. 2009.

[103] C. Liu, Z. Duan, G. Chen, and L. Huang, "Analyzing and controlling the network synchronization regions," *Physica A*, vol. 386, pp. 531–542, 2007.

[104] Y. Hong, G. Chen, and L. Bushnell, "Distributed observers design for leader-following control of multi-agent networks," *Automatica*, vol. 44, pp. 846–850, 2008.

[105] Y. Hong and X. Wang, "Multi-agent tracking of high-dimensional active leader with switching topology," *Preprint*, 2009.

[106] Y. Hong, X. Wang, and Z. Jiang, "Multi-agent coordination with general linear models: a distributed output regulation approach," in *Proc. Int. Conf. Control and Automation*, Xiamen, June 2010.

[107] G. Xie and L. Wang, "Consensus control for a class of networks of dynamic agents: switching topology," in *Proc. 2006 American Control Conference*, 2006, pp. 1382–1387.

[108] F. Lewis, *Applied Optimal Control and Estimation: Digital Design and Implementation*, ser. TI Series. New Jersey: Prentice-Hall, Feb. 1992.

[109] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans Circuits and Systems*, vol. 25, no. 9, pp. 772–781, 1978.

[110] F. Lewis and V. Syrmos, *Optimal Control*, 2nd ed. New York: John Wiley and Sons, 1995.

[111] F. Lewis, "A survey of linear singular systems," *Circuits, Systems, and Signal Proc*, vol. 5, no. 1, pp. 3–36, 1986.

[112] F. Lewis, L. Xie, and D. Popa, *Optimal & Robust Estimation: With an Introduction to Stochastic Control Theory*, 2nd ed. Boca Raton: CRC Press, 2007.

[113] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York: Springer Verlag, 2001.

[114] P. Antsaklis and A. Michel, *Linear Systems Primer*. Boston: Birkhauser, 2007.

[115] J. Baras and P. Hovareshti, "Effects of graph topology on performance of distributed algorithms for networked control and sensing," in *Proc. Workshop on Networked Distributed Systems for Intelligent Sensing and Control*, Kalamata, 2007.

[116] F. R. Gantmacher, *The Theory of Matrices*, ser. Matrix Theory. AMS Chelsea, 1959.

[117] P. K. Swamee and A. K. Sharma, *Design of Water Supply Pipe Networks*. Wiley Interscience, Jan. 2008.

[118] K. Y. Volokh, "On foundation of the hardy cross method," *International Journal of Solids and Structures*, vol. 39(16), pp. 4197–4200, 2002.

[119] D. J. Wood and C. O. A. Charles, "Hydraulic network analysis using linear theory," *J. Hydr. Div.*, vol. 98(7), pp. 1157–1170, 1972.

BIOGRAPHICAL STATEMENT

Abhijit Das was born in Kharagpur, India, in 1980. He received his B.E. degree from Bengal Engineering College, Shibpur, in 2003, his M.S. degree from Indian Institute of Technology, Kharagpur in 2006 and Ph.D. degrees from The University of Texas at Arlington in 2010, all in Electrical Engineering. From 2003 to 2006, during his masters degree he was involved with several projects with Defense Research and Development Organization (DRDO), India . In 2007, he joined , Automation and Robotics Research Institute as a Research Assistant to work with Frank Lewis, an eminent control engineer. His Ph.D. dissertation won Dean Dissertation Fellowship award in 2010. He is the author of 1 book, 3 book chapters and several journal and conference articles. He is life member of Systems Society of India, student member of AIAA, IEEE, SIAM. His profile is also appeared in Marquis Who's Who in America.