QUADCOPTER TRAJECTORY PREDICTION AND WIND ESTIMATION USING MACHINE LEARNING

By

SAM ALLISON

Bachelor of Science in Mechanical Engineering

Oklahoma State University

Stillwater, Oklahoma

2016

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2019

QUADCOPTER TRAJECTORY PREDICTION AND WIND ESTIMATION USING MACHINE LEARNING

Thesis Approved:

He Bai

Thesis Advisor

Balaji Jayaraman

Rushikesh Kamalapurkar

ACKNOWLEDGMENTS

I would like to thank my committee members for their invaluable input into this project, and in particular my advisor, Dr. He Bai. His guidance was instrumental in developing the work presented in this thesis. In addition, I would like to thank the members of the Coral and SCC research groups for the many helpful discussions on how to best accomplish this research.[1]

This work was supported in part by the OK NASA EPSCoR Research Initiation Grant, Oklahoma State University, and a subcontract from AFOSR SIRCUS project FA8650-15-D-1845.

[1]Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: SAM ALLISON

Date of Degree: JULY, 2019

Title of Study: QUADCOPTER TRAJECTORY PREDICTION AND WIND ESTIMATION USING MACHINE LEARNING

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: Small unmanned aerial systems are heavily impacted by wind disturbances. Wind causes deviations from desired trajectories, potentially leading to crashes. In this thesis, we consider two inherently related problems: predicting quadcopter trajectory deviations due to wind disturbances and estimating wind velocity based on quadcopter trajectory deviations. The former is addressed using linear difference equation identification as well as neural network (NN) modeling. Simulations validate the use of linear difference equation identification as a tool to predict trajectory deviations in crosswinds and machine learning (specifically, long short-term memory (LSTM) NNs) as an approach to predict trajectory deviations in multidimensional wind. We approach the wind estimation problem from a machine learning perspective due to easier generalization of the NN to multidimensional winds. As in the trajectory prediction case, we use LSTM NNs to identify a model. The trained NN is deployed to estimate the turbulent winds as generated by the Dryden gust model as well as a realistic large eddy simulation of a near-neutral atmospheric boundary layer over flat terrain. The resulting NN predictions are compared to a wind triangle approach that uses tilt angle as an approximation of airspeed. Results from this study indicate that the LSTM NN based approach results in lower errors in both the mean and variance of the local wind field as compared to the wind triangle approach.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 1
1. Motivation 1
2. Literature Review 4
3. Overview of the Proposed Approach 6
4. Contributions of Thesis 7

II. QUADCOPTER MODEL 9
1. Dynamics 9
2. Motor and Rotor Models 10
3. Controller 12
4. Simulation Environment 14

III. TRAJECTORY PREDICTION USING LINEAR DIFFERENCE EQUATIONS 17
1. Modeling the Quadcopter Response 18
2. Simulation Results 20
3. Conclusions 32

IV. TRAJECTORY PREDICTION USING A NEURAL NETWORK 33
1. Motivation for Nonlinear Modeling 33
2. Trajectory Prediction using a Neural Network 33
3. Simulation Results 36
4. Conclusions 40

V. WIND ESTIMATION USING NEURAL NETWORKS 43
1. The Wind Estimation Problem for a Quadcopter 44
2. Wind Estimation using a Neural Network 44
3. Simulation Results 46
4. Conclusions 58

VI. CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH 60

REFERENCES 62

LIST OF TABLES

Table Page

1. Quadcopter model parameters. 14

2. Comparison of mean absolute errors and error variances for a $[1, 2, 0]^T$ m/s mean Dryden wind with a quadcopter in hover. 54

3. Comparison of mean absolute errors and error variances for a $[2, -1, 0]^T$ m/s mean Dryden wind with a quadcopter in hover. 55

4. Comparison of mean absolute errors and error variances for a $[1, 2, 0]^T$ m/s mean Dryden wind with a quadcopter in straight line flight. . . . 55

5. Comparison of mean wind errors and covariance distances for a $[1, 2, 0]^T$ m/s mean Dryden wind with a quadcopter in hover. 56

6. Comparison of mean absolute errors and error variances for wind from a LES simulation of the ABL at three altitudes with a quadcopter in hover. 57

7. Comparison of mean wind errors and covariance distances for wind from a LES simulation of the ABL at three altitudes with a quadcopter in hover. 57

8. Comparison of mean errors and error variances for the direction and normalized speed of wind from a LES simulation of the ABL at three altitudes with a quadcopter in hover. 58

LIST OF FIGURES

Figure Page

1. General quadcopter model structure. 9
2. Comparison of the true trajectory and the fitted trajectories of different system orders ($n = 2, 3, 4$) for 4 m/s wind. 21
3. Comparison of the trajectory obtained from the least square fit with the true trajectory for 1 m/s wind. 21
4. Comparison of the trajectory obtained from the least square fit with the true trajectory for 7 m/s wind. 22
5. Comparison of the predicted trajectory and the true trajectory for 2 m/s wind. 22
6. Comparison of the predicted trajectory and the true trajectory for 6 m/s wind. 23
7. RMSE and maximum prediction errors for various wind speeds. . . . 23
8. Comparison of the identified system with the true trajectory for 4 m/s wind with the new control gain $\bar{K}$. 24
9. Comparison of the predicted trajectory and the true trajectory for 2 m/s wind with the new control gain $\bar{K}$. 24
10. RMSE and maximum prediction errors for various wind speeds with the new control gain $\bar{K}$. 25
11. Sample winds generated using Dryden turbulence model. 25

Figure Page

12. Comparison of the predicted trajectory and the true trajectory for a wind generated by the Dryden wind model with a 2 m/s constant component. 26

13. Comparison of the predicted trajectory and the true trajectory for a wind generated by the Dryden wind model with a 6 m/s constant component. 27

14. RMSE and maximum prediction errors for various winds generated by the Dryden wind model with different constant components. 27

15. RMSE and maximum prediction errors for various winds generated by the Dryden wind model for moderate turbulence with different constant components. 28

16. Comparison of the predicted trajectory and the true trajectory for 4 m/s wind in the Iris+ simulator. 29

17. Comparison of the predicted trajectory and the true trajectory for 2 m/s wind in the Iris+ simulator. 30

18. RMSE and maximum prediction errors for various constant winds using a 15th order identified model and the Iris+ simulator. 30

19. Comparison of the total errors obtained from $v_{w,e} = 1, \cdots, 5$ m/s with different orders of the DE in (III.1). 31

20. Decorrelation of the Simulink quadcopter in hover. 31

21. General form of a LSTM NN unit. 36

22. Training loss for the trajectory prediction NN trained on 1,800 seconds of data for the hover case. 38

23. Comparison of the true quadcopter velocity and the NN estimated quadcopter velocity using the test dataset for the hover case. 38

Figure Page

24. Quadcopter velocity prediction error histograms for the test dataset for the hover case. 38

25. Training loss for the trajectory prediction NN trained on 1,800 seconds of data for straight line flight. 39

26. Comparison of the true quadcopter velocity and the NN estimated quadcopter velocity using the test dataset for straight line flight. . . . 39

27. Quadcopter velocity prediction error histograms for the test dataset for straight line flight. 40

28. Comparison of the true quadcopter position and the NN estimated quadcopter position using the test dataset for the hover case. 40

29. Quadcopter position prediction error histograms for the test dataset for the hover case. 41

30. Comparison of the true quadcopter north velocity and east position and the NN estimated quadcopter north velocity and east position using the test dataset for straight line flight. 41

31. Quadcopter north velocity and east position prediction error histograms for the test dataset for straight line flight. 41

32. Training loss for the preliminary wind estimation NN trained on 1,800 seconds of data. 48

33. Comparison of the true wind velocity and the NN estimated wind velocity using the test dataset. 48

34. Wind estimation error histograms for the test dataset. Note that wind velocity errors greater than 4 m/s occurred near the large instantaneous jumps in wind velocity but were not shown here. 48

Figure Page

35. Training loss for the wind estimation NN trained on 4800 seconds of data generated with a quadcopter in hover. 49

36. Comparison of the true wind velocity and the NN predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover. 50

37. Histograms of error divided by the standard deviation of the wind for the NN predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover. 50

38. Comparison of the true wind velocity and the WT predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover. 51

39. Histograms of error divided by the standard deviation of the wind for the wind velocity estimates produced by the WT approach in [1, 2] using the Dryden wind model with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover. 51

40. Histograms of error divided by the standard deviation of the wind for the NN predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in straight line flight. 52

41. Histograms of error divided by the standard deviation of the wind for the WT predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in straight line flight. 52

Figure Page

42. Histograms of error divided by the standard deviation of the wind for a LES of the ABL wind with at an altitude of 50m for the waypoint navigation controller in hover 53

43. Histograms of error divided by the standard deviation of the wind for the WT predicted wind velocity for a LES of the ABL wind with at an altitude of 50m for the waypoint navigation controller in hover . . 53

CHAPTER I

INTRODUCTION

1. Motivation

Small unmanned aerial systems (sUAS) have become an increasingly significant topic of discussion in control systems and air traffic management (ATM) over the past decade. Advances in computing and control systems have resulted in cost reductions and performance improvements that have lead to widespread interest in sUAS in general, and multirotors in particular. Potential applications of sUAS for environmental sensing and package delivery are currently being investigated, and they are already used for filming, agricultural observation, and a number of other purposes. However, many of the commercial sUAS use proprietary controllers and do not offer specific performance guarantees for off-nominal conditions. This makes it difficult to establish reasonable regulations concerning their operation near buildings, people, and other aircraft. In order to understand the performance characteristics of these commercial sUAS, effective modeling techniques that can predict their performance based on flight data must be leveraged.

Furthermore, obtaining accurate wind velocity measurements is critical in a number of fields, such as Meteorology, Environmental Science, and Aviation. Such measurements are critical for improving our understanding of micrometeorology; environmental transport phenomena including pollutant and particulate dispersion. In aviation applications, wind information is used to assess the flight environment, devise bounds for safe operation and inform the controller for effective navigation.

This thesis considers two related problems: trajectory deviation prediction given wind velocities and wind estimation given trajectory deviations. We examine linear and nonlinear modeling techniques in an effort to capture the relationship between trajectory deviations and wind velocity.

Trajectory Prediction

Even though there is such widespread interest in sUAS applications, there are restrictions in place [3] that prevent some of their potential uses from being fully explored, e.g., package delivery. These restrictions are currently necessary due in part to insufficiently detailed trajectory models for quadrotors. Since much of the commercially used software uses proprietary controllers, an exact model is difficult to obtain. One of the major modeling uncertainties is how well a quadrotor can follow a specified trajectory in the presence of wind. A deviation of a few meters may be acceptable in sparsely populated areas, but can be the difference between a successful flight and a crash in urban centers. Therefore, a methodology needs to be developed to predict quadrotor responses to wind perturbations by analyzing the wind perturbation response of a quadcopter. Knowing the response of a quadrotor to particular wind conditions will allow the user to determine whether or not the current conditions are safe for flight.

For sUAS operations, NASA has been developing UAS Traffic Management (UTM) systems [4] that can provide infrastructure for low-altitude airspace management and monitoring. UTM systems have the potential to enable safe, efficient operations of sUAS and integration with manned aircraft at low altitudes. Future UTM systems will also include the real-time conflict management capability that requires modeling sUAS encounters and predicting sUAS trajectories. Research aimed at this capability for large UAS has been extensive. For example, the MIT Lincoln Lab generated the highest fidelity encounter model of large aircraft using radar data collected from sites across the United States [5]. However, for sUAS, modeling realistic sUAS encounter trajectories, particularly in the presence of wind disturbances, remains an open problem.

Wind Estimation

Wind turbulence is a significant factor in aerial flight, in particular for small unmanned aerial systems (sUAS), which operate at much lower airspeeds than larger aircraft [6]. Since the majority of sUAS flights are restricted to line-of-sight flight at low altitudes, they are generally operating near people or obstacles, such as buildings, power lines, trees, etc. While there is currently little data and understanding of the causes of sUAS accidents [7], a 2010 FAA study [8] determined that wind played a major role in a significant fraction of weather-related incidents for manned aircraft. This fraction is expected to increase in the case of smaller unmanned aerial vehicles. Therefore, it is imperative for sUAS to be aware of the local turbulent wind gust field in order to mitigate these deleterious effects [9] and in turn limit property damage and personal injury.

Turbulent winds, both in the mean and fluctuations (gusts), impact flight characteristics in different ways. Gusts cause sudden deviations from the prescribed trajectory while controllers in general begin correcting these deviations within a few seconds. In the mean, one needs to consider crosswinds and/or headwinds. Depending on the type of controller, crosswinds can cause drift from the desired trajectory. Likewise, headwinds impact the maximum flight speed, potentially causing unexpected delays along the trajectory. In addition to these trajectory deviations, one encounters increased power draw and reduction in battery life when flying in winds other than tailwinds.

In addition to its relevance in aviation applications, wind sensing is critical for meteorology. For environmental sensing, the problem is one of scale, as point sensors are incapable of capturing the large-scale turbulent atmospheric eddies that characterize the wind field even at low-altitudes. Although there are currently a number of sensors and approaches for measuring wind velocity, each of them has drawbacks preventing its general applicability. A common strategy is to use ground-based sensors such as a cup anemometer fitted to a meteorology mast. While these provide accurate measurements at the location of the mast, they cannot measure wind over a large area. Alternatives include SODARs and LiDAR, which are generally expensive (tens to hundreds of thousands of dollars) but provide accurate [10] wind measurements across extended spatial regions (i.e., vertical or horizontal planes or both). However, they are limited in their measurement frequency [11].

In order to measure wind velocities at altitudes higher than those obtained using ground based systems, mobile sensing platforms such as weather balloons, manned aircraft, and UAS instrumented with wind sensors are used. Weather balloons drift with the wind, but are useful for vertical profiling and measuring at very high stratospheric altitudes. Manned aircraft and large UAS can measure over extended domains, but are relatively expensive to operate while being most effective for high altitude sensing. sUAS are an attractive alternative as they bridge the gap between ground-based and larger aircraft-based platforms [12–14]. The downside is that the sUAS size limits payload capacity and sensor placement choices [15], resulting in lower measurement accuracy.

2. Literature Review

There are a number of works regarding sUAS navigation and estimation in windy environments. For example, Kothari et al. [16] and Escareno et al. [17] design guidance laws to mitigate the effects of wind and Waslander and Wang [18] develop a method of estimating wind speeds based on accelerometer measurements. Additionally, there are a number of works regarding system identification of quadcopters for use in trajectory prediction. Two of the most relevant references are by Bisheban and Lee [19] and by

Xue [20]. Bisheban and Lee identify system parameters in a quadrotor model with a specific controller. Xue trains a neural network on trajectories of a quadcopter model under constant wind disturbances using position, velocity, and wind inputs. They both compare the outputs of their identified models with the true outputs of their respective systems. Our goal is to develop an approach to predict trajectory deviations in turbulent wind without any knowledge of the quadcopter parameters or controller.

Concerning the wind estimation problem, there are a number of approaches that can be used to measure wind velocity with sUAS. The most straightforward of these is direct measurement of the wind velocity using sensors, such as the FT Technologies FT205 or Trisonica Mini, mounted on the sUAS. In ideal conditions, these sensors can measure wind speed to within 0.3 and 0.5 m/s, respectively. However, this approach has a number of drawbacks. Turbulence generated by props and the sUAS's body effects the accuracy of wind measurements; Added weight and power draw reduce battery life; And, further, the cost of multidimensional wind sensors can be several times the cost of the sUAS itself. These issues can be mitigated by using the sUAS themselves, with just their onboard sensors, to estimate the wind velocity. One of these indirect approaches is system identification of a quadcopter's dynamics parameters using linear or nonlinear approaches [21–24]. Alternatively, an Euler angle approximation can be derived using the wind triangle (WT) and a constant velocity flight assumption [1, 2]. A third option is to leverage recent advances in machine learning architectures, such as a neural network (NN), as explored in our recent conference article [25]. This machine learning approach allows for approximation of a nonlinear wind estimation model that does not require any knowledge of the controller. Recurrent NNs (RNNs) are commonly used for sequential and dynamic systems modeling due to their ability to incorporate data from previous time steps into their predictions [26, 27]. We seek to use RNNs' ability to model nonlinear dynamical system to estimate wind velocity using measurements from the navigation sensors on sUAS.

3. Overview of the Proposed Approach

Trajectory Prediction

Our work on quadcopter trajectory deviation prediction is comprised of two main pieces: linear and nonlinear modeling approaches. In Chapter III, we discuss the discrete-time linear systems approach to system identification proposed in our conference paper [28] that uses only wind velocity measurements as inputs. This is shown to be accurate for trajectory deviations due to a crosswind. In Chapter IV, we demonstrate that a neural network (NN) is capable of predicting quadcopter velocity deviations in 2-dimensional winds with high accuracy given wind velocity. These velocity deviations can be integrated to obtain position deviations; However, these position deviation estimates will drift over time due to accumulated integration errors. In practice, the predictions for position deviations could be used to obtain the gust response of the quadcopter while periodically correcting the position prediction with GPS measurements. For the specific case of straight line flight, however, the coordinate system can be designated such that there is zero mean deviation in one of the horizontal coordinates. Then, a neural network can be trained such that it captures the zero mean position deviation in that coordinate, which eliminates the error accumulation in that direction.

Wind Estimation

There are a number of reasons that make a RNN approach an appealing solution to sUAS wind estimation. First, Existing linear system ID approaches [21,22] work well for quadcopters at relatively low air-speeds. However, as roll and pitch angles increase, the nonlinearities in the dynamics become more prominent. Since NNs are effective at function approximation even in the presence of nonlinearity, they are capable of accurately capturing the quadcopter's behavior in aggressive flight. Second, Existing nonlinear modeling techniques [23] require knowledge of the form of controller, thus, limiting their usage on commercial sUAS with proprietary controls. Third, While the Wind Triangle (WT) approach [1, 2] is only reliable for steady state measurements, NNs can operate effectively with transient and steady data and, therefore, are more effective for turbulent wind estimation. Motivated by the advantages offered by the ML approach, we have chosen to use Long Short-Term Memory (LSTM) RNNs [29] to demonstrate the effectiveness of the wind estimation algorithm presented in this paper. Specifically, we use LSTM NNs to estimate horizontal, 2-dimensional wind given time series of Euler angles and the inertial position of the quadcopter.

4. Contributions of Thesis

In this thesis, we present a linear trajectory deviation prediction approach as well as machine learning (ML) aided trajectory deviation prediction methodology and a wind estimation strategy that leverages sensor data on factory configured sUAS. Linear system identification is a standard first step for any modeling problem and NNs have been widely used for function approximation [30] and to model complex nonlinear dynamic systems [31, 32].

The main contribution of the portion of this work concerning trajectory deviation prediction of sUAS in windy environments is development of a linear system identification approach (Chapter III) as well as a machine learning based approach to trajectory prediction(Chapter IV). We describe detailed fitting procedures for the difference equation approach and training procedures for the NN modeling.

The main contribution of the wind estimation part of this work (Chapter V) is to develop and demonstrate a machine learning framework for effective estimation of wind velocity. To achieve this, we provide a theoretical basis for our approach, describe training procedures for NNs, and demonstrate the viability of our approach using LSTM NNs. Our research is the first to leverage machine learning tools for wind estimation using quadcopter trajectory deviations, and further, the first to use only position and roll and pitch angles without their derivatives to estimate wind velocity. Position data in particular are becoming increasingly more accurate due to advances in navigation sensors, such as RTK-GPS units, allowing for higher accuracy wind estimation. We examine the performance of our approach using wind data generated from both the Dryden wind model and Atmospheric Boundary Layer (ABL) Large Eddies Simulations (LES). Plots and error charts are used to illustrate that our approach is effective in estimating constant and turbulent winds. The results from the NN wind estimation are compared to results obtained by using a WT approach, demonstrating that our approach results in smaller error variances and smaller mean errors for wind velocity. Further, the NN wind estimation correctly captures the correct correlation of north and east winds, whereas the WT approach does not.

CHAPTER II

QUADCOPTER MODEL

1. Dynamics

A general quadcopter model is shown in Figure 1.. The quadcopter dynamics model is made up of three pieces, described in detail in this chapter: the motor dynamics model, rotor aerodynamics models, and quadcopter rigid body dynamics. There are a number of possible controllers that can be used for this model, but the form of controller is not necessary for our trajectory prediction or wind estimation.

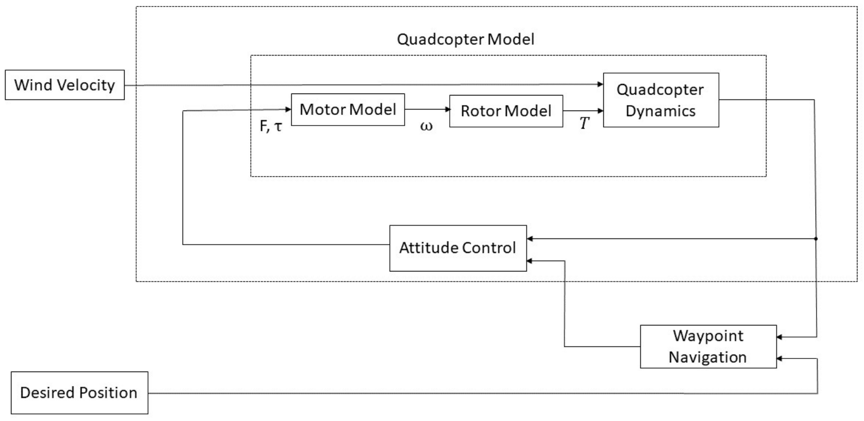

Figure 1.: General quadcopter model structure.

We use a standard formulation for the quadcopter dynamics [33], with the addition of a nonlinear drag term,

$$\begin{bmatrix} \ddot{p}_n \\ \ddot{p}_e \\ \ddot{p}_d \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} (-\cos\phi\sin\theta\cos\psi - \sin\phi\sin\psi)\frac{F}{m} + \frac{f_{d,n}}{m} \\ (-\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)\frac{F}{m} + \frac{f_{d,e}}{m} \\ g - (\cos\phi\cos\theta)\frac{F}{m} + \frac{f_{d,d}}{m} \\ \frac{J_y - J_z}{J_x}\dot{\theta}\dot{\psi} + \frac{1}{J_x}\tau_\phi \\ \frac{J_z - J_x}{J_y}\dot{\phi}\dot{\psi} + \frac{1}{J_y}\tau_\theta \\ \frac{J_x - J_y}{J_z}\dot{\phi}\dot{\theta} + \frac{1}{J_z}\tau_\psi \end{bmatrix}, \tag{II.1}$$

where (p_n, p_e, p_d) are the north, east, and down positions in the inertial frame, (ϕ, θ, ψ) are the roll, pitch, and yaw, $(F, \tau_\phi, \tau_\theta, \tau_\psi)$ are the force and the moments in the designated directions, and $(f_{d,n}, f_{d,e}, f_{d,d})$ are the drag forces in the specified directions. Since the drag is a function of the total airspeed of the quadcopter, i.e., the difference between the wind velocity and the groundspeed of the quadcopter, we adapt the standard one-dimensional drag equation to

$$f_d = C_d(V_w - \dot{p})|V_w - \dot{p}|, \tag{II.2}$$

where V_w is the wind velocity vector in the inertial frame and C_d is a diagonal drag coefficient matrix [22, 34]. The matrix C_d was determined by fitting an exponential to the calculated drag coefficient at different airspeeds of the 3DR Iris+ model in Gazebo [35], resulting in $C_d = \text{diag}(\min(1.1,\ (0.2 + 0.9\exp(-0.6|V_w - \dot{p}| - 2))))$.

2. Motor and Rotor Models

A motor model is typically given by a transfer function

$$H(s) = \frac{b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}, \tag{II.3}$$

where the b_0 and a_n are motor-dependent positive coefficients. This transfer function takes a pulse width modulation (PWM) input and outputs the rotor angular rate.

We implement and tune a PID motor controller to ensure that the motor reaches a desired angular rate as rapidly as possible while staying within the PWM limits. The specific motor model that we are using has coefficients $[a_3, a_2, a_1, a_0, b_0]^T = [1, 189.5, 13412, 142834, 2057342]^T$ and achieves convergence to a desired angular rate within 0.2 seconds. To convert the rotor speeds to force and torques on the quadcopter, we use the mapping

$$\begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} k_1 & k_1 & k_1 & k_1 \\ 0 & -Lk_1 & 0 & Lk_1 \\ Lk_1 & 0 & -Lk_1 & 0 \\ -k_2 & k_2 & -k_2 & k_2 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \tag{II.4}$$

where k_1 is the thrust coefficient, L is the quadcoptor arm length, and k_2 is the motor torque coefficient.

In addition to the quadcopter and motor dynamics given above, the rotors of the quadcopter have a number of aerodynamic effects that can impact the performance of the quadcopter. Our model incorporates two of the more significant aerodynamics effects: air-relative velocity and blade flapping [36]. Air-relative velocity effects are due to the flow of the air through the rotor. In the zero-airspeed case, the surrounding air flows through the rotors at the expected rate, resulting in a thrust calculated as

$$T = k_1\omega^2, \tag{II.5}$$

where T is the thrust of the rotor and ω is the angular rate of the rotor. However, if there is a non-zero airspeed, the thrust of each rotor needs to be corrected to account for the difference in the air flow. This corrected thrust can be calculated for each rotor as

$$T_{corrected} = \frac{Tv_i}{v_i + w}, \tag{II.6}$$

$$v_i = \frac{v_h^2}{\sqrt{u^2 + v^2 + (v_i + w)^2}}, \tag{II.7}$$

where v_h is the induced velocity of the rotor while the quadcopter is hovering and (u, v, w) are the air velocities in the body frame.

Blade flapping is a ‘tilting’ of the rotor due to unequal forces on the advancing and retreating edges of the blade while in flight. The advancing edge of the blade has a higher airspeed than the retreating edge of the blade. Since the higher airspeed results in a higher thrust on the advancing edge, the rotor bends slightly more on the advancing edge than the retreating edge, resulting in a shift in the thrust plane to be away from the body frame. This necessitates defining the thrust from each rotor as a vector in the body frame, given by

$$T_{flapping} = \begin{bmatrix} \frac{u}{\sqrt{u^2+v^2}} \sin\alpha \\ \frac{v}{\sqrt{u^2+v^2}} \sin\alpha \\ \cos\alpha \end{bmatrix} T, \tag{II.8}$$

$$\alpha = K_f \sqrt{u^2 + v^2}, \tag{II.9}$$

where α is the blade flapping angle and K_f is the flapping coefficient.

3. Controller

To control the quadcopter, we use a feedback linearized PD attitude controller and a saturated PID waypoint navigation controller. This is a common controller for quadcopters [37–39]. We hand-tune the gains to achieve a slightly under-damped response with as rapid convergence as possible for the quadcopter model in Section 1.. However, the exact performance of the controller is of little relevance for the trajectory prediction or wind estimation problems as it will be learned as part of the model in the following chapters. Our saturated PID controller is given by

$$\begin{bmatrix} \phi^d \\ \theta^d \\ \psi^d \\ \ddot{p}_d^d \end{bmatrix} = \begin{bmatrix} \mathrm{sat}(k_p e_{p_e} + k_d \dot{e}_{p_e} + k_i \int e_{p_e}, \phi\mathrm{max}) \\ \mathrm{sat}(k_p e_{p_n} + k_d \dot{e}_{p_n} + k_i \int e_{p_n}, \theta\mathrm{max}) \\ \psi \\ k_p e_{p_d} + k_d \dot{e}_{p_d} + k_i \int e_{p_d} \end{bmatrix}, \tag{II.10}$$

where the d superscript designates a desired value, k_p, k_i, and k_d represent the proportional, integral, and derivative control gains, $\phi\mathrm{max}$ and $\theta\mathrm{max}$ are the maximum allowed roll and pitch angles, and $e_p = p^d - p$ are the position errors in the designated directions. We use the saturation function

$$\mathrm{sat}(x, x\mathrm{max}) = \begin{cases} -x\mathrm{max} & \text{if } x < -x\mathrm{max} \\ x & \text{if } -x\mathrm{max} \leq x \leq x\mathrm{max} \\ x\mathrm{max} & \text{if } x > x\mathrm{max} \end{cases}, \tag{II.11}$$

to ensure that the waypoint navigation controller does not request a roll or pitch angle greater than $\phi\mathrm{max} = \theta\mathrm{max} = 0.8$ radians. The desired Euler angles and vertical acceleration generated by (II.10) are then used in the attitude controller below to generate desired force and moment commands

$$\begin{bmatrix} F \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} \frac{m(g+\ddot{p}_d^d)}{\cos(\phi)\cos(\theta)} \\ J_x(-K_1\dot{\phi} - \frac{J_y-J_z}{J_x}\dot{\theta}\dot{\psi}) + K_{p1}e_1 + K_{d1}\dot{e}_1 \\ J_y(-K_2\dot{\theta} - \frac{J_z-J_x}{J_y}\dot{\phi}\dot{\psi}) + K_{p2}e_2 + K_{d2}\dot{e}_2 \\ J_z(-K_3\dot{\psi} - \frac{J_x-J_y}{J_z}\dot{\phi}\dot{\theta}) + K_{p3}e_3 + K_{d3}\dot{e}_3 \end{bmatrix}, \tag{II.12}$$

where

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \phi^d - \phi \\ \theta^d - \theta \\ \psi^d - \psi \end{bmatrix}. \tag{II.13}$$

Using this controller allows us to set desired positions for the quadcopter to reach while applying wind disturbances to the system.

4. Simulation Environment

Quadcopter Parameters

In order to simulate a quadcopter with the dynamics given in above, we need to choose model parameters. The parameters that we use are based on the 3DR Iris+ and the controllers are hand tuned. Specific dynamics and controller parameters for our simulations are given in Table 1.. This model is realized in Simulink with a fixed time step of 0.001 seconds.

Dynamics parameters							
g	m	J_x	J_y	J_z	L	$k_1,\ k_2$	K_f
9.81	1.5	0.0348	0.0459	0.0977	0.235	5×10^{-5}	0.003

Position control gains		
k_p	k_d	k_i
0.3	0.25	0.0002

Attitude control gains					
$K_1,\ K_2$	K_3	$K_{p1},\ K_{p2}$	K_{p3}	$K_{d1},\ K_{d2}$	K_{d3}
21.93	48	4.65	3.77	0.1872	0.1496

Table 1.: Quadcopter model parameters.

Dryden Stochastic Wind Gust Model

The popular approach to representing small scale atmospheric gusts in aviation applications, such as trajectory estimation, rely on stochastic formulations [40–43], and its variants [44]. The Dryden turbulence model [42, 43, 45] is one such realization of stochastic wind gusts and is commonly implemented through a filtering operation on a white noise signal (see MATLAB documentation for details) with transfer functions as below:

$$H_u(s) = \sigma_u \sqrt{\frac{2V_{a0}}{L_u}} \frac{1}{(s + \frac{V_{a0}}{L_u})}, \quad H_v(s) = \sigma_v \sqrt{\frac{3V_{a0}}{L_v}} \frac{(s + \frac{V_{a0}}{\sqrt{3}L_v})}{(s + \frac{V_{a0}}{L_v})^2}, \tag{II.14}$$

$$H_w(s) = \sigma_w \sqrt{\frac{3V_{a0}}{L_w}} \frac{(s + \frac{V_{a0}}{\sqrt{3}L_w})}{(s + \frac{V_{a0}}{L_w})^2},$$

where s is the complex frequency (i.e. $s = 0 + i\omega$ where $i = \sqrt{-1}$), $\sigma = [\sigma_u, \sigma_v, \sigma_w]^T$ are the turbulence intensities, V_{a0} is an estimate of the quadcopter's airspeed, and $L = [L_u, L_v, L_w]^T$ are the turbulence lengths scales entering the model. Using this set of equations to generate turbulence results in a wind field that varies spatially but not temporally. This behavior is due to the reliance of the transfer functions on V_{a0} – i.e., if the mean airspeed is set to zero, then (II.14) will always equal zero. In our simulations, we consider four different turbulence intensities for the Dryden model based on standard values for turbulence intensity from literature [42, 43, 45]. Details of the intensities used for our simulations are given in the relevant results sections.

LES of the ABL

The limitations of the Dryden stochastic wind gust models are well known. In particular, these are: (i) treatment of turbulence as a stochastic process dictated only by the diagonal components of the spectral energy tensor and (ii) treatment independent of scale. This is inconsistent with the huge body of literature on turbulent coherent structures [46] and well-defined covariance statistical structure [47] in the atmospheric surface layer (ASL). In spite of these shortcomings and their impact on air traffic management(ATM) [48], they are popular on account of their computational efficiency for rapid estimation.

However, much more accurate models exist for modeling the behavior of wind at low altitudes. One of these formulations is large eddy simulation (LES) of the atmospheric boundary layer (ABL). In order to more accurately replicate the wind fields that a quadcopter would experience outdoors, we have incorporated wind data generated using the framework developed by [46] into our simulations.

CHAPTER III

TRAJECTORY PREDICTION USING LINEAR DIFFERENCE EQUATIONS

In this chapter, we provide a preliminary investigation of modeling the trajectory of a quadcopter under wind disturbances based on an input-output modeling approach. In particular, we employ a linear time-invariant difference equation model that takes the wind velocities as the input and outputs the trajectory deviation of the quadcopter due to the wind. Because of the nonlinearity of the quadcopter dynamics and the onboard controller, the linear model only serves as an approximation of the actual quadcopter dynamics. Using simulation data from a quadcopter trajectory with a constant crosswind, we identify the parameters in the model and make use of the identified model to predict the quadcopter trajectory in different wind speeds. The predicted trajectories are compared with the actual trajectories of the quadcopter under the same wind conditions.

We validate our approach using simulation results from a PID position controller and a PD attitude controller. Note that although the controllers are linear, the closed-loop system is still nonlinear due to the nonlinear dynamics of the quadcopter. Our results show that the obtained 4th order difference equation model is capable to predict the quadcopter trajectory with a reasonable error (on the order of 10% of the maximum east position overshoot). We also examine the prediction performance with a stochastic wind model consisting of a constant wind and a random wind. Prediction performance with similar errors is obtained. We further test the proposed approach with a 3DR Iris+ simulator. Possibly due to the complexity of the controller and the inaccuracies due to sensor noise, we obtain a higher order model and larger prediction errors. We are currently investigating nonlinear modeling techniques to reduce the system order and the prediction errors.

1. Modeling the Quadcopter Response

Objective

Given a wind v_w, the dynamic model (II.1) with the controller in (II.10)–(II.13) generates a quadcopter trajectory, $p(t; p_0, f_d, p^d)$. Our objective is to determine a simple model that predicts the trajectory $p(t; p_0, f_d, p^d)$ based on a specific wind without knowing the exact form of the controller. In this paper, we focus on modeling the trajectories due to a crosswind. Towards this end, we assume that the quadcopter is intended to fly towards a waypoint in the north direction and an east crosswind is injected on the path to the waypoint.

Difference Equation Modeling

In this section, we investigate a system identification method to model the response of a quadcopter to the east crosswind. Due to the wind, the quadcopter deviates from its intended path along the north direction towards the east. We model the deviation at time k as the output of a n-th order difference equation (DE). The input to the DE is the magnitude of the east crosswind. Specifically, we consider the following model

$$y_e(k) + a_1 y_e(k-1) + \cdots + a_n y_e(k-n) = b_0 v_{w,e}(k) + b_1 v_{w,e}(k-1) + \cdots + b_n v_{w,e}(k-n), \tag{III.1}$$

where $y_e(k)$ is the position deviation in the east direction at time step k, $v_{w,e}(k)$ is the wind magnitude at time step k, and a_i's, $i = 1, \cdots, n$ and b_j's, $j = 0, \cdots, n$, are constant coefficients to be identified. We assume that the wind is injected at $k = 0$. Thus, $v_{w,e}(k) = 0$ and $y_e(k) = 0$ for $k < 0$. It follows from (III.1) that the discrete time transfer function from $V_{w,e}(z)$ to $Y_e(z)$ is given by

$$G(z) = \frac{Y_e(z)}{V_{w,e}(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}. \tag{III.2}$$

Note that $G(z)$ models the combined effects of the dynamics and the onboard controller of the quadcopter. If prior knowledge of the controller is available, it can be incorporated into $G(z)$. For example, if the controller includes an integral term, we may factor $G(z)$ as $(z-1)\bar{G}(z)$, where the $z-1$ term represents the integral effect in the controller and

$$\bar{G}(z) = \frac{\bar{b}_1 z^{-1} + \cdots + \bar{b}_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}, \tag{III.3}$$

where $\bar{b}_i$'s are new coefficients after factoring $z-1$ out from the numerator of $G(z)$.

Identification of the Coefficients

Given a set of $v_{w,e}(k)$ and $y_e(k)$ data, we can identify a_i's and b_j's in (III.1) using standard techniques in system identification [49, 50]. In this paper, we use least square fit to identify the parameters. The obtained parameters are then used in (III.1) or (III.2) to represent the relationship between the wind and the response. To predict the response of the quadcopter to a new wind $\bar{v}_{w,e}(k)$, $k = 0, 1, 2, \cdots$, we replace $v_{w,e}(k)$ in (III.1) with $\bar{v}_{w,e}(k)$ and propagate $y_e(k)$ from $y_e(0)$ using the identified a_i's and b_j's.

In the special case where the wind $v_{w,e}(k) = v_{w,e}$ is a constant, (III.1) is reduced to

$$y_e(k) + a_1 y_e(k-1) + \cdots + a_n y_e(k-n) = \sum_{j=0}^{n} b_j v_{w,e}, \quad k \geq n, \tag{III.4}$$

which is used to identify a_i's and $\sum_{j=0}^{n} b_j$ given the data $y_e(k)$ and $v_{w,e}$, $k \geq n$. To identify b_j's, we note $v_{w,e}(k) = 0$ and $y_e(k) = 0$ for $k < 0$, using (III.1) for $0 \leq k < n$ allows us to identify b_j's, $j = 0, \cdots, n-1$.

If an integral term is included in the control, we apply standard $\mathscr{Z}$-transforms and obtain from (III.2)–(III.3)

$$Y_e(z) = (z-1)\bar{G}(z)\frac{v_{w,e}}{1-z^{-1}} = v_{w,e}\frac{\bar{b}_1 + \bar{b}_2 z^{-1} \cdots + \bar{b}_n z^{-n+1}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}. \tag{III.5}$$

Converting $Y_e(z)$ to time domain yields

$$y_e(k) + a_1 y_e(k-1) + \cdots + a_n y_e(k-n) = v_{w,e}\left(\bar{b}_1\delta(k) + \bar{b}_2\delta(k-1) + \cdots + \bar{b}_n\delta(k-n+1)\right), \tag{III.6}$$

where $\delta(k)$ is the standard dirac function, i.e., $\delta(0) = 1$ and $\delta(k) = 0$, $\forall k \neq 0$. After a_i's and $\bar{b}_i$'s, $i = 1, \cdots, n$, are identified, $\bar{G}(z)$ in (III.3) is obtained. We further use $G(z) = (z-1)\bar{G}(z)$ to obtain the coefficients in $G(z)$.

2. Simulation Results

We develop a simulation model of the dynamics and the controller in Chapter II in MATLAB Simulink. We use physical properties of the 3DR Iris+, as given by the PX4 SITL [51] Iris+ simulator, for the dynamics in the Simulink model.

In the simulations, the desired waypoint is set at 150 meters in the north direction in front of the quadcopter, resulting in a pitch saturation at the maximum allowed pitch angle for most of the flight.The sampling frequency of the quadcopter trajectory is 10 Hz.

Difference Equation Identification

To apply the discrete time system identification techniques discussed in Section 1., we apply a 4 m/s east crosswind in the simulation and collect the quadcopter trajectory data in the east direction. Since our controller consists of an integral term, we identify the a_i's and $\bar{b}_i$'s in (III.3) with the collected data. Fig. 2. compares the fitted trajectories for different system orders ($n = 2, 3, 4$) with the true trajectory and shows the corresponding fitting errors. We observe that the fitted trajectory for $n = 4$ closely matches the true trajectory. Further increasing the system order only

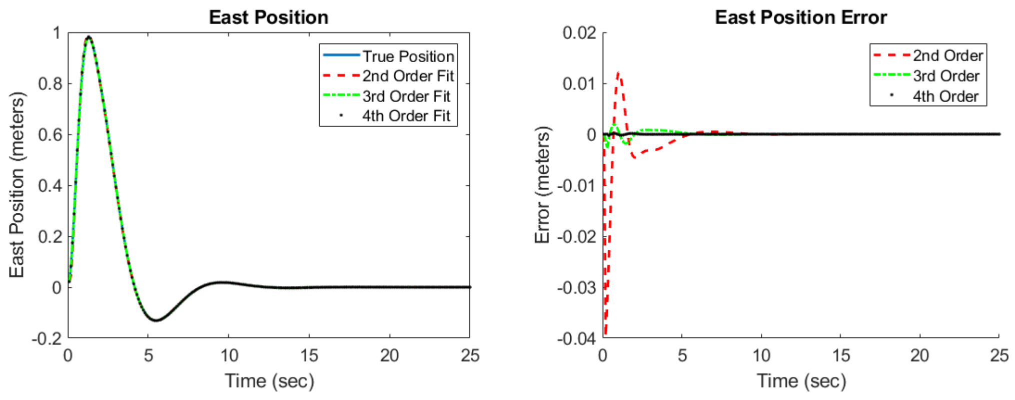

Figure 2.: Comparison of the true trajectory and the fitted trajectories of different system orders ($n = 2, 3, 4$) for 4 m/s wind.

results in small error reductions and could lead to overfitting of the data. Therefore, we choose $n = 4$ in the model. We also identify the coefficients for different wind speeds (1, 4, and 7 m/s) to confirm that the identified coefficients are similar.

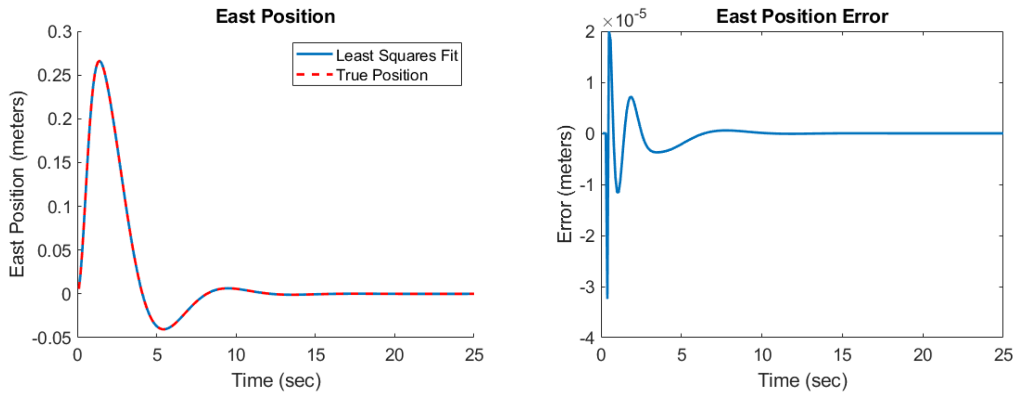

Figure 3.: Comparison of the trajectory obtained from the least square fit with the true trajectory for 1 m/s wind.

Difference Equation Model Validation

To validate that the model parameters identified in the previous section can be used to predict the behavior of the quadcopter, we compare the predicted trajectory and the true trajectory for additional wind speeds. We use the coefficients from the 4 m/s wind case to predict the trajectories of the quadcopter for 1 to 7 m/s winds

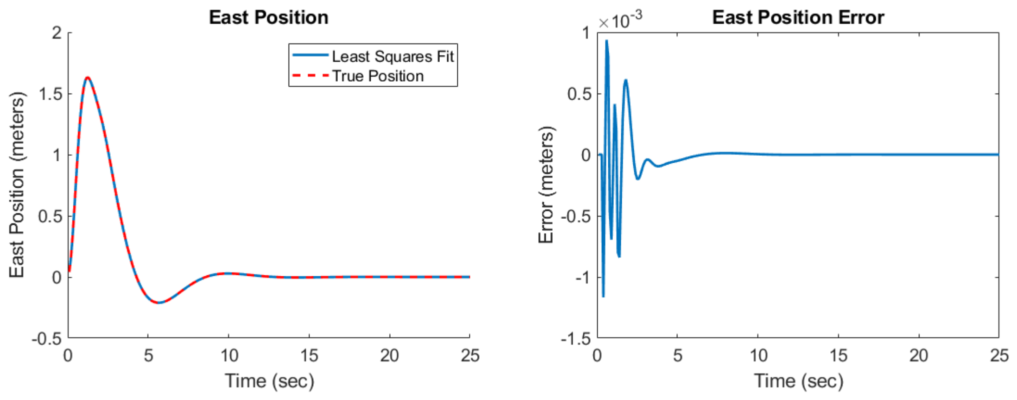

Figure 4.: Comparison of the trajectory obtained from the least square fit with the true trajectory for 7 m/s wind.

and compare the predicted trajectories with the true trajectories of the quadcopter for the same winds. Fig. 5. and 6. show the predicted trajectory and the actual trajectory for 2 m/s and 6 m/s winds, respectively. The predicted trajectories are within approximately 10% of the maximum east position overshoot for all of the test cases, as shown in Fig. 7..

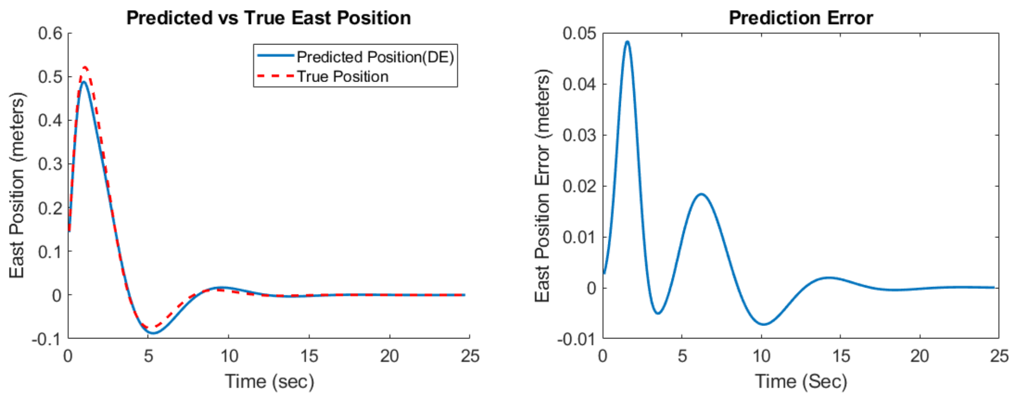

Figure 5.: Comparison of the predicted trajectory and the true trajectory for 2 m/s wind.

PID Gain Variation

We change the PID gains, $K = [k_p, k_d, k_i]$, to $\bar{K} = \frac{1}{2}K$ and re-identify the fourth order system coefficients for the controller with $\bar{K}$. The new control gains result in a

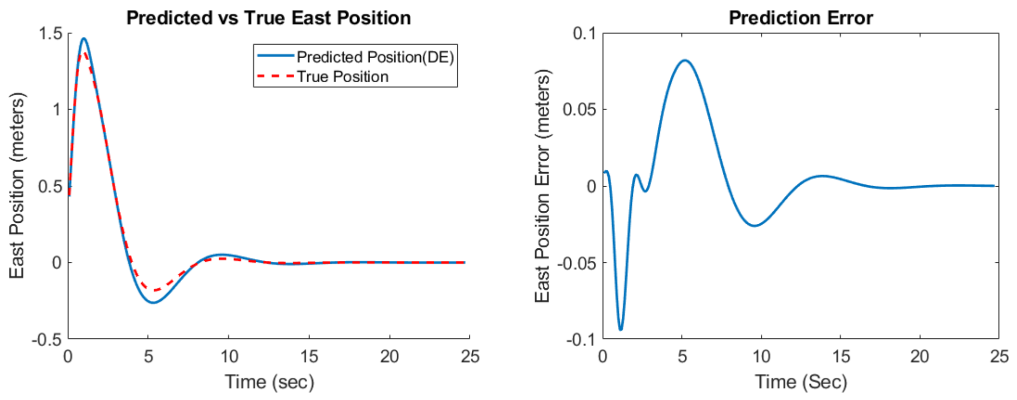

Figure 6.: Comparison of the predicted trajectory and the true trajectory for 6 m/s wind.

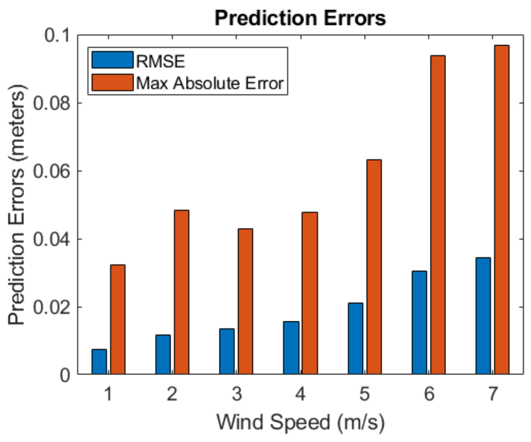

Figure 7.: RMSE and maximum prediction errors for various wind speeds.

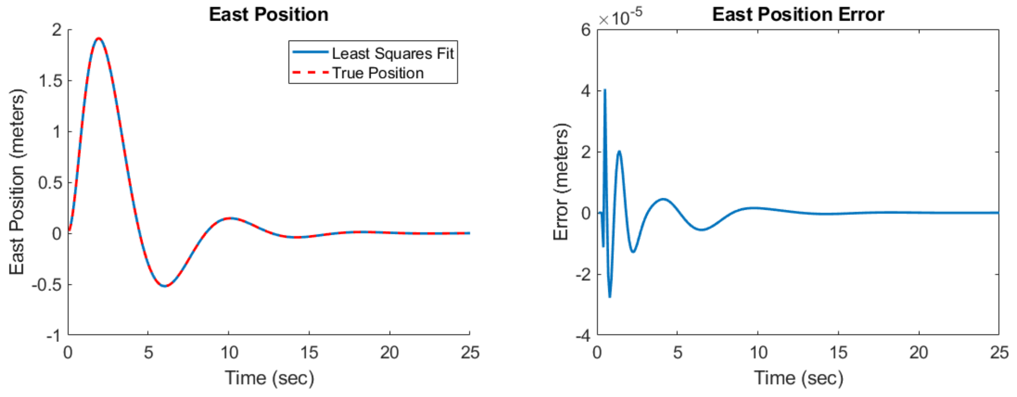

Figure 8.: Comparison of the identified system with the true trajectory for 4 m/s wind with the new control gain $\bar{K}$.

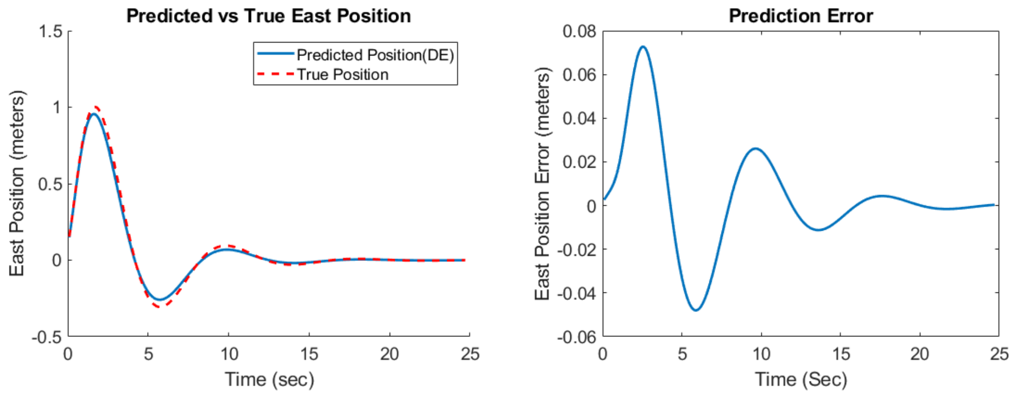

Figure 9.: Comparison of the predicted trajectory and the true trajectory for 2 m/s wind with the new control gain $\bar{K}$.

different system response for the 4 m/s crosswind. However, the trajectory generated from the new coefficients still matches the true trajectory well, as shown in Fig. 8.. Similar to the previous section, we can predict the response for different wind speeds. Fig. 9. shows one example where the predicted trajectory is compared with the true trajectory for a 2 m/s wind. Fig. 10. shows the prediction errors across different wind speeds. It should be noted that the maximum east deviation for the quadcopter using the adjusted PID gains is over 3 meters for the 7 m/s wind case. Therefore, although the magnitude of the error is significantly greater for this case than the case with the original gains, the percentage error remains similar.

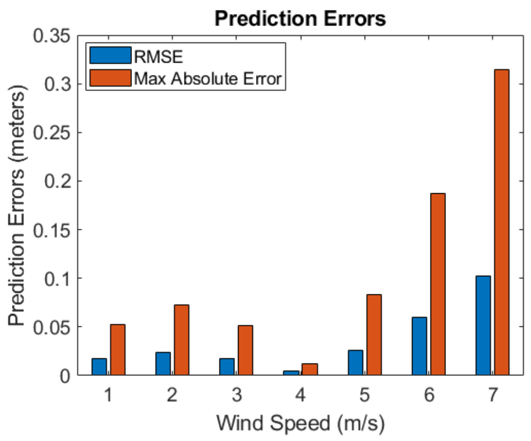

Figure 10.: RMSE and maximum prediction errors for various wind speeds with the new control gain $\bar{K}$.

Prediction with the Dryden Wind Model

Previous simulation results demonstrate that identifying a difference equation allows us to predict the trajectory deviation of a quadcopter disturbed by a constant wind. We now consider predicting the trajectory deviation due to a non-constant wind. A commonly used model for simulating wind turbulence is the Dryden model [12]. We have integrated the Dryden model into our simulation as described in Chapter II. Sample winds generated by this model are shown in Fig. 11..

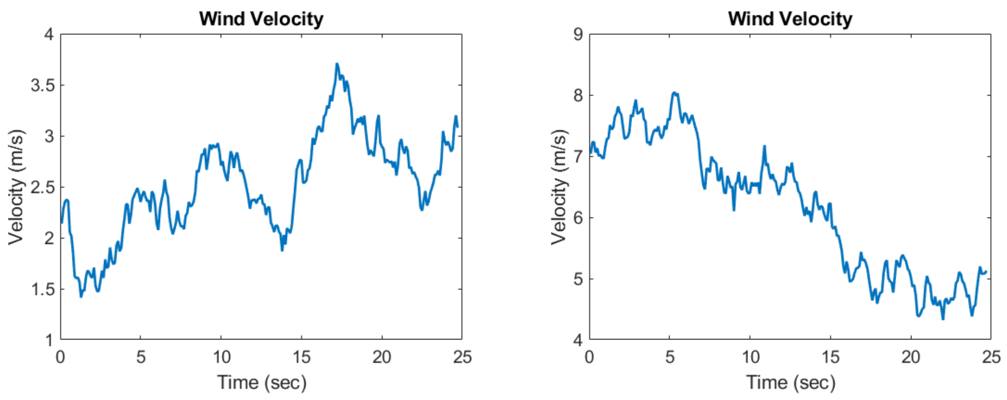

Figure 11.: Sample winds generated using Dryden turbulence model.

We examine whether our identified model in (III.1) is able to predict the deviation of a quadcopter in a time varying wind generated from the Dryden model. We employ

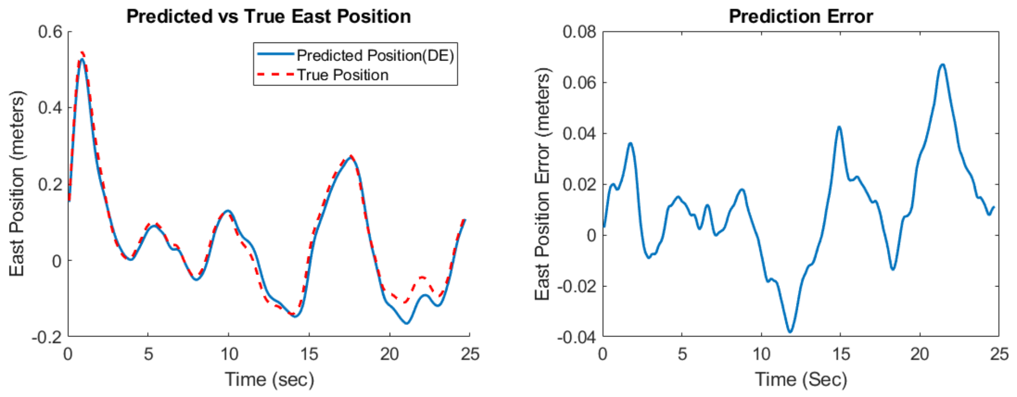

Figure 12.: Comparison of the predicted trajectory and the true trajectory for a wind generated by the Dryden wind model with a 2 m/s constant component.

the 4th order difference equation model identified in the 4 m/s wind case to predict the quadcopter's position given a time varying wind from the Dryden model. The predicted trajectory is then compared with the true trajectory of the quadcopter in the same turbulent wind. Fig. 12. and 13. show the comparison for the winds with 2 m/s and 6 m/s constant east wind, respectively. Since the Dryden wind model has a stochastic component, we conduct Monte-Carlo simulations with different constant wind speeds. Fig. 14. shows the RMSE and maximum errors averaged over the 50 simulations. As one can see, for the turbulent wind for each wind case, the errors from the Dryden wind model are similar to their constant wind counterparts.

We also ran simulations with moderate turbulence. For the moderate turbulence case, the turbulence scale lengths remained constant, but we increased the turbulence intensities, σ, by a factor of two. We averaged the RMSE and maximum errors over 50 simulations as in the previous case. Using the increased turbulence results in similar errors to the light turbulence model for wind speeds below 6 m/s, but there is a significant increase in the errors for the 6 m/s and 7 m/s wind cases, as shown in Fig. 15.. This increase in the errors for higher wind speeds occurs because the controller cannot compensate for the increased turbulence at higher wind speeds.

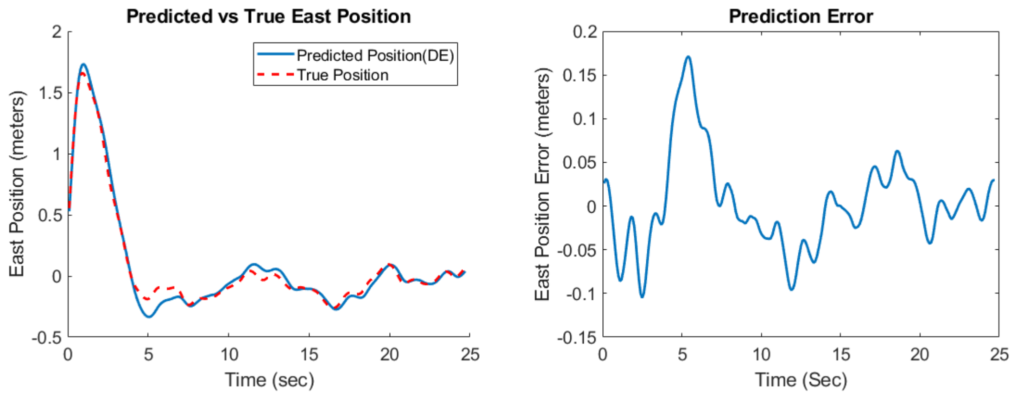

Figure 13.: Comparison of the predicted trajectory and the true trajectory for a wind generated by the Dryden wind model with a 6 m/s constant component.

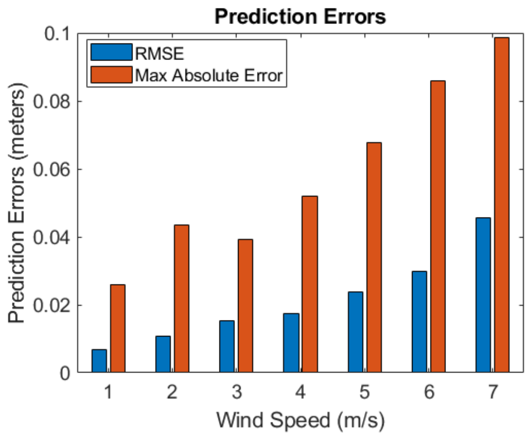

Figure 14.: RMSE and maximum prediction errors for various winds generated by the Dryden wind model with different constant components.

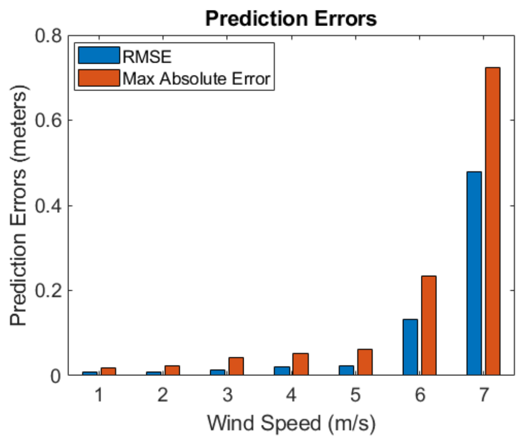

Figure 15.: RMSE and maximum prediction errors for various winds generated by the Dryden wind model for moderate turbulence with different constant components.

Prediction Using Iris+ PX4 Model

Our simulation uses a simple PID controller for waypoint navigation. A number of more sophisticated simulations of quadcopters are freely available. We use one of the simulators designed for use in Gazebo due to the maturity of the sensor models and physics engines and its close ties to ROS (Robot Operating System [52]). Since ROS is used not only for simulations, but also for the control of a number of commonly used quadcopters (3DR, DJI, AscTec, and others), using a Gazebo based simulation allows for easier transition to a physical platform. RotorS was developed in ROS and gazebo by Furrer, et al. [35] as a realistic multirotor simulator. Sensors and motors have noise and inaccuracies added to give a more representative view of how a quadcopter would actually behave. The PX4 [51] firmware developers have modeled a 3DR Iris+ using these simulated components along with a CAD model of the structure of the Iris+. By combining the PX4 firmware developed for use on the actual Iris+ with this simulation model of the Iris+, an accurate simulation of the quadcopter has been created by the PX4 developers (PX4 SITL). Using this simulator allows us to test our modeling approach with a completely unknown controller.

The general system identification and validation procedures used in this section are the same as in the previous sections. quadcopter trajectory and wind velocity data from Iris+ flight simulations are collected and a difference equation model is identified from the data. For the PX4 simulations, we increase the system order n to 15 to obtain more accurate predictions. The identified model predicts the quadcopter trajectory with less accuracy compared to the results in the previous sections. Fig. 16. and 17. show the predicted trajectory compared with the true trajectory for 5 m/s and 1 m/s constant wind, respectively. Fig. 18. shows the prediction errors of the 15th order model for various constant winds. Fig. 19. shows how the total prediction error obtained from the five wind cases ($v_{w,e} = 1, \cdots, 5$) varies against the order of the DE in (III.1).

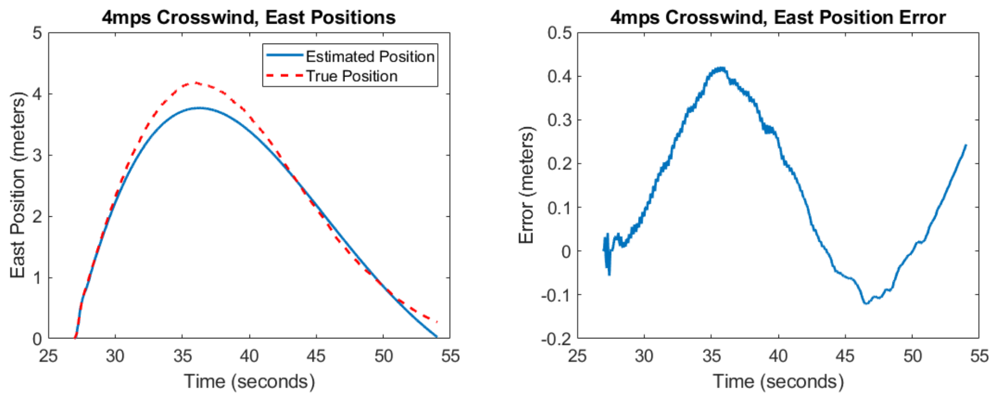

Figure 16.: Comparison of the predicted trajectory and the true trajectory for 4 m/s wind in the Iris+ simulator.

The less accurate prediction results obtained for the Iris+ simulation may be due to the unknown and potentially complex controller used in the PX4 simulation. Additionally, the sensor noise in the measurements in the Iris+ simulator would contribute to inaccuracies in the prediction.

Time Correlation

An approach that could potentially be used to improve the accuracy of these predictions is decorrelation analysis. Decorrelation is the measure of correlation across

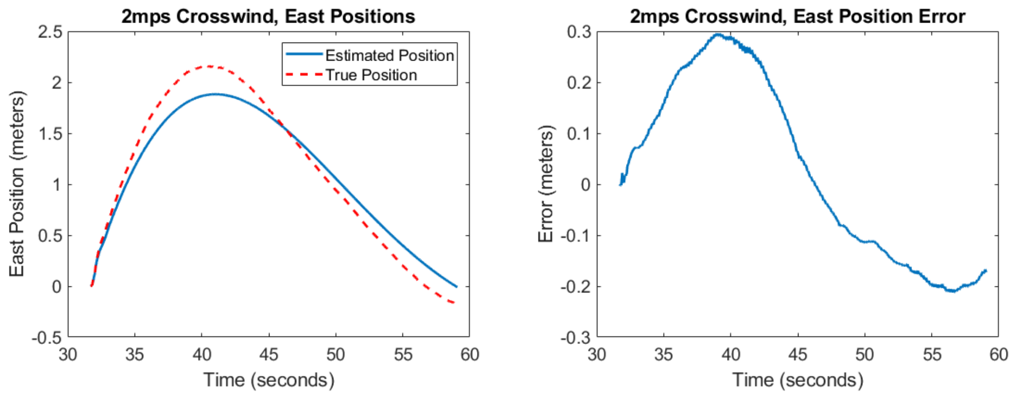

Figure 17.: Comparison of the predicted trajectory and the true trajectory for 2 m/s wind in the Iris+ simulator.

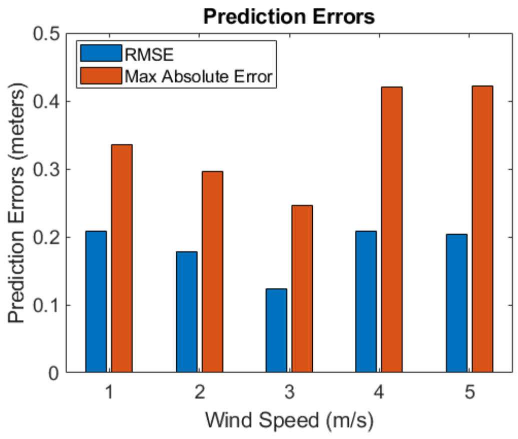

Figure 18.: RMSE and maximum prediction errors for various constant winds using a 15th order identified model and the Iris+ simulator.

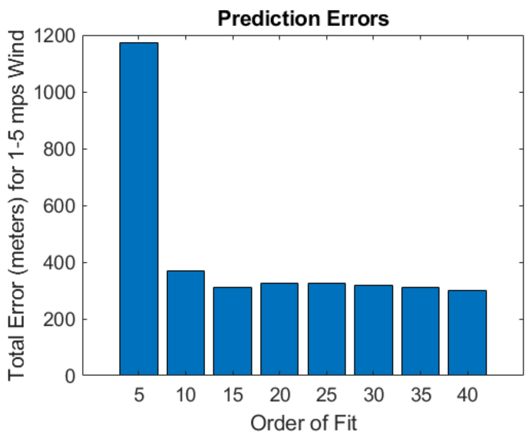

Figure 19.: Comparison of the total errors obtained from $v_{w,e} = 1, \cdots, 5$ m/s with different orders of the DE in (III.1).

time that a signal has. Therefore, the decorrelation of a signal can be used to determine the optimal order for the difference equation identification. In Figure 20., the decorrelation for the Simulink quadcopter in hover is shown for acceleration, velocity, and position. The horizontal line drawn at five percent correlation demonstrates that the position experiences decorrelation at a distance of roughly 25 time steps from the current time when considering 10 Hz data.

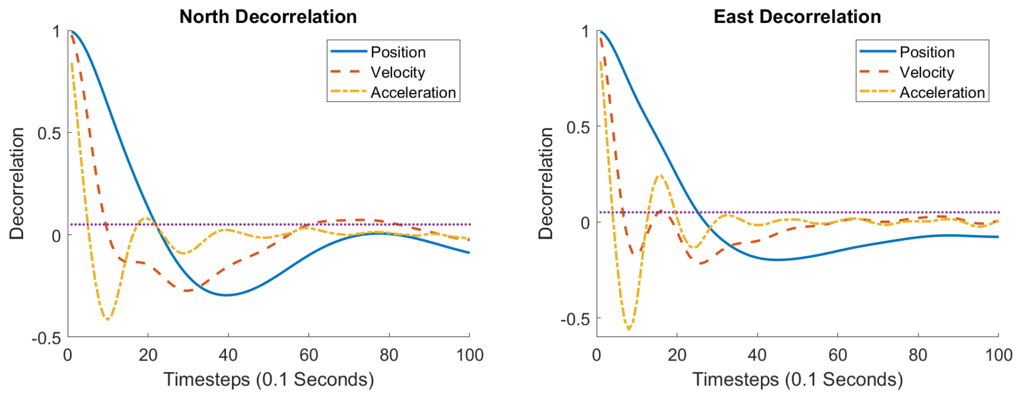

Figure 20.: Decorrelation of the Simulink quadcopter in hover.

3. Conclusions

We investigate trajectory modeling of quadcopters in the presence of a crosswind disturbance. We model the trajectory deviation due to the wind using a linear difference equation approach. By identifying the coefficients of the difference equation, we construct the trajectory model and use it to predict the trajectory deviation of a quadcopter flying in both constant and time-varying winds. For the case of a PID controller, we obtain a low order (4th order) model that predicts the trajectory with errors on the order of 10% of the maximum east position overshoot for the controller used and constant wind speeds of less than 7 m/s. We further validate the trajectory model against time-varying winds consisting of a constant component and a random component. We can also predict trajectory deviations of the 3DR Iris+ in Gazebo, although the resulting model order is much higher and the prediction error is larger. This implies that, for more complex controllers, this linear approach may not be sufficient. Therefore, in the following chapter, we investigate nonlinear modeling approaches.

CHAPTER IV

TRAJECTORY PREDICTION USING A NEURAL NETWORK

1. Motivation for Nonlinear Modeling

As was seen in the previous chapter, linear difference equations (LDE) are effective at predicting trajectory deviations of a quadcopter with relatively simple trajectories and in pure crosswinds. However, in more complex scenarios, the linear approach has reduced accuracy. Since the linear difference equation approach to sUAS gust response modeling has reduced accuracy for the multidimensional wind case, the logical next step is to consider nonlinear system modeling techniques. In this chapter, we present an approach to trajectory deviation prediction that leverages the nonlinear modeling capabilities of an LSTM NN.

2. Trajectory Prediction using a Neural Network

Derivation of Trajectory Prediction Formulation

To determine the states necessary for a NN to predict trajectory deviations given wind velocity, we seek to convert the translational dynamics of the quadcopter in (II.1) into the form $p = f(\cdot)$, where $f(\cdot)$ is a nonlinear function of the wind. To accomplish this, we note from (II.1) that

$$\ddot{p} = ge_3 - \frac{F}{m}Re_3 + C_d(V_w - \dot{p}_n)|V_w - \dot{p}_n|, \tag{IV.1}$$

where $e_3 = [0, 0, 1]^T$ and R is the rotation matrix from the body frame to the inertial frame. Using Euler discretization, we rewrite (IV.1) as

$$\dot{p}(k) \approx \dot{p}(k-1) + \Delta t \ddot{p}(k-1) \tag{IV.2}$$

$$= \dot{p}(k-1) + \Delta t[ge_3 - \frac{F(k-1)}{m}R(k-1)e_3 + C_d(V_w(k-1)... \tag{IV.3}$$

$$- \dot{p}(k-1))|V_w(k-1) - \dot{p}(k-1)|]. \tag{IV.4}$$

Assuming a fixed trajectory, the control input and orientation can be learned, resulting in

$$\dot{p}(k) = f(V_w(k-1), P), \tag{IV.5}$$

where P are constant parameters, including the drag coefficient, mass, and gravity. In order to obtain position deviations, the velocity can be integrated over time. However, integration errors will accumulate, resulting in drift from the actual values. In the case that position in a direction is zero mean, the integration can be learned as part of (IV.5), allowing

$$p(k) = f(V_w(k-1), P) \tag{IV.6}$$

to be learned for that direction instead of (IV.5). This will eliminate the drift issue associated with the integration, resulting in accurate position deviation predictions.

We focus on 2-D wind for our trajectory prediction approach. For each trajectory, we train the RNN on 1,800 seconds of data sampled at 10 Hz. The specific inputs used are north and east wind velocity with targets of north quadcopter velocity and east quadcopter position.

LSTM Neural Network

LSTM NNs are a type of RNN frequently used for dynamic system modeling due to their ability to learn information about long term dependencies in data [29]. One of the distinguishing features of LSTM NNs is their gated self-loops. Since the weighting on the gates is trained, it allows for the retention of information over long sequences of data. This mitigates the exploding/vanishing gradient problems experienced with some other forms of RNNs.

LSTM NNs make predictions based on sequences of data. Each set of input sequences with n time steps of the input is paired with a single time step of the target vector. An LSTM NN is trained on the input sequences and target vector. After training, it can be used to generate predictions when given a new input sequence.

The structure of a LSTM unit is shown in Figure 21.. LSTM units are composed of three gates that serve different purposes, as well as a memory cell to store previous inputs. In the following equations, variables specific to each gate are designated using the superscripts i, f, and o for the input, forget, and output gates, respectively. Each gate has a sigmoid activation function, resulting in a vector with values between zero and one. Training with this activation function allows the gate to prioritize the data most relevant to the target values. The input gate given by

$$i_i^k = \sigma\left(b_i^i + \sum_j U_{i,j}^i x_j^k + \sum_j W_{i,j}^i h_j^{k-1}\right) \tag{IV.7}$$

acts on a combination of the k^{th} input data point and n previous data points (stored in the memory cell). The subscript i represents the i^{th} unit of the NN, j designates the input feature, and the superscript k represents the time step for the prediction. The variable b_i^i is the input gate's bias for the i^{th} unit of the NN, U_i^i is the input weight, W_i^i is the recurrent weight, x^k are the inputs to the LSTM unit at the k^{th} time step, and h^{k-1} are the internal states of the LSTM unit from the previous time step for j input features. To determine which of the previous inputs remains stored in the cell, the forget gate output given by

$$f_i^k = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^k + \sum_j W_{i,j}^f h_j^{k-1}\right) \tag{IV.8}$$

is multiplied by the data in the memory cell at each time step. Thus, the internal

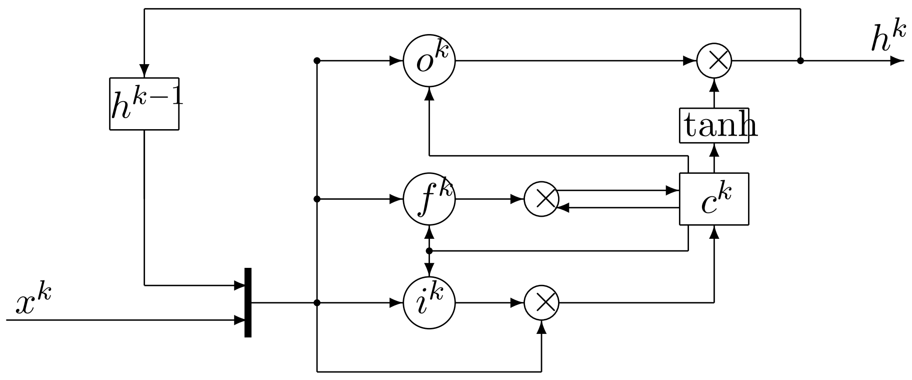

Figure 21.: General form of a LSTM NN unit.

state of the LSTM cell is given by

$$c_i^k = f_i^k c_i^{k-1} + i_i^k \sigma \left(b_i + \sum_j U_{i,j} x_j^k + \sum_j W_{i,j} h_j^{k-1} \right). \tag{IV.9}$$

Once the new input and previous inputs have been appropriately scaled by the input and forget gates, they are sent through a hyperbolic tangent function as

$$h_i^k = \tanh(c_i^k) o_i^k, \tag{IV.10}$$

and multiplied by the output of the output gate as

$$o_i^k = \sigma \left(b_i^o + \sum_j U_{i,j}^o x_j^k + \sum_j W_{i,j}^o h_j^{k-1} \right), \tag{IV.11}$$

to ensure that the output of the LSTM unit corresponds to the desired output.

3. Simulation Results

Data from the simulation environment described in Chapter II are used to train a LSTM NN. The preliminary training case used 1,800 seconds of training data and uniformly random winds between -7 and 7 m/s, each applied for a random amount of time between 0 and 15 seconds. A sequence length of $n = 20$ that overlaps with 5 previous inputs of the previous training sequence is used. Ten percent of the sequences are randomly chosen as the validation sequences for training purposes, and the NN is trained for 30 epochs.

For the size of the NN, we choose to use two hyperbolic tangent hidden layers with 50 units each and 10 percent dropout based on hand-tuning. Since the hyperbolic tangent function has a range of $(-1, 1)$, the inputs and targets must be scaled to this range. In order to accomplish this, we use the normalization $x_{\text{norm}} = \frac{x - \text{mean}(x)}{\max(\text{abs}(x - \text{mean}(x)))}$. The specific numbers used for the normalization, $\text{mean}(x)$ and $\max(\text{abs}(x - \text{mean}(x)))$, are then saved and used to normalize the test datasets to ensure proper scaling for the NN.

NN training results can vary significantly depending on the choice of loss function and optimizer. Common loss function choices include mean square error (MSE), mean absolute error, and mean squared error, and we chose to use MSE based on its performance relative to the other choices. Regarding optimizers, there are a number of common options, such as Stochastic Gradient Descent (SGD) variants, RMSPROP, and Adaptive Moment Estimation (Adam) [53]. We tested RMSPROP and Adam since they have been shown to have excellent convergence properties and are available in Keras. Adam with a learning rate of 0.001 and a batch size of 10 resulted in the smallest losses for our problem and thus is the optimizer that we use to train the NNs for this paper.

Velocity Predictions

The training method described above is used to train a LSTM NN to predict quadcopter velocities given wind velocity inputs, resulting in the losses shown in Figure 22.. In Figure 23., example predictions are shown for a 300 second segment of data with a hovering quadcopter. Figure 24. shows the error histograms for 3000 seconds of testing data with a hovering quadcopter.

After confirming that the velocity prediction works for a hovering quadcopter, we

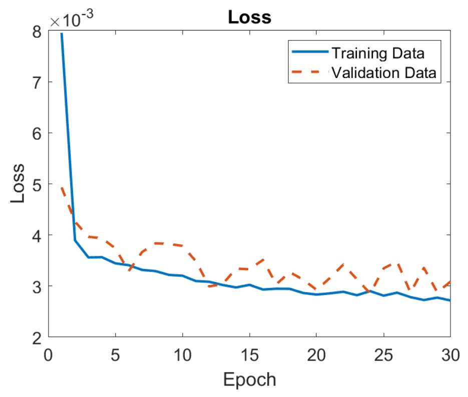

Figure 22.: Training loss for the trajectory prediction NN trained on 1,800 seconds of data for the hover case.

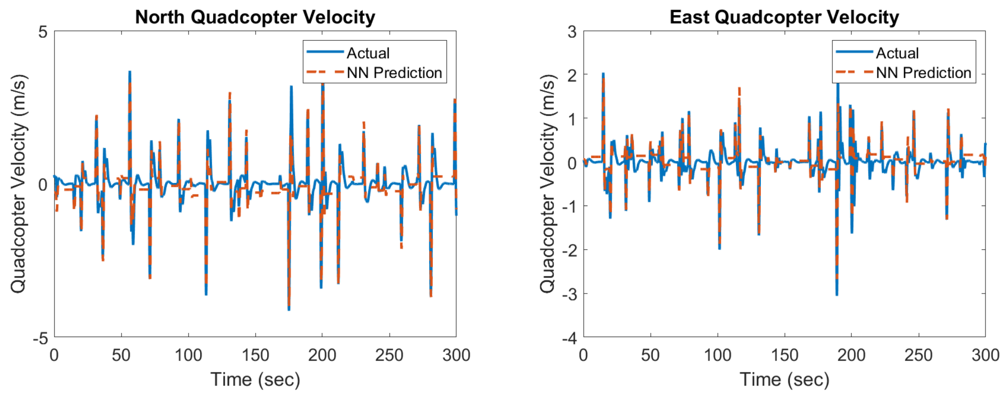

Figure 23.: Comparison of the true quadcopter velocity and the NN estimated quadcopter velocity using the test dataset for the hover case.

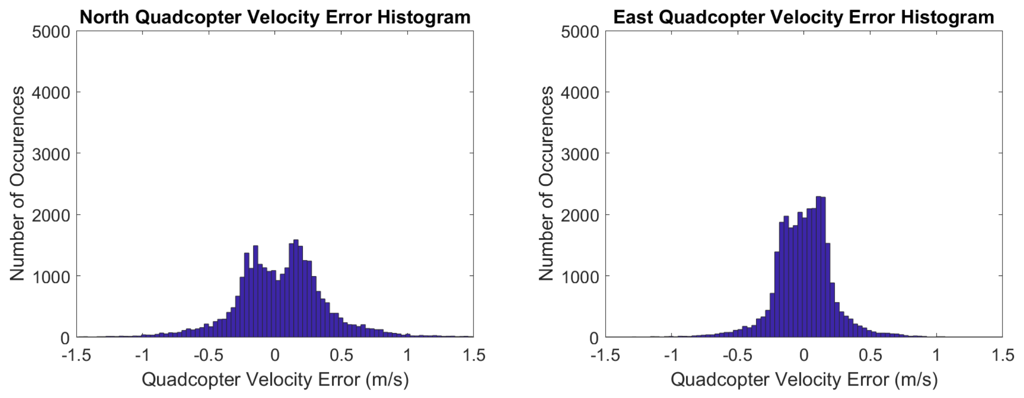

Figure 24.: Quadcopter velocity prediction error histograms for the test dataset for the hover case.

trained a NN on data from a quadcopter in straight line flight, resulting in losses shown in Figure 25.. As we saw in the hover case, this results in accurate prediction of the quadcopter's velocity in both the north and east directions for a variety of different wind speeds and directions (Figures 26.-27.).

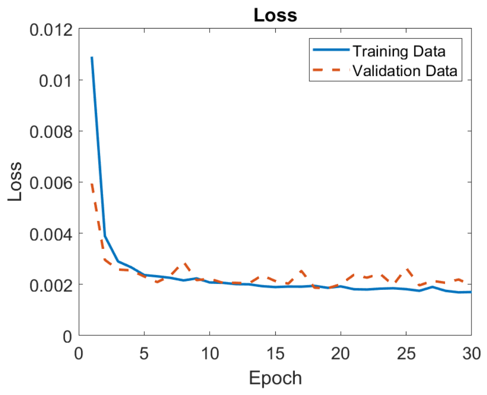

Figure 25.: Training loss for the trajectory prediction NN trained on 1,800 seconds of data for straight line flight.

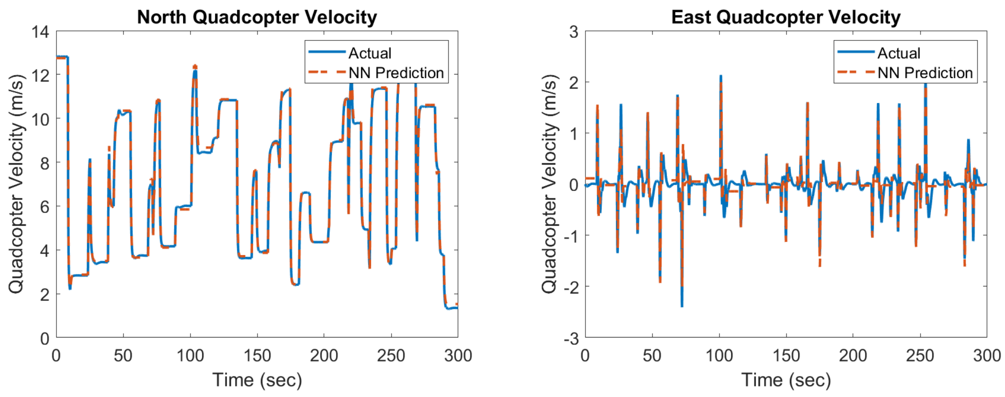

Figure 26.: Comparison of the true quadcopter velocity and the NN estimated quadcopter velocity using the test dataset for straight line flight.

Position Prediction for Zero Mean Deviations

As described in Section 2., there are two methods that can be used to obtain position deviations given wind velocity inputs: integration of the velocity predictions, or, for the case of zero mean deviation in a direction, the integration can be learned. In

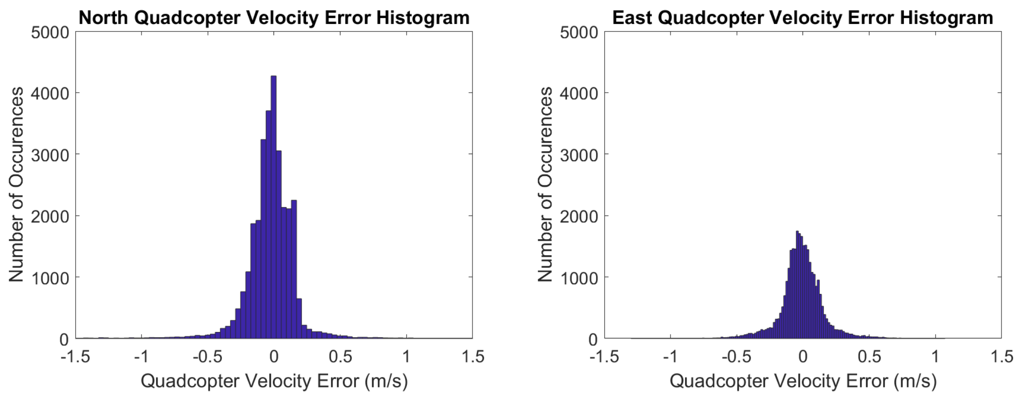

Figure 27.: Quadcopter velocity prediction error histograms for the test dataset for straight line flight.

this section, we consider the latter case, which avoids accumulation of errors due to integration. Figures 28.-29. show a comparison of the predicted and actual position deviations and an error histogram for the hover case. Figures 30.-31. show the predictions and actual values for the straight line case with the NN trained on north velocity targets and east position targets.

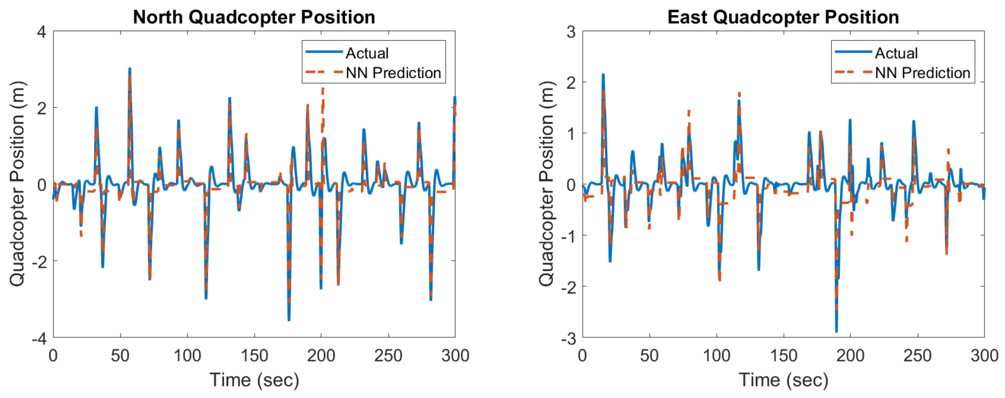

Figure 28.: Comparison of the true quadcopter position and the NN estimated quadcopter position using the test dataset for the hover case.

4. Conclusions

As shown in Section 3., the NN approach results in accurate predictions of quadcopter trajectory deviations due to wind. Using the NN instead of the linear difference

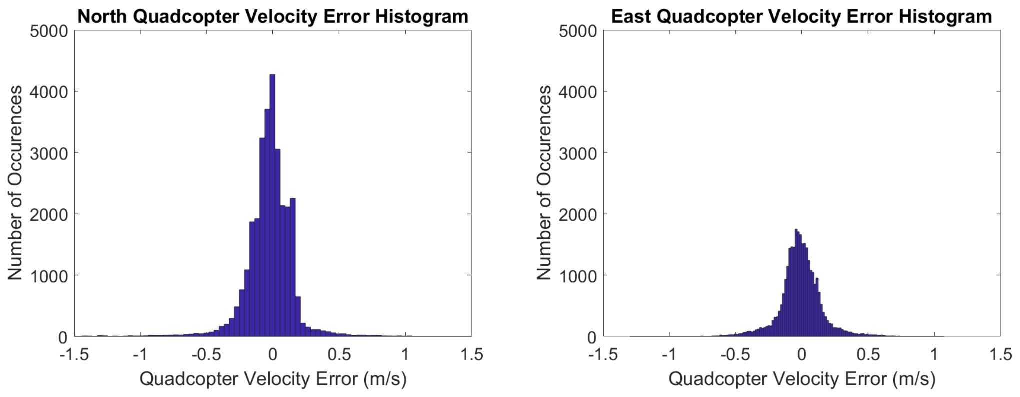

Figure 29.: Quadcopter position prediction error histograms for the test dataset for the hover case.

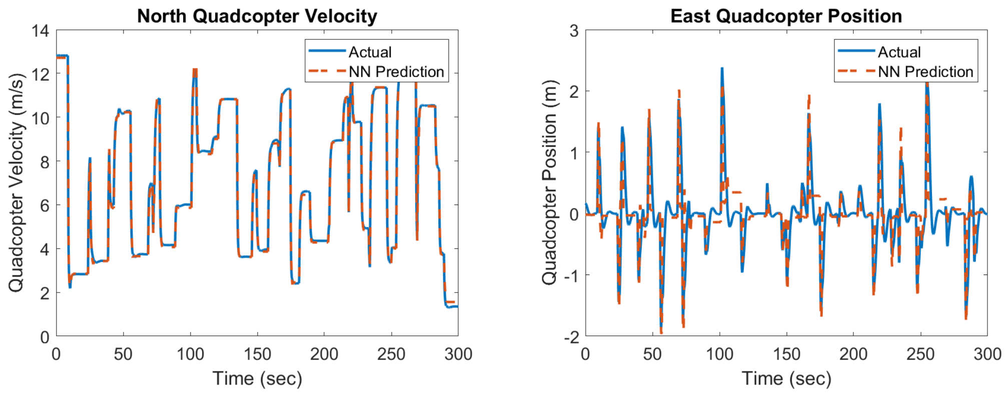

Figure 30.: Comparison of the true quadcopter north velocity and east position and the NN estimated quadcopter north velocity and east position using the test dataset for straight line flight.

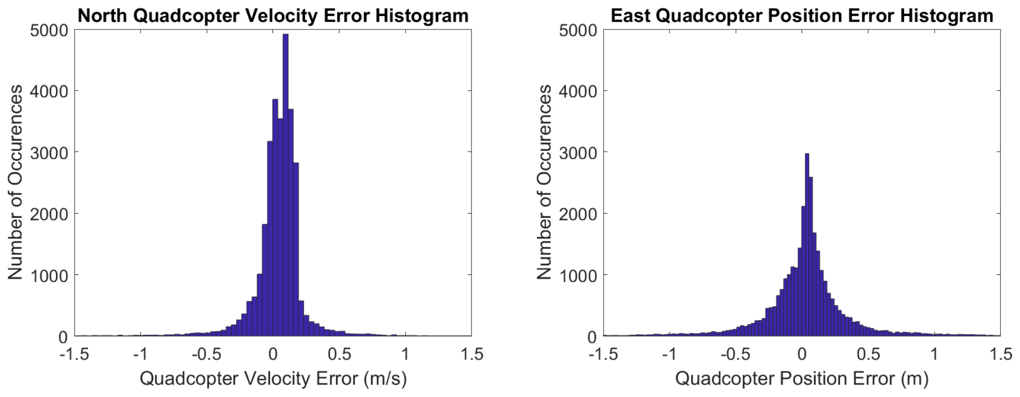

Figure 31.: Quadcopter north velocity and east position prediction error histograms for the test dataset for straight line flight.

equation approach (Chapter III) allows for predictions in winds from any direction using a single model.

CHAPTER V

WIND ESTIMATION USING NEURAL NETWORKS

In this chapter, we present a machine learning (NN) aided wind estimation strategy that leverages sensor data on factory configured sUAS. NNs have been widely used for function approximation [30] and to model complex nonlinear dynamic systems [31, 32]. Recurrent NNs (RNNs) are commonly used for sequential and dynamic systems modeling due to their ability to incorporate data from previous time steps into their predictions [26, 27]. There are a number of reasons that make the proposed NN approach an appealing solution to sUAS wind estimation. Existing linear system ID approaches [21, 22] work well for quadcopters at relatively low air-speeds. However, as roll and pitch angles increase, the nonlinearities in the dynamics become more prominent. Since NNs are effective at function approximation even in the presence of nonlinearity, they are capable of accurately capturing the quadcopter's behavior in aggressive flight. Existing nonlinear modeling techniques [23] require knowledge of the form of controller, thus, limiting their usage on commercial sUAS with proprietary controls. While the Wind Triangle (WT) approach [1, 2] is only reliable for steady state measurements, NNs can operate effectively with transient and steady data and, therefore, are more effective for turbulent wind estimation. We have chosen to use Long Short-Term Memory (LSTM) RNNs [29]. Specifically, we use LSTM NNs to estimate horizontal, 2-dimensional wind given time series of Euler angles and the inertial position of the quadcopter.

1. The Wind Estimation Problem for a Quadcopter

To realize the machine learning wind estimation approach discussed above, we first need a model of the quadcopter's dynamics. Using these dynamics, we can reformulate the wind estimation problem into a form conducive to machine learning approaches. The specific dynamics that we use are given in Section 1. and a description of the problem reformulated for machine learning is given in this section.

Wind Estimation Formulation

The quadcopter model (II.1) shows that the two primary forces impacting a quadcopter's translational motion are the drag force and the forces due to rotor thrust. The drag force is dependent on the airspeed of the quadcopter and the rotor thrusts are dependent on the Euler angles of the quadcopter. Thus, to estimate the wind velocity with minimal time delay, approximations of the velocity and the Euler angles are required, implying that the problem can be formulated as

$$V_w = f(\dot{p}, \phi, \theta, \psi). \tag{V.1}$$

There are a number of approaches that could be used to approximate the formulation in (V.1). The most straightforward approach is to attempt the approximation using linear system identification. However, considering the nonlinearities in (II.1), this would be limited to conditions relatively near hover. Thus, a more intuitive approach would be to use nonlinear modeling approaches, such as neural networks.

2. Wind Estimation using a Neural Network

Derivation of Wind Estimation Formulation

To determine the states necessary for a NN to estimate wind velocity, we seek to convert the translational dynamics of the quadcopter in (II.1) into the form $V_w = f(\cdot)$, where $f(\cdot)$ is a nonlinear function of the states. To accomplish this, we note from (II.1) that

$$\ddot{p} = ge_3 - \frac{F}{m}Re_3 + C_d(V_w - \dot{p}_n)|V_w - \dot{p}_n|, \tag{V.2}$$

where $e_3 = [0, 0, 1]^T$ and R is the rotation matrix from the body frame to the inertial frame. Using Euler discretization, we rewrite (V.2) as

$$\begin{aligned} \dot{p}(k) &\approx \dot{p}(k-1) + \Delta t \ddot{p}(k-1) && \text{(V.3)} \\ &= \dot{p}(k-1) + \Delta t[ge_3 - \frac{F(k-1)}{m}R(k-1)e_3 + C_d(V_w(k-1)... \\ &\quad - \dot{p}(k-1))|V_w(k-1) - \dot{p}(k-1)|]. && \text{(V.4)} \end{aligned}$$

Rearranging (V.4) leads to

$$\begin{aligned} &(V_w(k-1) - \dot{p}(k-1))|V_w(k-1) - \dot{p}(k-1)| \\ &\quad = C_D^{-1}\left(\frac{\dot{p}(k) - \dot{p}(k-1)}{\Delta t} + \frac{F(k-1)}{m}R(k-1)e_3 - ge_3\right). \end{aligned} \tag{V.5}$$

We let $V_a(k) = V_w(k) - \dot{p}(k)$ be the airspeed of the quadcopter at time $k\Delta t$ and further simplify (V.5) as

$$\begin{aligned} V_a(k-1)|V_a(k-1)| &= C_d^{-1}\left(\frac{\dot{p}(k) - \dot{p}(k-1)}{\Delta t} + \frac{F(k-1)}{m}R(k-1)e_3 - ge_3\right) && \text{(V.6)} \\ &= f(\dot{p}(k), \dot{p}(k-1), F(k-1), R(k-1), P), && \text{(V.7)} \end{aligned}$$

where P are constant parameters, including the drag coefficient, mass, and gravity. From (V.7), we observe that $V_a(k-1)$ is a function of the ground velocity $\dot{p}$ at times k and $k-1$ as well as the rotor thrust, F, and the rotation matrix, R, at time $k-1$.

Since $F(k-1)$ is calculated based on the controller and a constant desired waypoint, $p^d(k-1)$, it can be learned given different $p^d(k-1)$. Thus, $F(k-1)$ can be considered a part of the parameters, P, for a distant waypoint. Because the wind velocity is a difference between the ground velocity and the airspeed, the wind velocity $V_w(k-1)$ can be learned as a nonlinear function of $\dot{p}(k)$, $\dot{p}(k-1)$, $R(k-1)$, and P.

For a RNN, e.g., a LSTM NN, $\dot{p}$ can be approximated by using multiple time steps of p. This results in our final formulation

$$V_w(k-1) = f(p(k), p(k-1), ..., p(k-n), R(k), R(k-1), ..., R(k-n), P) \quad \text{(V.8)}$$

which illustrates that $V_w(k-1)$ depends on sequence data of positions and orientation.

We focus on 2-D wind and two types of trajectories for our wind estimation approach – hover and straight line. For each trajectory, we train the RNN on 4,800 seconds of data sampled at 10 Hz. The specific inputs used are north and east quadcopter positions and roll and pitch angles with north and east wind velocities as the targets. These four inputs are sufficient for estimating wind velocity in the north and east directions with zero yaw. Since we are not estimating vertical winds in this case, the altitude of the quadcopter is not necessary. Furthermore, since yaw is assumed to be constant, it will be learned as part of P, leaving us with

$$\begin{aligned} V_w(k-1) = f(&p(k), p(k-1), ..., p(k-n), \phi(k), \phi(k-1), ..., \\ &\phi(k-n), \theta(k), \theta(k-1), ..., \theta(k-n), P). \end{aligned} \quad \text{(V.9)}$$

3. Simulation Results

Data from the simulation environment described above are used to train a LSTM NN. The preliminary training case used 1,800 seconds of training data and uniformly random winds between -7 and 7 m/s, each applied for a random amount of time between 0 and 15 seconds. A sequence length of $n = 10$ that overlaps with 5 previous inputs of the previous training sequence is used. Ten percent of the sequences are randomly chosen as the validation sequences for training purposes.

For the size of the NN, we choose to use two hyperbolic tangent hidden layers with 100 units each and 10 percent dropout based on hand-tuning. Since the hyperbolic tangent function has a range of $(-1, 1)$, the inputs and targets must be scaled to this range. In order to accomplish this, we use the normalization

$x_{\text{norm}} = \frac{x - \text{mean}(x)}{\max(\text{abs}(x - \text{mean}(x)))}$. The specific numbers used for the normalization, mean(x) and max(abs($x -$ mean(x))), are then saved and used to normalize the test datasets to ensure proper scaling for the NN.

NN training results can vary significantly depending on the choice of loss function and optimizer. Common loss function choices include mean square error (MSE), mean absolute error, and mean squared error, and we chose to use MSE based on its performance relative to the other choices. Regarding optimizers, there are a number of common options, such as Stochastic Gradient Descent (SGD) variants, RMSPROP, and Adaptive Moment Estimation (Adam) [53]. We tested RMSPROP and Adam since they have been shown to have excellent convergence properties and are available in Keras. Adam with a learning rate of 0.001 and a batch size of 10 resulted in the smallest losses for our problem and thus is the optimizer that we use to train the NNs for this approach.

Training and Testing Results for Constant Winds

A plot of the training and validation loss is shown in Figure 32.. In general, having a significantly higher validation loss than training loss indicates that a NN is overfitting. As can be seen from the loss, the validation loss is slightly lower than the training loss, which indicates that the model is not overfitting. The reason that the validation loss is lower than the training loss is due to the dropout layer. Keras applies dropout to the training data but not the validation data, which can result in this behavior. Wind estimation results from the test dataset are shown in Figure 33. and an error histogram for this case is shown in Figure 34..

After validating the results using the randomly varying wind, we trained NNs on 4,800 seconds of data for two cases: First, a quadcopter in hover; Second, a quadcopter in straight line flight. These two NNs were then used to estimate the wind velocity for their respective cases on data obtained while simulating a quadcopter in turbulent

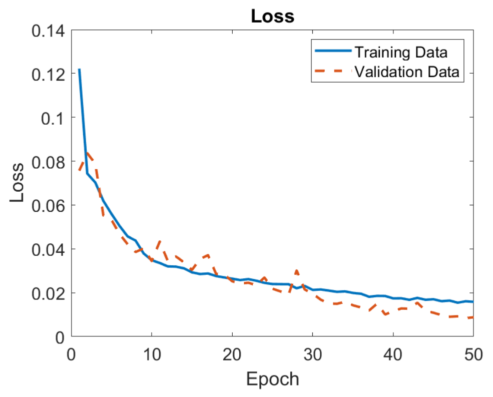

Figure 32.: Training loss for the preliminary wind estimation NN trained on 1,800 seconds of data.

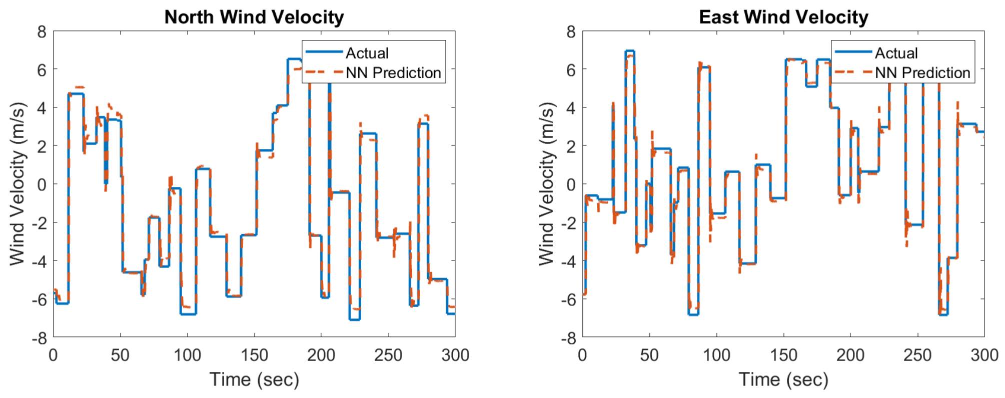

Figure 33.: Comparison of the true wind velocity and the NN estimated wind velocity using the test dataset.

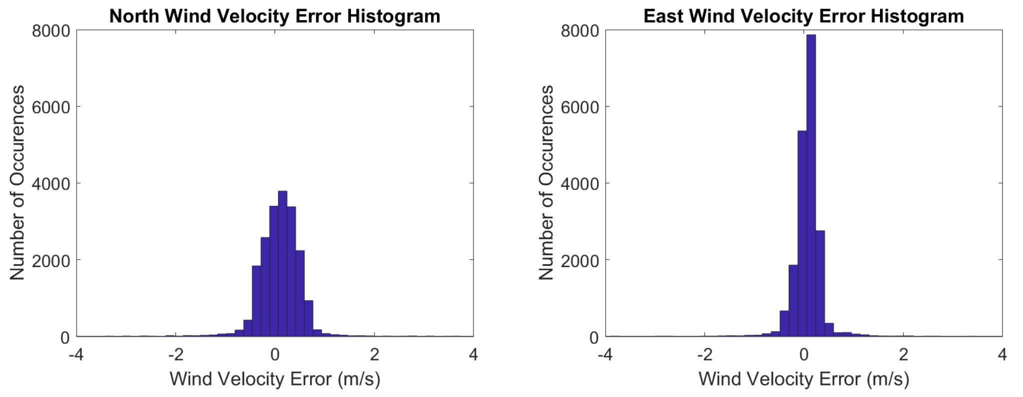

Figure 34.: Wind estimation error histograms for the test dataset. Note that wind velocity errors greater than 4 m/s occurred near the large instantaneous jumps in wind velocity but were not shown here.

wind. The loss plot for the hover case is shown in Figure 35..

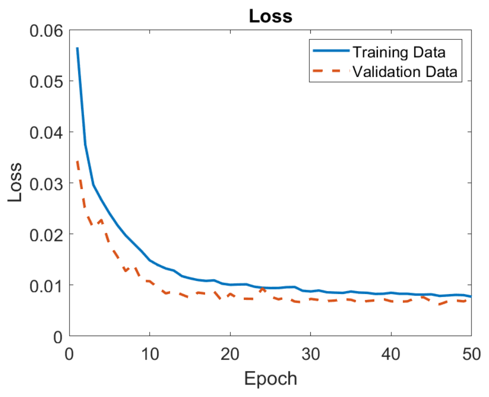

Figure 35.: Training loss for the wind estimation NN trained on 4800 seconds of data generated with a quadcopter in hover.

Dryden Wind Results

Hover Case

We consider a quadcopter in hover. This is the simplest case for wind estimation and provides an appropriate baseline for comparison. In order to generate wind estimation results, we train a LSTM NN as described in Section 2. on data collected for a hovering quadcopter. After training the NN on the wind velocity for constant winds with random jumps, we used the same NN to estimate the wind velocity for a wind with Dryden turbulence, using a range of turbulence intensities. An example segment of this estimation with Dryden turbulence and a mean wind velocity of $[1, 2, 0]^T$ m/s is shown in Figure 36.. Corresponding error histograms are shown in Figure 37. for 5,000 seconds of data.

To demonstrate the improved performance offered by the NN approach, we compare with the WT approach described in Neumann and Bartholmai [1] and Palomaki et al. [2]. In the WT approach, the Euler angles are known and assumed to correspond directly to the airspeed of the quadcopter. Then, if the ground velocity is known from a GPS, the difference between the airspeed vector and the ground velocity vector is a direct estimation of the wind velocity. Results are generated using this approach with the same datasets used for the NN approach. Wind estimation plots and error histograms using the WT are shown in Figures 38. and 39.. As can be seen from the histograms in Figures 37. and 39., the NN approach has a narrower distribution with a mean closer to zero than the WT approach. This indicates that the NN more accurately estimates the wind velocities.

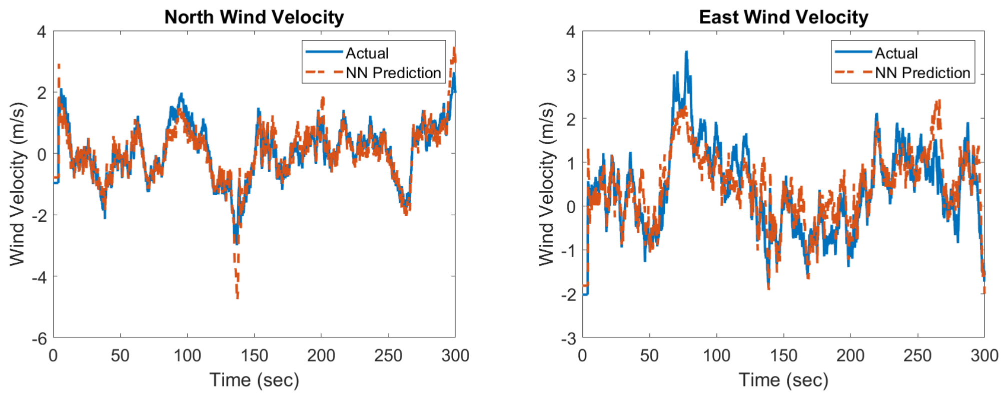

Figure 36.: Comparison of the true wind velocity and the NN predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover.

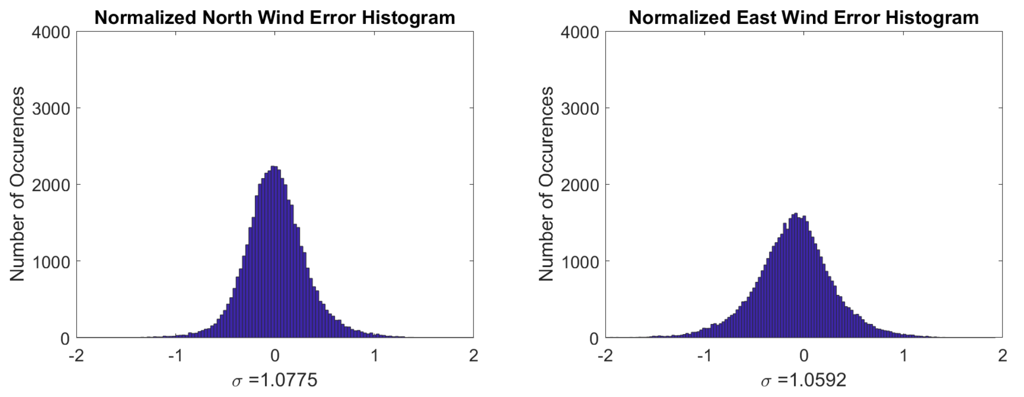

Figure 37.: Histograms of error divided by the standard deviation of the wind for the NN predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover.

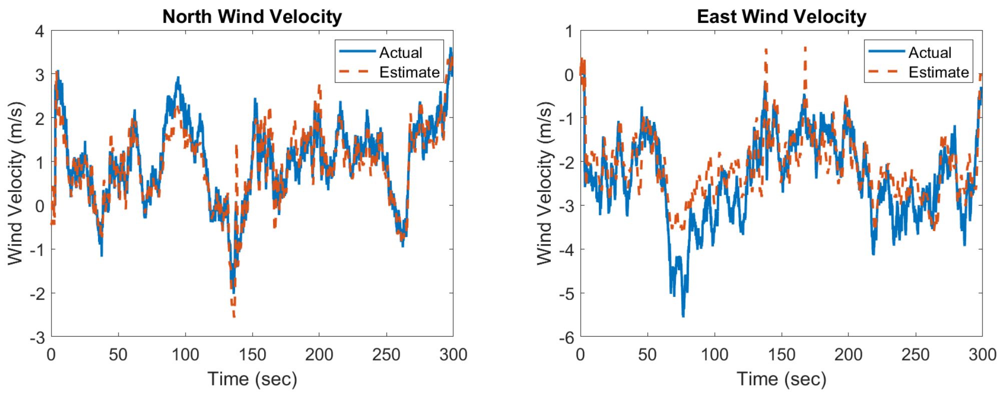

Figure 38.: Comparison of the true wind velocity and the WT predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover.

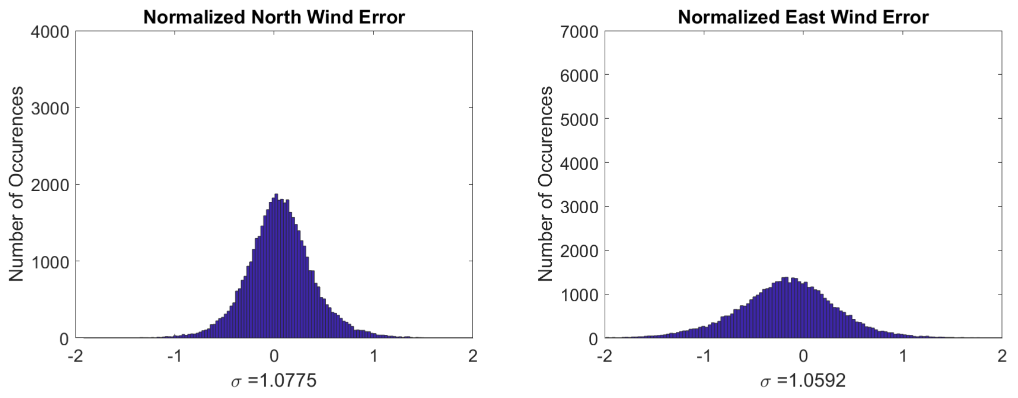

Figure 39.: Histograms of error divided by the standard deviation of the wind for the wind velocity estimates produced by the WT approach in [1,2] using the Dryden wind model with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in hover.

Straight Line Flight

After ensuring that the wind estimation strategies worked for the hover case, we applied the NN and WT approaches to data generated while a quadcopter is flying to a waypoint directly north of its initial position. We see similar behavior for this case as in the hover case, as can be seen by the NN error histogram, Figure 40., and the WT error histogram, Figure 41., with the NN approach resulting in smaller errors.

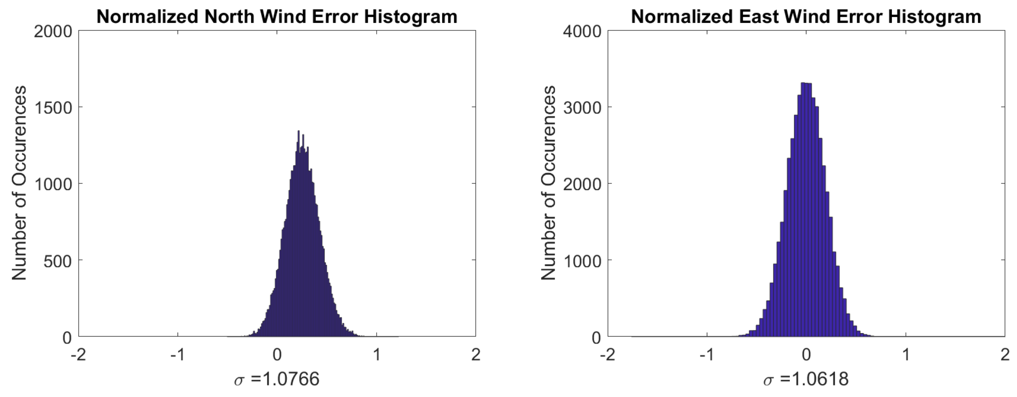

Figure 40.: Histograms of error divided by the standard deviation of the wind for the NN predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in straight line flight.

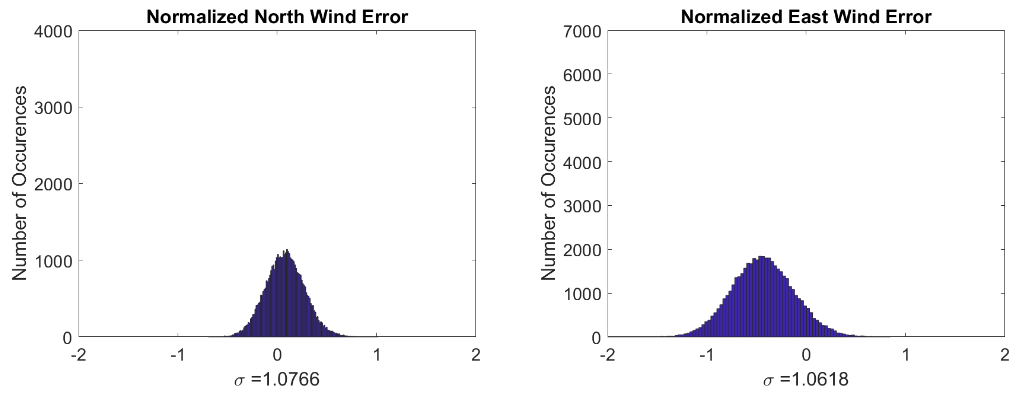

Figure 41.: Histograms of error divided by the standard deviation of the wind for the WT predicted wind velocity in Dryden wind with $\sigma_{u,v,w} = [1.06, 1.06, .7]^T$ for the waypoint navigation controller in straight line flight.

ABL Wind Estimation

Using wind and quadcopter data from a LES of the ABL gives results of a similar level of accuracy as for Dryden winds. Figure 42. displays the error histogram for the NN estimation of an ABL wind at 50m above ground level and Figure 43. shows the WT estimation error histogram for the same dataset.

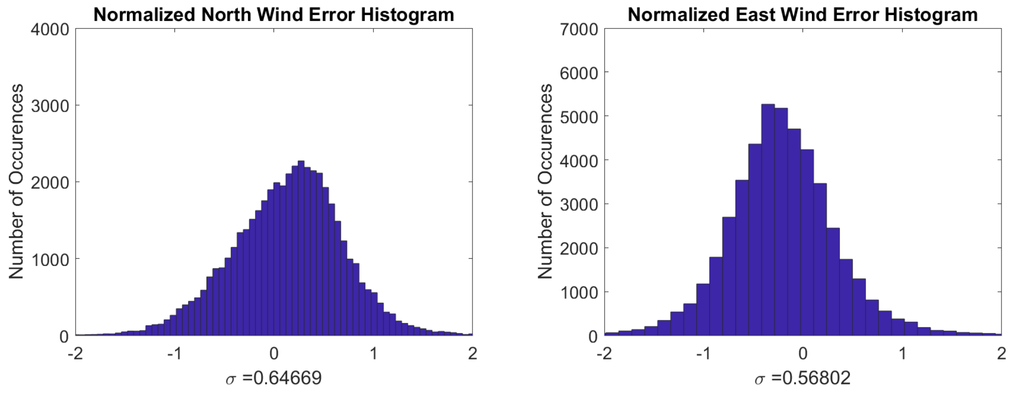

Figure 42.: Histograms of error divided by the standard deviation of the wind for a LES of the ABL wind with at an altitude of 50m for the waypoint navigation controller in hover

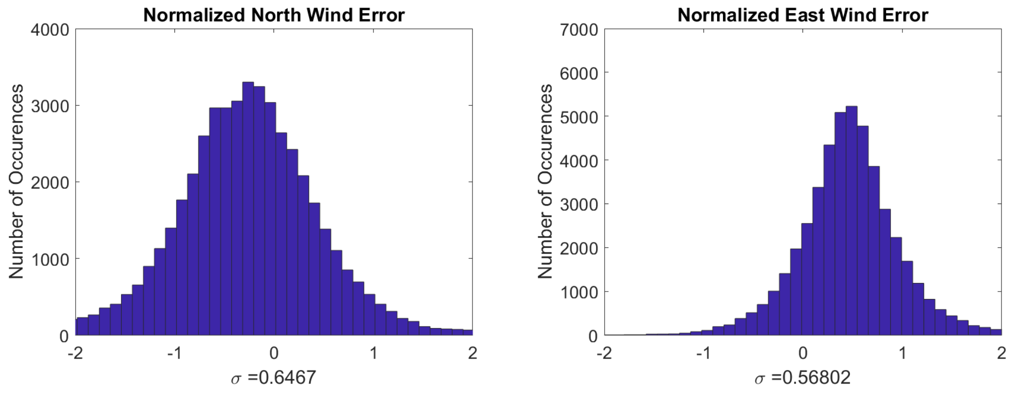

Figure 43.: Histograms of error divided by the standard deviation of the wind for the WT predicted wind velocity for a LES of the ABL wind with at an altitude of 50m for the waypoint navigation controller in hover

Comparison with the WT Approach

In order to compare the accuracy of the two methods, we calculate the mean absolute error (MAE) and the variance of the error of the estimates in the north and east directions using the same training and test datasets for the NN and for the WT. For the hover case (Tables 2. and 3.) and the straight line flight case (Table 4.) in Dryden wind, we estimate the wind velocity for two different mean winds, $V_w = [1, 2, 0]^T$ and $V_w = [2, -1, 0]^T$, and four different turbulence intensities, each with 5,000 seconds of data. In almost all cases, the NN estimates have lower MAE and variance than the WT. However, this difference becomes less pronounced as the turbulence decreases, with the WT having slightly lower north variance in Table 2. for the lowest turbulence case and slightly lower north MAE in Table 4. for the lowest turbulence case. Since the WT assumes that the tilt angle of a quadcopter corresponds directly to the airspeed of the quadcopter, errors primarily occur during the transient response of the quadcopter to a gust. Thus, as the gust intensity becomes smaller and less frequent, the WT becomes more accurate. Gust intensity affects the NN as well, but since the NN uses position measurements to approximate velocities, there is less of an impact on the wind estimation during the transient response.

Turbulence Intensities	$\sigma = [0.53, 0.53, 0.35]$		$\sigma = [1.06, 1.06, 0.6]$		$\sigma = [1.59, 1.59, 1.05]$		$\sigma = [2.12, 2.12, 1.4]$	
	NN	WT	NN	WT	NN	WT	NN	WT
North MAE (m/s)	0.268	0.3203	0.2608	0.4351	0.4354	0.5587	0.7024	0.8724
East MAE (m/s)	0.2689	0.5254	0.3234	0.551	0.5188	0.647	0.76	0.8503
North Variance $(m/s)^2$	0.128	0.0944	0.1231	0.3104	0.4	0.6304	1.175	1.7186
East Variance $(m/s)^2$	0.1222	0.1841	0.1736	0.3989	0.4989	0.7203	1.3229	1.5418

Table 2.: Comparison of mean absolute errors and error variances for a $[1, 2, 0]^T$ m/s mean Dryden wind with a quadcopter in hover.

In addition to the mean absolute errors and error variances for the Dryden wind, we considered two other metrics: mean wind velocity error and the covariance error,

Turbulence Intensities	$\sigma = [0.53, 0.53, 0.35]$		$\sigma = [1.06, 1.06, 0.6]$		$\sigma = [1.59, 1.59, 1.05]$		$\sigma = [2.12, 2.12, 1.4]$	
	NN	WT	NN	WT	NN	WT	NN	WT
North MAE (m/s)	0.173	0.5884	0.3356	0.6162	0.5137	0.7074	0.7825	0.9963
East MAE (m/s)	0.145	0.2714	0.265	0.3776	0.4558	0.5626	0.6873	0.8222
North Variance $(m/s)^2$	0.046	0.1572	0.1943	0.4422	0.5249	0.8752	1.354	2.0126
East Variance $(m/s)^2$	0.0209	0.0646	0.1175	0.2455	0.4278	0.6391	1.2426	1.6187

Table 3.: Comparison of mean absolute errors and error variances for a $[2, -1, 0]^T$ m/s mean Dryden wind with a quadcopter in hover.

Turbulence Intensities	$\sigma = [0.53, 0.53, 0.35]$		$\sigma = [1.06, 1.06, 0.6]$		$\sigma = [1.59, 1.59, 1.05]$		$\sigma = [2.12, 2.12, 1.4]$	
	NN	WT	NN	WT	NN	WT	NN	WT
North MAE (m/s)	0.1228	0.0967	0.1613	0.184	0.2176	0.2603	0.2745	0.3675
East MAE (m/s)	0.1061	0.4706	0.1597	0.499	0.2236	0.537	0.296	0.6534
North Variance $(m/s)^2$	0.0074	0.0116	0.0267	0.109	0.0564	0.0976	0.1236	0.1838
East Variance $(m/s)^2$	0.008	0.0285	0.0323	0.0461	0.0708	0.228	0.1397	0.4443

Table 4.: Comparison of mean absolute errors and error variances for a $[1, 2, 0]^T$ m/s mean Dryden wind with a quadcopter in straight line flight.

as measured by the distance metric presented in [54],

$$d = \sqrt{(\sum_{i=1}^{n} \ln^2 \lambda_i)}, \tag{V.10}$$

where d is the measured distance, λ_i are eigenvalues from $|\lambda A - B| = 0$, A is the actual covariance matrix, and B is the estimated covariance matrix for north and east wind velocities. Results for the Dryden wind case are shown in Table 5.. As can be seen from this table, the mean errors are much lower for the NN than for the WT. One interesting characteristic of the NN covariances was that the cross terms consistently had the correct sign in our tests, whereas the WT estimates did not. This is an important characteristic of turbulence to capture, and further indicates than NNs are more effective in turbulent environments.

Turbulence Intensities	$\sigma = [0.53, 0.53, 0.35]$		$\sigma = [1.06, 1.06, 0.6]$		$\sigma = [1.59, 1.59, 1.05]$		$\sigma = [2.12, 2.12, 1.4]$	
	NN	WT	NN	WT	NN	WT	NN	WT
North Mean Error (m/s)	0.0071	-0.2383	0.0112	-0.1881	0.0132	-0.1277	0.0146	-0.1422
East Mean Error (m/s)	-0.0958	-0.3915	-0.1039	-0.3291	-0.0729	-0.2514	-0.0737	-0.1769
Covariance Distance	0.1605	1.1083	0.2255	0.732	0.1592	0.5141	0.2617	0.5746

Table 5.: Comparison of mean wind errors and covariance distances for a $[1, 2, 0]^T$ m/s mean Dryden wind with a quadcopter in hover.

For our tests using data from the LES simulation of the ABL, the NN estimates of the wind velocity were more accurate than the WT estimates of the wind velocity for all of the tested cases (Table 6.). We used data from three altitudes in the ABL – 50m, 100m, and 150m. As altitude increases, the mean wind velocity increases and the turbulence deceases.

The NN approach results in better overall wind estimation accuracy than the WT for the flight scenarios the NN is trained on. This is primarily due to the fact that the WT relies on the assumption that tilt angle directly corresponds to the airspeed of the quadcopter, which is only true in the steady state condition. Since

Altitude	50m		100m		150m	
	NN	WT	NN	WT	NN	WT
North MAE (m/s)	0.2897	0.3047	0.2539	0.2807	0.2278	0.3055
East MAE (m/s)	0.2567	0.3985	0.2212	0.5432	0.2044	0.6688
North Variance $(m/s)^2$	0.1378	0.1795	0.1113	0.1276	0.0924	0.0985
East Variance $(m/s)^2$	0.1224	0.1522	0.079	0.1357	0.0664	0.1131

Table 6.: Comparison of mean absolute errors and error variances for wind from a LES simulation of the ABL at three altitudes with a quadcopter in hover.

this assumption causes a delay in the wind estimation proportional to the speed with which the attitude controller responds to a wind gust, the difference between the errors is more pronounced in higher turbulence. Therefore, the NN is more effective at measuring higher frequency components of wind velocity and the errors have significantly lower variance that the WT, albeit only on the specific trajectory type the NN is trained on. Mean errors and covariance distances for the ABL winds are shown in Table 7.. It should be noted that, although the covariance distances are larger for the NN than the WT, the NN again correctly captures the sign of the cross terms in the covariance matrices.

Altitude	50m		100m		150m	
	NN	WT	NN	WT	NN	WT
North Mean Error (m/s)	0.0496	-0.047	0.542	0.1543	0.0559	0.2785
East Mean Error (m/s)	-0.1094	-0.3111	-0.1029	-0.5318	-0.1009	-0.6804
Covariance Distance	0.7882	0.3248	0.6165	0.2548	0.5446	0.2952

Table 7.: Comparison of mean wind errors and covariance distances for wind from a LES simulation of the ABL at three altitudes with a quadcopter in hover.

In addition to the mean absolute errors and error variances for the north and east ABL wind, we considered two other metrics: wind direction and magnitude mean error and error variance. Again, we see that the NN approach results in better overall approximation of the wind than the WT approach, as demonstrated in Table 8.. Although the error variances are generally similar, the mean errors are significantly higher for the WT than for the NN, indicating better overall estimation from the NN. In this table, the direction error is the value given by $\lambda = \arccos(\cos(\arctan(V_{w,n}, V_{w,e})_{actual} - \arctan(V_{w,n}, V_{w,e})_{estimated}))$ to avoid quadrant-related errors.

Altitude	50m		100m		150m	
	NN	WT	NN	WT	NN	WT
Direction Error Variance (rad)	0.001544	0.002701	0.001315	0.001905	0.00102	0.001428
Direction Mean Error (rad)	0.0388	0.06539	0.03149	0.0715	0.02705	0.07854
Speed Error Variance $(m/s)^2$	0.1732	0.1695	0.1196	0.1189	0.1016	0.0956
Speed Mean Error $(m/s)^2$	0.1416	0.1149	0.1206	0.3701	0.1126	0.5284

Table 8.: Comparison of mean errors and error variances for the direction and normalized speed of wind from a LES simulation of the ABL at three altitudes with a quadcopter in hover.

4. Conclusions

In this chapter, we have shown a novel approach to wind estimation using quadcopter trajectory measurements. Accurate wind sensing is critical in fields such as Aviation and Meteorology, but there are gaps and inefficiencies in the current approaches – e.g., limited range with ground based towers, high cost and limited low altitude measurements when using manned aircraft, and low accuracy and limited use cases with existing sUAS wind estimation approaches. Our machine learning approach to wind estimation using state measurements seeks to mitigate these problems by providing an accurate, low cost approach to wind estimation using commercial sUAS in their default configurations.

In order for our machine learning approach to improve on the existing approaches, it must provide higher accuracy wind measurements than existing sUAS wind sensing approaches without increasing the cost of measurements or requiring knowledge of the control algorithm used. Since machine learning can provide an approximation of arbitrary nonlinear functions, this approach should, in theory, provide more accurate wind estimation than linear approaches. In particular, due to the highly nonlinear dynamics of quadcopters in aggressive flight and in turbulent wind, the machine learning approach can be expected to significantly outperform the linear approaches.

Our simulations confirmed that wind estimation using a LSTM NN outperforms linear approaches, particularly in highly turbulent wind. As shown in Section 3., the mean absolute errors and error variances for the NN were lower than those of the WT on identical datasets. This behavior was consistent for multiple turbulence levels for hover and straight line trajectories, although the difference was more marked in higher turbulence. Furthermore, we have shown that the NN has more accurate results than the WT in winds generated using both the Dryden (stochastic) wind model as well as a realistic wind field generated using a LES of the ABL.

CHAPTER VI

CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

In this thesis, we have considered two inherently related problems: quadcopter trajectory deviation prediction given wind velocity inputs and wind velocity estimation given quadcopter trajectory deviations. For the trajectory prediction problem, two approaches are considered. We began with linear difference equation identification and demonstrated its efficacy for a pure crosswind. However, the identified LDE's have reduced accuracy when predicting trajectory deviations for wind directions other than the identification case. Therefore, to aid in generalization, we presented a machine learning approach – and, in particular, we use long short-term memory neural networks for validation of the approach. Simulation results verify that the LSTM NNs result in accurate trajectory predictions, with MAE of 0.1-0.25 m/s and error variances of 0.05-0.15 $(m/s)^2$.

For the wind estimation problem, we considered nonlinear approaches exclusively, presenting a formulation of the problem suitable for general machine learning approaches, and validating the approach using LSTM NNs. Simulation results demonstrate that we are able to obtain more accurate wind velocity estimates using our approach than can be obtained using the wind triangle to estimate wind velocity.

The most immediately apparent direction for continuation of this research is experimental validation of the proposed approaches. Code for the LSTMs is written to read data from a text file in a particular format, and thus the NNs could be readily trained on properly formatted data with no required changes to the NN. A second area of research would be to use wind velocity estimates from the NNs in a closed loop controller to minimize trajectory deviations. This could allow for almost immediate trajectory corrections, resulting in very little deviation from the desired trajectory.

REFERENCES

[1] P. P. Neumann and M. Bartholmai, "Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit," *Sensors and Actuators A: Physical*, vol. 235, pp. 300–310, 2015.

[2] R. T. Palomaki, N. T. Rose, M. van den Bossche, T. J. Sherman, and S. F. De Wekker, "Wind estimation in the lower atmosphere using multirotor aircraft," *Journal of Atmospheric and Oceanic Technology*, vol. 34, no. 5, pp. 1183–1191, 2017.

[3] J. Planning and D. Office, "Unmanned aircraft systems comprehensive plan," tech. rep., Department of Transportation.

[4] P. H. Kopardekar, "Unmanned aerial system (UAS) traffic management (UTM): Enabling low-altitude airspace and uas operations," *NASA Technical Report*, 2014.

[5] M. J. Kochenderfer, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, "A comprehensive aircraft encounter model of the national airspace system," *Lincoln Laboratory Journal*, vol. 17, no. 2, pp. 41–53, 2008.

[6] E. Ranquist, M. Steiner, and B. Argrow, "Exploring the range of weather impacts on uas operations," in *18th Conference on Aviation, Range and Aerospace Meteorology*, p. 11, 2016.

[7] C. M. Belcastro, R. L. Newman, J. Evans, D. H. Klyde, L. C. Barr, and E. Ancel, "Hazards identification and analysis for unmanned aircraft system opera-

tions," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3269, 2017.

[8] FAA, "Weather-related aviation accident study 2003–2007," 2010.

[9] J. Solberg, "Susceptibility of quadcopter flight to turbulence," 2018.

[10] S. Lang and E. McKeogh, "Lidar and sodar measurements of wind speed and direction in upland terrain for wind energy purposes," *Remote Sensing*, vol. 3, no. 9, pp. 1871–1901, 2011.

[11] I. Suomi, S.-E. Gryning, E. J. O'Connor, and T. Vihma, "Methodology for obtaining wind gusts using doppler lidar," *Quarterly Journal of the Royal Meteorological Society*, vol. 143, no. 706, pp. 2061–2072, 2017.

[12] J. W. Langelaan, N. Alley, and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *Journal of Guidance Control and Dynamics*, vol. 34, no. 4, p. 1016, 2011.

[13] G. W. Donnell, J. A. Feight, N. Lannan, and J. D. Jacob, "Wind characterization using onboard imu of suas," in *2018 Atmospheric Flight Mechanics Conference*, p. 2986, 2018.

[14] S. Prudden, A. Fisher, M. Marino, A. Mohamed, S. Watkins, and G. Wild, "Measuring wind with small unmanned aircraft systems," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 176, pp. 197–210, 2018.

[15] J. Elston, B. Argrow, M. Stachura, D. Weibel, D. Lawrence, and D. Pope, "Overview of small fixed-wing unmanned aircraft for meteorological sampling," *Journal of Atmospheric and Oceanic Technology*, vol. 32, no. 1, pp. 97–115, 2015.

[16] M. Kothari, I. Postlethwaite, and D. W. Gu, “Uav path following in windy urban environments,” *Springer Journal of Intelligent and Robotic Systems*, vol. 74, pp. 1013–1028, 2013.

[17] J. Escareno, S. Salazar, H. Romero, and R. Lozano, “Trajectory control of a quadrotor subject to 2d wind disturbances,” *Springer Journal of Intelligent and Robotic Systems*, vol. 70, pp. 51–63, 2013.

[18] S. L. Waslander and C. Wang, “Wind disturbance estimation and rejection for quadrotor position control,” in *AIAA Infotech@Aerospace Conference*, AIAA, 2009.

[19] M. Bisheban and T. Lee, “Computational geometric identification for quadrotor dynamics in wind fields,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1153–1158, Aug 2017.

[20] M. Xue, “Uav trajectory modeling using neural networks,” in *17th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA, 2017.

[21] J. Gonzalez-Rocha, C. A. Woolsey, C. Sultan, and S. F. De Wekker, “Model-based wind profiling in the lower atmosphere with multirotor uas,” in *AIAA Scitech 2019 Forum*, p. 1598, 2019.

[22] J. GonzÃ¡lez-Rocha, C. A. Woolsey, C. Sultan, and S. F. J. De Wekker, “Sensing wind from quadrotor motion,” *Journal of Guidance, Control, and Dynamics*, pp. 1–18, Feb 2019.

[23] M. Bisheban and T. Lee, “Computational geometric identification for quadrotor dynamics in wind fields,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1153–1158, IEEE, 2017.

[24] J.-Y. Wang, B. Luo, M. Zeng, and Q.-H. Meng, “A wind estimation method with an unmanned rotorcraft for environmental monitoring tasks,” *Sensors*, vol. 18, no. 12, p. 4504, 2018.

[25] S. Allison, H. Bai, and B. Jayaraman, “Estimating wind velocity with a neural network using quadcopter trajectories,” in *AIAA Scitech 2019 Forum*, p. 1596, 2019.

[26] M. C. Phan, M. H. Beale, and M. T. Hagan, “A procedure for training recurrent networks,” in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1–8, IEEE, 2013.

[27] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *CoRR*, vol. abs/1409.2329, 2014.

[28] S. Allison, H. Bai, and B. Jayaraman, “Modeling trajectory performance of quadrotors under wind disturbances,” in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, p. 1237, 2018.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[30] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[31] K. S. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, March 1990.

[32] S. CHEN and S. A. BILLINGS, “Neural networks for nonlinear dynamic system modelling and identification,” *International Journal of Control*, vol. 56, no. 2, pp. 319–346, 1992.

[33] R. Beard, “Quadrotor dynamics and control rev 0.1,” 2008.

[34] T. Luukkonen, “Modelling and control of quadcopter,” *Independent research project in applied mathematics, Espoo*, vol. 22, 2011.

[35] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. Cham: Springer International Publishing, 2016.

[36] N. Sydney, B. Smyth, and D. A. Paley, “Dynamic control of autonomous quadrotor flight in an estimated wind field,” in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pp. 3609–3616, IEEE, 2013.

[37] Z. He and L. Zhao, “A simple attitude control of quadrotor helicopter based on ziegler-nichols rules for tuning pd parameters,” *The Scientific World Journal*, vol. 2014, 2014.

[38] G. M. Qian, D. Pebrianti, Y. W. Chun, Y. H. Hao, and L. Bayuaji, “Waypoint navigation of quad-rotor mav,” in *2017 7th IEEE International Conference on System Engineering and Technology (ICSET)*, pp. 38–42, IEEE, 2017.

[39] R. M. Criado and F. R. Rubio, “Autonomous path tracking control design for a comercial quadcopter,” *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 73–78, 2015.

[40] T. Von Karman, “Progress in the statistical theory of turbulence,” *Proceedings of the National Academy of Sciences*, vol. 34, no. 11, pp. 530–539, 1948.

[41] F. M. Hoblit, *Gust loads on aircraft: concepts and applications.* AIAA, 1988.

[42] Dryden, *Flying qualities of piloted aircraft.* Department of Defense, 2012. MIL-HDBK-1791B.

[43] Dryden, *Flying qualities of piloted aircraft.* Department of Defense, 1990. MIL-STD-1797A.

[44] J. R. Schiess, "Composite statistical method for modeling wind gusts," *Journal of Aircraft*, vol. 23, no. 2, pp. 131–135, 1986.

[45] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft - Theory and Practice.* Princeton University Press, 2012.

[46] B. Jayaraman and J. G. Brasseur, "Transition in atmospheric boundary layer turbulence structure from neutral to moderately convective stability states and implications to large-scale rolls," *arXiv preprint arXiv:1807.03336*, 2018.

[47] S. B. Pope, "Turbulent flows," 2001.

[48] D. A. Forrester and G. C. Dean, "Improvement of meteorological data for air traffic management purposes," *Air Traffic Control Quarterly*, vol. 2, no. 2, pp. 85–101, 1994.

[49] K. J. Keesman, *System identification: an introduction.* Springer Science & Business Media, 2011.

[50] L. Ljung, *System identification: Theory for the user.* Prentice Hall, New Jersey, 1999.

[51] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *IEEE International Conference on Robotics and Automation*, IEEE, 2015.

[52] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, (Kobe, Japan), May 2009.

[53] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[54] W. Förstner and B. Moonen, “A metric for covariance matrices,” in *Geodesy-The Challenge of the 3rd Millennium*, pp. 299–309, Springer, 2003.

VITA

Sam Allison

Candidate for the Degree of

Master of Science

Thesis: QUADCOPTER TRAJECTORY PREDICTION AND WIND ESTIMATION USING MACHINE LEARNING

Major Field: Mechanical and Aerospace Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Mechanical and Aerospace Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2019.

Completed the requirements for the Bachelor of Science in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in 2016.

Experience:

Research Assistant at Oklahoma State University, Aug. 2015 - May 2019

Teaching Assistant at Oklahoma State University, Aug. 2018 - May 2019

Professional Memberships:

American Institute of Aeronautics and Astronautics (AIAA)