RUNWAY SAFETY IMPROVEMENTS THROUGH A DATA DRIVEN APPROACH FOR RISK FLIGHT PREDICTION AND SIMULATION

A Thesis
Presented to
The Committee Members

By

HyunKi Lee

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering
Department of Engineering

Georgia Institute of Technology

December 2022

RUNWAY SAFETY IMPROVEMENTS THROUGH A DATA DRIVEN APPROACH FOR RISK FLIGHT PREDICTION AND SIMULATION

Approved by:

Prof. Dimitri N. Mavris
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Alexia P. Payan
Aerospace Systems Design Laboratory
Georgia Institute of Technology

Prof. Daniel P. Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Angela Campbell
William J. Hughes Technical Center
Federal Aviation Administration

Dr. Tejas G. Puranik
Ames Research Center
National Aeronautics and Space Administration

Dr. Young Jun Choi
Advanced Technology Group
United Parcel Service

Date Approved: December 9, 2022

to Ji-Eun and Tan Tan

ACKNOWLEDGMENTS

To start, I would like to express my deepest gratitude to Jesus Christ, my Lord. Looking back, He has always provided me with the strength I need to overcome all of the challenges during the graduate program. I give all of the glory and honor that I have earned to Him. I would also like to express my deepest gratitude to my wife Ji-Eun Kim. I thank her for all of the support and patience. She helped me have faith in my work and push on until the end. Also, without her I would never have started the graduate program.

I am extremely grateful to my committee members. First of all, I would like to thank Professor Mavris. He has accepted me to the Aerospace Systems Design Laboratory (ASDL) and taught me all the knowledge I need to work on this thesis. I appreciate Professor Schrage and Dr. Payan for reviewing my work. I thank Dr. Campbell for sharing her expertise and knowledge. They were critical for interpreting the problem in the regulator's perspective. I thank Dr. Choi for the frequent meeting where he shared his knowledge on how I could approach the research questions. Lastly, I am grateful to Dr. Puranik for bringing me to the world of flight safety. I would never have thought of doing research related to flight safety without doing my first project with him in the graduate program. He introduced me to the world of python, machine learning, and guided me through multiple publications.

I would like to give my special thanks to David Anvid from the ASDL and Captain John Lande from the Norwegian Airlines. They shared their perspective on my research questions from the operators perspective. Their knowledge provided me with a clear goal to strive for based on the need from the aviation industry. I learned through our meetings that it is crucial for researchers to always communicate and learn.

HyunKi Lee

Atlanta, GA

December 2022

TABLE OF CONTENTS

Acknowledgments iv

List of Tables ix

List of Figures xi

List of Acronyms xvii

Summary xxii

Chapter 1: Motivation 1

Chapter 2: Background and Literature 9

2.1 Landing Process Analysis 11

2.1.1 Approach 13

2.1.2 Landing 16

2.1.3 Ground Roll 19

2.1.4 Overrun 21

2.2 Literature Review 21

2.2.1 Prediction 22

2.2.2 Simulation 27

2.2.3 Prediction and Simulation for Mitigating Runway Overruns 42

Chapter 3: Problem Formulation 49

3.1 Research Objective 49

3.2 Research Questions 51

3.2.1 Accident Precursor Prediction 52

3.2.2 Aircraft System Identification 63

Chapter 4: Proposed Methodology 73

4.1 Increase Awareness to Touchdown Condition with Prediction Model 73

4.2 Improve Flight Control with Dynamics Model 82

4.3 Maximize the Model Accuracy with Hyperparameter Optimization Algorithm 86

Chapter 5: Hyperparameter Optimization Algorithm Development 91

5.1 Development of Optimization Algorithm 94

5.2 Algorithm Performance 96

5.3 Optimizing Prediction and Dynamic Models 98

Chapter 6: Development of Prediction Model for Aircraft States above Touchdown 103

6.1 Data Pre-Processing 104

6.2 Base Model Structure 109

6.3 Initial Model Evaluation 111

6.4 Prediction Model From A Different Perspective 118

6.5 Clustered Model Evaluation 123

6.6 Input Optimized Model Evaluation 127

6.7 Prediction Model Sensitivity 131

6.7.1 Significant Parameter Identification 132

6.7.2 Flare 134

6.7.3 Wind 138

Chapter 7: Development of Dynamics Model for Final Approach Simulation . . 142

7.1 Data Pre-Processing 143

7.2 Base Model Structure 149

7.3 Initial Model Evaluation 150

7.4 Input Optimized Model Evaluation 158

7.5 Dynamic Model Performance 162

7.5.1 Flare 164

7.5.2 Wind 167

7.5.3 Feasible Control Space 169

Chapter 8: Demonstration of the Proposed Model for Improving Final Approach 172

8.1 Interpreting the Prediction and Simulation Results Based on Metrics Associated to Overrun 173

8.2 Evaluating Potential Risk with Overrun Precursors 176

8.3 Improving Control Trajectory to Minimize Potential Risk 179

Chapter 9: Conclusion 183

9.1 Findings and Recommendations 186

9.2 Future Works 191

References 194

Vita 209

LIST OF TABLES

1.1 Accident Abbreviations 3

1.2 GAPPRE Recommendations 7

2.1 Approach Stabilization Criteria 14

2.2 Percent Contribution to Overrun 21

2.3 Ground Speed Prediction with Machine Learning 24

2.4 Percent Contribution of External Threats 47

2.5 Landing Distance Multipliers 48

3.1 Ground Speed Prediction with Neural Network Models 55

3.2 Prediction Model Parameters and Accuracy 59

3.3 FOQA Aircraft Parameters 66

5.1 model parameters 97

5.2 model parameters 98

5.3 model parameters 99

6.1 state abbreviations 107

6.2 model parameters 128

7.1 control abbreviations 145

7.2 model parameters 159

9.1 Ground Speed Prediction with Machine Learning 187

9.2 Results comparison of machine learning simulation models of final approach 188

LIST OF FIGURES

1.1 Accidents From 2017 to 2021 2

1.2 Flight Accident by Type and Flight Phase From 2017 to 2021 2

1.3 Fatal Accident Rate Per Million Flights 7

2.1 Precursors Contribution to Runway Excursion 12

2.2 Continuous Descent Final Approach 15

2.3 Ground Speed Distribution 16

2.4 Aircraft Dynamics Notations 30

2.5 Runway Excursion Precursors 45

3.1 Overarching Methodology 51

3.2 Experiment 1.1 57

3.3 Input Structure 58

3.4 Updated RQ1 62

3.5 Experiment 2.1 68

3.6 Model Design Method 69

3.7 Updated RQ2 72

4.1 Example Long Short-Term Memory (LSTM) neural network cell 77

4.2 Example Gated Recurrent Units (GRU) neural network cell 78

4.3 Example CNN cell 79

4.4 Example Neural Network Nonlinear Autoregressive with Exogenous (NNARX) model 84

4.5 Example Recursive Neural Network (RNN) model 85

4.6 Example Gated Neural Networks (GNN) model 86

5.1 Illustration of the genetic algorithms (GA) processes 91

5.2 Illustration of the Quantum Inspired Genetic Algorithm (QIGA) processes . 92

5.3 Illustration of the Ackley (left) and Schwefel (right) test functions 93

5.4 Illustration of how samples are made and optimized with Hybrid Genetic Algorithm (HGA) algorithm 95

5.5 Example illustration of hyperband 101

6.1 Prediction model development process overview 104

6.2 Prediction model architecture 104

6.3 Aircraft longitudinal position, velocity (left), and attitude (right) recorded for all of the final approach data 105

6.4 Aircraft longitudinal acceleration and wind speeds recorded for all of the final approach data 106

6.5 Data structure for the prediction model input (left) and output (right) 107

6.6 Error distribution of prediction data index due to difference in how the data is collected in flight 111

6.7 Test prediction results (red) vs actual (blue) Calibrated Air Speed (CAS) and True Air Speed (TAS) 112

6.8 Test prediction results (red) vs actual (blue) Inertial Vertical Velocity (IVV) and Pitch Angle (PTCH) 112

6.9 Test prediction Root Mean Square Error (RMSE) (left) and relative (right) error distributions for CAS and TAS 113

6.10 Minimum and maximum range of each prediction features across prediction altitudes 114

6.11 Polynomial prediction model of order six selected (green) based on the improvements in RMSE with increasing order 115

6.12 Polynomial regression model test prediction RMSE (left) and relative (right) error distributions 116

6.13 Test prediction RMSE (left) and relative (right) error distributions for IVV and PTCH 116

6.14 Polynomial regression model of PTCH RMSE (left) and relative (right) error distributions 117

6.15 Polynomial regression model of IVV RMSE (left) and relative (right) error distributions 118

6.16 Neural network model of CAS and TAS, trained based on time index, test prediction RMSE (left) and relative (right) error distributions 119

6.17 Neural network model of CAS and TAS, trained without averaging, test prediction RMSE (left) and relative (right) error distributions 120

6.18 RMSE (left) and relative (right) error distributions of CAS and TAS neural network model trained and tested with autopilot flights 121

6.19 Neural network model of IVV and PTCH, trained with data indexed by time, test prediction RMSE (left) and relative (right) error distributions . . . 121

6.20 Neural network model of IVV and PTCH, trained without averaging, test prediction RMSE (left) and relative (right) error distributions 122

6.21 RMSE (left) and relative (right) error distributions of IVV and PTCH neural network model trained and tested with autopilot flights 123

6.22 Step wise difference matrix between trajectories of different size and the difference minimizing path shown with blue errors [165] 124

6.23 Group 1 of the final approach data clustered in frequency domain 125

6.24 Group 2 of the final approach data clustered in frequency domain 125

6.25 Group 3 of the final approach data clustered in frequency domain 126

6.26 Group 4 of the final approach data clustered in frequency domain 126

6.27 Neural network model for CAS and TAS, trained with clustered data, test prediction RMSE (left) and relative (right) error distributions 126

6.28 Neural network model for IVV and PTCH, trained with clustered data, test prediction RMSE (left) and relative (right) error distributions 127

6.29 Hyerparameter optimized neural network model for TAS, test prediction RMSE distribution 129

6.30 Hyperparameter optimized neural network model for IVV and PTCH, test prediction RMSE distributions 129

6.31 Permutation importance plot of the top 11 model features for the hyperparameter optimized prediction model, the cumulative percent contribution curve (red) of the model features based on their importance and the prediction MSE (yellow) at which the permuted feature would have no impact . . 133

6.32 Flare initiation identification process, where the red lines indicate zero pitch and pitch rate and the green lines indicate the points at which pitch of zero, pitch rate of zero, and difference in elevator control of zero are crossed. . . 135

6.33 Distribution of Height Above Touchdown (HAT) and TAS at flare initiation time 136

6.34 Distribution of flare initiation altitude in 20 ft interval 136

6.35 Prediction RMSE for IVV based on predictions above or below the start of flare and based on flare initiation altitudes 137

6.36 Prediction RMSE for TAS and PTCH parameters based on predictions above or below the start of flare and based on flare initiation altitudes 137

6.37 Prediction RMSE for IVV based on predictions different wind speed conditions 139

6.38 Prediction RMSE for PTCH and TAS parameters based on predictions different wind speed conditions 140

7.1 Dynamics model development process overview 143

7.2 Dynamics model architecture 143

7.3 Aircraft control parameters recorded for all of the final approach data. Here controls 1 and 2 represent two different surfaces for each control. 144

7.4 Data structure for the dynamics model input (blue) and output (yellow) . . . 147

7.5 Summary plots of the NNARX based simulation prediction results 150

7.6 Trajectory plots of the NNARX based simulation prediction results 151

7.7 Summary plots of the NNARX based simulation prediction results 152

7.8 Summary plots of the RNN based simulation prediction results 153

7.9 Two sample plots of the RNN based simulation prediction results 153

7.10 Summary plots of the LSTM based simulation prediction results 154

7.11 Two sample plots of the LSTM based simulation prediction results 155

7.12 Summary plots of the GRU based simulation prediction results 156

7.13 Two sample plots of the GRU based simulation prediction results 156

7.14 Summary plots of the LSTM based simulation prediction results trained with earth frame data 157

7.15 Two sample plots of the LSTM based simulation prediction results trained with earth frame data 157

7.16 Summary plots of the LSTM based simulation prediction results trained with hyperparameter optimized configuration 160

7.17 Two sample plots of the LSTM based simulation prediction results trained with hyperparameter optimized configuration 160

7.18 The distribution of ELEV feature for all of the flights 164

7.19 The error distribution based on the start of flare and elevator deflection . . . 165

7.20 The data distribution and model performance map based on ELEV and HAT features 166

7.21 The error distribution based on large wind speed or wind direction change . 167

7.22 The data distribution and model performance map based on WLAT and WLONG features 168

7.23 The data distribution between ELEV, ROLL, PTCH, DA, and PLA features 169

7.24 The model performance map based on ELEV, ROLL, PTCH, DA, and PLA features 170

8.1 Overview of how the prediction and simulation models can be used for pilot training 173

8.2 CAS distribution for all the data 177

8.3 confusion matrix of high approach speed (right) and the time difference between simulated high approach speed trigger and actual high approach speed trigger (left) 177

8.4 confusion matrix of high approach speed based on results below 200 ft HAT 178

8.5 Demonstration: Prediction 180

8.6 Demonstration of how the prediction and simulation models can be used in a training environment to get in flight and post-flight feedback to reduce the risk of overrun 181

9.1 Overarching Methodology 184

LIST OF ACRONYMS

ANN Artificial Neural Network

AOA Angle Of Attack

ASDE-X Airport Surface Detection Equipment model X

ASK Attitude Skill and Knowledge

ATAP Airport Taxiway Arrival Prediction

ATL Hartsfield–Jackson Atlanta International Airport

BO Bayesian Optimization

CAS Calibrated Air Speed

CDFA Continuous Descent Final Approach

CNN Convolutional Neural Network

DE Differential Evolution

DTW Dynamic Time Warping

EBT Evidence Based Training

EMAS Engineered Material Arresting System

ET Guidance Material for Instructor and Evaluator Training

FAA Federal Aviation Administration

FAF Final Approach Fix

FF Feed Forward

FOQA Flight Operational Quality Assurance

FSF Flight Safety Foundation

GA genetic algorithms

GAPPRE Global Action Plan for the Prevention of Runway Excursions

GNN Gated Neural Networks

GRU Gated Recurrent Units

GS Ground Speed

HAT Height Above Touchdown

HGA Hybrid Genetic Algorithm

ICAO International Civil Aviation Organization

IMC Instrumental Meteorological Conditions

IVV Inertial Vertical Velocity

JAA Joint Aviation Authorities

LSTM Long Short-Term Memory

MIMO Multi Input Multi Output

MISO Multi Input Single Output

MSE Mean Squared Error

NNARX Neural Network Nonlinear Autoregressive with Exogenous

NTSB National Transport Safety Board

PTCH Pitch Angle

QIGA Quantum Inspired Genetic Algorithm

RAAS Runway Awareness and Advisory System

RIM Runway Incursion Mitigation

RMSE Root Mean Square Error

RNN Recursive Neural Network

ROAAS Runway Overrun Awareness and Alerting System

ROD Rate Of Descent

ROPS Runway Overrun Prevention System

RSM Runway Safety Metric

RWSL RunWay Status Lights

SA Simulated Annealing

SOP Standard Operating Procedure

TAS True Air Speed

TCH Threshold Crossing Height

VMC Visual Meteorological Conditions

SUMMARY

Runway overrun is one of the most frequently occurring flight accident types threatening the safety of aviation. Sensors have been improved with recent technological advancements and allow data collection during flights. The recorded data helps to better identify the characteristics of runway overruns. The improved technological capabilities and the growing air traffic led to increased momentum for reducing flight risk using artificial intelligence. Discussions on incorporating artificial intelligence to enhance flight safety are timely and critical. Using artificial intelligence, we may be able to develop the tools we need to better identify runway overrun risk and increase awareness of runway overruns. This work seeks to increase attitude, skill, and knowledge (ASK) of runway overrun risks by predicting the flight states near touchdown and simulating the flight exposed to runway overrun precursors.

To achieve this, the methodology develops a prediction model and a simulation model. During the flight training process, the prediction model is used in flight to identify potential risks and the simulation model is used post-flight to review the flight behavior. The prediction model identifies potential risks by predicting flight parameters that best characterize the landing performance during the final approach phase. The predicted flight parameters are used to alert the pilots for any runway overrun precursors that may pose a threat. The predictions and alerts are made when thresholds of various flight parameters are exceeded. The flight simulation model simulates the final approach trajectory with an emphasis on capturing the effect wind has on the aircraft. The focus is on the wind since the wind is a relatively significant factor during the final approach; typically, the aircraft is stabilized during the final approach. The flight simulation is used to quickly assess the differences between fight patterns that have triggered overrun precursors and normal flights with no abnormalities. The differences are crucial in learning how to mitigate adverse flight conditions. Both of the models are created with neural network models. The main challenges of

developing a neural network model are the unique assignment of each model design space and the size of a model design space. A model design space is unique to each problem and cannot accommodate multiple problems. A model design space can also be significantly large depending on the depth of the model. Therefore, a hyperparameter optimization algorithm is investigated and used to design the data and model structures to best characterize the aircraft behavior during the final approach.

A series of experiments are performed to observe how the model accuracy change with different data pre-processing methods for the prediction model and different neural network models for the simulation model. The data pre-processing methods include indexing the data by different frequencies, by different window sizes, and data clustering. The neural network models include simple Recurrent Neural Networks, Gated Recurrent Units, Long Short Term Memory, and Neural Network Autoregressive with Exogenous Input. Another series of experiments are performed to evaluate the robustness of these models to adverse wind and flare. This is because different wind conditions and flares represent controls that the models need to map to the predicted flight states. The most robust models are then used to identify significant features for the prediction model and the feasible control space for the simulation model. The outcomes of the most robust models are also mapped to the required landing distance metric so that the results of the prediction and simulation are easily read. Then, the methodology is demonstrated with a sample flight exposed to an overrun precursor, and high approach speed, to show how the models can potentially increase attitude, skill, and knowledge of runway overrun risk.

The main contribution of this work is on evaluating the accuracy and robustness of prediction and simulation models trained using Flight Operational Quality Assurance (FOQA) data. Unlike many studies that focused on optimizing the model structures to create the two models, this work optimized both data and model structures to ensure that the data well capture the dynamics of the aircraft it represents. To achieve this, this work introduced a hybrid genetic algorithm that combines the benefits of conventional and quantum-inspired

genetic algorithms to quickly converge to an optimal configuration while exploring the design space. With the optimized model, this work identified the data features, from the final approach, with a higher contribution to predicting airspeed, vertical speed, and pitch angle near touchdown. The top contributing features are altitude, angle of attack, core rpm, and air speeds. For both the prediction and the simulation models, this study goes through the impact of various data preprocessing methods on the accuracy of the two models. The results may help future studies identify the right data preprocessing methods for their work. Another contribution from this work is on evaluating how flight control and wind affect both the prediction and the simulation models. This is achieved by mapping the model accuracy at various levels of control surface deflection, wind speeds, and wind direction change. The results saw fairly consistent prediction and simulation accuracy at different levels of control surface deflection and wind conditions. This showed that the neural network-based models are effective in creating robust prediction and simulation models of aircraft during the final approach. The results also showed that data frequency has a significant impact on the prediction and simulation accuracy so it is important to have sufficient data to train the models in the condition that the models will be used. The final contribution of this work is on demonstrating how the prediction and the simulation models can be used to increase awareness of runway overrun.

CHAPTER 1

MOTIVATION

The aviation industry has faced a steady increase in air traffic volume worldwide until 2019. In 2020, the total number of air passengers decreased by 60% after the coronavirus (COVID-19) struck the world. The International Civil Aviation Organization (ICAO) anticipates between 1.2% to 3.6% compound annual growth rate of revenue passenger kilometers between 2018 and 2028 [1]. This is below the 4.2% compound annual growth rate anticipated before COVID. Despite the dampened air passenger traffic, the air freight traffic demand remains strong with a relatively insignificant 6.2% decrease in October 2020, from 2019 [2], and an 8.1% increase in February 2022 [3].

The accident rate remains unchanged, unlike the number of flights, at about 1 accident per million for 2019 and 2021, for aircraft with maximum takeoff weight over 12600 lbm [4], as shown in Figure 1.1. The distribution of flight accidents by type, as shown in Figure 1.2, shows that runway excursion is the accident type that occurs most frequently. A description of the abbreviations used in the Figure 1.2 is shown in Table 1.1. As indicated, runway excursion contributes to 25 percent of all accidents [4]. The fatal accident distribution, as shown in Figure 1.2, shows that runway excursion contributes close to 20 percent of all fatal accident flights. A fatal accident is defined as any accident that resulted in at least one fatality or an injury leading to death. The frequency and fatality of runway excursions make it one of the most frequently investigated accidents to reduce flight risk.

The Economic impact of runway excursions is also significant. For flights between the years 1985 and 2010, Airbus-Willis estimated the cost due to hull loss from landing excursion is $5,429 MUSD [5]. This is nearly double the cost of accidents that occurred during the approach phase ($2,937 MUSD), which is the second highest in cost due to hull loss. Including liability, the cost due to the landing excursion is estimated at $6,562 MUSD.

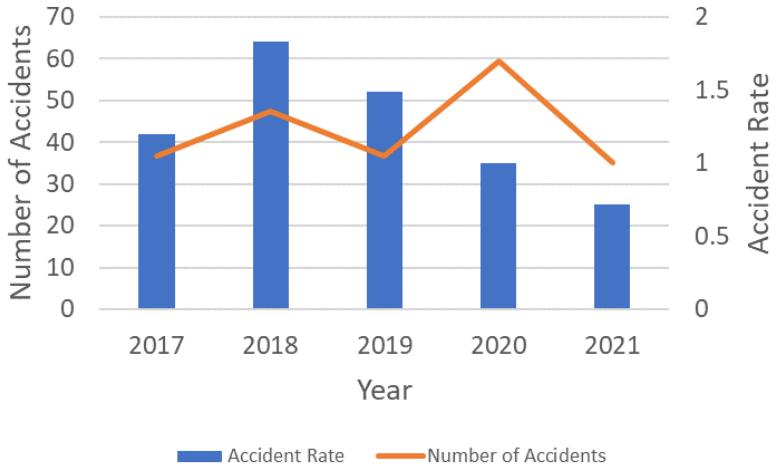

Figure 1.1: Number of flight accidents and accident rates from 2017 to 2021 for both turbojet and turboprop aircraft with maximum takeoff weight over 12600 lbm [4].

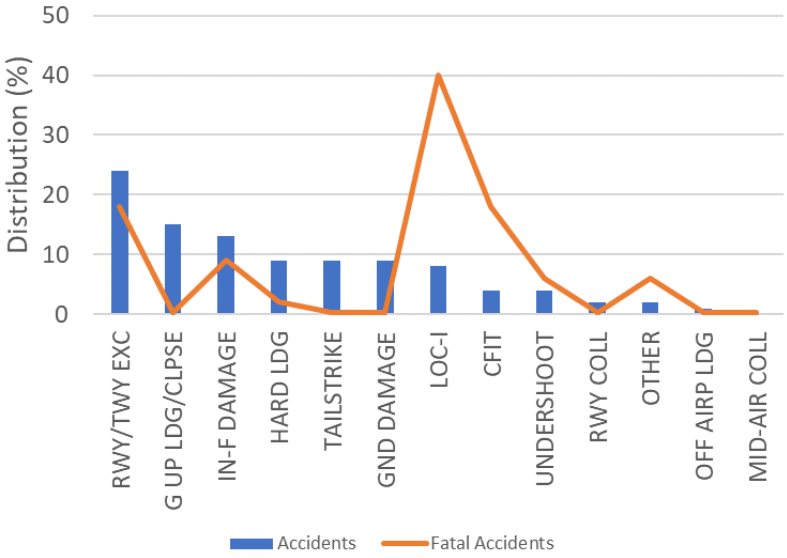

Figure 1.2: All flight accident (blue) and fatal flight accident (orange) distribution from year 2017 to 2021 for both turbojet and turboprop aircraft with maximum takeoff weight over 12600 lbm [4].

This is over 39 percent of the total cost ($16,542 MUSD) due to flight accidents at landing between 1985 and 2010. In addition, the costs from hull loss have been trending upward since 1985 and will continue to rise without a change that will shift the trend. Therefore,

Table 1.1: Accident Type Abbreviations

Abbreviation	Accident Type
RWY/TWY EXC	Runway/Taxiway Excursion
G UP LDG/CLPSE	Gear-Up Landing/Gear Collapse
IN-F Damage	In-Flight Damage
HARD LDG	Hard Landing
GND DAMAGE	Ground Damage
TAILSTRIKE	Tail Strike
LOC-I	Loss of Control - In-Flight
UNDERSHOOT	Undershoot
RWY COLL	Runway Collision
OTHER	Other End State
CFIT	Control Flight into Terrain
OFF AIRP LDG	Off-Airport landing
MID-AR COLL	Mid-Air Collision

the need to evaluate runway excursions and reduce flight risk is evident.

Typical causes for runway excursions are crosswind, contaminated runway, pilot behavior, and more. As shown in Figure 1.2, runway excursion is the dominant category for runway accidents. In total, 58 runway excursions occurred between 2017 and 2021 and resulted in 79 fatalities [4]. This compares to the 31 gear-up landings, the second most frequent accident, that occurred with no fatalities. The corresponding accident rates for excursion and gear-up landings are 0.32 and 0.17 per million flights.

Runway excursions can further divide between accidents that occur during takeoff and landing. The distribution of runway excursions from flights between 2016 and 2021 [4] shows that the ratio of landing and takeoff excursions is close to 5 to 1. This indicates that runway excursions during landing are of a bigger concern. Runway excursions can also divide between veer-offs and overruns. At the takeoff phase, the percent contribution to runway excursion by veer off and overrun is 37 and 63 percent. At the landing phase, the percent contribution to runway excursion by veer off and overrun is 53 and 47 percent [6]. These accident data suggest that runway veer-offs and overruns during landing are both dominant categories of runway accidents. However, studies identified flight behaviors in the final approach, such as the altitude at threshold crossing and touchdown speed, to have a strong correlation to runway overruns [7]. This makes overrun predictable and trainable through final approach simulations. These observations indicate a need to investigate

methods for reducing runway overruns during landing.

Runway overrun accidents were surveyed to identify the factors that need to be addressed to runway overrun. A runway overrun accident occurred on 2005 December 8th in Chicago Midway International Airport [8]. B737 aircraft with 103 occupants ran off the departure end of the runway. The aircraft rolled onto an adjacent runway and struck an automobile. A child in that automobile was killed and over 18 people in the aircraft were injured in the accident. Another runway overrun accident occurred on 1999 June 1st in Little Rock National Airport [9]. An MD82 aircraft overran runway 4R. During this accident, 10 passengers were killed and 105 passengers received injuries. The accident at LaGuardia airport on 2015 March 5th is another example of a runway overrun [10]. National Transport Safety Board (NTSB) analysis on these three accidents shows that multiple contributing factors lead to these accidents. The factors include, and are not limited to, inappropriate application of braking devices, poor weather, and runway contamination. Not a single factor was the sole dominating cause of the accident. This finding shows all of the contributing factors need to be addressed together to effectively reduce overrun accidents.

Multiple organizations are actively reducing multiple factors contributing to runway accidents. Organizations include aerodrome operators, air navigation service providers, aircraft operators, aircraft manufacturers, regulators, and R&D. Key principles addressed by these organizations for reducing runway accidents include safety assurance, safety risk management, safety policy, and safety promotion [11]. The safety assurance principle is focused on identifying the data monitoring and analysis methods to characterize flight accidents and identify risks. A program designed around this principle is the Flight Operational Quality Assurance (FOQA) program [12]. The objective of this program is to collect and analyze aggregate flight data so that regulators, aircraft operators, and aircraft manufacturers can identify and reduce risk. The objective of the safety risk management principle is to develop plans and technology to reduce flight risks. This is realized by efforts such as the application of Engineered Material Arresting System (EMAS). EMAS was developed

to stop aircraft that overrun the runway. The safety policy principle is concentrated on establishing and maintaining policies to ensure adequate measures are taken to reduce flight risk. Lastly, the safety promotion principle is focused on educating the aviation industry to promote best practices and reduce flight risk.

Efforts are made based on the four principles to reduce flight risk. The effectiveness of these efforts was difficult to measure without a metric that quantifies the risk level of a flight. So Runway Safety Metric (RSM) like tools are developed with an effort to monitor the safety performance near runway environment [13]. The RSM assumes fatal injury as the worst outcome and assigns weight to flights based on their ”proximity” to a fatal accident. Investigation on the risk of commercial and noncommercial flights, based on RSM, shows that the flight risk was reduced for commercial flights but increased for noncommercial flights between 2009 to 2014. The difference in risk trends indicates that commercial and noncommercial operations should be considered independently. As discussed, Metrics such as RSM are capable of characterizing global trends in flight safety. RSM like tools may be developed to further analyze the weak spots in the runway overrun.

More direct efforts are made by aerodrome operators to reduce flight risk through programs such as; EMAS, Airport Surface Detection Equipment model X (ASDE-X), Airport Taxiway Arrival Prediction (ATAP) enhancements, and installment of RunWay Status Lights (RWSL) [11]. These efforts reduce the severity of accidents by increasing pilots' awareness, in flight, of external factors, and current flight conditions. Currently, at the national level, these technology solutions are targeted toward 475 towered airports without surface surveillance systems. The case for installing EMAS at 115 runway ends and installment of RWSL at 20 airports nationwide are some outcomes. Combined efforts from regulators and aerodrome operators to reduce runway risk include the Runway Incursion Mitigation (RIM) program. This program modifies the taxiway design to reduce runway accidents. In one case the runway incursion reduced from 15 (between the years 2008 and 2014) to 0 before and after implementing the RIM program.

Aircraft manufacturers are also making efforts that directly address the issues with runway accidents. Runway Overrun Awareness and Alerting System (ROAAS) [14] is a tool developed by Honeywell and Gulfstream to reduce overrun risk. Factors leading to the development of ROAAS include the lack of awareness of energy state, current autobrake setting, braking capabilities in actual runways, and approached runway length. ROAAS improve flight safety through a two-step process. While the aircraft is in the air, ROAAS computes a model landing distance based on runway data, aircraft height relative to the glide slope, and aircraft ground speed. While the aircraft is on the ground, various braking device usage is considered to estimate the landing distance. When the required landing distance is beyond the available runway, the pilot is alerted of this condition. A ROAAS like system developed by Airbus is called Runway Overrun Prevention System (ROPS) [15]. A ROAAS like system currently in development by Boeing is called Runway Awareness and Advisory System (RAAS) [16]. The ability of these systems to predict landing distance during multiple phases of flight allows the pilots to get a real-time evaluation of landing distance and predictive measure of the landing performance.

Efforts introduced so far are proven to be effective based on accident data shown in Figure 1.3. The accident rate distribution from the year 2011 to 2020 by IATA also shows a decrease in accident rate and hull loss [4]. These two accident data also show that the accident rate is leveling off and a change in how accidents are handled is required to further improve flight safety. The Global Action Plan for the Prevention of Runway Excursions (GAPPRE) [17] is developed by 40 different international organizations to set a list of recommendations to improve runway excursion. These recommendations are made based on representing industry best practices, for aerodrome operators, air navigation service providers, aircraft operators, aircraft manufacturers, regulators, and R&D organizations. As shown in Table 1.2, there are frequent mentions in the GAPPRE to increase the awareness of the flight condition and energy states to reduce overruns. ROAAS like tools are frequently mentioned as a potential tool to realize this goal. In the review of this work

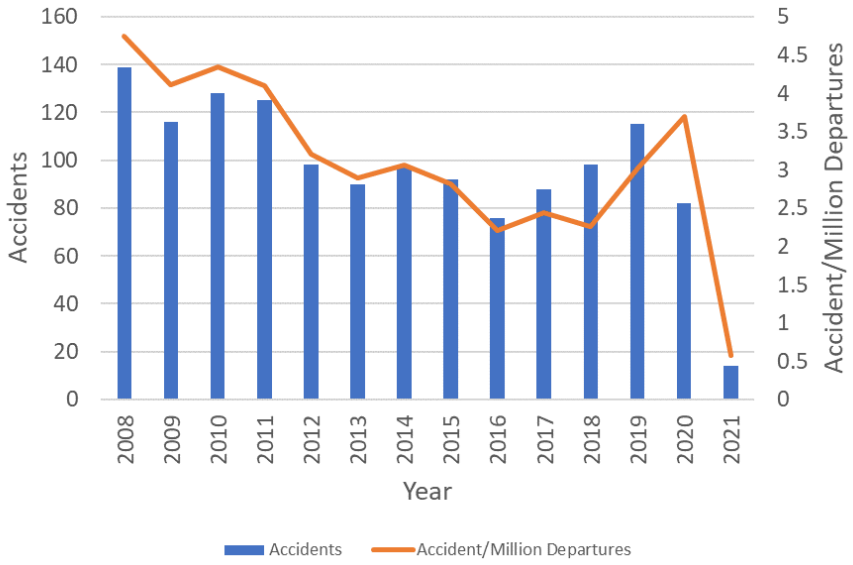

Figure 1.3: Yearly fatal accident rate per million flights [18].

Table 1.2: GAPPRE recommendations on the need for an awareness tool for monitoring flight condition

REF	Recommendation	Implementation Date
R&D1	Investigate awareness and alerting system for significant lateral deviation during final stages of landing	End of 2030
R&D7	Develop functions that provide flight path and energy information	End of 2030
MAN4	Develop a system to easily and reliably calculate landing distances for normal and non normal conditions	End of 2025
MAN12	Make available systems that provide flight path and energy state awareness	End of 2027
OPS4	Incorporate appropriate technical solutions to reduce runway excursion risks	End of 2027
OPS29	Foster a culture that stimulates safe behavior	Ongoing

and other relevant research, the methodology to improve flight conditions and energy state awareness is important for reducing runway overrun. These tools currently rely on human inputs and limited flight parameters to determine the aircraft landing performance metrics. They are ill-suited for rating flight conditions in a holistic manner to account for overrun precursors that are triggered early on in the final approach. Also, these tools are designed to reduce the severity of potential accidents during flight. They do not provide pilots feedback on the flight behavior post-flight. A methodology that may potentially reduce runway overruns by closing these gaps is proposed in this study. The goal that summarizes the research objective for the proposed is as below.

Research Objective

Develop a methodology that increases pilots' awareness of the risk of runway overrun by predicting and simulating flight states that define a stable approach

The following chapters of this document discuss the following:

- Chapter 2 presents a review of relevant background to characterize the runway overrun accidents and improve flight safety
- Chapter 3 presents the problem formulation in terms of gaps found in chapter 2 and proposes methods to close the gaps
- Chapter 4 presents the overall framework of the proposed methodology that increase awareness to risk through prediction and simulation
- Chapter 5 proposes a hyperparameter optimization framework to optimize the models used to close the gaps found in chapter 2
- Chapter 6 discusses the process of developing the prediction model
- Chapter 7 discusses the process of developing the simulation model
- Chapter 8 demonstrates the potential of the proposed methodology for increasing awareness of runway overrun risks
- Chapter 9 presents concluding remarks on the proposed methodology

CHAPTER 2
BACKGROUND AND LITERATURE

Flight safety and accident mitigation require efforts from all aspects of aviation. An important aspect of aviation for accident mitigation is to increase the pilots' awareness of accident precursors so that they can avoid risks. A concept at the foundation of pilot training is the Attitude Skill and Knowledge (ASK) triangle. The attitude skill knowledge related to evidence-based training is a concept used in a wide range of industries. [19, 20, 21]. The idea behind ASK for pilot training is that you can "survive" if you are well equipped with two of the three aspects. For example, a pilot may have poor landing skills during crosswind. However, the pilot can "survive" by knowing that it is against guidance to land at the high crosswind and having the right attitude to avoid threats through go around. So it is important to train the pilots in all three aspects of ASK and maintain a balance between the three.

The goal of this study is to support the pilots in these three aspects of training to mitigate runway overruns. To do so pilots need to be given enough but not too much information to have a good attitude when facing runway overrun. Pilots also need to be given the opportunity to learn and practice from historical flights to have good knowledge and skill against conditions that have led to runway overrun. If so, what leads to runway overrun? Flight accidents, in general, are frequently characterized by flight precursors based on threshold exceedance of one or few flight metrics [12]. The occurrence of a precursor does not guarantee an accident, but precursors are strong indicators of an accident occurring. NTSB analysis of flight accidents shows that a typical accident results from multiple compounding precursors. For example, NTSB identified the following precursors as probable causes for the accident at Chicago Midway on December 8, 2005 [8]: tailwind over 5 knots, runway operational uncertainties, and proper application of braking devices. Other factors that

contributed to the accident were: a lack of clear guidance on landing distance calculation procedures and a lack of knowledge of the assumptions used in an onboard computer that is critical for the pilot's decision-making. These indicate the lack of training in all three aspects of ASK and an enhancement in the training program is needed. Another accident analyzed by NTSB is the accident at Little Rock, Arkansas on June 1, 1999. [9]. Precursors that contributed to the accident were: high crosswind and excess reverse thrust application (> 1.3 engine pressure ratio). Other factors that contributed to the accident were: pilot's fatigue and reduced situational awareness. These indicate the lack of training in all three aspects of ASK as well. A more recent accident analyzed by NTSB was the accident at La-Guardia airport on March 5, 2015 [10]. Precursors that contributed to the accident were: a lack of lateral stability and inadequate thrust reverser control. Other factors that contributed to the accident were: stress due to concerns about stopping performance and fatigue during landing. Again this accident points to the lack of all three aspects of ASK. These accidents show that there is a correlation between overrun precursors and overrun. Due to this reason, regulatory guidance is in place at specific approach altitude gates so that the pilots stay within the safe flight parameter bounds for landing [22]. Flights that remain within these parameter bounds during the approach are defined as a stable approach. The use of flight parameter threshold to define a stable approach again shows that precursors are a good way to describe overruns. So the focus of this study is on investigating methods to identify and increase awareness of overrun precursors.

Some of the overrun precursors are defined by flight parameters during the final approach. The correlation between runway overruns and final approach parameters lead to studies that measure the risk of runway overrun based on these parameters. The next sections survey the runway overrun-related flight parameters identified by other studies. These parameters are analyzed and collected to form a satisfactory set that characterizes the risk of overrun during the final approach. The methods of improving flight safety were investigated based on these parameters.

2.1 Landing Process Analysis

Stabilized approach and stabilized ground roll are important factors to consider when reducing runway overruns. An approach is considered stable when nine conditions are met either by 1000 ft above touchdown in Instrumental Meteorological Conditions (IMC) or by 500 ft above touchdown in Visual Meteorological Conditions (VMC) [6]. Some of these conditions are as follows:

- Aircraft is on the correct flight path.
- Small corrections in heading and pitch to maintain the flight path
- Aircraft speed $< V_{Ref} + 20$ and indicated airspeed $> V_{Ref}$
- Aircraft is on correct landing configuration.
- Sink rate below $1000 ft/min$
- Engine is on appropriate power settings

Meeting these conditions is important for reducing the occurrences of overrun precursors.

So what are some of the overrun precursors and how much do they contribute to overruns? This can be identified from studies that analyzed runway excursions. The distribution of top precursors that contributes to runway excursion is shown in Figure 2.1 [6]. Go-around not conducted is the most important factor for landing excursions that contributes to over 35 percent of excursions. There are multiple factors, at or near touchdown, considered to conduct a go-around. These factors include and are not limited to alerts, visibility, runway contamination, wind, side slip angle, vertical velocity, and lateral position to the center of the runway. If these factors can be predicted ahead of touchdown, then the need for go around can be predicted, hence the risk for overrun can also be predicted. The same rule applies to other precursors such as touchdown: long, fast, and hard. These three precursors result from a lack of energy management, which is defined as maintaining the right

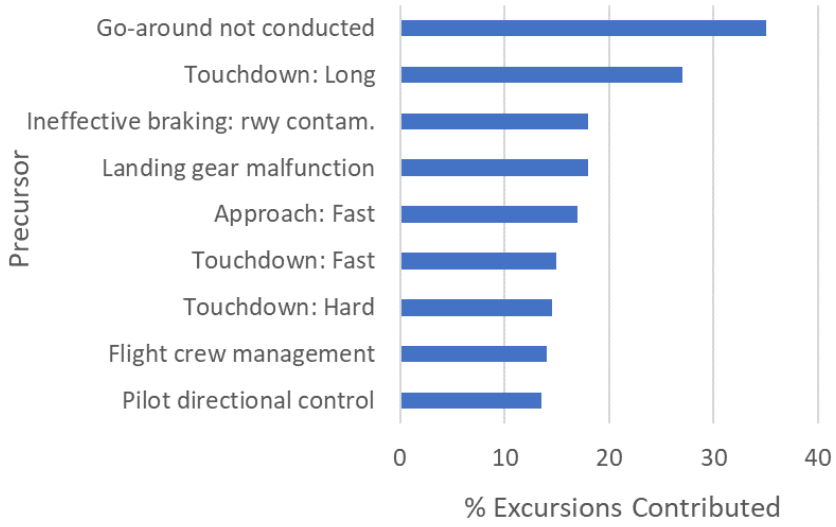

Figure 2.1: Distribution of percent excursions that each precursors contributed to [6]

speed and altitude throughout the approach. If the aircraft speed and altitude trajectories can be predicted, the likelihood of long, fast, and hard touchdown precursors can also be predicted, hence the risk for overrun can also be predicted. Other precursors such as pilot directional control precursors have more direct relation veer-offs and they are not considered for this study. However, the same logic applies when analyzing runway veer-offs. For example, lack of directional control means asymmetry of braking devices deployed from the time of landing. The likelihood of pilot directional control precursor to occur may be identified in flight based on the side slip angle near touchdown and the level of cross-wind fluctuation. If the aircraft side slip and crosswind near touchdown can be predicted then the likelihood of pilot directional control precursor can also be predicted, hence the risk for veer off can also be predicted.

The precursors mentioned in Figure 2.1 show the importance of ASK triangle and this can be explained with a few scenarios. During the approach phase, one may be skilled at landing, but a poor attitude towards go around and a lack of knowledge of the aircraft condition may lead to excursions. One may also have the knowledge that the aircraft is exceeding the energy thresholds but the lack of attitude and skill to drain enough energy

before touchdown may lead to excursions. During the rollout phase, one may have the attitude to use lateral control to resist crosswind but does not have the knowledge or skill to do so. An overview of the precursors that trigger runway excursion shows that the following may increase awareness of runway overruns:

1. Prediction of flight parameters near touchdown

2. Simulation of final approach at normal and adverse flight conditions

The runway excursion precursors associated with the full landing process are observed in the following sections. To do so, the landing process is divided into three different flight phases since aircraft dynamics vary significantly at each flight phase. The flight phases are defined based on Boeing's breakdown on the runway overrun precursors by flight phase: approach, touchdown/landing, deceleration/ground roll [16]. Different runway excursion precursors are identified at each of these flight phases. The precursor analysis by phase showed that different precursors contribute to runway overrun for each phase [23, 24]. Hence the precursors are further evaluated for each of the flight phases.

2.1.1 Approach

The aircraft landing process starts with the approach phase. The approach phase includes the base leg and final approach [25]. In the base leg, the velocity and glide slope is decided to maintain stabilized descent while reaching the desired touchdown point. A general rule is to descend with reduced power and maintain an airspeed of approximately 1.4 V_{stall}. This phase is continued until a shallow turn can align the airplane's path with the runway center line and the accurate touchdown point can be estimated. At the start of the final approach, the aircraft's longitudinal axis is aligned with the landing surface. The alignment is followed by a correction of the pitch to match the desired descent rate. Here, for every engine power setting, there is one speed, flap, and wind condition to land at the landing point. Flap setting is a significant factor at this stage of the flight since it is capable of

increasing lift and drag to reduce the aircraft's speed while remaining in flight.

Studies show that safety at the approach phase is characterized by maintaining stability at various altitude gates. The main parameters to stabilize are glide slope deviation, V_{ref} deviation, and Rate Of Descent (ROD). Other parameters and their stabilization thresholds are shown in Table 2.1. These parameters are expected to be stabilized at altitudes 1,000 ft above touchdown for IMC and 500 ft for VMC. Failure to stabilize the aircraft at these altitudes may trigger multiple precursors, which may result in an accident. A study with flights between 2012 and 2016 shows that aircraft with undesired states are contributing factors to 50 percent of hard landings, 27 percent of excursions, 9 percent of tail strike, and more [22]. Analysis showed that for approaches with undesired aircraft states, 74 percent

Table 2.1: Approach stabilization criteria for safe landing [26]

Parameter	Criteria
Stabilization height	1,000 ft IMC, 500 ft VMC
Glideslope	3 degrees
V_{ref}	$V_{ref} - 5$ to $V_{ref} + 20$ kts
ROD	$<$ 1,000 ft/min
Engine Power	stable above idle

were due to incorrect manual handling, 47 percent due to noncompliance with Standard Operating Procedure (SOP), and 53 percent due to failure to go around. These precursors are compound with the fact that 97 percent of unstable approach flights did not result in go around to increase flight risk [24]. This emphasizes the significance of ASK to prevent accidents.

Multiple regulatory guidelines are developed to encourage operators to approach the runway in a stabilized manner. Continuous Descent Final Approach (CDFA) approach is encouraged for Instrumental Approach Procedures, Approach Procedures with Vertical Guidance, and Non-Precision Approaches [27]. This technique starts from Final Approach Fix (FAF) to flare and increases safety by abiding by the stabilized approach criteria. This method utilizes a constant glide slope to land at the desired location on the runway. The constant glide slope is calculated based on the altitude difference between FAF and run-

way Threshold Crossing Height (TCH) divided by the descent gradient. This approach is shallower, more efficient, and has a lower risk of triggering accident precursors (such as hard landing) compared to the conventional method of nonprecision approach (the dive drive approach) as shown in Figure 2.2. Some parameters relevant to CDFA include ground

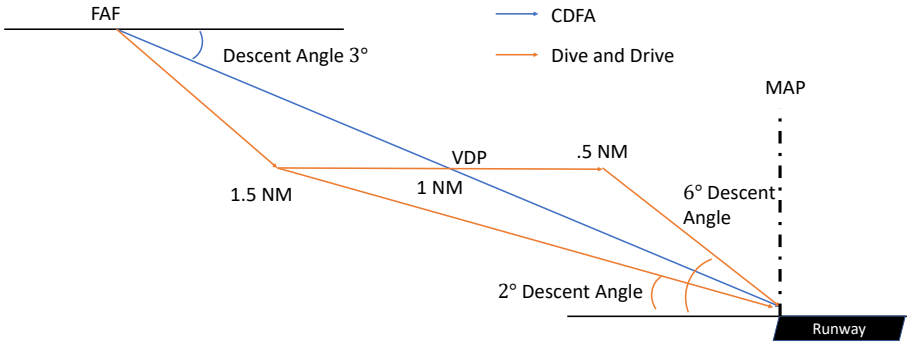

Figure 2.2: Examples comparing dive and drive approach vs CDFA [27]

speed, glide slope, ROD, FAF altitude, TCH, and minimum descent altitude, which is the decision altitude to proceed with a final approach based on visual ques. These parameters are also relevant to the parameters identified by Federal Aviation Administration (FAA) for mitigating runway overrun upon landing [28]. Other factors to consider for mitigating runway overrun that is not considered for CDFA include aircraft landing weight, landing airport altitude, runway slope, and delayed use of braking devices.

Based on CDFA guidance, there should be small deviations in the approach parameters below the stabilization window. In reality, the approach parameter may vary significantly at each stabilization gate and near touchdown. For example, the distribution of ground speed is shown for 3688 regional turbojet flight data at 50 ft above touchdown in Figure 2.3. One study shows that a 10 percent increase in final approach speed results in a 20 percent increase in landing distance at normal touchdown conditions [6]. The study also shows that for every 10 ft above standard TCH, the landing distance increases by 200 ft.

Multiple external factors can deviate the approach parameters such as wind and control sequence. Campbell et al. [26] studied the significance of deviation in some of these

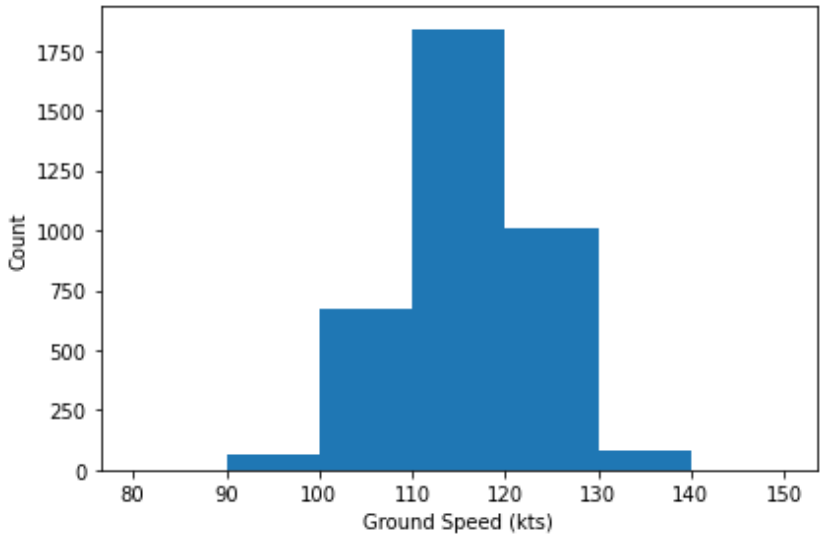

Figure 2.3: Ground speed distribution at 50 ft above touchdown for regional turbojet flights.

approach parameters (localizer, glide slope, ROD, and ground speed) to the touchdown condition at three approach gate altitudes (500 ft, 300 ft, 100 ft). Touchdown performance was analyzed based on longitudinal, lateral touchdown location, and sink rate at touchdown. The analysis showed that the deviation in relative velocity at the 100 ft gate had the most significant impact on touchdown performance. Deviations in approach parameters at 300 ft and 500 ft gates were less significant because the pilots were able to correct the deviations. This brings into question if the flight parameters near touchdown can be predicted based on aircraft states and external factors known well in advance of touchdown. If this can be realized, the pilots may be notified of the risk of landing well in advance, above the 300 ft altitude gate.

2.1.2 Landing

The aircraft approach phase is followed by the landing phase. The landing phase includes flare and touchdown [25]. Flare is a continuous process that smoothly transitions the aircraft from approach to touchdown. During a flare, pitch and Angle Of Attack (AOA) are increased to increase lift while reducing ROD and ground speed. Here, a proper level of elevator application is needed to prevent the aircraft from stalling. Touchdown is a gentle

settling of the aircraft on the runway. This is accomplished by the right amount of AOA so that at touchdown the ROD is near zero. An important factor to remember for a touchdown is that the longitudinal axis needs to be parallel to the runway at touchdown to avoid veer-off.

The performance of the flare and how well the excess airspeed is bled has a significant impact on the landing distance. As mentioned earlier, landing distance increases by 10 percent for every 5 percent increase in airspeed during the final approach. For delayed touchdown due to extended flare, the landing distance may increase by 30 percent [6]. Touchdown may be delayed due to bounced/rejected landing. Bounced landing is one of the precursor events to runway excursions, that occurs during touchdown. For these situations remaining landing distance, aircraft pitch attitude, and asymmetry in thrust needs to be considered [29]. A study tried to estimate the aircraft touchdown parameters such as aircraft attitudes, angular rates, aileron deflection angle, elevator deflection angle, rudder deflection angle, and engine throttle setting [30]. This study shows that undesired aircraft states through long / floated / bounced / firm / off-center / crabbed landing contribute to 40 percent of runway excursion, which includes overrun. Measuring the touchdown performance based on ”proximity” to these events may be done based on parameters measured from the study by Wu et al. [30].

Crosswind is one of the threats associated with flare and touchdown phases that contribute to runway overrun. IATA safety report shows that for runway excursion accidents between years 2017 to 2021, the wind is a threat that contributes to 33 percent of excursions [4]. Unnecessary weather penetration is an undesired aircraft state that contributes to 22 percent of excursions. Based on the frequent contribution of wind on runway excursions, the regulators computed the maximum recommended crosswind for a safe landing. There are also two techniques developed to account for crosswind: crabbed approach and side slip approach. The crabbed approach is when the aircraft approaches the runway with wings leveled while the longitudinal axis is corrected so the flight path aligns with the runway

center line. The side slip is when the aircraft approach with a longitudinal axis aligned with the runway centerline while correcting the aileron and rudder to account for the crosswind. Typically Airbus recommends using the crabbed approach for fighting crosswind [31]. For example, a strong 30 kts crosswind is typically accounted with 13 degrees of crab angle. To reduce the potential for landing gear damage, a combination of crabbed angle and side slip may be considered. Parameters to consider during crosswind include pitch attitude, bank angle, crosswind, ground speed, runway condition, aileron deflection, and rudder deflection. How these parameters interact at the time of landing defines the aircraft's condition near touchdown. One such interaction is between crosswind and ground speed. The contribution of crosswinds to the aircraft's aerodynamics may be reduced by increasing the ground speed. Another interaction is between the aileron and rudder deflection. The difference in airspeed, at the swept wings, generated by crosswind results in a rolling moment. Ailerons are deflected to make up for this moment, but this also generates a difference in drag across the wing span which is countered by the rudder deflection. Interaction between flight parameters and external factors also exist such as limited pitch and bank angle to account for a tail strike, engine nacelle contact, and wingtip contact. Crosswind is an external factor with a significant impact on landing performance. Therefore an accurate representation of the impact of external factors on flight dynamics is needed to train pilots in these adverse situations.

It is important to mention that the recommended flight condition near touchdown varies with the runway. The approach and touchdown performance should consider the landing guidance provided per airport. For example, at Newark Liberty International Airport (EWR) the glide slope indicator is at 3.1 degrees for runway 22R but 3 degrees for 22L. The TCH TCH is 47 ft for EWR runway 22R but 40 ft for EWR runway 22L.

2.1.3 Ground Roll

At the ground roll phase, the aircraft speed is reduced to full stop or taxi in speed on the runway[25]. Main gear brakes, spoilers, and thrust reversers are some of the controls used in civil aircraft to bring the aircraft to a full stop. During this process, the effectiveness of directional controls and the timeliness of the braking devices applied contributes significantly to runway excursion accidents. Weight on the main gear is significant for establishing the effectiveness of directional control and braking devices. Since higher weight on the main gear means the landing gears have full contact with the runway. The same applies to the brake pressure due to its contribution to skidding. Even with perfect weight on the main gear and brake pressure, the braking performance may degrade from external factors.

The Flight Safety Foundation (FSF) mapped each factor, that contributes to landing, to a landing distance multiplier [6]. The contribution of the increase in ground speed and altitude at threshold crossing on the required landing distance was already introduced above. The analysis shows that runway parameters such as runway gradient, condition, and elevation have a considerable impact on the landing performance as well. Runway altitude has a significant contribution to the air density, which directly impacts lift. Any lift lost due to an increase in altitude is compensated by an increase in speed. So for an airport located at 1000 ft above sea level, the landing distance is expected to increase between 5 to 10 percent. An increase in runway gradient by 1 percent increases the runway distance by 10 percent as well. It is important to note that airport runway gradient may not be constant for the full distance. So the aircraft may experience a higher runway slope locally as the aircraft runs down the runway. This study shows that the contribution from various parameters can be condensed into the required landing distance metric.

Understanding runway condition plays a significant role in the success of reducing overrun. Studies show a significant reduction in tire friction for contaminated vs uncontaminated runways [32]. The Joint Aviation Authorities (JAA) defined contaminated runways with the presence of the following contaminants:

- More than 3 mm deep surface water, slush, loose snow, or the equivalent.
- Compressed solid snow.
- Ice or wet ice.

During braking, these contaminants reduce friction between the tires and the runway surface by reducing the contact surface [6]. Fluid contaminants may also contribute to braking by reducing the forward movement of wheels through displacement drag and by spraying fluid at the airframe to form impingement drag. The level of impact the contaminants have on the braking performance is categorized into seven levels. The dry runway is denoted with a six while a wet ice runway with minimal braking deceleration is denoted with a zero [28]. A wheel braking coefficient is assigned to each of the contamination categories to modify the required landing distance [33]. Runway pavement is another factor to consider when considering overrun especially for a contaminated runway condition. Studies show that micro and macro textures on the runway determine the drainage of water and are inversely proportional to the hydroplaning [34]. Macro structures on the runway such as grooving affect the slope of the tire friction and speed gradient curve. Microstructures on the runway affect the magnitude of friction at a given speed. Analysis of hydroplaning showed that the pressure or percent contact between the tire and flooded pavement varies with vehicle speed squared. When this pressure exceeds the pressure between the tire and the pavement surface, the tire is detached from the surface.

An overview of the ground roll phase shows that for this flight phase it is important to accurately represent the impact of external factors on aircraft dynamics. Especially, an accurate representation of an aircraft's performance during poor weather may improve the attitude, knowledge, and skill of the pilot by simulating landing under adverse conditions.

2.1.4 Overrun

The factors that contribute to runway overrun have been summarized. Boeing investigated the precursor events to runway overruns and generated an overview of overrun accidents [16]. Based on the investigation in the planning and approach flight phases, stabilization at various altitude gates is the key factor to consider. The approach phase is followed by a touchdown, which includes flare and landing processes. Long landing, high touchdown speed, and proper deployment of speed brakes are precursors that contribute to overrun during touchdown. A key consideration in the touchdown phase is to decelerate the rate of descent so that the aircraft "kiss" the runway. The last phase is the landing phase. During this phase, proper application of braking, runway contamination, and timing of braking devices are identified as precursors that contribute to overrun. Collective analysis of accident data from 2003 to 2010 is shown in Table 2.2. The analysis shows that excursion is a complex problem with multiple contributors. Reducing runway excursions requires an investigation of the full landing process.

Table 2.2: Percent flights with overruns that exceeded the critical flight metrics [16]

Flight Metrics	Overrun Flights (%)
Unstable Approach	32
Exceeded Touchdown Zone	45
Contaminated Runway	90
Tailwind ($>$ 5 kts)	42

2.2 Literature Review

The precursor analysis of runway overrun points to two themes for improving flight safety: prediction and simulation. These themes revolve around enhancing the three aspects of pilots training: ASK. The prediction of flight conditions near touchdown may increase pilots' awareness of potential risks for landing and enhance their attitude to reduce risk. The flight simulation in regular and adverse weather may be used to enhance the pilots' knowledge of

the threats in these conditions. Studies regarding these two themes are investigated in the following sections.

2.2.1 Prediction

Flight condition prediction may be in the form of metrics prediction and precursor events prediction. A survey was conducted by NASA to evaluate what, when, and where the prediction should be made [35]. The survey was conducted by 22 airline transport pilots with flight hours ranging from 2,000 hrs to 17,500 hrs. Results from the survey showed that the pilots wanted information on the abnormal changes in flight parameters and the presence of threshold exceedance. The survey also showed that the pilots preferred that the predictive information is shared before touchdown so that enough time is given to respond. Another study conducted by NASA discusses the stress experienced by the pilots from the predicted information [36]. The experiment considered the stress contributed by data type (baseline, direction, timed) and the prediction time (1min, 5min, 15min, 45min) until the alerted issue occurred. The study showed that although introducing predictive system information did increase stress, this was mainly due to the increased workload to correct flight paths. What's interesting is that pilots did not respond differently to alerts made 1 min before a problem occurred because not enough time was given to the pilots to diagnose the problem. The pilots did respond when enough time was given to respond and the stress level decreased with increasing prediction time. The experiment was not conducted for the pilots' response to the prediction of runway overrun precursors. However, a similar result can be expected for runway overrun precursor prediction. The pilots will need enough time to utilize the prediction data or else the prediction will only add to the stress. In fact, the study by Campbell et al. also showed that pilots were able to correct deviations in flight parameters when alerts were made at least 300 ft HAT. Based on these results, some criteria can be made for the expectation on the prediction tool to mitigate runway excursion accidents.

1. Show potential flight parameter with threshold exceedance.

2. Show expected trend for the significant flight parameters.

3. Sufficient prediction time to react to the alerts.

Based on these criteria, FOQA data may be suitable for this analysis since the purpose of FOQA is to identify threshold exceedance trends. The events defined by FAA identified some of the threshold exceedance events to be monitored by the basic FOQA program [12]. The temporal manner in which the FOQA data is recorded also makes it capable of analyzing the direction that the flight parameters are changing and how they vary from the norm. The following parameters are most frequently mentioned in the earlier survey of flight parameters significant to runway overrun: ground speed, vertical speed, and pitch angle. So these parameters were considered to be the most meaningful parameters to predict. In terms of how the prediction is made, multipoint and sequential prediction models of the aforementioned three parameters are most beneficial based on the survey results. The key is that the prediction is made as early as possible. Studies that attempt to perform this prediction task are surveyed.

Metrics Prediction

Multiple studies are focused on predicting aircraft speed. This is because speed is a significant factor in determining the kinetic energy of the aircraft. Proper energy management plays a significant role in flight safety due to its contribution to various accidents including loss of control (LOC) [37], controlled flight into terrain (CFIT) [38], and runway excursions (RE) [4]. A study by Puranik et al. [39] identified total energy and total energy rate metrics to track energy management. As shown in this study the contribution that speed has on energy management makes predicting this metric meaningful.

The predictive power of various studies that predict ground speed are shown in Table 2.3. Here the predictive power is defined based on the accuracy in terms of root mean

square error (RMSE), prediction time, and prediction type. Existing studies on ground speed prediction show that many of the prediction models are generated with neural networks. Tong et al. [40] used LSTM neural network to generate a state space model of ground speed. This model predicts ground speed at time-step (t) based on ground speed and other flight metrics at the previous time-step (t-1). The flight parameters considered for the analysis are selected based on their significance to the ground speed. The significance here is defined based on the prediction sensitivity to the perturbation in flight parameters. Diallo [41] predicted landing speed under various gust conditions using a feed-forward back propagation (FFBP) neural network. Kang et al. [42] combined gradient boosting decision tree with LSTM model to do sequence-to-sequence prediction of ground speed. A common theme between the methods introduced here is the use of LSTM models. LSTM models are a type of recurrent neural network model that can handle diminishing or exploding feedback propagation errors by using memory gates. This allows LSTM models to capture the spatial-temporal behavior of an aircraft and generate a more accurate model of the data it represents as shown in Table 2.3. However, neural network models are not the only

Table 2.3: Machine learning prediction models for aircraft ground speed [43]

Measure	Puranik et al. [44]	Tong et al. [40, 45]	Diallo [41]	Kang et al. [42]
RMSE (kts)	2.98	0.0065	3.5	1.71
Prediction Time (s)	18 (300 ft)	1	Retrospective	3 (50 ft)
Prediction Type	Single-Point	Single-Point	Sequential	Sequential

method used for predicting ground speed. Puranik et al. [44] generated a prediction model of ground speed with random forest regression. A random forest model is a combination of tree predictors where at each node of the tree the data set is split to maximize homogeneity within each group. Although the ground speed accuracy of this study is lower than that of the rest of the studies, the prediction range is much higher, making this the more feasible tool for the pilots to use.

Ground speed or airspeed is not the only metric predicted from current studies. Zhang and Zhu [46] used LSTM, support vector machine (SVM), back-propagating neural net-

work (BP), and logistic regression (LR) models to predict hard landing based on vertical acceleration at touchdown. The generated model is capable of predicting the vertical acceleration 2 seconds before touchdown. The results show that LSTM prediction model had the highest accuracy followed by the SVM model. Jarry et al. [47] used LSTM and simple dense models to estimate fuel flow rate, landing gear setting, and flaps configuration. The state space prediction model was used to predict each metric for up to 70 nm. The prediction accuracy is below 10 percent of the normalized RMSE and Pearson's value of the predicted metric trajectories is above 0.90. These studies show the potential for machine learning models to predict multiple flight metrics.

Precursors Prediction

Flight metric prediction provides quantitative analysis of the level of abnormality and the direction of the change in the flight parameters. A quick way to alert the pilots of a threshold exceedance is by directly predicting the threshold exceedance labels or precursors. FAA has identified a list of base exceedances to be used for identifying anomalies in the FOQA data [12]. These exceedances are typically used to find trends in the aggregate flight data. Some studies used these exceedances to generate labels that count the number of exceedances that occurred for each predefined metric threshold. These labels are used to train a supervised machine learning method to predict future exceedances. This is because supervised machine learning models are capable of identifying the relationship between the input data and the labels.

Exceedances may also be identified by using unsupervised machine learning methods by detecting anomalies directly from the input data. Some of the anomaly detection methods used in the aviation industry include the exceedance method, distance method, cluster method, distance method, and neural network method. The exceedance method identifies thresholds based on reference data to evaluate anomalies in the current flight [48, 12, 49, 50]. Multiple Kernel Anomaly Detection (MKAD) and Orca algorithms are examples of

exceedance methods of anomaly detection. Orca utilizes the k nearest neighbors approach and defines the reference data based on a nominal data set [51]. Euclidean distance to the nominal vector space is used to identify anomalous data. A downside to orca is that it does not learn the sequential dependencies. MKAD, on the other hand, is capable of identifying anomalies for both discrete and continuous data but it is more sensitive to discrete parameters [52]. MKAD finds the probability of a data point to lie outside of the dataset [49]. A study by Lv et al. [50] used the physics of the overrun risk to identify the risk margin for speed and remaining runway distance. The euclidean distance between flight data and risk margin was used to identify anomalous flights. This method of anomaly detection limits analysis to a few parameters that are considered the most significant for an accident type.

The cluster method finds clusters at a reduced dimensional space to group similar flights. SequenceMiner and Cluster Anomaly Detection (ClusterAD) algorithms are examples of cluster methods of anomaly detection. SequenceMiner selects the longest common sequence as the reference and identifies data that is not similar to anomalies [53]. This algorithm does consider the sequence of how the data parameters change but is only capable of handling discrete parameters. ClusterAD utilizes density-based spatial clustering of application with noise (DBSCAN) on input data at a latent space. DBSCAN identifies the nominal data points and groups them based on data point density near the nominal point [54]. Li et al. [52] compared ClusterAD and MKAD to show that ClusterAD is more capable of detecting anomalies in continuous parameters than MKAD.

Neural Network methods of anomaly detection used supervised machine learning techniques to map flight data to threshold exceedance labels. Nanduri and Sherry [55] used LSTM and GRU to classify flight data to various threshold precursors. The benefit of using a neural network is that the algorithm does not map the data to a latent space. The multivariate sequential input data is used directly for classifying the input data. Other types of neural networks such as Convolutional Neural Network (CNN), RNN, and hybrid models are also used for capturing the hidden features in complex data [56].

While it is possible to make an algorithm identify precursors on its own, some studies are focused on detecting specific precursors. Multiple studies used SVM, radial basis function neural network (RBF), and LSTM models to classify hard landing flights [57, 58, 46]. These methods either directly predict hard landing or predict a flight metric. These methods also classify hard landing flights based on threshold exceedance of the predicted metric. A similar study is performed by Martinez et al. [59] to predict an unstable approach with supervised machine learning. Ensemble and LSTM models are used to generate prediction models. The results show that the LSTM model has higher accuracy for predicting unstable approaches. Overall, the literature survey shows the potential of neural network models to accurately predict anomalous flights. The neural network models can directly use the input data to simultaneously predict flight metrics and exceedance events. Multiple gaps can be identified regarding the development of flight condition prediction models.

- Gap 1 – current models that predict flight conditions optimized model-related parameters and not data-related parameters

- Gap 2 – current studies on predicting flight conditions lack analysis on the sensitivity of the model to external conditions

2.2.2 Simulation

Aircraft fly in various flight conditions including strong wind, rain, snow, and more. The aircraft performance and pilot behaviors may or may not change for different flight conditions. Capturing the potential difference in flight performance for various flight conditions requires sufficient data for flights under each condition. The same goes for capturing aircraft dynamics experiencing accident precursors. Unfortunately, studies show a significant imbalance between flights with and without specific flight events. For example, flight data analyzed by the author in an earlier work [60] shows that the ratio of tire speed exceedance event to nonevent flights may be close to 1 to 4. Data balancing techniques were created

to balance the number of events and nonevent flights to improve analysis results. So depending on the flight data available, it may be difficult to generate highly accurate aircraft models tailored for each flight condition. This imbalance in the data indicates a need for an aircraft dynamics model to simulate adverse flight conditions. For example, a runway with snow contamination may be difficult to come by for pilots flying from Hartsfield–Jackson Atlanta International Airport (ATL). The aircraft deceleration model during ground roll on the snow-contaminated runway may be used to improve pilots' awareness of this type of runway condition. Currently, regulatory guidance points toward the use of multipliers on the point mass aircraft dynamics model to consider various ground roll conditions [33, 6]. Studies by van Es [32, 61] also show that these physics-driven dynamic models can be used as performance margins.

Simulation models can be created by identifying aircraft system dynamics with linear and nonlinear models [62]. Locally linear aircraft models are generated with methods such as Kalman Filter variants to estimate short-period dynamics, upset detection, and identify key flight parameters [63, 64, 65]. These methods are used iteratively to update the model with every small change in system states when the assumptions for the linear models are no longer applicable. Nonlinear aircraft models rely on methods such as neural networks, fuzzy logic, GA, swarm intelligence, and more. Neural network system identification methods have the benefit of combining different neural network models to accurately model the system. This method of system identification is used in various applications including the biomedical and aviation industries [66, 67]. The benefit of neural network models is that they can generate Multi Input Multi Output (MIMO) system. This is important because even the simplified longitudinal and lateral dynamics model requires the tracking of four states. Multiple variants of fuzzy logic are used for system identification as well [68, 69, 70, 71]. Benefits of the fuzzy logic identification method include the ability to model large time delay system and Multi Input Single Output (MISO) system. The ability to model time delay is an important consideration for generating the aircraft model because aircraft con-

trols have a different impact on each state for the long and short term. For example, when the aircraft is in trim condition increasing thrust will initially increase speed but eventually increase the altitude at a steady state. Based on existing aircraft system identification methods, some criteria can be made for the expectation of the simulation model to mitigate runway overruns.

1. Capture the temporal aspect of the aircraft behavior
2. Capture the coupling of multiple inputs and multiple outputs
3. Accurately model the change in the dynamics at various flight conditions

Based on these criteria, the current methods of aircraft system identification are surveyed to identify methods that are capable of modeling regular and anomalous flights. These methods may have the potential to develop a single neural network model that captures non-linearity and temporal dependencies on aircraft behavior.

Physics Driven

A conventional way of generating an aircraft state space model starts with the force moment diagram of assuming a point mass system. To discuss the force moment diagram of an aircraft, an illustration of the notations used in this section is shown in Figure 2.4 In the illustration, the capital letters (X, Y, Z, L, M, N) denote the forces and moments in the body fixed reference frame. u, v, w, and p, q, r denote the aircraft's linear and angular velocities. x, y, z, and θ denote the aircraft position and pitch angle. Not labeled in the illustration are ϕ, ψ variables, which represent the bank angle and the yaw angle. Below are the forces and moments on the aircraft assuming a rigid body, symmetry in the xz plane, and negligible

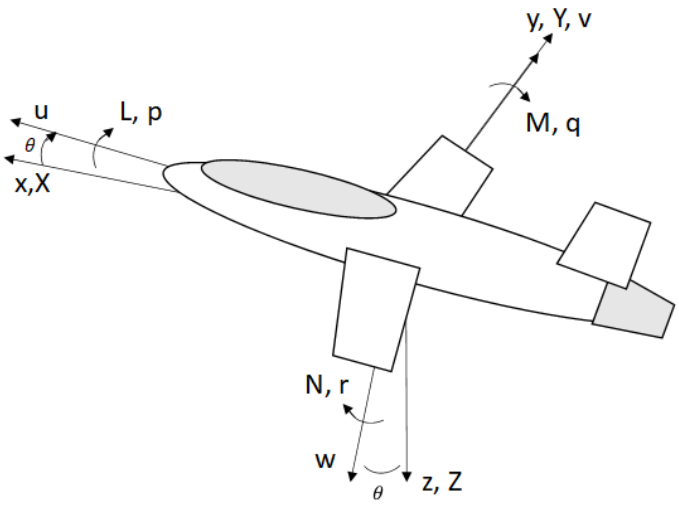

Figure 2.4: Illustration of the aircraft dynamics notation

contribution from the engine spinning rotors, and no wind condition.

$$X - mg\sin\theta = m(\dot{u} + qw - rv) \tag{2.1a}$$

$$Y + mg\cos\theta\sin\phi = m(\dot{v} + ru - pw) \tag{2.1b}$$

$$Z + mg\cos\theta\cos\theta = m(\dot{w} + pv - qu) \tag{2.1c}$$

$$L = I_x\dot{p} - I_{zx}\dot{r} + qr(I_z - I_y) - I_{zx}pq \tag{2.2a}$$

$$M = I_y\dot{q} + rp(I_x - I_z) + I_{zx}(p^2 - r^2) \tag{2.2b}$$

$$N = I_z\dot{r} - I_{zx}\dot{p} + pq(I_y - I_x) + I_{zx}qr \tag{2.2c}$$

Here the m is the mass of the aircraft, g is the acceleration due to gravity, and I is the inertial matrix of the aircraft. The force and moment terms are normalized by aircraft geometry (wing plan-form area, mean cord, and span) and flight condition (total velocity, u, and local air density) to neglect the impact of these factors. These normalized aerodynamic coefficients are linearized in terms of states and inputs to analyze the static and dynamic

stability of the aircraft. For an aircraft with no bank angle and assuming a small disturbance in the dynamics, the aircraft dynamics can be linearized in terms of longitudinal and lateral motion. The resulting form is as follows:

$$\dot{s} = As + Bc \tag{2.3}$$

where s is the state vector, c is the input vector, A is the state matrix, and B is the input matrix. The state vectors for the longitudinal, lateral dynamics and the input vectors are as follows:

$$s_{long} = [\Delta u,\ w,\ q,\ \Delta\theta]^T \tag{2.4a}$$

$$s_{lat} = [v,\ p,\ r,\ \phi]^T \tag{2.4b}$$

$$c = [\delta_a,\ \delta_e,\ \delta_r,\ \delta_p]^T \tag{2.4c}$$

here the δ of a, e, r, and p are the inputs applied for the aileron, elevator, rudder, and thrust controls. In some studies only p, e are considered for the longitudinal dynamics and a, r are considered for the lateral dynamics assuming the two are completely decoupled. Physics-driven system identification methods typically focus on finding the best aerodynamic coefficients to characterize the motion for longitudinal or lateral dynamics. The identified coefficients may be linear or nonlinear depending on the method used.

There are many studies that utilize the general equations of motion to update the aerodynamic coefficients from flight data [72, 73, 74]. The reason for modeling the aerodynamic coefficients instead of generating the state space model is because the result can be generalized to other flight conditions [75]. However, the state space model does generate a model more accurate to the local condition. Some of the system identification methods identify the aerodynamic coefficient regression models in the frequency domain. This is done by updating the aerodynamic coefficients in a recursive manner that maximizes the objective function such as the Baysian model, Fisher model, or least squares model. The

set of equations solved by these methods is focused on identifying the unknown vector θ, which is a list of aerodynamic coefficients, with the following set of equations [72].

$$\dot{s} = f[s(t), c(t), \theta] \tag{2.5a}$$

$$z = h[s(t), c(t), \theta] + v \tag{2.5b}$$

In equation Equation 2.5, f is the dynamic equations, h is the aircraft response, z is the observed states, and v is the measurement noise. For the methods that use the least squares model, the objective function (J) finds the sum of error squared in the measured and estimated aircraft states.

$$J = \sum_{i=1}^{N} [z(i) - h(i)\theta]^2 \tag{2.6}$$

This objective function does not assume uncertainty in the unknown variables. This is different from the Baysian and Fisher models which consider uncertainty and tries to maximize the likelihood of the estimation. A benefit of fitting models in the frequency domain is that the frequency ranges relevant to aircraft dynamics can be modeled[76].

Some studies identified aerodynamic coefficients of the aircraft dynamics through Kalman filters [77, 78, 79]. A Kalman filter estimates the optimal states of a linear system subject to zero mean noise through a two-step process: prediction and correction. The prediction (Equation 2.7) and correction (Equation 2.8) processes are defined by the following set of equations [79].

$$\hat{s}_k = As_{k-1} + Bc_{k-1} \tag{2.7a}$$

$$\hat{P}_k = AP_{k-1}A^T + Q_{k-1} \tag{2.7b}$$

Here P is the covariance matrix, Q is the covariance of the process noise, and hat denotes the predicted variable. The covariance matrix P is a definition of the accuracy of the states and it is referred to as the Riccati variable. As shown by Equation 2.8, the Kalman filter

updates the state variable by the solution of the Riccati equation.

$$K_k = \hat{P}_k H_k^T [H_k \hat{P} H_k^T + R_k]^{-1} \tag{2.8a}$$

$$s_k = \hat{s}_k + K_k[z_k - H_k \hat{s}_k] \tag{2.8b}$$

$$P_k = [I - K_k H_k]\hat{P}_k \tag{2.8c}$$

Here K is the Kalman gain, H is the observation model, and R is the covariance of the observation matrix. Simply put, the Kalman gain reduces the residual error of the predicted and actual states based on the accuracy of the observation, defined by the covariance. Because Kalman filter models are designed to estimate the linearized system, it is often insufficient to use a single model to predict the dynamics of the full flight. Variants of Kalman filters such as extended Kalman filter (EKID), unscented Kalman (UKID) filter, and neural network based Kalman filters are used to model the aerodynamic coefficients in a nonlinear manner. Extended and unscented Kalman filter work with the nonlinear f so that z and s are in the form equivalent to Equation 2.5 [78]. The neural network based Kalman filter uses a feed-forward neural network to generate a nonlinear aerodynamic coefficient model and train this model with a Kalman filter [77].

Studies show that neural networks are capable of generating nonlinear aerodynamic coefficient models with different methods of training. The most common training method is the least squares algorithm. This is also known as the steepest descent learning rule and it guarantees the convergence to a local minimum. Levenberg-Marquardt technique is used by Bagherzadeh [75] to train the Artificial Neural Network (ANN). This technique, also known as damped least squares, updates the weights (w) of the ANN by assuming a quadratic error with model weights. First, the Jacobian matrix of the error in terms of model weights is found. Then the model weights are updated assuming a locally linear relationship to the error. The update is modified by a damping term to improve the solution convergence. The set of equations that define the Levenberg-Marquardt technique is as

follows:

$$e(w) = e^T e \tag{2.9a}$$

$$J = \frac{\partial e}{\partial w} = \begin{bmatrix} \frac{\partial e}{\partial w_1} & \frac{\partial e}{\partial w_2} & \frac{\partial e}{\partial w_3} & \dots \end{bmatrix} \tag{2.9b}$$

$$w_{k+1} = w_k - [J_k^T J_k + \mu I]^{-1} J_k^T e_k \tag{2.9c}$$

Here e is the error in desired and output states of the model and μ is the damping term that modifies the weight update. In this study, six different force and moment coefficients of an aircraft are modeled with a feed-forward ANN. The ANN generated a model with an acceptable range of control commands and the model accurately captured the maneuvers at high AOA. An interesting conclusion made by Bagherzadeh is that the model accuracy increased with the number of inputs (time steps). In another study by Amin et al. [80], a filtered error model learning method for recurrent high-order neural networks (RHONN) is used to find the pitching moment coefficient gradients of an aircraft online. In summary what the filter-based method does is it finds the bounded optimal weight vector of the model as shown in Equation 2.10. The bounds are added to avoid numerical problems when the weight is too large. [81].

$$w^* = \arg \min_{|w| \leq M} [\sup_{(s,c) \in e} |f(s,c) + as - w^T z(s,c)|] \tag{2.10}$$

Here the M is the numerical bounds to the model weights and a is the model constant. The study pointed out that the algorithm had significantly improved identification behavior for dynamical systems with a feed-forward neural network model.

The physics-driven aircraft system models may be used for identifying accident precursors. For example, observer and extended variants of Kalman filters are used in studies to identify aircraft upset conditions [79, 65]. This is accomplished by monitoring flight parameters or aerodynamic coefficients such as coefficient of lift, coefficient of moment,

velocity, or AOA of the Kalman filter variants. The study by Caliskan et al [82] identified aircraft icing with Kalman filter based on the flight performance degradation. Multiple studies focused on identifying braking performance parameters, at ground roll, based on aircraft physics. Van Es [83] evaluated braking performance based on select parameters such as the hydroplaning speed. The relationship between tire pressure and hydroplaning speed is used to generate hydroplaning speed bounds on the contaminated runways. Wahi [84] evaluated braking performance based on the braking distance instead. Braking distance was predicted by estimating the braking energy based on the aircraft's gross weight, temperature, altitude, aerodynamic forces, and braking forces. The change in speed, on a point mass aircraft, due to applied braking energy was summed iteratively until the velocity converged to zero. Pasindu et al. [85] also evaluated braking performance based on the braking distance. Here the braking distance was calculated by integrating the incremental braking distance from the finite element simulation of the tire resistance. Overall the studies show that aircraft models may be used to evaluate flight performance or identify precursors. However, a single linearized dynamics model may not be sufficient to model the full aircraft flight performance, especially at threshold exceedance events where the perturbation in the dynamics model is expected to be large. So models capable of modeling the nonlinear dynamics of an aircraft may be needed.

Data Driven

The physics-driven approach to identifying aircraft systems is focused on generating nonlinear models of aerodynamic coefficients that characterize flight behavior. At the core, this approach relies on the equations of motion to map the aircraft inputs to the states. This allows a generated model to be used for estimating flight states in multiple flight conditions but may risk the accuracy of estimating flight states in each flight condition. A data-driven approach for identifying aircraft systems generates a black box (state space model) that directly maps a vector of inputs to a vector of outputs. This approach does not require

aerodynamic or propulsion information to generate the state space model.

Studies investigated so far show that the Kalman filter can identify the nonlinear aerodynamic coefficients of an aircraft. More studies show that the Kalman filter is also capable of directly identifying the nonlinear dynamics of an aircraft. These studies solve Equation 2.3 by generating the A and B matrices. One variant of the Kalman filter, known as the observer Kalman filter (OKID) is used to estimate the aircraft states during flight [63]. This is made possible by adding and subtracting the observer term Gy_{k-1} to the Equation 2.7a. Here G is an arbitrary observer matrix and y is a vector of Markov parameters. The observer term would modify the A and B matrices so the dimension of the A and B matrices are reduced. This process reduces the storage space and computation time of estimating the aircraft states. Nonlinear Kalman filters (NKF) are also used online and compared to the parameter prediction of the inertial navigation system (INS) of an unmanned aerial vehicle (UAV) [86]. The result shows a 91 percent correction in longitudinal parameters accuracy for NKF optimized with GA over INS. A similar study by Soal et al. [87] used the Kalman filter to improve the sensitivity of a research vessel model to environmental conditions, system configuration, and boundary conditions. While Kalman filters are capable of generating MIMO model of an aircraft, other methods generate a MISO model of flight parameters.

Stelmach [88] has generated a polynomial model of true airspeed. The model basis parameters are second-order full factorial parameters based on time, position, velocity, and distance. As shown in Equation 2.11, the model is trained with a quadratic error cost which modifies the model coefficients (a) in the steepest descent direction.

$$y_j(t) = \sum_{i=1}^{n} a_{ji} t^i \tag{2.11a}$$

$$J = \sum_{i=1}^{N} \epsilon_i^2 \tag{2.11b}$$

$$a = (J^T J)^{-1} J^T y \tag{2.11c}$$

Here y is the output vector (true airspeed trajectory), a is the model coefficient vector, t is time, and ϵ is the model error. Although this study does generate an accurate and stable velocity model, it does not consider the impact of aircraft control. Fuzzy systems are also data-driven methods that generate MISO models. They are commonly used for modeling low-level controls and are applied to various system identification applications [89, 90, 91, 92, 93, 94]. The fuzzy logic system consists of a fuzzifier, rule base, fuzzy inference engine, and defuzzifier. This system is characterized by a mapping between adjustable parameters (θ) and a vector of fuzzy basis (ζ) as shown in Equation 2.12 [94].

$$y(x) = \theta^T \zeta(x) \tag{2.12a}$$

$$\theta = \arg \min_{\theta \in R^N} \left[\sup_{x \in \Omega} |y(x) - f(x)| \right] \tag{2.12b}$$

Here y is the fuzzy output, x is the input, Ω is a set of R^N, and f is the actual output. A fuzzy logic system has been used in multiple studies to identify an aircraft system. Kouba et al. [90] used Sugeno and Mandani fuzzy logic methods to model the trailing edge flap for F/A-18 to observe flutters. Sugeno method arranged the input data into packages and used Gaussian memberships to fuzzify each package. The output is the sum of first-order polynomial of the fuzzy inputs normalized by the weights. Mandani fuzzy method adds basis functions to the defuzzification process on top of the Sugeno method. A model of fit coefficient of over 98 percent is generated. Some studies used the capability of fuzzy logic systems to control the aircraft's longitudinal dynamics. The study by Wahid and Rahmat [93] compared the stabilization time for LQR and fuzzy controller to a step change in the pitch attitude. A difference in how LQR and fuzzy logic systems train the model is by assuming a quadratic objective function in terms of states (s) and inputs (c) as shown in

Equation 2.13.

$$J = \int_0^t [s^T Q s + c^T R c] dt \tag{2.13a}$$

$$u = R^{-1} B^T S s \tag{2.13b}$$

$$0 = A^T S + S A + Q^T Q - S B R^{-1} B^T S \tag{2.13c}$$

Here Q is the state weight matrix, R is the input weight matrix, A is the system state matrix, B is the system input matrix, and S is the solution to the Riccati equation. In the study, the fuzzy logic controller behaved like an over-damped system with no overshoot while the LQR controller behaved like an under-damped system that stabilized faster. Another study by Gao et al. [94] proposed a fuzzy adaptive control for hypersonic aircraft with unknown dynamics. The controller is capable of generating a stable close loop trajectory that followed the command flight path angle and altitude. A study by Furey et al. [91] studied a fuzzy logic controller and neural network controller to maximize the L/D (lift over drag) of a delta wing for a given maneuver. The study found that both fuzzy logic and neural network models are capable of capturing nonlinear behaviors in aircraft pitch. Fuzzy logic may also be applied to other machine learning models such as a neural network. Lee and Teng [89] investigated a recurrent fuzzy neural network (RFNN) that can identify and control nonlinear dynamic systems. A feedback layer is added on top of the membership layer where the nodes use the basis function to act as a memory unit. The study found that RFNN is capable of solving dynamic and static problems based on how θ is modified.

Studies presented above lead to the potential of neural networks to be used for modeling the dynamics of a system [95, 96, 97, 98, 99, 100, 67, 82, 101]. A study by Harris et al. [95] used a neural network to model A and B matrices. The study concluded that the ANN system identification (SID) successfully generated longitudinal and lateral linear aircraft models. In fact, the ANNSID is capable of learning linear models for both small (F-16XL) and large (Anaconda SUAS) aircraft. However, linear models are restricted to

the local flight condition with small perturbations, which means the model needs to be updated iteratively to be accurate. Multiple studies are focused on identifying nonlinear neural network state space models. Roudbari and Saghafi [96] considered three different neural network architectures to identify linear and angular acceleration of aircraft states. First, is the NNARX model. This model predicts one step ahead of the input signal. Here the input signal is defined by a sequence of states and controls as shown in Equation 2.14.

$$s_n(k+1) = f(c(k), c(k-1), ..., s_p(k), s_p(k-1)) \tag{2.14}$$

Here n is the predicted states and p is the input states. Second, is the internal recurrent neural network based on Elman and Jordan models. Elman network is a two-layer network with hidden layers that use outputs of its own layer as feedback inputs. Jordan network is similar to the Elman network except the output from the model is used as model feedback inputs. Last, is the hybrid recurrent network that combines the internal and external recurrent neural network models. Roudbari and Saghafi also introduced GA like neural network model called ANN simultaneous optimization algorithm (NNSOA) as a means to train the aircraft model. The key steps behind GA are random selection, mutation, and crossover. In summary, the algorithm generates a random list of algorithm parameter combinations. At each iteration, the combination set with poorer performance mutates (randomly modify some algorithm parameters) and then crossover (randomly exchange algorithm parameters) with some of the existing population. The process repeats until the best algorithm parameter combinations is selected. The GA may also be used to update the neural network structure to update the internal or self-feedback connections. The study showed that the hybrid model is best at learning the effects of coupling among aircraft control inputs due to the improved performance from self-feedback connections. The study also showed that the NNSAO model significantly improved the generalization of the model for the output of sample data. Besides aircraft modeling, the NNARX architecture is used generically

for identifying the nonlinear dynamics of various systems. Boussaada et al. [101] used NNARX, neural network model, to identify the direct solar radiation on sailboats based on the theoretical solar radiation level, predicted cloud cover, and global solar radiation.

Kumar et al. [98] compared the performance of three different neural network architectures similar to the ones introduced. The three architectures introduced here are the multilayer feedforward neural network (MLFFNN), diagonal recurrent neural network (DRNN), and NNARX. The identification models of MLFFNN and DRNN, shown in Equation 2.15, differ from that of NNARX, shown in Equation 2.14.

$$S_n(k+1) = f(c(k), c(k-1), ...) + \sum_{i=0}^{m-1} b_i S(k-i) \tag{2.15a}$$

$$S_n(k+1) = f(c(k), c(k-1), ..., s_p(k)) \tag{2.15b}$$

The structural output layer for all three architectures is the sum of induced states from each layer to the corresponding weight as shown in Equation 2.16a. The difference appears from how the induced states are derived. The induced states of MLFFNN are a sum of the inputs multiplied by the corresponding weight. Induced states of NNARX add the MLFFNN model to the sum of feedback states multiplied by their corresponding weights. DRNN goes a step further to add NNARX model to the feedback-induced state multiplied by the corresponding weight. The mathematical model of the three architectures is shown in Equation 2.16, in the respective order.

$$y(k+1) = \sum_n w_n f[V_n(k)] \tag{2.16a}$$

$$V_{MLFFNN}(k) = \sum_l w_{ln}(k) x_l(k) \tag{2.16b}$$

$$V_{NARX}(k) = V_{MLFFNN}(k) + \sum_l w_{ln}(k) y_{NARX}(k) \tag{2.16c}$$

$$V_{DRNN}(k) = V_{NARX} + W_n(k) f[V_{DRNN}(k)] \tag{2.16d}$$

Kumar et al. [98] identified that the performance of each architecture, in terms of robustness and fit, for various nonlinear functions ranked in the order of DRNN, NNARX and MLFFNN. So based on this study the nonlinear system model performance scale with complexity.

Some more complicated neural network architectures are introduced to better model the nonlinearity in the time series data. These models include GRU and LSTM models. An issue with the RNN is that they rely on back-propagating feedback terms to memorize the historical data. This may result in back-propagating errors blowing up or vanishing. LSTM [102] and GRU [103] models resolve this by introducing memory gates. LSTM models open and close these feedback memory gates in the network through nonlinear multiplicative units to approximate long-term information with significant delays. GRU models use two RNN layers in encoder-decoder fashion. The encoder uses input data to form a representative sequence of fixed-length data. This data transform the representative data to minimize the error between output and target states. Wang [97] compared the performance of conventional RNN and LSTM models for identifying a nonlinear dynamic system function. The study found that both RNN and LSTM models converged but the LSTM model converged much faster. This result can be validated by another study by Ogunmolu et al. [99] where the convergence of RNN, GRU, and LSTM models are compared. However, this study also noted that the conventional RNN models resulted in a higher fit than GRU and LSTM models and the need for these more complex models should be re-evaluated for each problem.

Overall studies show the potential of data-driven methods to identify the nonlinear dynamics of an aircraft. Aykan et al. [82] presented the potential and the need to identify the nonlinear dynamics of an aircraft accurately. Here neural network model is used to evaluate the impact of icing to reconfigure the flight control system and advise the pilots. Unfortunately, icing is not the only adverse weather condition that can significantly impact flight performance. Unexpected gusts, crosswinds, and contaminated runways are some

of the significant contributors to runway overruns. Based on the studies introduced above, the impact of gusts and crosswinds may be characterized by the change in the aircraft velocity vector. Similarly, contaminated runways may be characterized by a multiplier on the required landing distance or braking coefficient. However, a simple multiplier does not fully capture the change in aircraft behavior and this error may have an impact on the overrun analysis. Hence, accurate nonlinear dynamic models of an aircraft that considers the impact of external factors would provide more accurate feedback to the pilots. This improvement can enhance the ASK of how the pilots respond to these overrun precursors in future flights. Multiple gaps can be identified regarding the development of aircraft dynamic system models.

- Gap 3 – current studies that model aircraft systems do not simultaneously optimize the model-related parameters and data-related parameters
- Gap 4 – current studies on simulating flight states lack analysis on the sensitivity of the model to external factors
- Gap 5 – current studies that generate aircraft system models lack the evaluation of the model control space

2.2.3 Prediction and Simulation for Mitigating Runway Overruns

So far the literature survey focused on identifying methods for characterizing certain flight parameters. This section will survey how the characterized flight parameters are used currently to improve flight safety. Pilot behaviors are known to have a significant contribution to runway excursion. IATA safety report shows that turbojet and turboprop aircraft flights between 2017 to 2021 saw 58 runway excursions, which corresponds to an accident rate of 0.32 per million flights [4]. Flight crew errors (and their percent contribution to the runway excursion) include manual handling (41), SOP adherence (31), and failure to go around after destabilized on approach (24). Flight crew errors have a significant contribution to

other accident types as well. For loss of control, the SOP adherence and manual handling both have a 50 percent contribution. For controlled flight into terrain the SOP adherence, and manual handling have 57 percent and 43 percent contributions. These data show that there may be a need to increase awareness and training to reduce risk.

Global organizations, such as IATA and ICAO, have put multiple efforts to identify the areas of concentration to reduce runway overruns [17, 104, 105, 22, 106]. As mentioned in chapter 1, the GAPPRE recommended actions in six aspects of aviation to improve runway overruns. These action plans provide guidance for long and short-term improvements. Many of the recommendations discuss a need to develop, improve, or include realistic training scenarios to train the SOPs and awareness to external risks (OPS3, MAN2). One recommendation (REG6) also mentioned improved training programs through the use of performance indicators. These recommendations require a quantified means of evaluating flight performance based on how well the pilots respond to external factors.

IATA also released guidance regarding new and modified training methods. One of which is the Data Report for Evidence Based Training (EBT) [107] and the other is Guidance Material for Instructor and Evaluator Training (ET) [104]. ET document emphasizes competency-based training that checks and balances the competency of the trainee, instructor, and training. One of the behaviors that the competency is evaluated on is how well the trainee adapts to situations that deviate from the planned sequence of events. This behavior is not only focused on the trainees' behavior but also the realism and attitude generated by the training tool and the trainer. Another behavior mentioned in this document is the need to provide clear feedback to trainees based on their performance grading and the outcome of the assessment.

EBT document emphasizes on some strong cases to change the pilot training methods. These cases were made based on flight data from six different aircraft generations. Some of the major findings from this report are listed below.

1. Pilot handling skill decay is observed with increased reliance on automation, espe-

cially for adverse weather conditions

2. Majority of unstable approach flights still proceed with landing instead of go around and the risk of go around is high due to the low frequency and awareness of this maneuver

3. Although high severe landing precursors events are more frequent for unstable approaches, the overall rate of flights with landing precursors is similar for stable and unstable approaches

4. Focused attention during training to improve compliance behavior based on the observed relationship between the number of errors per flight to the number of intentional noncompliance error

5. Lack of situational awareness (SA) resulting in an increase of SA problems for later generation aircraft (gen 3 and 4) with advanced tools

In summary, there is a lack of evolution in pilot training for later-generation aircraft. The training should enhance the use of sophisticated tools to improve risk awareness and provide clear feedback to the pilots on their performance. The question is then how to provide clear and meaningful performance feedback.

Flight Evaluation

Multiple regulatory guidance is released over time to improve risk awareness and aircraft handling [27, 12, 28]. The FOQA program is one example. This program allows regulators to observe the trend in flight safety with aggregate flight data based on events defined by flight parameter thresholds. Multiple studies by international organizations, manufacturers, and regulators, presented so far, identified precursor events significant to runway overrun [9, 8, 10, 23, 15, 16]. The Figure 2.5 is an illustration of some of the precursors identified by European Aviation Safety Agency [23] and Boeing [16] for landing overrun and veer off

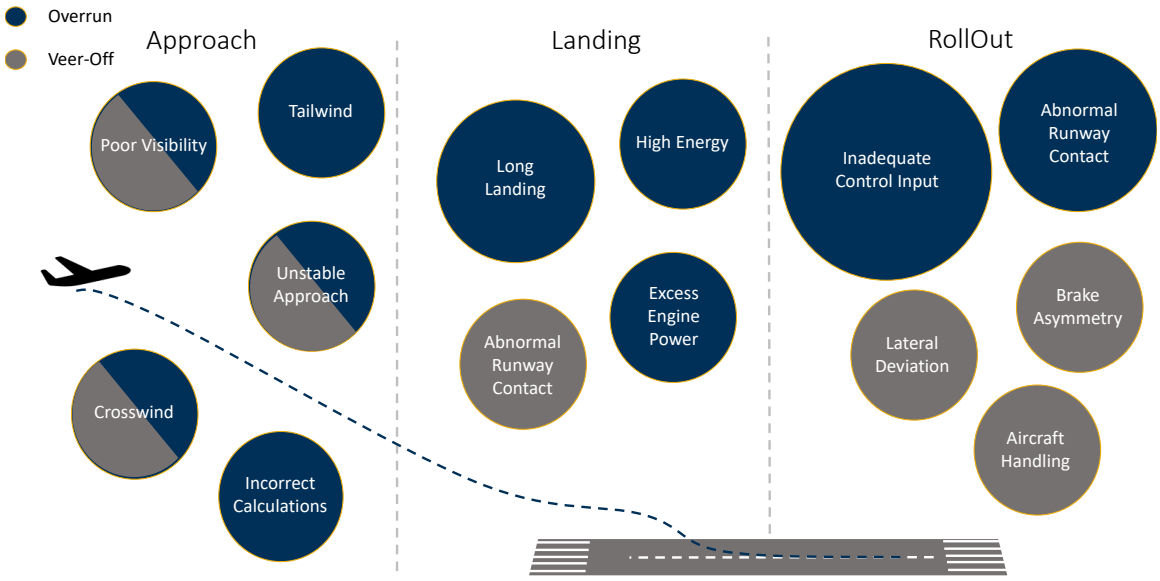

Figure 2.5: Illustration of precursors that contributes to runway excursion for each phases of the landing process.

accidents. A study by Puranik et al. found methods for identifying threshold exceedance events based on FOQA data [108]. This study defined the flight metrics, for characterizing the runway braking performance, based on flight parameters recorded in FOQA data. The flight metrics are categorized into metadata metrics, single-point metrics, and calculated metrics. These metrics show the potential for FOQA data to quantify accident precursors. On the other hand, there are multiple studies focused on monitoring the data to acquire knowledge about the precursors and how these parameters are perceived by the flight operators [105, 109, 110, 111, 83, 61, 40, 41, 42, 44]. Monitored flight data is sometimes used to generate flight condition models to evaluate the braking performance based on speed and braking coefficients on contaminated runways. In another study by Oehling and Barry, a flight scoring tool is generated based on the level of flight anomaly observed from unsupervised machine learning methods. Memarzadeh et al. used unsupervised learning to augment missing precursor event labels to characterize flights with unknown precursors. These studies show a move towards machine learning to quantify and evaluate flight performance.

Other releases in the guidance include approach and landing procedures to mitigate

risk. These regulations identified difficulties in landing flight phases, quick methods to plan for landing, and significant parameters to consider. Then there are also documents that go into details of SOPs for landing [112, 6, 113, 106, 16]. These documents discuss in detail the sequence of control from approach to landing, call signs, go-around condition thresholds for various flight parameters, and surveys from pilots regarding landing such as the perceived lowest altitude for a safe go-around. Much of this regulatory guidance utilizes stable approach parameters, mentioned by Campbell et al. [26], and braking parameters, mentioned by Konrad et al. [14], to measure flight safety. These are expected to improve awareness of precursors that may lead to accidents.

These precursors and flight metrics are expected to change whether the aircraft is in the air or on the ground due to the change in aircraft dynamics. In the air, the flight performance typically emphasizes the aircraft stability at various gates. These gates are typically arranged based on the aircraft altitude such as 1000 ft, 500 ft, or altitude at threshold crossing. However, evaluating the approach on select altitude gates may not be sufficient because the aircraft stabilization criteria are designed to ensure stable flight throughout the entire approach. Also, the aircraft configuration may deviate significantly in adverse weather compared to clear weather. So the flight performance evaluation needs to consider the aircraft energy and configuration trajectories during the approach phase. For example, the engine configuration may be set to idle at 1000 ft altitude gate but the energy may remain high due to wind. This indicates that the proper evaluation of the approach would require an independent flight metric for each criterion. The weight of each flight metric may also need to change with altitude since their contribution to various accident precursors changes. A study by Ackley et al. [114] identified significant flight parameters to unstable approach by sequentially identifying the significant parameters at higher altitudes. The study showed that the contribution of vertical speed and ground speed increases with a decrease in altitude. In-air flight performance evaluation also needs to be sensitive to external factors and how well the pilot reacts to them. The current guidance on the approach process encour-

ages a stable continuous descent approach [27]. Based on this guidance, the variance in the landing technique should cause large changes in aircraft states assuming that the aircraft is stabilized at 1000 ft gate. Instead, the variation in the landing procedure mainly results from external factors. A shortened list of external threats, identified by IATA [4], that contributes to runway excursion is shown in Table 2.4.

Table 2.4: Percent contribution of external threats to runway/taxiway excursion for flights between 2016 and 2020 [4]

Threats	Percent Contribution
Airport Facilities	55
Meteorology	55
Wind	37
Contaminated Runway/Taxiway	37
Thunderstorms	30

This data and the evidence-based training analysis [107] point to unexpected external factors as the cause of many accidents. So sensitivity to external factors would be key to properly evaluating flight performance and giving pilots feedback that can enhance attitude, skill, and knowledge of these external factors. Flight performance may also be evaluated based on training flights defined by regulatory guidance. For one of the training tool evaluation processes, the pilot is to perform a circular approach with maximum landing weight, night scene, low visibility, and at least 90 degrees turn to the final approach [115]. The simulated flight is evaluated based on the need for ”unusual maneuvers” and if the approach is viable. This criterion identifies the significance of considering the entire approach trajectory over specific points in the approach process for evaluating in-flight performance.

On the ground, multiple studies emphasize the braking capability of the aircraft under various conditions. One of the most significant conditions to consider for landing is the contaminated runway. Multiple guidance characterized the impact of runway condition on the braking distance with a multiplier [33, 116], an example is shown in Table 2.5. This method of evaluating landing may be useful for identifying performance limits for overruns based on the maximum distance required for the aircraft to come to a full stop. However, it

Table 2.5: Landing distance modifiers based on aircraft architecture and runway condition [116]

Runway Condition	6	5	4	3	2	1
	good			⇒		poor
No Reverse	1.67	2.6	2.8	3.2	4.0	5.1
With Reverse	1.67	2.2	2.3	2.5	2.9	3.4
Turboprop	1.67	2.0	2.2	2.4	2.7	2.9

is insufficient for evaluating the performance throughout the ground roll.

Providing the right feedback is one concern. It is important to emphasize that the feedback has to be quick as well. A potential solution to this is mobile devices. Conventional automobile navigation systems provide immediate feedback on the quality of a drive to the driver. This kind of feedback can be brought to the aviation industry with properly designed prediction and simulation models. To achieve this the flight evaluation tool has to be light and precise. Overall current methods of flight evaluation is focused on identifying flight metrics that characterize the performance of interest. Also, flight performance is evaluated on discrete settings based on maximum and minimum metric bounds. The final set of gaps can be identified by surveying the need for a prediction and a simulation model to increase awareness to runway overrun.

- Gap 6 – current studies on identifying metrics correlated to runway overrun lacks analysis based on machine learning models
- Gap 7 – there is a need to give direct and quick feedback on the flight performance to help pilots enhance their flight experience
- Gap 8 – current studies lack how the prediction and the simulation model outputs can be mapped to proximity to runway overrun
- Gap 9 – there is a need to continuously investigate how prediction and simulation models can increase awareness to runway overrun

CHAPTER 3
PROBLEM FORMULATION

Runway overrun is a significant threat to flight safety. The frequency of runway overrun makes it a top concern for the aviation industry. Efforts from various organizations have been fruitful in reducing fatal accident rates in aviation history. However, as presented in Chapter 1 the runway overrun accident rate has remained fairly constant for the past few years indicating the need for a change in flight safety. A survey on current methods for runway overrun mitigation showed that current goals focus on providing clear feedback to the pilots to increase awareness of overrun precursors. Providing clear feedback to the pilots is important for strengthening the ASK of the pilots to adverse flight conditions. So methods of providing clear feedback to the online and offline applications were surveyed in Chapter 2. It is important to mention that the feedback also has to be quick. One comparison would be the feedback from automobile navigation systems that can quantify the quality of a drive immediately after shutting down the car. The methods surveyed for this research are flight performance quantification based on flight behavior prediction and flight simulation. Current studies show the potential for these methods to improve flight safety, but there are challenges unaccounted for. This chapter is organized to formulate the research objective based on identified challenges. The research objective will be followed by several research questions that can solve each individual challenge identified.

3.1 Research Objective

The research objective of this dissertation and the related assumptions are as follows:

1. The range of vehicles considered for this study has not been identified. Due to the significant difference in the controls and the aircraft dynamics, it may be difficult to

Research Objective

Develop a methodology that increases pilots' awareness of the risk of runway overrun by predicting and simulating flight states that define a stable approach

generate a tool generic for all airframes. This work will focus on a regional turbojet aircraft with a single tail number.

2. It is acknowledged that technical errors do contribute to runway excursion [4]. However, for this study, the aircraft is assumed to be fully functional.

3. In this study the accident precursors are defined based on the threshold exceedance events identified by FAA in Advisory Circular 120-82 [12]. It is assumed that the precursors defined by threshold exceedance do have a significant contribution to overrun.

4. Since the purpose of FOQA program is to observe aggregate trends, these data lack identifiers. This compounded with limited access to weather and runway contamination data limits the capability to identify weather conditions for each flight. So only high-speed wind will be considered for adverse weather conditions.

With these assumptions in mind the overarching hypothesis for the research objective is as follows:

Overarching Hypothesis

If past FOQA data is used to create a prediction and a simulation model of flight states that define a stable approach, then these models can be used to improve the awareness of the risk of runway overrun for civil transport aircraft.

The overarching hypothesis is based on the methodology developed for this research to mitigate runway overrun. The methodology includes the prediction of accident precursors, simulation of flights exposed to overrun precursors, and post-flight training through prediction and simulation. An overview of the methodology is shown in Figure 3.1. As shown, the methodology consists of three building blocks: Prediction, SysID (simulation),

and Training. A four step process is developed for the prediction building block and the four steps are: data pre-processing, baseline model design, prediction model training, and testing. Then the same four step process is used for the simulation building block. For the prediction building block, the first three steps are performed offline, and the last step is performed online. For the simulation building block, the entire process is performed offline. For the testing step of both models, the accuracy of estimated states is measured with the test flight data. In the pre-processing step, flight data are used to identify high-risk flights, based on precursor triggers. The risk flights will be used for enhancing ASK to enable mitigation of runway overrun. The contributions from this study are shown as blocks 1 to 5. Details on each of the blocks will be introduced below.

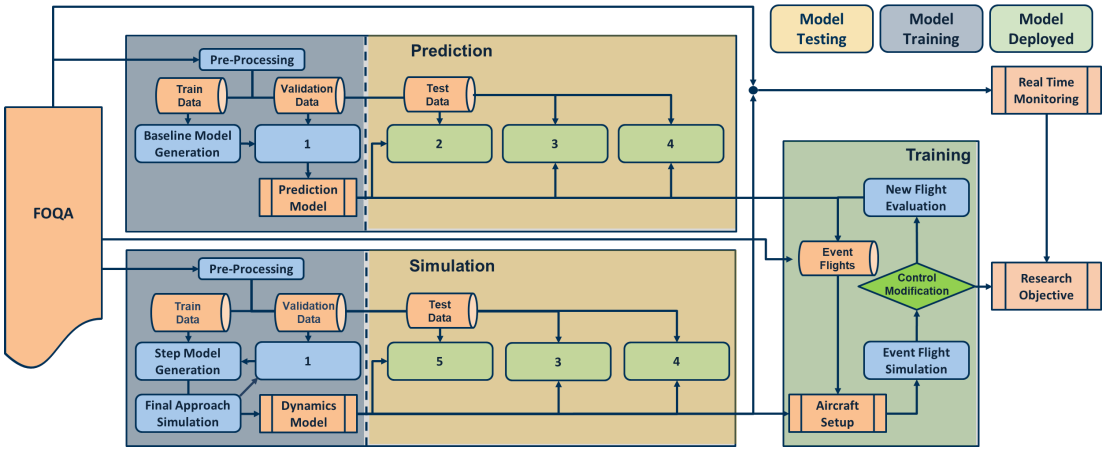

Figure 3.1: Overview of the proposed methodology

3.2 Research Questions

In this section, research questions are developed for each of the prediction and sysID blocks to identify the gaps that need to be closed to realize the research objective. Each question identifies the basic requirements for developing the corresponding building block. Then a set of sub-questions are generated based on the gaps identified, from the research survey in chapter 2, for realizing the research questions. Lastly, hypotheses and experiments are developed to close the identified gaps. The gap closures are focused on the clarity and

accuracy of the feedback provided to the pilots to help improve awareness towards runway excursion precursors.

3.2.1 Accident Precursor Prediction

The findings from subsection 2.2.1 point toward a multi-point or sequential prediction model of aircraft states with advanced prediction time that aids pilots' operations during landing. This requirement was supported by a study on the change in pilots' stress levels when prediction alerts are added at a different point in the flight. Multiple machine learning tools were surveyed to find one that can predict flight metrics such as ground speed. Most of the prediction models were focused on predicting the flight metrics in the near future (below 3 seconds), which is not enough time for the pilots to respond to [26]. However, the study by Puranik et al [108] showed the potential for a machine learning model to perform advanced prediction (below 18 seconds) of ground speed while maintaining an accuracy of 2.98 knots. Further review of other prediction tools showed the potential for machine learning models to predict other flight metrics such as fuel flow and flight control. Although ground speed, true airspeed, fuel flow, and flight control are significant flight metrics for runway excursions, there is a lack of investigation on predicting flight metrics relevant to the runway overrun precursors.

Runway overrun precursors may be predicted instead of flight states relevant to runway overrun. Models that directly predict runway overrun precursors based on approach data are surveyed and found studies that predict hard landing and unstable approach precursors. These are two of a larger set of precursors that characterize runway overrun. Anomaly detection methods, based on unsupervised machine learning, that are capable of categorizing a flight by various accident precursors are found. However, these methods are typically strong in identifying precursors from a specific type of data, such as continuous or discrete. This is not favorable since FOQA data is heterogeneous data with continuous, discrete, binary, and text data types (in some cases). An alternate method of precursor prediction

is by predicting flight metrics and mapping them to the precursors by their definition. A downside of this method is that the accuracy of the predicted metrics needs to be high in order to accurately label the flight with precursors. Optimizing the prediction accuracy of a machine learning model is a difficult process that requires consideration of three factors: model architecture, model parameters, and data structure. Current studies lack a discussion on the model optimization process that considers all three factors. Based on these observations, the following research question is posed by this work.

Research Question 1

What is the prediction model that can accurately predict runway excursion events, and the metrics that define them, far enough into the future for the pilots to respond.

There are two key considerations to answer this research question. The first is accuracy. An aircraft is a large system with six degrees of freedom. It is controlled by at least four inputs and influenced by a large number of external factors such as altitude, temperature, wind, and more. These parameters exhibit Spatiotemporal behaviors and are coupled in a nonlinear way. Common examples include the elevator-thrust relationship to the longitudinal dynamics during trim and lift-AOA relationship at high AOA. Another issue is that machine learning models require a large amount of data to train, validate, and test the model. Also, the flight pattern will have to change with the flight condition. Especially if the prediction is done across multiple flight phases. This lead to the following sub-question.

Research Question 1.1

How can an accurate prediction model be created to predict metrics that define runway overrun precursors

The sub-question can be addressed if only a single flight phase is considered for analysis. In fact, pilots are encouraged to perform a stable approach and are recommended to achieve stable flight by 1000 ft height above touchdown [27, 22]. So the flight trajectories

below 1000 ft are expected to be more consistent. Another study showed that the approach condition at 100 ft height above the touchdown has a significant contribution to the touchdown performance. So flight data from the final approach, where the flight is stable, can be used to predict flight metrics below 100 ft to label runway excursion precursors. As discussed, machine learning models require a large set of data for training and validation. FOQA data [12] typically contains thousands of flight parameters recorded sequentially between 0.25 and 16 Hz frequency. These parameters include sensor readings, control switch settings, ground measurements, and more. The parameters are arranged in a hierarchical structure with atmosphere, altitude, aircraft state, GPS, engine, and fuel data at the system level and various recordings in different units at the subsystem level. A subset of this data is used in other studies for predicting ground speed, and this data will be used for predicting other flight metrics. The data set utilized in this work is obtained from NASA's open-source FOQA data available at the DASHlink website [1].

The second consideration for research question 1 is the prediction range. Studies show that pilots cannot react to predictions made too close to landing. So predictions must be made well ahead of event occurrence. Advanced prediction can only be realized when the model can accurately map the full input data to the aircraft states. Complex data may be accurately mapped with machine learning models through supervised learning and by using labels that characterize the data far into the future from the input data. Simply modifying the labels, to perform advanced prediction, hurts the accuracy of the prediction, especially for simpler models that cannot capture the full dynamics of an aircraft. How well the model represents the aircraft data is dependent on how the input data is pre-processed. Data pre-processing needs to consider multiple factors such as input data window, input data parameters, and input data index. The model architecture is another factor that determines how well the model represents the input data. With these considerations, the following two sub-research questions were formulated. Research questions 1.1, 1.2, and 1.3 can be

[1] https://c3.nasa.gov/dashlink/resources/?page=3&sort=-created&type=28

Research Question 1.2

How to increase the prediction range to provide pilots with enough time to respond to the precursor predictions

Research Question 1.3

How to identify the proper data structure to pre-process the data used to train the prediction model with the goal of maximizing the range and accuracy of the model

combined to address gap 1.

A study by the author [117] was on ground speed prediction with various combinations of machine learning models. The purpose of this study was to generate a model that can capture the spatiotemporal behavior of flight data. To do so, the flight data was pre-processed with three steps so that the machine learning model can recognize the spatiotemporal behavior of flight data. The First step is data dimension reduction by removing erroneous data and text data. The Second step is data index modification by rearranging the data based on altitude instead of time. The Last step is data target selection by using prediction metrics on four altitude points of interest. The study also used a hyperparameter optimization tool, called hyperband, to optimize the model design. Hyperband is a method of finding the optimal configuration through successive halving [118]. The result was a successful CNN-LSTM model that can predict touchdown ground speed from 300 ft above touchdown with an accuracy of 3.16 kts. A table that compares the result of this study to other ground speed prediction neural network models are shown in Table 3.1. This result

Table 3.1: Neural network prediction models for aircraft ground speed

Measure	Lee et al. [117]	Tong et al. [40, 45]	Diallo [41]	Kang et al. [42]
RMSE (kts)	3.16	0.0065	3.5	1.71
Prediction Time (s)	12 (300 ft)	1	Retrospective	3 (50 ft)
Prediction Type	Multi-Point	Single-Point	Sequential	Sequential

is on par or less accurate compared to other studies but predicts much further. The result may improve by considering more factors, alternative machine learning models, and other model design space exploration methods [119, 120, 121]. For example, the study by Ogun-

molu et al. [99] showed that although LSTM-based models converge faster, their accuracy may be lower than that of simpler RNN models.

Various machine learning structures and models may be considered to capture this complex flight data. One of the benefits of the machine learning model is that they can be combined to add the strengths of each model as shown from many other prediction models [122, 123, 124, 125, 126].

The following hypothesis is formed based on these observations.

Hypothesis 1.1

If the prediction range and accuracy are evaluated in terms of model structure, and data structure and the prediction model was trained with FOQA data, then the response surface model will find an optimal prediction model that is capable of predicting flight states far away with accuracy.

To address hypothesis 1.1, one experiment will be performed. The purpose will be to find a hyperparameter optimization algorithm that can simultaneously optimize the prediction model design space. For the algorithm, The design space will be in terms of model and data structural parameters and the goal will be to maximize accuracy and range. During the optimization process, the prediction results for each configuration tested will be compared to find the best configuration. A test may be performed after the completion of hyperparameter optimization to determine if the optimized neural network model is successful in learning the flight behavior to improve prediction. The test will be conducted by comparing the neural network prediction model to a polynomial regression model of predicted features. The title and illustration of the first sub-experiment are shown below.

Experiment 1.1

Flight states prediction with neural network models

Experiment 1.1 starts with data pre-processing and parameter selection. Here, each of the flight data will be modified based on the selected input window, index, and parameters of interest. The data structure is an important factor to determine the ”goodness” of the

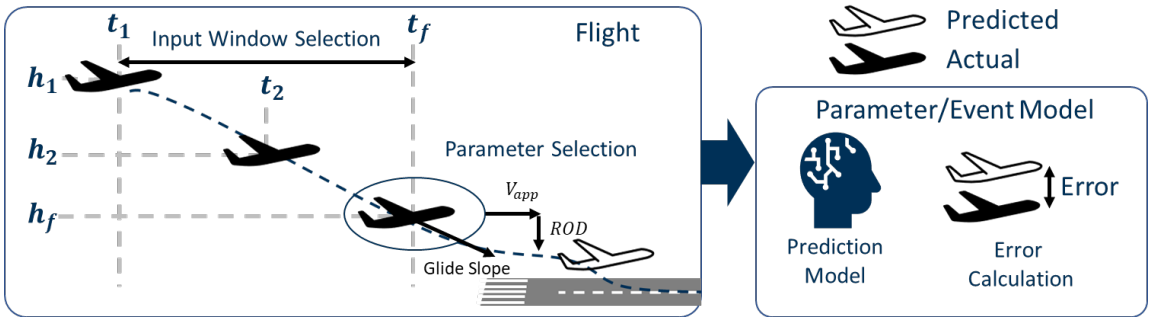

Figure 3.2: Flight parameter prediction experimental setup

input data for representing the predicted data. This is because the machine learning models are trained assuming that the labels used for training are accurate. The base input window will be 1000 ft to 300 ft above touchdown. This input range is determined based on the study by Campbell et al. [26], where the minimum altitude required during approach for pilots to respond to deviations is identified as 300 ft above touchdown, and the fact that aircraft are stabilized from 1000 ft during final approach. The base index will be altitude above touchdown. Altitude is used as an index since decision gates are typically determined based on altitude and the time it takes for different flights to fly between two decision gates may differ. Experience shows that most supervised machine learning models require an equal length for each set of data. So the input data needs to be configured in such a way to reflect this nature. Another factor to consider is the frequency of the input data. As introduced, FOQA data records each sensor's readings between 0.25 to 16 Hz frequency. Observations from the FOQA data show that using the highest frequency data may increase the contribution from sensor noise. So for the base configuration, 1 Hz frequency was used to train the prediction models. However, prediction models will be retrained with input data will be re-indexed between 1 and 2 Hz frequency. When altitude is used as the data index the altitude step will be varied between 10 and 20 ft to find the best input data frequency for maximizing prediction accuracy. The 10 ft is selected because if the step size is below 10 ft then it will require a lot of interpolation during the start of the final approach where the vertical speed is high, which may impact model accuracy. The 20 ft is selected because

this step size has been identified to generate a good prediction model from past study [117].

Most of the parameters provided by the FOQA data will be used initially to optimize the prediction accuracy. The parameters removed will be those that either does not change between the input windows or have erroneous recordings. The optimized prediction model will be used to find the correlation between each parameter to the variance in the predicted metrics. This will be calculated based on the permutation of importance to find the top features for predicting the flight states.

The overall shape of the input data is as follows. The input data sequence and prediction data sequence will be stacked by flight to form feature vector matrices. An illustration of the input data format is shown in Figure 3.3.

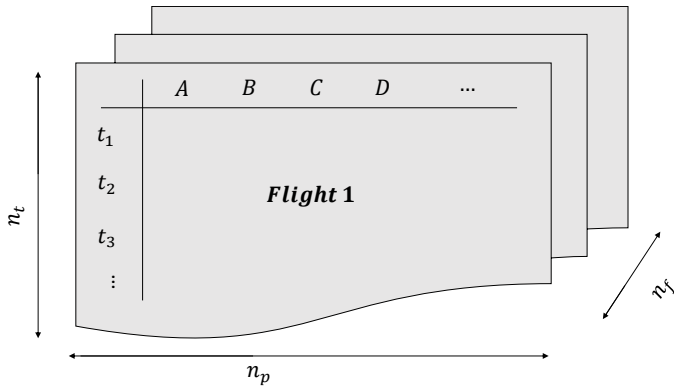

Figure 3.3: Input structure for the machine learning model [117].

The following nomenclature is used.

- n_f is the number of flights/samples
- n_t is the number of altitudes/index for a given flight/sample
- n_p is the number of parameters/features recorded for a given flight/sample

In supervised machine learning, target metrics called labels need to be determined. Based on the literature survey the metrics selected are airspeed, vertical speed, and pitch angle of the aircraft. The prediction model will predict these metrics at various altitudes below the input window. The accuracy of the model will be determined based on RMSE

between predicted and actual trajectories. Only the approach phase data will be used for prediction since post-touchdown the aircraft dynamics change significantly and information provided after touchdown may only add stress to the pilots.

Prediction models will need to be modified to optimize prediction accuracy. The goal is to identify the model design that best captures the spatiotemporal nature of an aircraft. RNN, LSTM, and GRU models will be considered based on the success of these neural network models for capturing the behaviors of nonlinear dynamic systems. The significance of model parameters to the accuracy based on the earlier study by the author is shown in Table 3.2. Note that this table does not show all the model parameters considered and the RMSE is calculated from normalized data so they will seem relatively small. Once the design space is determined, the optimal configuration will be selected. The output of this sub-experiment will be a response surface of the accuracy in terms of data and model configuration.

Table 3.2: RMSE of the ground speed prediction model in terms of model depths.

Rank	CNN Layers	LSTM Layers	RMSE	val. RMSE
0	1	1	0.0616	0.0212
1	1	2	0.0576	0.0236
3	1	3	0.0552	0.0239
8	0	1	0.0793	0.0272
9	0	3	0.0768	0.0281

The prediction model will also need to be optimized to maximize the prediction range. While the accuracy of the model will be optimized by changing the prediction model structure, the prediction range will be optimized by changing the data structure. The input data will be shifted away from the touchdown point. The input window range will also be modified to find the best window to predict flight states below 300 ft HAT. For model and data structure design space exploration a hyperparameter optimization algorithm will be considered. Once the input data configuration is optimized the accuracy of the model will be validated with test data.

Having a prediction model with a long prediction range and high accuracy is not enough

to understand how the model behaves with flights not represented by the training data. Additional studies will be performed to validate the prediction model. Observation from an earlier study on significant parameter identification shows that for a given accident precursor, not all parameters have equal contribution [60]. The parameters most contributing to tire wheel rotation speed exceedance events are identified as aircraft speed, weight, thrust, and air density (altitude). In fact, most of the studies introduced in this document, regarding metrics prediction, used select parameters for generating their models. This is because adding parameters that do not represent a metric will only increase the noise to the prediction. This shows the need to explore significant parameters for runway overrun precursors and the identified parameters may be weighted higher for quantifying the landing performance. The explored metrics may also be used as a reference for future studies that try to predict flight metrics near touchdown. The sub-research question, hypothesis, and experiments to identify the significant parameters are as follows:

Research Question 1.4

What are the flight parameters that have significance in predicting airspeed vertical speed, and pitch of the aircraft near touchdown

Hypothesis 1.2

If the input data with randomized order, by feature, is used to predict flight states, then a difference in the prediction RMSE will show the significance of each input feature to the predicted flight states.

Experiment 1.2

Significant parameter identification

Research question 1.4 will be used to address gap 6.

The last addition to evaluating the prediction model will be prediction model sensitivity analysis. The robustness of the model is an important factor to consider especially when the flight environment can vary significantly. Also by nature, the prediction model cannot

consider external factors beyond the input data. External factors include, and are not limited to, control and wind. So the model performance must be evaluated with control and wind trajectories that are different from the training data. This observation leads to the following research sub-question. Research question 1.5 will be used to address gap 2.

Research Question 1.5

How to evaluate the sensitivity of the prediction model to external factors to ensure that the select model performance does not degrade significantly with large changes in the external factors beyond the input data

A few scenarios that represent large changes in external factors can be considered. The large change in control may be evaluated by observing the prediction accuracy at the flare. Flare is a complex technique that varies significantly for every flight and for every operator. It has a significant impact on runway overrun due to its contribution to energy management. The complexity of this technique makes it a difficult maneuver to predict from the final approach data. In some cases, the pilot may start flaring early to drain the energy in the system. In other cases, the pilot may not even flare to initiate a hard landing on contaminated runways and ensure full contact between the runway and the wheel. In fact survey on the final approach so far shows that there is not a clear method of defining flare initiation. One attempt of doing so is by finding the maximum change in vertical speed and significant change in pitch near touchdown. However, even this cannot guarantee an accurate definition of when the flare is initiated especially when there is noise in the recorded data. Experience shows that the prediction accuracy degrades significantly near touchdown. So it is necessary to identify the trend in prediction error with altitude, especially around flare, so that the prediction outcome can be properly interpreted. To do so the start of flare altitude must be determined for each flight and the accuracy of each flight near flare needs to be evaluated. For the wind external factor, prediction model sensitivity to at least two different wind conditions need to be evaluated. This is because the wind has two components in FOQA data: speed and direction. These two components of wind can be derived

into wind speed in body longitudinal and lateral directions. Then the model performance at large wind speed and large change in wind speed, called adverse wind in this document, must be considered. One more analysis to add for the sensitivity analysis is to evaluate the model accuracy in terms of the data count. This is because data count has significant implications on the prediction accuracy of neural network models. Based on these observations, the following hypothesis and experiments can be considered for prediction model sensitivity analysis.

Hypothesis 1.3 and 4

If the definition of flare and adverse wind can be made from FOQA data and if the data count can be identified at various external conditions, then the different in the prediction RMSE will show the sensitivity of the prediction model to flare and adverse wind.

Experiment 1.3 and 4

Prediction model sensitivity analysis

With this, a complete analysis of runway overrun precursor prediction will be performed and an illustration of research question 1 is shown in Figure 3.4.

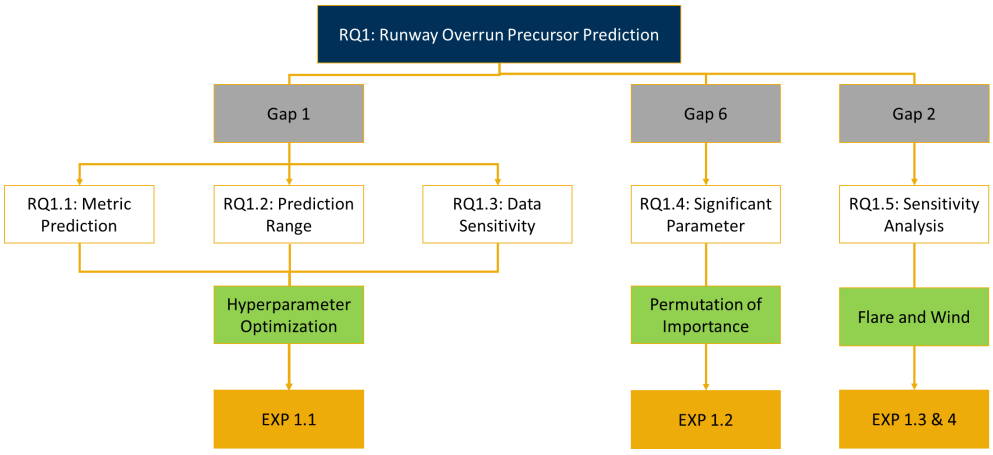

Figure 3.4: Updated diagram of research question 1.

3.2.2 Aircraft System Identification

The goal of this study is to enhance the pilot training process and increase pilots' exposure to overrun risk. So far simulation environment was identified to be used as the training ground to realize this goal. A simulation model is required to generate the simulation environment. The model needs to accurately map the nonlinear relationship between aircraft control, external factors, and aircraft states. The sensitivity of the aircraft model to external factors is extremely important based on the knowledge that the landing SOP, once the aircraft is stabilized, is not expected to change much unless external factors are at play. Based on the survey conducted in Chapter 2, it is identified that there is a potential to develop a single neural network model that captures the nonlinearity and spatiotemporal dependence on aircraft behavior.

Multiple studies identified aircraft dynamic coefficients for adverse conditions such as aircraft upset and aircraft braking on contaminated runways. These coefficients are used on the generic aircraft dynamics model to predict aircraft behaviors. These kinds of physics-based modeling methods map the external factors to one of the parameters in the dynamics equation. For example, tailwind or headwind is mapped to the dynamics with the change in the true airspeed of the aircraft. These methods do not use flight data to directly modify the state space model. This is why they have the benefit of generalizing the model to be used for multiple flight conditions and aircraft types. However, it is harder for these methods to estimate the full impact of external factors on aircraft dynamics.

On the other hand, there are studies that generate the state space model of an aircraft based on flight data. These models are black boxes that map the relationship between controls and external factors to the change in aircraft states. Machine learning models are frequently used for generating these black boxes from time series data such as FOQA data. Studies have compared various neural network architectures to check the accuracy of the state space model against nonlinear functions. The key to capturing the nonlinear dynamics is through recursive feedback to update the model error and identify the full system as

shown by Kumar et al. [98]. To be successful there needs to be enough data to represent the full nonlinear dynamics of the aircraft. So past studies used test flights to collect the necessary data. Based on these observations, gaps are identified in current methods to generate the state-space model of an aircraft. First is that there is a lack of investigation on developing an aircraft simulation model based on FOQA data from nontest flights. As discussed above, FOQA data is a recording of all the flight parameters during the entire flight. This data can provide a view of aircraft dynamics for the flight condition and the control range flown. The second gap is a lack of study on flight safety simulation based on aircraft models identified from in-flight data.

The simulation model is generated in this study to simulate the precursors to runway overrun. Since precursors are defined by thresholds on various flight parameters, the simulation model must be accurate As mentioned above, aircraft maneuver is not expected to change too much post-stabilization. The main cause of variation between flights is external factors such as wind and control. These external factors are well represented in regular flights instead of experimental flights. From these gaps, the second research question is formulated.

Research Question 2

What is the simulation model that can accurately simulate runway overrun precursors and can this model be generated from existing flight data.

Two sub-questions are considered for this research question. The first sub-question is on whether FOQA data is enough to simulate flights under adverse conditions. The flight represented by the FOQA data may not capture the full nonlinear dynamics of the aircraft. Based on this observation the following sub-question is derived.

Research Question 2.1

Can an aircraft dynamics model be made with FOQA data to simulate flights under adverse flight conditions.

The feasibility for FOQA data to model the aircraft state space is dependent on whether

the data fully capture the kinematic and dynamics of the aircraft. Figure 2.4 denotes the notation for all the parameters required to capture the kinematic and dynamics of the aircraft. Additional aircraft parameters to consider include structural components such as weight and fuel flow. External factors that affect flight parameters must also be included. These include deflection in control surfaces, engine thrust, landing gear position, wind velocity, and wind direction. Observation of the FOQA data shows that all of these parameters are included. The sub-system level breakdown of the FOQA parameters are shown in Table 3.3.

Given sufficient data, neural network models may generate a dynamic model but it is known to have some errors. It is guaranteed with the universal approximation theorem that neural network models can approximate a nonlinear model with bounded error [127]. The theorem assumes an activation function $\psi(t)$ that is a nonconstant, bounded, and monotonically increasing continuous function. For this activation function, there exists $N \in \mathbb{Z}$, $v_i, b_i \in \mathbb{R}$, and $w_i \in \mathbb{R}^m$ such that the following is true:

$$NN(x) = \sum_{i=1}^{N} v_i \psi(w_i^T x + b_i) \tag{3.1a}$$

$$e > |NN(x) - f(x)| \tag{3.1b}$$

Where NN is a neural network neuron, f is the true value independent of the activation function, x is the input, and e is the arbitrary error term. This theorem brings key insight that any nonlinearity may be approximated by introducing more activation functions. This may be true when the prediction of a nonlinear function is made for a single step, but for this study, the model is simulated to generate a full trajectory. Harrison [128] summarized Haddad and Chellaboina's work and stated that for a reasonably accurate dynamic model, the resultant trajectory predicted by the model with a small time horizon will be similar to the true trajectory. To prove this, for two dynamic systems $f : D \to \mathbb{R}^m$ and $g : D \to \mathbb{R}^m$, both f and g are assumed to be Lipschitz continuous on D. The error between these two

Table 3.3: List of aircraft states, controls, and external factor parameters identified in FOQA data

No.	System	Metric
Aircraft States		
1	Acceleration	Vertical
2		Longitudinal
3		Lateral
4	Attitude	Roll
5		Pitch
6		True Heading
7		Angle of Attach
8	Engine	N1
9		Fan and Core RPM
10	Velocity	Ground and True Airspeed
11		Altitude Rate
12	Position	Longitude Position
13		Latitude Position
14		Pressure Altitude
15	Structure	Fuel Quantity
Atmosphere		
1	Wind	Speed
2		Direction
3		Pressure
Control		
1	Surface	Elevator
2		Flap
3		Spoiler
4		Aileron
5		Rudder
6	Engine	Throttle

dynamic systems is bounded over the full domain, D, by some arbitrary constant error term, e. In fact, let's say the initial conditions satisfy the following:

$$||x_0 - y_0|| \leq \gamma \tag{3.2}$$

Where γ is a small arbitrary error term for the initial states of functions f and g. When

x and y state trajectories of functions f and g are propagated to future time step t, the difference between x and y at time step t is bounded by the following:

$$||x(t) - y(t)|| \leq \gamma e^{L|t-t_0|} + \frac{e}{L}(e^{L|t-t_0|} - 1) \tag{3.3}$$

Where L is Lipschitz constant. This shows that the aircraft trajectory generated by the system identification model can be trusted for small time intervals. L term provides an upper bound on the growth rate of the error term. A large L term represents a system with high sensitivity to perturbations. This term is assumed to be sufficiently low for stabilized fixed-wing aircraft in the final approach. Based on this assumption there are multiple studies that used neural network models to generate a dynamic model of nonlinear systems including aircraft [67, 129, 95, 130, 47].

Just like the prediction model the simulation model also has model and data structural parameters that impact the model's accuracy. So the ideal configuration that can accurately estimate the flight states while being robust to external noise must be identified. Research

Research Question 2.2

How to identify the proper model and data structure to consistently identify the configuration that can maximize the simulation model accuracy.

questions 2.1 and 2.2 can be combined to solve gap 3

Based on the observations made so far, a similar hypothesis as hypothesis 1.1 can be formulated for research sub-questions 1.1 and 1.2.

Hypothesis 2.1

If both event and nonrisk flights are used for training the system model and if the simulation accuracy is evaluated in terms of model and data structures then flight simulation model can be generated from the FOQA data.

To answer and test this hypothesis, the design space of the simulation model has to be evaluated. In this sense, this hypothesis is related to hypothesis 1.1. The difference

between hypotheses 1.1 and 2.1 is that the simulation model only needs to predict the flight states one step into the future. Also to simulate a flight, the control trajectory of the flight needs to be known. So for the simulation model the state and control features are both used to train the model. The input window size is short since the input window size only needs to be long enough to represent the low-frequency dynamics. and output range is also short because the flight state estimation only needs to be one step far. For hypothesis 2.1 the window size is varied to ensure that the input data capture the full dynamics behavior. The same hyperparameter optimization algorithm used to test hypothesis 1.1 will be used to find the optimal model configuration for the simulation model.

The experiment to test hypothesis 2.1 needs to generate a neural network model that can use FOQA data to map the aircraft controls and external factors at time step (t) to the aircraft states in the next time step (t+1). This is illustrated in Figure 3.5. To test hypothesis

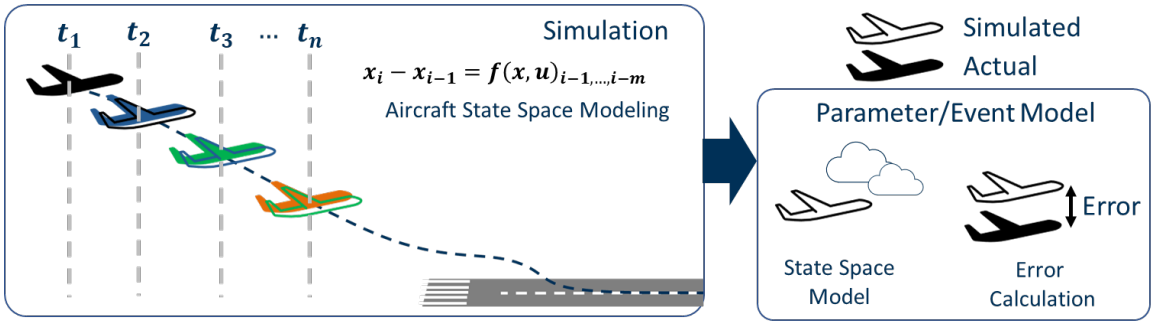

Figure 3.5: Experimental setup for flight simulation through aircraft state space model generation

2.1 the size of the input window (m) needs to be optimized. During the data pre-processing stage, the input data sequence and the prediction data will be stacked and combined for all of the flights. When the simulation is run the final approach data for each flight is provided to simulate each flight. The model will continue to simulate as far as the length of the control trajectory and external factors are defined. This is shown in Figure 3.5 where the simulation ran until t+n time step. An illustration of this process is shown in Figure 3.6 The accuracy of the simulation will be determined based on the RMSE between simulated

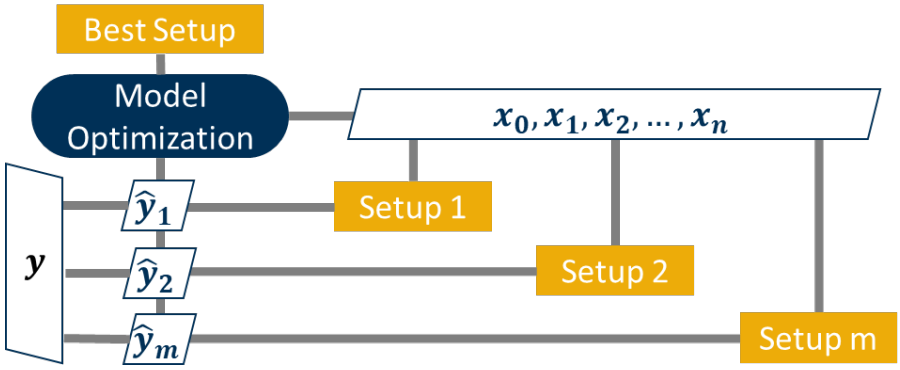

Figure 3.6: Overview of the model design and optimization method

states and actual states for n time steps. The study by Tong et al. [45] used a similar process to predict vertical acceleration. As shown in Table 3.1, the RMSE of the predicted vertical acceleration is very low showing the potential for the neural network to generate the aircraft state space model. The title of the experiment for hypothesis 2.1 is shown below.

Experiment 2.1

Aircraft simulation model design and optimization

This experiment is also set up to optimize the model in terms of accuracy. The study by Kumar et al. [98] showed that neural networks with feedback improved the prediction of nonlinear dynamics. Hence, various RNN are considered for modeling the aircraft state space including fully connected RNN, partially connected RNN, GRU, and LSTM. Although a single-layer neural network model may capture the dynamics, additional layers may be needed to fully capture the coupling and low-frequency dynamics. Two layers of neural network model may be used as encode decoder format or more layers of neural network models may be used to generate a deep learning model. All of these models need to be tested to find the optimal neural network structure for modeling the aircraft state space.

Just like the prediction model, the simulation model will also require sensitivity analysis. This is important for the simulation model because the error is compounded throughout the simulation. So the large errors will cause the simulation to fail. The sensitivity analysis can help validate the model performance at regions of control space where the model is

most likely to fail. This is especially true when using a neural network model where the model behavior is unknown without testing it. The first sensitivity analysis is based on the flare maneuver. Testing the simulation model with flare maneuver can help identify the model performance at extremes of the control space. The second sensitivity analysis is based on wind. Again, the wind has two components: speed and direction. To the simulation model, both of these wind components are just like the aircraft control. So testing the simulation model at various wind conditions can help identify the model performance at different extremes of the control space. Based on these observations a research sub-question can be formulated. Research question 2.3 will be used to answer gap 4.

Research Question 2.3

How to evaluate the sensitivity of the simulation model to external factors to ensure that the mode performance does not degrade significantly at certain regions of the control space that lacks training data

To answer this sub-question, the design space exploration will consider the accuracy of the simulation for the full final approach and flare separately. To do so, flare initiation will be defined based on altitude, runway threshold, velocity, pitch, and control surface deflection. The simulation accuracy before and after the flare will be measured. The simulation accuracy at different levels of longitudinal and lateral wind speed will also be measured. And to check the correlation between the RMSE and data count, a data frequency map in terms of flare and wind parameters will be generated The hypothesis and the experiment titles to answer the research sub-question 2.3 is as follows:

Hypothesis 2.2 and 3

If the definition of flare and adverse wind can be made from FOQA data and if the data count can be identified at various external conditions then the difference in the simulation RMSE will show the sensitivity of the simulation model to flare and adverse wind.

Flare and wind are the main external factors with large contributions to the aircraft dynamics during the final approach. However, the model accuracy in the extremes of the other

Experiment 2.2 and 3

Simulation model sensitivity analysis

control space features must also be identified. There is a lack of study on the robustness of the simulation model at different points in the control space. A downside of neural network models is that they are not suited for extrapolation. This gap leads to the following research sub-question. Research question 2.4 will be used to answer gap 5.

Research Question 2.4

How to identify the control space of the simulation model so that the control configurations that the model can be trusted.

Observations showed that there is a lack of flight data for extreme maneuvers during the final approach. This means the model accuracy and data frequency at different points in the control space must be compared. The ideal outcome is that the model accuracy is independent of the control space. The hypothesis and experiments to answer this research sub-question are as follows:

Hypothesis 2.4

If the data count can be identified at the entire control space
then the difference in the simulation RMSE will show the sensitivity
of the simulation model to different control configurations.

Experiment 2.4

Simulation control space analysis

Separate research questions are not formulated for the training building block. However, the presence of gaps 7, 8, and 9 is acknowledged. Also, developing the prediction and the simulation models is not enough to show that they can increase awareness of overrun precursors. So this objective will be achieved by taking three different steps. The first step is introducing how the prediction and simulation model outputs can be mapped to the

proximity of overrun. The second step is demonstrating how accurately the two models can predict overrun precursors. The last step is demonstrating how the two models can be used to predict overrun precursor in flight and learn from simulated flights post-flight. The computation time for both prediction and simulation will be discussed to show the feasibility of the designed model for providing quick feedback to the pilots.

With this, a complete design and analysis framework of the final approach simulation will be generated. An illustration of this framework, for research question 2, is shown in Figure 3.7.

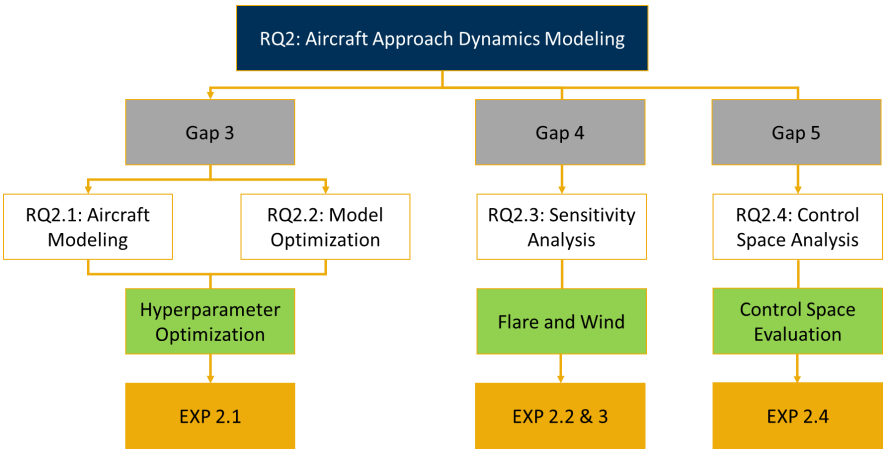

Figure 3.7: Updated diagram of research question 2.

CHAPTER 4
PROPOSED METHODOLOGY

Studies so far showed that a prediction model can potentially increase pilots' awareness of runway overruns. The prediction model uses the final approach data to show the predicted aircraft states near touchdown. These states are selected from the definitions of overrun precursors. These precursors are threshold exceedance events that relatively frequently occurred before a runway overrun. Each precursor may be given different threshold levels to help visualize the severity of the potential risk. Pilots may use this indicator as feedback, during training, to learn the sequence of control that minimize the severity of each precursor. During training, a simulation model may also be used to evaluate the impact of different flight behaviors on landing performance. To make this feasible the aircraft dynamic models need to accurately simulate flight behavior from the influence of external factors such as wind. The prediction and simulation models combined can provide a platform for pilots to test different landing scenarios and increase awareness of risks associated with runway overrun. This chapter will discuss an overview of the development and analysis processes of the two models.

4.1 Increase Awareness to Touchdown Condition with Prediction Model

The goal of the first research question is to alert overrun precursors based on the final approach data. There are two key factors for answering this question:

1. whether the pilot behaviors during the final approach are related to the pilot behaviors near touchdown

2. whether the prediction model can model the pilot behavior to accurately predict aircraft states near touchdown

The first factor is important for developing the prediction model because if there is no relationship between pilot behavior during the final approach and near touchdown, then the prediction model is nothing but a polynomial regression model of aircraft states in terms of altitude. The hypothesis is if there is a relationship then aircraft states can be predicted near touchdown. Based on this hypothesis, the goal is to identify a framework that can design the prediction model so that the prediction is made early and accurately for pilots to respond to any adverse conditions.

Five research sub-questions are addressed based on the gaps identified in the current methods of predicting aircraft states near touchdown. The first step of the process is addressed and validated from an earlier study [117]. The result showed the potential for the neural network models to accurately predict the aircraft speed near touchdown. The second step is focused on sub-question 1.2, how early can this prediction be made so the pilots can utilize the prediction early on. One feature that directly relates to this is the input window size. Understanding how much data above the touchdown has relevant information for determining the aircraft states near the touchdown is the key to this question. By addressing this question, sub-questions 1.2 and 1.3 can also be answered. This is because the prediction accuracy and range may also change with the frequency of the input data provided. The hypothesis is that depending on the smoothness of the input data, the prediction model may treat changes in the aircraft states as noise. The more that the machine learning model treats changes in input data as noise the more the model will perform like a simple polynomial regression model. An important factor to remember is that no information is provided regarding the predicted aircraft states in the input data. This means the prediction model is creating a complex regression model of the aircraft states at each predicted altitude. So the prediction models may be generalized models of aircraft states. This means the prediction model will not be able to predict the small changes in the predicted states instead it will predict the overall trend. So if the model is unable to find a correlation between input data and the predicted states, the resulting model will be equivalent to a polynomial regression

model of the aircraft states in terms of altitude. Increasing the frequency of the input data may resolve this issue but the maximum benefit of increasing the frequency needs to be determined. So a hyperparameter optimization algorithm will be used to find the ideal data and model configuration with the highest prediction accuracy and range.

To determine if the developed prediction model is behaving like a simple polynomial regression model, a simple polynomial regression model of aircraft states in terms of altitude will be generated. The prediction accuracy of this regression model to the neural network model will be compared to ensure the two factors for answering the first research question. It is observed that if the prediction model has the characteristic of a regression model then the prediction accuracy will increase by clustering similar flights. So a clustering step will be added to the data pre-processing step and generate a separate prediction model for each cluster and observe if the prediction accuracy improves. Here the final approach trajectories will be clustered based on their flight path, velocity, and acceleration. The result will provide additional feedback on the sensitivity of the prediction model accuracy to the training data.

Researchers have recently considered data-driven models for predicting flight trajectories by learning from the historical flight trajectories with machine learning and data mining algorithms [131, 132]. Zhang and Mahadevan [133] have used bayesian neural networks for one-step-ahead prediction on the deviation along latitude and longitude between the actual flight trajectory and target flight trajectory. This application focuses on en-route flight trajectory prediction and does not include higher fidelity trajectory data like FOQA. However, it does provide the value of predicting trajectories by assessing a safety measure that calculates the potential loss of separation. Che et al. [134] have used multiple deep-learning models to assess system conditions and classify faults in aircraft systems. This application focuses on individual aircraft systems rather than the trajectory as a whole but provides insights into the utility of using multiple deep-learning models. Ayhan and Samet [132] used Hidden Markov Model to learn from historical flights and predict flight trajectory while

taking environmental uncertainties into account. This study generated flight trajectories on a discretized map that represents a weather observation at each discretized space. Shi et al. [130] used LSTM neural network to predict the flight trajectory. Sliding windows are applied to the LSTM model to maintain a high level of accuracy in the dynamics in the adjacent states. Wu et al. [131] developed recurrent neural network models that can model trajectories constrained by the topological structure of road networks. Among all these applications, there is an overarching theme that uses neural network models to generate prediction models. What are some of the neural network models used specifically to predict flight states? The study by Kang et al.[42] presented this by developing a sequential prediction model of aircraft speed from 4 to 14 seconds after crossing 50 ft height above touchdown based on the data 4 seconds before and after crossing the same altitude. The authors tested machine learning models including LSTM, decision tree, linear regression, gradient boosting decision tree, random forest, and support vector machine models. The results showed that LSTM based prediction model has the lowest error. Comparison of the prediction model accuracy of past models in Table 3.1 further shows the potential for the gated neural network model, such as LSTM, to most accurately predict the aircraft speed compared to other machine learning models. Based on the prediction models developed in past studies, aircraft states prediction models based on LSTM, GRU, and CNN are considered.

LSTM is a type of recurrent neural network that tries to resolve the issue of back-propagating errors blowing up or vanishing. Information for LSTM models flow through four steps as shown in Figure Figure 4.1:

1. Forget gate, which changes the state from t-1 to t

$$f_t = \sigma(W_f(h_{t-1}, x_t) + b_f)$$

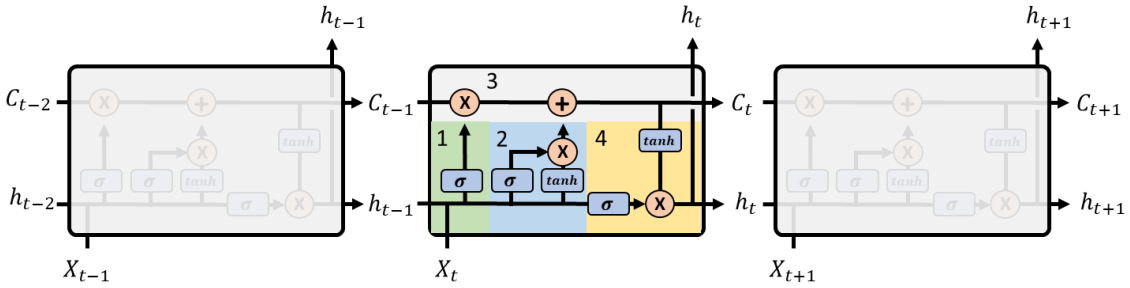

Figure 4.1: Example LSTM neural network cell

2. Input gate and updated values, which decide which states are added by how much

$$i_t = \sigma(W_i(h_{t-1}, x_t) + b_i)$$
$$\tilde{C}_t = \tanh\left(W_C(h_{t-1}, x_t) + b_C\right)$$

3. State update, based on f_t, C_{t-1}, i_t, and $\tilde{C}_t$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

4. Output gate, where the output to the next cell and layer is determined

$$O_t = \sigma(W_O(h_{t-1}, x_t) + b_O)$$
$$h_t = O_t * \tanh\left(C_t\right)$$

Here the f is the forget gate, i is the input gate, O is the output gate, C is the state, h is the hidden layer state, x is the input, W is the weights, b is the bias, and σ and $\tanh$ are the activation functions. For further details of LSTM models, the readers are referred to Hochreiter et al. [135].

GRU is another gated recurrent neural network model with a similar goal as LSTM. Information for GRU models flow through four steps as shown in Figure Figure 4.2

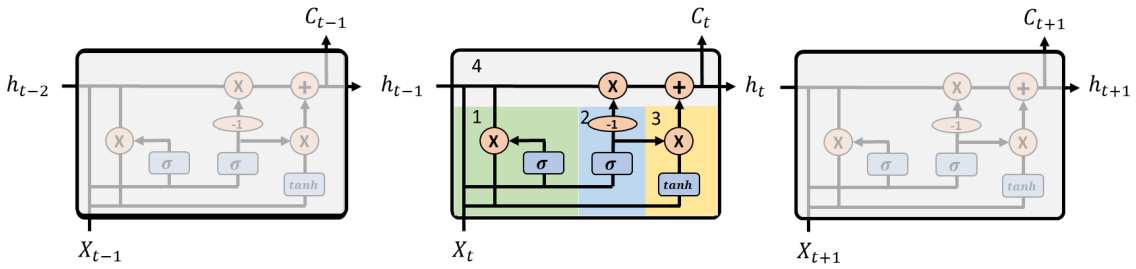

Figure 4.2: Example GRU neural network cell

1. Update gate determines the past information that is passed

$$z = \sigma(W_z * x_t + U_z * h_{t-1} + b_z)$$

2. Reset gate determines the past information to forget

$$r = \sigma(W_r * x_t + U_r * h_{t-1} + b_r)$$

3. Update, based on z, h_{t-1}, and $\tilde{h}_t$

$$h_t = z * h_{t-1} + (1 - z) * \tilde{h}_t$$
$$\tilde{h}_t = \tanh\left(W_h * x_t + U(r \odot h_{t-1})\right)$$

Here the z is the update gate, r is the Reset gate, h is the hidden layer state, C is the state, x is the input, W are the weights for the input, U are the weights for the hidden states, b are the biases, σ, and $\tanh$ are the activation functions. For further details of GRU models, the readers are referred to Cho et al. [103]. GRU and LSTM have both structural and functional differences. Structurally GRU have two gates while LSTM has three. Functionally, the study by Yin et al. [136] show that GRU is capable of learning a model faster, while LSTM can learn longer sequences.

CNN model extracts spatial relationships in the sequence of data and maps them to

the internal features, flight parameters in this case. The information for the CNN model flows through a set of filters, also known as kernels, and is convoluted across the input volume. The filters are activated when a specific feature is triggered at the spatial input. Some parameters that affect the performance of the CNN model are kernel size and stride. Kernel size determines the size of the spatial area that is convoluted simultaneously. Stride determines the spatial step size so that the filter passes the input volume. An example figure for CNN models is shown in Figure Figure 4.3. For further details of CNN, the readers are referred to Indolia et al. [137].

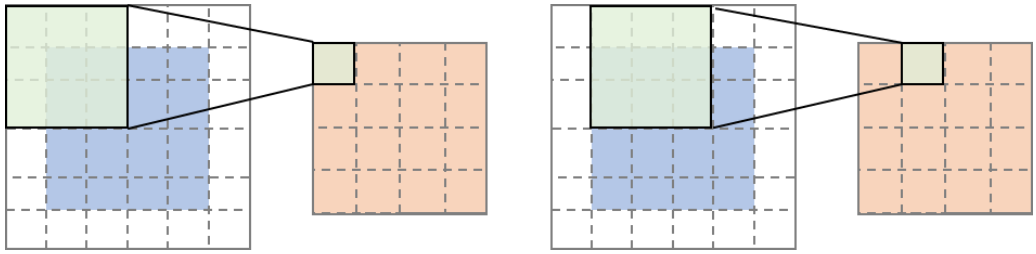

Figure 4.3: Example CNN cell

1. Convolution operation

$$C_q^l = (\Sigma_{p=1}^{n} S_p^{l-1} * k_{p,q}^l + b_q^l)$$

2. Pooling

$$S_q^l(i,j) = \frac{1}{4}\Sigma_{u=0}^{z}\Sigma_{v=0}^{z} C_q^l(2i-u, 2j-v)$$

Here p and q are the feature map (used to convert an area of the input data into a single value) indices. i and j are the image indices. u and v are the pooling window dimension. l is the layer, b is the bias, z is the pool window size, S is the image to be convoluted or the pooled image, and C is the convoluted image. With a CNN model, input data is divided into smaller segments and each segment is converted to a single value with the convolution

operation. The converted values are sampled or pooled with the most representative value. This process repeats for multiple layers until the final output data, which most represents the features of interest, is generated.

These models are configured in various combinations to find the best model for predicting the risk of overrun for a given flight. How will these different combinations of models be compared? Safety is a concept that may be difficult to quantify because it is associated with the absence of something rather than its presence [138]. However, since this work is focused on predicting precursors by predicting flight states, the accuracy of flight states will be used to compare the models. The average error of the predicted and actual states may be calculated from all the test flights to find the best model combination. The RMSE and relative error metrics are calculated for each flight. The RMSE will measure the absolute error of the prediction in the unit of the predicted parameter. This metric will provide an estimate of how accurate the model is compared to peers and provide an error distribution range. This metric will also tell if the prediction model is learning from the flight behavior or is generating a simple regression model of the predicted parameter. If the prediction model is a simple prediction model, the RMSE is expected to be constant or reduce at each altitude. This is because a regression model is a curve that connects the mean or the median of the prediction parameter distribution at each target altitude. The relative error will measure the error residual normalized by the recorded min-max range of the prediction parameter at each target altitude. This metric will provide an estimate of how accurate is the model relative to the prediction data distribution at each target altitude. This metric will be used to compare the prediction model performance at various target altitudes. Now that the accuracy metric is defined, how much of the final approach should be predicted? A study by Campbell et al. [26] 300 ft HAT as the lowest altitude at which the pilots could respond to deviations from the ideal approach trajectory. Based on these observations and aforementioned studies, longitudinal dynamic parameters are predicted at various target altitudes between 300 ft and 0 ft HAT. So the configurations are compared

by finding the predicted and actual states below 300 ft HAT

In summary, the prediction model investigated in this study should address some limitations of the existing supervised learning approaches, such as providing predictions for multiple features. The model needs to accurately predict aircraft states with enough time for pilots to respond and take preventative actions. The prediction model will be generated with various combinations of neural network models. To address the research sub-question 1.4 the significant parameters of the predicted states will be identified from the final model. The significant parameters will be identified based on the permutation of importance. Then to address the research sub-question 1.5 the model sensitivity to flare will be analyzed. This is because, throughout the majority of the final approach, changes in these parameters are relatively small until the pilot initiates flare. The exact altitude at which the flare is initiated depends on the technique used during landing. The flare technique used may have significant implications on the amount of speed bled during the flare and the final touchdown location. So the variation in the techniques makes flare an important process to consider when analyzing landing performance. The wind is another factor to consider when evaluating model sensitivity to external factors. When the aircraft is in close proximity to the ground, the change in airflow results in erroneous measurements in sensor readings and changes in aerodynamics. Also by nature, an aircraft rely on airspeed to generate lift. So when an unexpected change in the wind speeds is observed, large maneuvers will be made to account for these changes and deviate the flight from a typical landing. The wind effect coupled with flare makes the aircraft's behavior just before touchdown a complex system to model. Based on a survey of prior work, it is unclear how well a supervised machine-learning algorithm can deal with the flare and wind. To answer this question, flare and adverse wind must be defined from the FOQA data. Then the model accuracy before and after flare or with and without adverse wind will be compared.

4.2 Improve Flight Control with Dynamics Model

While the goal of the prediction model is to predict the aircraft states near touchdown, the goal of the simulation model is to sequentially predict the aircraft states in the near future. This type of model may be trained with supervised machine learning models that map the input data to the truth label one step into the future. The input data is the initial aircraft states and the full control trajectories. The truth data is the predicted aircraft state trajectories. With a well-trained model, the system control inputs may be modified to generate alternate paths. These alternate paths may be used to analyze the impact of changes in aircraft behavior on the exposure of overrun precursors. For improving awareness of overrun precursors, alternate paths may be used to train the control sequence to return to a safe path.

A dynamic model may be generated with artificial neural networks (ANN) based on the universal approximation theorem [127]. The theorem states that neural network models with nonconstant, bounded, and monotonically increasing continuous activation functions can approximate any system. Based on this knowledge, aircraft systems have been modeled with neural networks in numerous studies. In general, aircraft systems are modeled based on the aerodynamic coefficients or black-box approach. The aerodynamic coefficient approach generates nonlinear coefficients with neural network models for generic aircraft equations of motion [139, 140]. The benefit of this method is that it works for any generic system that follows the same equations of motion. This also means that the system model strictly follows the equations of motion used, which may not capture the full dynamics of the aircraft. External factors such as wind or ground effect need to be factored into the equations of motion for the dynamic model to capture these factors. The black-box approach assumes the model as a black box that estimates future states based on current states and controls [141, 142, 143, 144]. Systems identified with the black-box approach require the input data to be from the same aircraft type to ensure that the model stays accurate. So the black-box model is tailored to the airframe and the flight conditions that

the data represents. The downside of a black-box approach is that identifying the system for the full dynamics requires flight data that includes various maneuvers to ensure that the data represent the aircraft well. However, as long as sufficient data is obtained, an accurate system model of an aircraft may be generated. With this model, control trajectories that trigger threshold exceedance precursors based on aircraft states may be analyzed. The risk of a flight may be measured based on the magnitude and frequency of threshold exceedance to provide feedback on the flight performance to the pilots.

In terms of the architecture, ANN based dynamic models may be feed-forward or recurrent. Some examples of a feed-forward neural network include a wavelet neural network, radial basis function network, and rectified-linear unit networks. The aforementioned models such as LSTM and GRU are recurrent neural networks. Multiple studies show the potential for recurrent neural networks to model nonlinear dynamic behaviors [145, 67, 99]. In exchange, recurrent neural networks are more costly to generate. ANN based dynamic models may also be offline or online networks depending on whether the network parameters are fixed or adjustable in real-time. Offline models are smoother in their approximations, making them a better choice as a reference model, and not constrained in time for training while the online models are more adaptive to changes in the inputs [146]. Since the goal of this study is to develop an aircraft final approach model that most accurately approximates the behavior of an aircraft, offline recurrent architecture is selected. This means a large number of final approach data that well represent the full dynamics of the aircraft is needed to train the model. The FOQA data is a well fit for this purpose because it includes the real-time recording of aircraft states, controls, and wind. The concerns are how much FOQA data is needed to model the full aircraft, what is the scope of the FOQA data required for modeling aircraft behavior during the final approach, and how the control space change with the training data.

To address these concerns multiple time series system identification structures are surveyed, one of which is the NNARX. A sample illustration of the NNARX model is shown

in Figure Figure 4.4. In the illustration, y is the states, u is the controls, w is the model

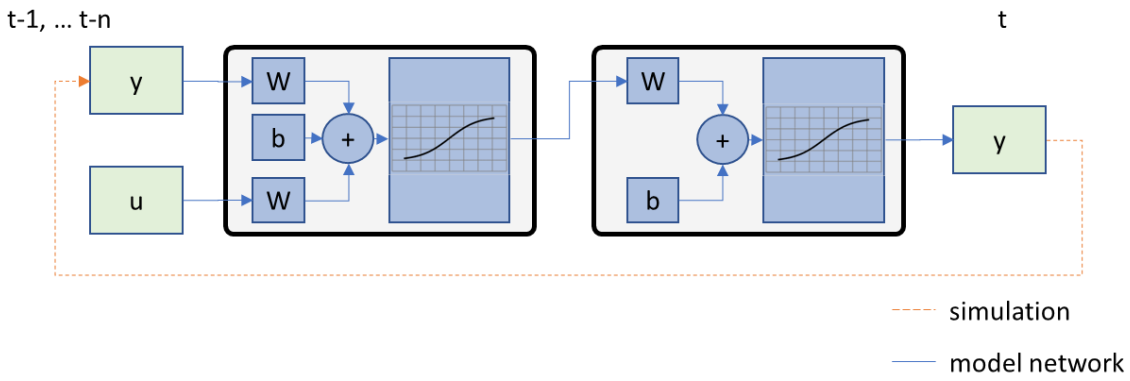

Figure 4.4: Example NNARX model

weights, b is the model biases, the blue arrow is the model network connections, and the orange dashed arrow is the feedback loop that simulates a flight. NNARX maps the past states and controls to the next state with nonlinear regression functions. As explained in subsubsection 2.2.2, multiple variants of NNARX exist based on how the feedback loop is connected in the network. Variants of NNARX also exist by only using the states or the controls as inputs to the model. The nonlinear regression functions may also be replaced by gated neural network models so that the network memory can reduce the need to identify the lag in the states and the controls. Multiple studies already showed the potential for a neural network based system identification model for representing the dynamics of an aerial vehicle [143, 144]. In fact, the study by Roudbari and Saghafi[96] showed that NNARX based model can accurately model the linear and angular acceleration of an aircraft. However, this study utilized test flight data recorded at 10 Hz frequency with specific maneuvers. The FOQA data collected in regular flights have errors and noise issues that limit the data available for training the neural network model. In fact, the flight data for large aircraft motions are expected to be scarce depending on the pilot, aircraft, and runway combination. When addressing the risk of certain flight trajectories based on the threshold exceedance of flight parameters, it is important that the model accurately approximates the

aircraft states. So the control space that the simulation model accurately represent needs to be evaluated. This includes the simulation model's sensitivity to external factors such as wind. To do so the model will be first trained with an open loop structure. So the true controls and states at t-1 will be mapped to the true states at t. The trained model will be tested with a close loop structure where only the initial states and the full control trajectories are provided. The approximated aircraft states will be used as the input for the next state approximation until the end of the control trajectory.

A simpler architecture considered for modeling the aircraft system dynamics is the RNN. A sample illustration of the RNN model is shown in Figure 4.5. The key difference

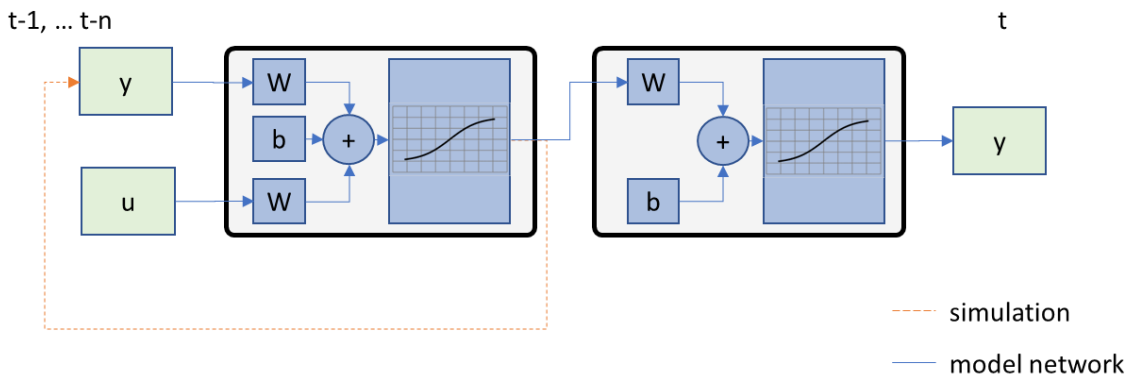

Figure 4.5: Example RNN model

between RNN and NNARX is the location of the feedback loop. While the NNARX model feedback around both layers of the model, RNN model feedback only loops around the top layer. So in essence the RNN is a single-layer model with a bias and a multiplier added to the output of each time step. The purpose of the second output layer may be as simple as unit conversion or it could add a higher-order term, in the aircraft equations of motion, that is time-independent in nature. For the first example, the normalized data could be used to as input and the regular data as output to train the RNN. Then the aircraft system could be modeled with the recurrent layer and the second, nonrecurrent, layer could be responsible for converting the normalized output data to the regular output data. For the

second example, if the RNN was modeling the drag polar of an aircraft, the recurrent layer would model the induced drag while the nonrecurrent layer would model the parasitic drag. The benefit of RNN is that it is a simpler model, but whether this model is complex enough to represent the dynamics of an aircraft has to be addressed.

The aforementioned GNN models would look like Figure 4.6. Essentially the simple

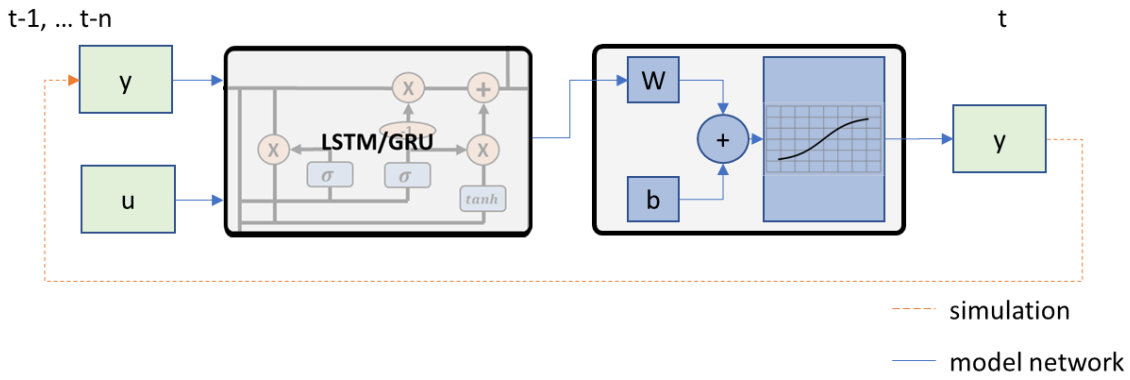

Figure 4.6: Example GNN model

top layer of the NNARX model would be replaced by a complex gated layer. The benefit of GNN models is that the gates can reduce the issues with exploding or diminishing feedback error. The gated layer is also recurrent, so it will better represent the nonlinear behaviors of the aircraft. However, GNN models are much more complex and require a large number of data to train. So the pros and cons of using GNN over NNARX has to be addressed.

4.3 Maximize the Model Accuracy with Hyperparameter Optimization Algorithm

The overarching theme behind this thesis is modeling aircraft behavior with neural network models. Past studies using neural network models to predict or simulate aircraft states used multiple layered models due to the complexity of aircraft behavior. The simplest neural network model contains a dense layer and its model parameters are the number of hidden units, activation function, and bias. LSTM and GRU model parameters include: recurrent activation function, dropout, recurrent dropout, and unit forget bias on top of the model

parameters for the dense model. The CNN model parameters include filters, kernel size, strides, padding, activation, and bias. As the number of layers increases to approximate more complex behaviors, there is an explosion of the model parameter combinations to test. On top of this, the design space of the neural network models is unknown because they are not derived from the physics of the aircraft. This means the neural network design space may be multi-modal making it difficult to find the global optimum configuration with the highest prediction and state approximation accuracy. Not to mention that machine learning models are sensitive to how the training data is pre-processed. Hence it is important to select the right optimization algorithm to explore the neural network model design space to search for the global optimum.

The goal of an optimization algorithm is to find the minimum (or maximum) in a design space. The dimensions of the design space are the parameters of the objective function. A point in the design space is considered as a local optimum if $f(x*) < f(x)$ where $x*$ is the local optimal configuration and x are its neighbors. Conditions for optimum are supported by two conditions: first and second-order conditions. The first order necessary condition states that $x*$ is locally optimal if $\nabla f(x*) = 0$ where ∇ represents the gradient. The second order necessary condition states that $x*$ is locally minimal if $\nabla f(x*) = 0$ and $H(x*)$ is positive semi-definite. Here the H represents the hessian of the point $x*$. The sufficient variant of the second order condition requires the $H(x*)$ to be positive definite. For a large design space that is multi-modal, global optimum needs to be identified. Global optimum is when $f(x*) \leq f(x)$ for all ranges of x.

Optimization algorithms are classified based on their method of searching for the optimum.

1. Direct search methods – search for the optimum without gradient information (ex. grid search)

2. Line search methods – search for the optimum along successive lines in a design space (ex. steepest descent)

3. Trust region methods – search for the optimum by approximating the local design space (ex. quasi newton direction)

Direct search methods search for the design space randomly or in a pattern. These methods may not find $x*$ but are better at exploring the design space. For the line search methods, the search line is defined based on the gradient and the hessian at x. These methods will more likely converge to a local optimum, but the solution is sensitive to the initial x. The trust region methods are also likely to converge to a local optimum, but a lot of information about the design space is needed to create the trust region. Also, both line search and trust region methods search for the gradient of the design space and this may be costly, especially with increasing dimensions.

In terms of neural network model optimization metaheuristic optimization algorithm that utilize direct search is considered. This way the design space can be quickly explored. Examples of metaheuristic optimization algorithms include Simulated Annealing (SA), GA, and particle swarm algorithms. SA models the behavior of heat dissipation described by the following equation:

$$P(\Delta f) = e^{\frac{\Delta f}{kt}} \tag{4.1}$$

where k is the Boltzmann constant, t is temperature, and f is the objective function. Initially, when the temperature is high the algorithm accepts small improvements in the objective function. As the temperature is reduced, the algorithm searches for a steeper path to the optimum. SA is an example of a stochastic optimization algorithm. Another example of a stochastic optimization algorithm is GA.

GA is a population-based stochastic algorithm. It models the behavior of how chromosomes share information and generate new cells during meiosis. GA follows a 5 step process:

1. sampling – sample population randomly

2. evaluation – evaluate each sample

3. selection – select the samples (parents) that will move on to the next generation

4. crossover – generate new samples (children) by interchanging information between parents

5. mutation – random change of information within each sample.

The benefit of stochastic optimization is that the algorithm explores a large design space while converging to an optimum. The limitation to SA is that repeated annealing may be costly. The limitation to GA is that the solution may converge to the local optimum depending on the initial sample.

Particle swarm optimization is a population-based stochastic algorithm. This algorithm model the behavior of a swarm of animals such as bees or a flock of birds. Initially, a random set of samples are used to search the design space. The cost of the objective function for each sample is compared and the search direction is found based on the samples, with the lowest objective function results.

$$v^{t+1} = av^t + bU_1^t(\rho b^t - x^t) + cU_2^t(gb^t - x^t) \tag{4.2}$$

Here a, b, and c are coefficients, $(\rho b^t - x^t)$ is the search direction based on the sample x^t, and $(gb^t - x^t)$ is the search direction based on other samples. The particle swarm optimization is easy to fall into the local optimum and may have local convergence. This is because the initial set of samples converges to the local optimum of the best sample found in this first iteration.

Bayesian Optimization (BO) is another metaheuristic optimization algorithm and it relies on the probability of improving the model to optimize the design space. The equation

that describes the BO algorithm is Equation 4.3.

$$\hat{\eta} = \arg\max_{\eta} P(\eta | f(x_{1:n})) \tag{4.3}$$

As noted in Equation 4.3, the BO algorithm evaluates a configuration (η) where the variation of the configuration to be on the base function selected from past observations ($f(x_{1:n})$) is maximum. The algorithm continues to search for new configurations (evaluation points) until the expected improvement is small.

An important characteristic to consider when deciding the right metaheuristic optimization algorithm is that most neural network model parameters are either discrete or can be discretized. So the algorithm needs to be good at solving combinatorial problems. Past studies show GA being used frequently for training neural network models [147, 148, 149, 150, 151, 152]. A study compared GA with two other metaheuristic optimizers and found benefits in using GA [153, 154]. However, GA is known to converge to a local optimum with a significant correlation to the initial condition. This study will further explore GA variants to increase the design space exploration and use the developed algorithm to optimize neural network models for aircraft state prediction and simulation.

CHAPTER 5

HYPERPARAMETER OPTIMIZATION ALGORITHM DEVELOPMENT

GA is one form of evolutionary-based optimization algorithms [155]. GA has shown success in optimizing and designing multiple neural networks in the past. The five-step process of the GA allows it to effectively test the combinations that converge to a local optimum. But the nature of the GA makes it difficult to increase the explored design space. Because the mutation rate is typically set as a low value so that the algorithm can converge to a solution. Also, the randomness of the mutation step does not guarantee that the mutated sample can survive. So the solution is sensitive to the initial samples selected in the first step. Nevertheless, GA can explore a large design space, relatively fast with parallel computing, even for an unknown design space. This makes the algorithm a great selection for designing neural network models because each model is unique to the training data so each time the design space has to be re-evaluated. An illustration of the conventional GA is shown in Figure 5.1.

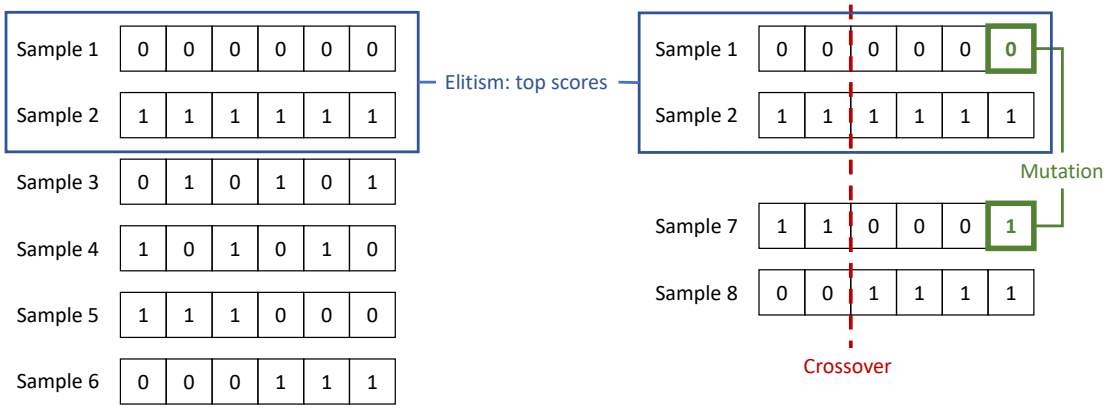

Figure 5.1: Illustration of the GA processes

QIGA is a variant of the genetic algorithm inspired by quantum mechanics to explore the design space. Multiple studies show the improvements in the optimum search process

when using QIGA over GA [156, 157, 158, 159, 160]. QIGA achieves this by using qubits that replace a value in the binary code for the GA. Unlike in GA where a binary value must represent a 1 or a 0, a qubit represents a distribution of 1s and 0s and this distribution is defined by Equation 5.1.

$$\phi_i = \alpha_i|0) + \beta_i|1) \tag{5.1a}$$

$$where\ 1 = |\alpha|^2 + |\beta|^2 \tag{5.1b}$$

So a quantum chromosome is represented by a list of αs and βs. And each chromosome represents a combination of samples for a feature of the design space. Notice Equation 5.1b represents the function of a circle which means α and β can represent $\sin\theta$ and $\cos\theta$. So a feature that distinguishes QIGA from GA is that after each generation, the QIGA algorithm will update θ for each qubit to converge to the optimum. This way the mean of the new set of samples generated will shift towards the identified optimum in the quantum realm. However, the randomness of the samples generated makes it possible to increase the design space explored. An illustration of how qubit operates to achieve this is shown in Figure 5.2.

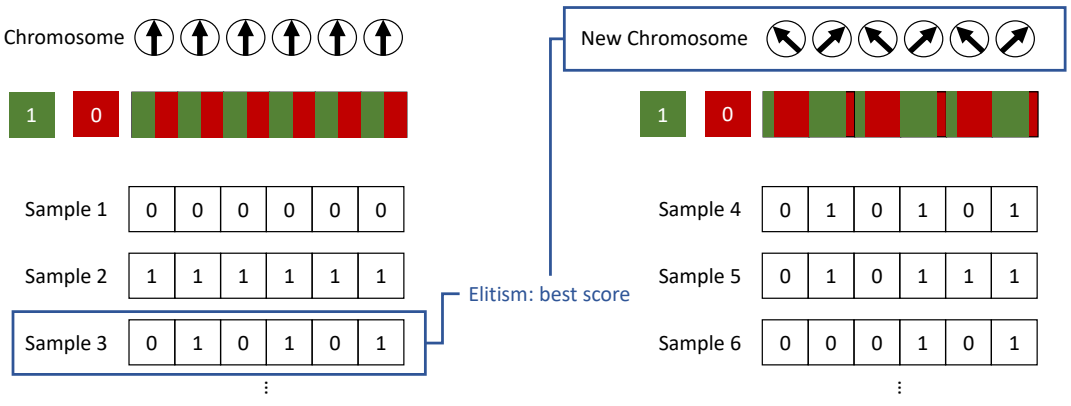

Figure 5.2: Illustration of the QIGA processes

The potential for QIGA to optimize a neural network structure can be evaluated based on a known design space, or a test function. A test function is a complex multimodal

function with a single peak. The multimodal functions represent local optimums and the single peak represents the global optimum. With the test function, the ability of the optimization algorithm to identify and converge to the global optimum may be observed based on the number of function evaluations. A list of test functions is listed by Surjanovic and Bingham [161]. For this study, Ackley and Schwefel functions are selected to test the optimization algorithm generated for this study. These two functions are selected because they are multimodal with known global optimums. However, the Ackley function has a distinct optimum while the global optimum for the Schwefel function is not too different from the other optimum. Equation 5.2 show the two test functions.

$$f(x) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos cx_i\right) + a + \exp(1) \tag{5.2a}$$

$$f(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin\sqrt{\|x_i\|} \tag{5.2b}$$

The illustrations of the test functions can be found in Figure 5.3.

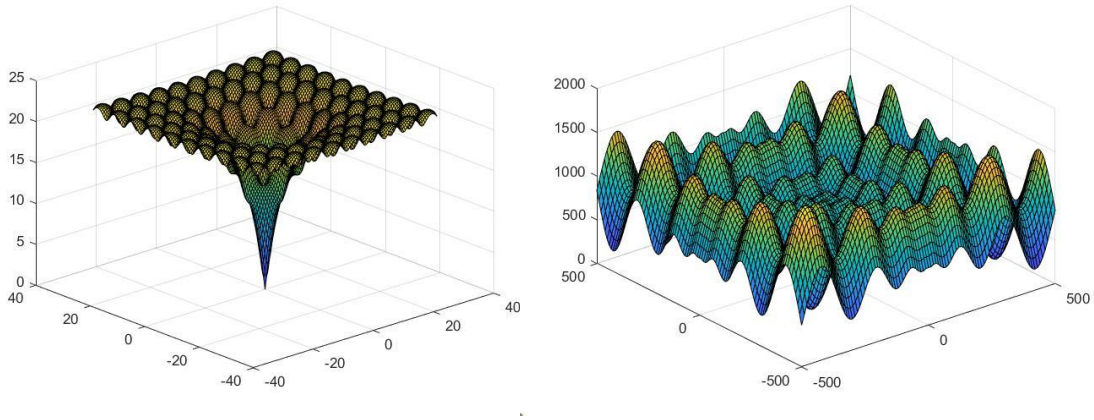

Figure 5.3: Illustration of the Ackley (left) and Schwefel (right) test functions

5.1 Development of Optimization Algorithm

As introduced, the benefit of QIGA is the increase in design space exploration while simultaneously converging the new samples to the optimum. This is made possible by varying the θ so that the updated qubit converges to the optimal. The corresponding pseudo-code is shown in Algorithm 1. Here Q is the string of qubits that represent a single chromosome.

Algorithm 1 QIGA algorithm with caption

```
t ← 0
initialize Q(t)
generate P(t) from Q(t)
evaluate P(t) and find b
update Q(t) by dθ to converge to b
while t ≤ max generation do
    t ← t + 1
    generate P(t) from Q(t)
    evaluate P(t) and find b
    update Q(t) by dθ to converge to b
end while
```

Each qubit is used as a threshold for uniform distribution samples to select between zero and one. A distribution of values from each cubit is combined to form a population, denoted as P. Each sample is evaluated and the best sample, b, is identified. Then the qubit is updated by changing θ to converge to b. Depending on the problem solved, the magnitude of θ change (dθ) and the direction of change may differ. Because of this, dθ is a critical factor to the convergence rate and the optimum identified. If dθ is high then the algorithm will converge to b faster but the chance of b being the true optimum is reduced.

The QIGA's dependence on dθ and the lack of its ability to search for the local optimum lead to multiple variants of QIGA. While some improved the qubits with a density function [162, 163], others made dθ dynamic [164, 157]. QIGA was also combined with other optimization algorithms, such as gradient descent, for numerical problems with continuous variables. By combining QIGA with gradient descent, the algorithm can behave like a simulated annealing algorithm with qubits converging to the optimum instead of heat metric

”cooling down”. As such, a HGA is investigated to improve the optimization process of the prediction and simulation models.

The design space to optimize with both models include data and model features. The features to optimize are mainly discrete with parameters such as the method of indexing the input data, indexing step size, the number of prediction model layers, and the types of prediction model layers. There are some parameters such as dropout % that are not discrete so the optimization algorithm should be capable of handling these parameters as well. Past experiment [117] shows that the main contributors to improving the model accuracy come from having the right combination and number of neural network layers. So for this study, GA is selected since it is capable of optimizing the different combinations of model layers. As mentioned above, the optimum found with GA is limited mainly with the initial samples generated. This limitation could be reduced by generating new samples each generation with QIGA. So a HGA algorithm is developed and an illustration of this algorithm is shown in Figure 5.4.

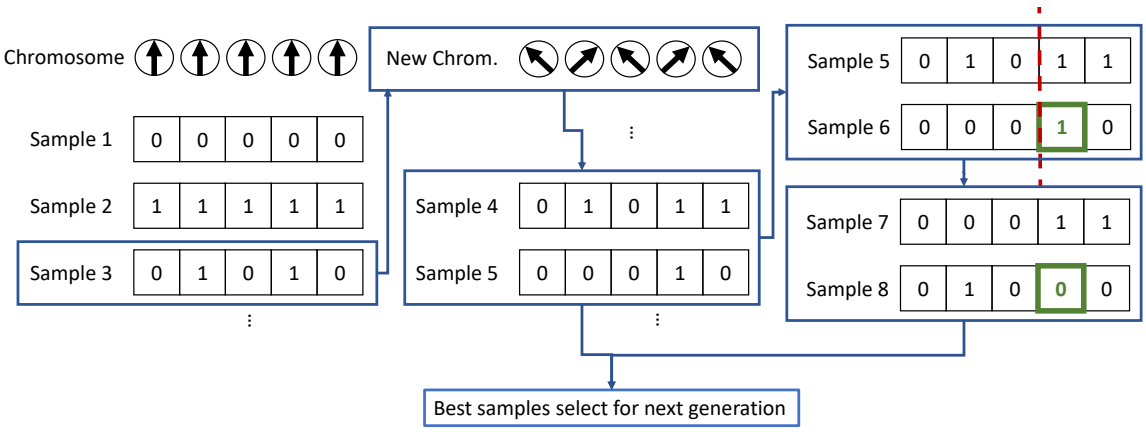

Figure 5.4: Illustration of how samples are made and optimized with HGA algorithm

The HGA algorithm starts with initial qubits ($Q(t)$) with $\theta = 45°$. Here t represents a generation. Samples of the population ($P(t)$) are generated from $Q(t)$ with a uniform random number generator for each qubit. After evaluating $P(t)$, the best sample (b) is saved. Then each of the qubits is updated by $d\theta$ to converge to b. Until here the HGA is

identical to the conventional QIGA. With HGA, instead of generating the entire population $P(t+1)$ with $Q(t+1)$, only half of the population is generated. The other half is populated by the surviving samples from $P(t)$ and they will be called $P'(t+1)$. $P'(t+1)$ samples are processed with the conventional genetic algorithm methods such as crossover, and mutation. In short, what happens with HGA is as follows: after each generation of samples is evaluated, the top-performing samples are kept for mutation and cross-over, while the bottom-performing samples are replaced by updated $Q(t+n)$. This process continues until Q is converged. The pseudo-code of the proposed HGA is shown in Algorithm 2

Algorithm 2 HGA algorithm with caption

```
t ← 0
initialize Q(t)
generate P(t) from Q(t)
evaluate P(t), find b, and identify P'(t+1)
update Q(t) by dθ to converge to b
while t ≤ max generation do
    t ← t + 1
    crossover P'(t)
    mutate P'(t)
    generate P(t) from Q(t)
    evaluate P(t) and P'(t) to find b and P'(t+1)
    update Q(t) by dθ to converge b
end while
```

5.2 Algorithm Performance

The HGA algorithm and a few other optimization algorithms are run on two different test functions in Equation 5.2. The other algorithms compared are conventional GA, Differential Evolution (DE), SA, and BO. The SA and DE algorithms compared are dual_annealing and differential_evolution optimization functions from the scipy library. The BO algorithm compared is gp_minimize optimization function from the scikit optimize library. The GA algorithm is self-developed based on the algorithm introduced above. The performance of the optimization algorithms is compared by the number of evaluations they used to con-

verge to an optimum and the optimum identified. The optimization problem is set up as follows. 5 parameters are required to be optimized each with a min-max range between -100 and 100 for the Ackley function and -1000 and 1000 for the Schwefel function. Initial states are found by repeating the random number generator. All algorithms are run for 5000 evaluations each. Note, the Ackley function is used as shown in Equation 5.2a. However, the Schwefel function is modified to represent a test function with discrete variables with a step size of 10. So when the Schwefel function is evaluated, each input value is rounded to the 10s. This is because the data and model features of both the prediction and simulation models are mostly discrete. So the Schwefel function is discretized to represent such behavior. The results are shown in Table 5.1.

Table 5.1: list of model parameters to optimize for prediction model

Result	Test f(x)	Global	HGA	GA	DE	SA	BO
Opt.	Ackley	0	13.7	19.5	20.7	3.7	16.8
	Schwefel	0	550	570	0	0	900
Eval.	Ackley	–	5000	5000	300	5000	100
	Schwefel	–	2000	4000	3000	1250	300

Based on the trial runs, the SA algorithm performed the best in terms of finding the optimum with the least number of function evaluations. The HGA performed second in terms of finding the optimum with similar performance observed from DE. This is because while HGA found the better optimum for the Ackley function with a higher number of iterations than DE, DE found the better optimum for the Schwefel function with a number of higher iterations than HGA. The BO was able to converge with very few function evaluations but it converged far from the global optimum. The GA had the opposite behavior where it took a lot of iterations to converge to a local optimum. Based on this observation the SA is a better selection. However, neural network optimization requires the algorithm to handle discrete features that are not ordinal. Indeed experimental results showed that SA took longer iterations to optimize the design space for this thesis and found a less optimum configuration. So HGA is selected to optimize the data and model features. Note for all of the algorithms tested the results do vary with each iteration and Table 5.1 is made based on

the typical behavior observed after 10 runs.

5.3 Optimizing Prediction and Dynamic Models

The developed HGA algorithm will be used to optimize the prediction and simulation models. The prediction model features to be optimized are shown in Table 5.2. The simulation

Table 5.2: list of model parameters to optimize for prediction model

No.	Data Parameters	Options
1	input index	time (s), altitude (ft)
2	min altitude for training data	300 - 500 (ft)
3	max altitude for training data	600 - 1500 (ft)
4	resolution: time	0.5, 1, 2, 4 (s)
5	resolution: altitude	10, 20, 30 (ft)
6	mean pooling size for training data	0, 2, 3, 4
7	normalization	none, min max, standard deviation
No.	Model Parameters	Options
8	activation functions	none, relu, tanh, sigmoid
9	CNN encoder	True, False
10	CNN filter	16, 32, 64
11	number of prediction layers	1 - 4
12	prediction models	RNN, GRU, LSTM
13	dropout	0 - 50 (%)
14	mean pooling size for CNN output	0, 1, 2, 3, 4
15	training batch size	8, 16, 32, 64

model features to be optimized are shown in Table 5.3.

For the prediction model, there are seven data parameters selected for optimization. The input index parameter is where the progress is defined by time or altitude. Time is considered because this world flows in time. Altitude is considered because during the final approach the aircraft descends until touchdown and height above touchdown is an indicator of progress. Depending on the input index parameter selected the resolution of the training data may be in seconds or ft. For example, if the resolution is 1 ft the input data is collected for every ft. The min and max altitude for training data is the input altitude windows used to perform the prediction. The min altitude range is selected based on the gate heights used by Campbell et al. [26] since the lower bound of 300 ft is selected based on this study. The max

Table 5.3: list of model parameters to optimize for simulation model

No.	Data Parameters	Options
1	input window length	1 - 10 (s)
2	output window length	1 - 5 (s)
3	resolution	0.5 - 2 (Hz)
4	normalization	none, min max, standard deviation
No.	Model Parameters	Options
5	activation functions	none, relu, tanh, sigmoid
6	number of prediction layers	1 - 4
7	number of hidden units	10 - 500
8	prediction models	RNN, GRU, LSTM, NNARX
9	dropout	0 - 50 (%)
10	training batch size	8, 16, 32, 64, 128

altitude range is selected based on an observation that the dynamics change significantly per flight above 1500 ft for the data obtained for this study. Some flights had a short cruise for a long time just above 1500 ft and other flights descended past 1500 without cruising. To minimize the design space, the maximum altitude of 1500 ft is selected. The mean pooling size is the window size for averaging the data at every step the data is collected. This process is necessary because there is noise in the data. By averaging the data for each step, the noise in the collection can be minimized. Finally, the normalization parameter determines how the data is normalized in each column. The methods considered include ”none,” ”min and max,” and ”standard deviation.” The ”none” method is when the data is not normalized. The ”Min and max” method normalizes the column with the difference between the minimum and maximum values. After normalization, the min and max values for each data feature are zero and one. The ”standard deviation” method normalizes the column by the standard deviation of each data feature. In short, this method converts the recorded data into z-scores.

There are also eight model parameters to be optimized for the prediction model. The activation function parameter is the list of activation functions considered for each neural network layer. Only the continuous activation functions are included. The CNN encoder parameter indicates whether a CNN will be used as an encoder. Then, the CNN filter indi-

cates the number of filters used by each CNN encoder. The mean pooling size parameter indicates the average pooling layer size after the CNN encoder. Here zero average pooling means this layer is not considered for that configuration. The number of prediction layers indicates the number of internal layers in the prediction model. The higher the number of prediction layers means the deeper the neural network model. The prediction model parameter is the list of recurrent neural network model layers considered for this study. Only the recurrent neural network models are considered due to the nonlinearity of the aircraft dynamics. The dropout parameter indicates the range of dropout percentages considered after each prediction layer. Typically larger dropout percentages are used to mitigate overfitting. Finally, the training batch size indicates the different batch sizes tested for the study.

There are four data parameters considered for the simulation model. The resolution and normalization parameters are identical to the data parameters introduced above. The input and output window lengths indicate the window size of the input and output data for each simulation step. There are six model parameters considered for the simulation model. The activation functions, number of prediction layers, dropout, and training batch size are identical to the model parameters introduced above. The number of hidden units is a unique parameter for the simulation model. It is the number of units in each model layer to increase the complexity of the model. Two more model layers are considered for this study. Feed-forward neural network and NNARX models are added because multiple studies successfully modeled the dynamics of a physical system with these models.

For both prediction and simulation models, the HGA is used after the initial models are generated. The parameters introduced above are identified after analyzing the performance of the initial models by manually varying each parameter and retraining the initial models. To simplify this process, the hyperband optimization algorithm is used to find the optimal model structure. The hyperband algorithm combines random search and successive halving to search for the optimal hyperparameter configuration [118]. An example is shown in Figure Figure 5.5. The concept of successive halving is as follows: Each configuration

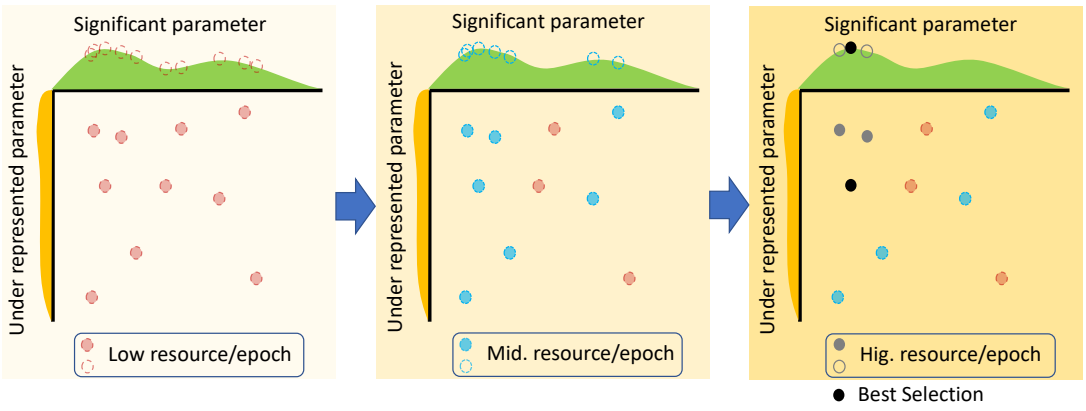

Figure 5.5: Example illustration of hyperband

is given the same amount of resources to search for the optimum, then half of the configurations with higher accuracy metric is chosen for the next round of trials until only one hyperparameter configuration survives. The hyperband algorithm solves the problem of n vs B/n with successive halving. The hyperband algorithm tries several configurations, n; if the convergence is fast, it quickly filters out the configurations with poor performance. However, if the convergence is slow, the algorithm allocates more resources, B, to each hyperparameter configuration. Hyperband is selected as the machine learning model's performance is unknown, and the algorithm can handle categorical hyperparameters.

The parameters optimized with the hyperband algorithm are as follows:

- $epoch$: number of times data is trained on the model
- N: number of RNN, GRU or LSTM layers
- M: number of CNN layers
- N_{units}: number of hidden units in RNN, GRU or LSTM layer N
- N_{bias}: presence of bias in RNN, GRU or LSTM layer N
- M_{filter}: number of filters in CNN layer M
- M_{kernel}: size of the kernel in CNN layer M

- $M_{strides}$: number of data points each filter is applied to in CNN layer M
- M_{bias}: presence of bias in CNN layer M

As shown the hyperband algorithm is mainly used for optimizing the model structure of RNN, GRU, or LSTM model layers. For the NNARX model the default structure provided by the deep learning toolbox was used as the initial model.

The findings and contributions from this chapter can be summarized as follows:

1. A HGA has been developed by combining GA and QIGA
2. HGA was observed to converge to a better solution than GA by exploring a larger design space at a similar speed
3. Experimental runs showed HGA finding a better optimum at a faster rate than SA, which performed best with the test functions

CHAPTER 6
DEVELOPMENT OF PREDICTION MODEL FOR AIRCRAFT STATES ABOVE TOUCHDOWN

So far runway overrun has been identified as one of the most significant contributors to runway accidents. Surveys on various runway accidents showed that overrun is highly correlated to the flight states near touchdown. It is also noted that the procedure is standardized during the final approach and that the near-touchdown flight states may be predicted with final approach data. However, prediction results may or may not be helpful to the pilots depending on the prediction accuracy and range. With accurate predictions made far ahead of touchdown, pilots could adjust the flight to mitigate potential risks. This ties back to a pilot training concept focused on enhancing the attitude, skill, and knowledge to reduce risk, also known as ASK. It also ties back to the global initiative GAPPRE which mentioned a need to increase pilots' awareness of potential risks. Past studies introduced the potential for neural network models to predict accurately and early. The problem is large design space needs to be explored so that accurate neural network models can be created. In this chapter, a framework is proposed to create a prediction model that can predict accurately and early. A summary of the framework is illustrated in Figure 6.1.

This framework is designed to create a prediction model that can predict flight states near touchdown. The predicted flight states are used to predict runway overrun precursors based on the threshold exceedance criteria introduced in AC 120-82 [12]. The predictions are made based on the flight data collected during the final approach. An illustration of the prediction model architecture is shown in Figure 6.2.

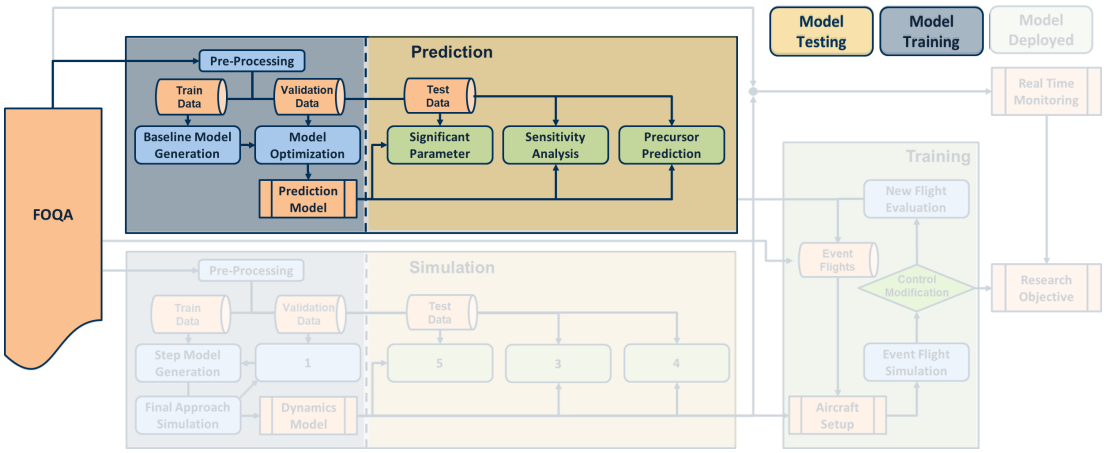

Figure 6.1: Prediction model development process overview

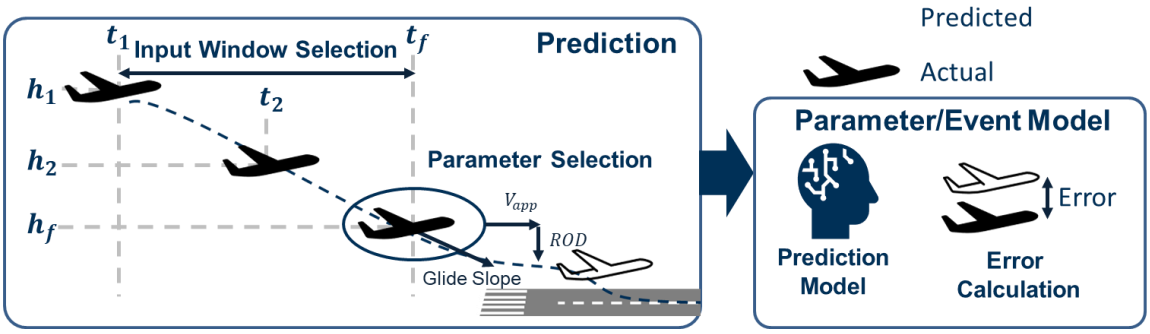

Figure 6.2: Prediction model architecture

6.1 Data Pre-Processing

The first step for developing a neural network is data preprocessing. As with any machine learning model, the quality of the training data has a strong implication on the model's accuracy. In the first step of preprocessing the aggregate data set with all flights is collected and cleaned. This data contains flight from 1500 ft height above touchdown to touchdown at the resolution of 4 Hz. 4 Hz instead of 16 Hz is selected because this is the lowest frequency that all of the flight states are collected. This way there is no need to consider the data augmentation process for flight states collected at lower frequencies. During the data collection process, flights with the following errors are removed.

1. Missing flight phases: flights missing core flight segments such as takeoff, cruise, and approach

2. Missing steps: flights missing recordings or significant jumps at random points in flight

3. Missing features: flights with missing recordings for random features during the entire approach or mission

4. Shifted readings: flights with significant jumps for a large duration during the approach

Some of the flight data features are also modified so that the data is more interpretable. All of the speed and acceleration parameters are modified so that their units are in ft/s. Wind speed and wind direction parameters are updated to represent wind speed in the aircraft body axis longitudinal and lateral directions. Finally, the fuel quantity metric is replaced by the cumulative sum of the fuel flow metrics. This is because the fuel quantity readings have significant noise from the sloshing effect. The counts of longitudinal position, velocities, accelerations, attitude, and wind speeds recorded for all of the final approach data, after the data cleanup process, are shown in Figure 6.3 and Figure 6.4. The Table 6.1 lists out all

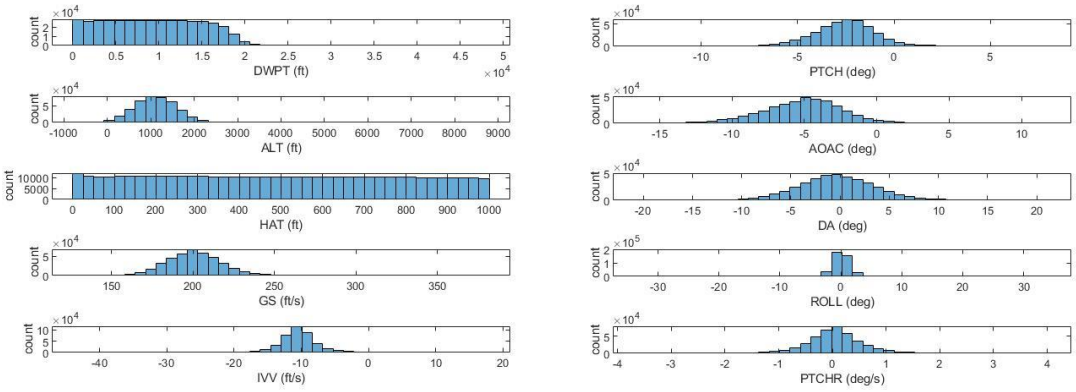

Figure 6.3: Aircraft longitudinal position, velocity (left), and attitude (right) recorded for all of the final approach data

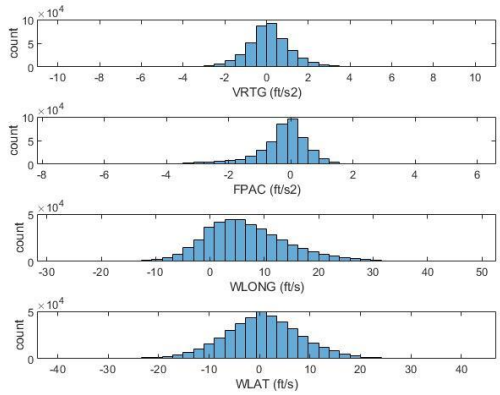

Figure 6.4: Aircraft longitudinal acceleration and wind speeds recorded for all of the final approach data

the abbreviations used for the prediction model development. In addition to these features

The data distributions show that almost all of the final approaches occurred below 2000 ft indicating that the airports landed are not significantly different in altitude. The ground speed during the final approach is between 150 to 250 ft/s, which is equivalent to 88 to 148 kts, a typical range expected near touchdown. The average rate of descent observed is at an expected level of 600 ft/min as well. All of the attitude recordings are as expected since the average of DA and ROLL distribution is at 0 degrees with aircraft pitching down (-) for most of the final approach. Interestingly the distribution of WLONG indicating that a larger number of flights experience headwind.

Once all of the flight data is cleaned, they are indexed so that the data structure matches the neural network model requirements. The input and output data structure is illustrated in Figure 6.5. Here n is the number of time stamps, m is the number of data features, l is the number of samples, o is the predicted features, and p is the predicted time stamps. As introduced in section 5.3, the flight data may be indexed with time or altitude. When the data is indexed by time, it is padded at the end of each flight such that the length of each flight is identical. This is a requirement for a typical neural network model. When data is indexed by altitude, each altitude point is used as a threshold to find the data row with HAT just above the threshold. The downside of this method is that it is difficult to space out the

Table 6.1: List of all of the flight state metric abbreviations

No.	Abbreviations	Definitions	$units$
1	DWPT	Distance to touchdown	ft
2	ALT	Altitude	ft
3	HAT	Height above touchdown	ft
4	RALT	Radial altitude	ft
5	GS	Ground speed	ft/s
6	CAS	Computed airspeed	kts
7	TAS	True airspeed	kts
8	MACH	Mach number	$-$
9	IVV	Vertical speed	ft/s
10	PTCH	Pitch angle	deg
11	AOAC	Corrected angle of attack	deg
12	DA	Drift angle	deg
13	ROLL	Roll angle	deg
14	PTCHR	Pitch rate	deg
15	VRTG	Vertical acceleration	ft/s^2
16	FPAC	Flight path acceleration	ft/s^2
17	BLAC	Body longitudinal acceleration	ft/s^2
18	CTAC	Cross track acceleration	ft/s^2
19	WLONG	Longitudinal wind speed	kts
20	WLAT	Latitudinal wind speed	kts
21	N1	Fan speed	$\%RPM$
22	N2	Core speed	$\%RPM$

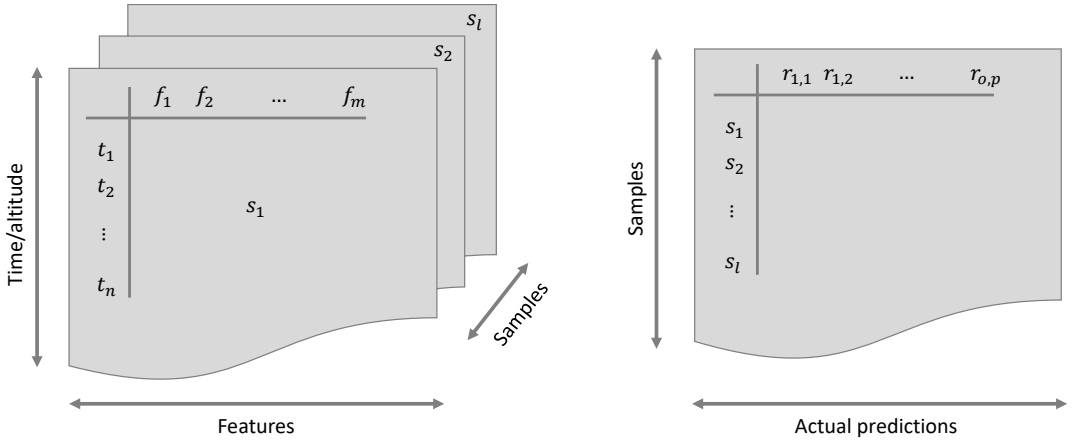

Figure 6.5: Data structure for the prediction model input (left) and output (right)

index evenly. Because flight data is not recorded by altitude but by time. Although input data can be indexed with both altitude and time, output data is only indexed with altitude. This is because time does not directly represent the progress to touchdown and the goal is to predict flight states and precursors before touchdown. During the indexing process, the

selected data row is augmented by taking the average of the data a few seconds before and after. So a sliding window mechanism, that represents the average pooling layer of a neural network, has been used to augment both the input and output data. This is done to account for any white noise in the measurement due to sensor accuracy. The averaging is done from 0 - 4 seconds to ensure that a model is also trained with data that did not go through this process. After this, each feature of the input and output data is normalized with a min-max scale and standard scale. This is frequently done on machine learning data to improve the prediction results. Both of the normalization techniques have their own benefits. Min-max scale is capable of modifying each number to a percentage scale where 0% represents the minimum value observed and 100 percent represents the maximum. This is a risk when a large change in the approach process causes a shift in the min-max range, resulting in values above 100% or below 0%. Also, the min-max range is sensitive to outliers meaning the data will get skewed if extremely large or small measurements are made. Standard scale takes care of this issue by converting each value to its corresponding z-score. However, the normalized data is not scaled between 0 and 1, which is frequently discussed as the best data range for training machine learning models. Once the data has been preprocessed it is used to train a neural network based prediction model.

For this study, flight data was obtained from NASA's open-source FOQA data available at the DASHlink website[1] for single tail number aircraft. A total of 5376 flights were collected, out of which 4128 flights were considered usable. The unusable flights are ones with missing flight phases, missing steps, missing features, or large shifts in some readings. Flights with missing flight phases have certain flight phases missing from the mission data. For example, the flight data would be missing cruise, takeoff, or approach phases. Flights with missing steps recorded large jumps in readings that are physically impossible. For example, the true airspeed during the final approach may jump from 120 kts to well over 200 kts over a second. Flights with missing features are flights without recordings in certain

[1] https://c3.nasa.gov/dashlink/resources/?page=3&sort=-created&type=28

features while the other features are recorded properly. Finally, flights with a large shift in some readings are flights with step changes in the recorded flight controls and states. Because the goal of this study is on creating neural network models, flights that observed these issues were removed from prediction model training. This is because the prediction model training process requires the model to interpret the full final approach dynamics to predict the aircraft states near touchdown. For prediction model training, 90 % of the data was used for training, 5 % for validating, and the last 5 % for testing the model. All of the results discussed in this study are generated based on the 5 % of the test data set.

6.2 Base Model Structure

With the preprocessed data, the next task for prediction model development was to test the model architectures. As discussed in section 5.3, there are multiple factors to consider to develop a prediction model that accurately predicts aircraft states far away from a touchdown. So creating the best model requires an iterative process that optimizes both model and data parameters. To start this process model parameters were explored in depth with the hyperband algorithm to identify the best architecture. There are multiple neural network architectures and at the core are the neurons. The neurons are connected to each other to form basic neural network models such as feed-forward models. When the system that the models are trying to represent has nonlinearity, the feedback loops are added to each of the neurons or neuron layers to form RNN. A common issue with RNN is the exploding or diminishing feedback. To resolve this issue, recent studies added gates to the RNN models and these are the GNNs. Depending on the type of data that is being handled, some neural networks would add convolution and pooling layers to process the data through multiple filters. Convolution layers act as if a person is looking at an artwork through colored lenses. These different neurons and layers may be combined in different ways to make architectures such as encoders and decoders, and deep neural networks. Tensorflow and Keras packages were used for developing and testing the neural network models.

The hyperband algorithm was used to answer what kind of architecture and which neural network model is best suited for predicting aircraft states during the final approach. This way the design space to be explored by the HGA can be reduced. To also simply the design space, two separate prediction models were created; one for predicting aircraft longitudinal speeds (CAS and TAS) and one for prediction of vertical speed (IVV) and pitch angle (PTCH). For the first model, two different longitudinal speeds are predicted because CAS is independent of wind speed while TAS is dependent on wind speed. So by comparing the errors in prediction, the model sensitivity to wind may be observed. For the second model, the two predicted features are two separate states in the aircraft dynamics. However, they both have significant representations of the aircraft lift, which is what an aircraft is all about. So by comparing the prediction results for both factors the predictability of the two features can be compared and the more predictable feature may be selected.

The objective function used for designing the neural networks is the RMSE of the predicted and actual states for all predicted altitudes. The objective function has significant implications for how the neural networks are trained. For example, the RMSE may be weighted by altitude or feature so that emphasis is put on predictions closer to touchdown with a higher correlation to overrun. Different functions, such as Mean Squared Error (MSE), may also be considered. Since the focus of this work is not on the neural network but on creating a framework to improve runway safety, the regular RMSE was used as the objective function.

Since the hyperband algorithm is only optimizing the model parameters, the data parameters are preset based on past experience with developing air speed prediction models. In this work, predicted features are predicted at a range of target altitudes every 20 ft from 200 ft height above touchdown to touchdown. The prediction at touchdown is not considered because of the recording noise observed at touchdown, also addressed by prior work [117].

The prediction data was collected from 1000 ft to 300 ft HAT. The 1000 ft upper limit

is the decision altitude used by pilots to proceed with the final approach. The 300 ft lower limit is the lowest altitude at which pilots can respond to changes from a study by Campbell et al. [26] The prediction data was indexed by altitude and the resolution of the data was also set at 20 ft. 20 ft was selected here because at the start of the final approach when the ROD is high, data points can be collected at approximately every 20 ft. Note this does mean some of the errors in prediction are due to the indexing error. The distribution of index errors from 1000 ft to 10 ft HAT is shown in Figure 6.6. As shown, the indexing error at all altitude range between 2.5 and 2.8 ft RMSE. This means the relative error at each altitude increase linearly with decreasing altitude. So this should be a factor to consider when analyzing prediction results, which are between 200 and 10 ft HAT. The

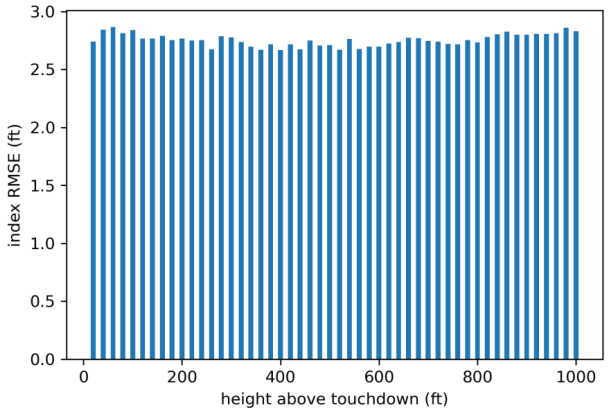

Figure 6.6: Error distribution of prediction data index due to difference in how the data is collected in flight

ground speed of prediction data and predicted data are both averaged using the data two seconds before and after to account for noise and fluctuations. Both prediction data and predicted data are normalized by the min-max range of each feature.

6.3 Initial Model Evaluation

The trained prediction models were used to predict the aircraft states near touchdown with the test data. The example prediction results for the CAS and TAS can be found in Figure 6.7. The example prediction results for the IVV and PTCH predictions can be found

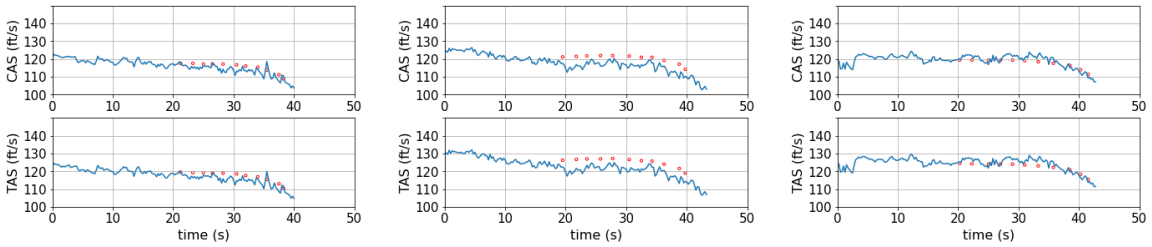

Figure 6.7: Test prediction results (red) vs actual (blue) CAS and TAS

in Figure 6.8. In the figures, the blue lines represent the actual trajectory and the red dots

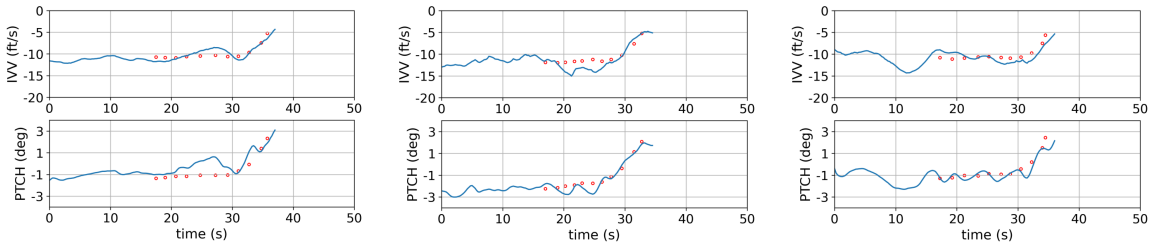

Figure 6.8: Test prediction results (red) vs actual (blue) IVV and PTCH

represent predicted states. The CAS and TAS prediction model shows that the prediction can be accurate, but more importantly, the predicted state trajectories follow the overall flight direction with small offsets. These examples also show similar patterns. This finding indicates that the neural network model is creating a regression of the aircraft speed as expected. However, this is not a simple regression model that takes the mean or the median distribution at each predicted altitude. This is because the prediction results are still representative of flight speeds with slight differences in the trajectories. The neural network is still strictly accurate to the data used to train the model. The same is true for the IVV and TAS prediction results. The results indicate that for the IVV prediction, the neural network model highly resembles a simple polynomial regression model. The IVV trajectories frequently show a cyclic behavior since pilots are constantly trying to adjust the aircraft states to match the ideal touchdown condition. The prediction model is not capable of predicting the details of the cyclic behavior. However, the prediction results resemble a path that goes through the center of each IVV trajectory. This can also be determined from the randomness in the direction of the error, at a given altitude, from one flight to

another. Potential causes include the lack of data or the lack of the selected model's ability to represent the cyclic behavior. PTCH prediction results show a similar behavior as IVV prediction because PTCH directly impact AOA, and AOA has a linear relationship with lift, which is also defined by the difference of IVV. However, the magnitude of the cyclic changes for PTCH is different and this does affect how much the neural network model resembles a simple polynomial regression model based on the error plots. This difference can be observed from RMSE and rel error distributions across the touchdown altitudes.

The prediction errors for CAS and TAS prediction, based on the test data set, are shown in Figure 6.9. The RMSE distribution plot is shown to the left and the relative error distri-

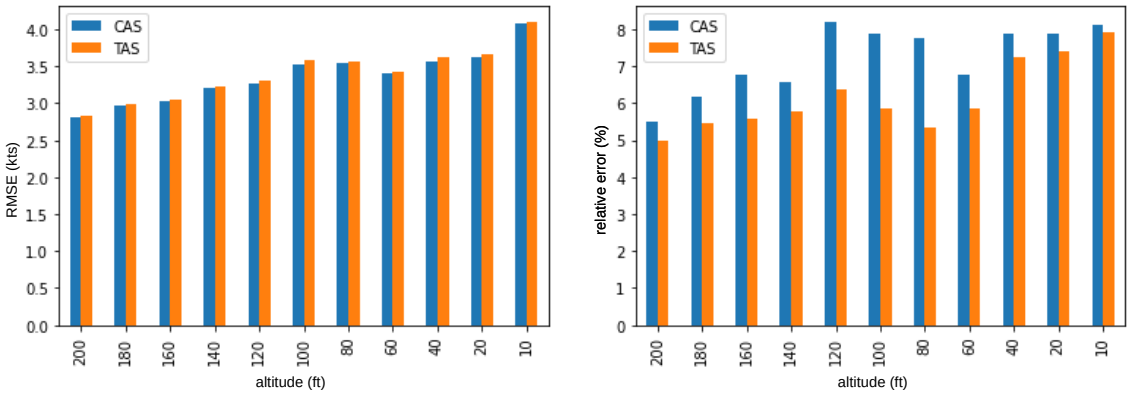

Figure 6.9: Test prediction RMSE (left) and relative (right) error distributions for CAS and TAS

bution plot is shown to the right. The equations used to calculate the errors are shown on the y-axis of each plot. For each target prediction altitude, the RMSE and relative error are calculated at each prediction altitude from the prediction residual. The RMSE is used to measure the spread of the residual in the unit of the predicted parameter. The relative error is used to measure the residual normalized by the min-max range of the recording at each predicted altitude. The reason why both RMSE and relative error plots are observed is that the speed parameters converge to ideal touchdown conditions with decreasing altitude and this may have a positive effect on the RMSE measurements. So relative error must also be observed to show the error magnitude relative to the true data range. An increase in

the residual with decreasing altitude is expected based on previous studies with prediction models.

The prediction result shows that for CAS and TAS model the RMSE increases with decreasing altitude. The RMSE ranges between 2.7 kts, at 200 ft HAT, and 4.0 kts, at 20 ft HAT. This error behavior is expected because prediction states further away are affected by more external factors such as control or wind. There is a local RMSE maximum at 100 ft HAT but the potential cause of this behavior will be discussed in the subsection 6.7.2. Similar behavior is observed for the relative error. This is important to note because there was a concern that the neural network model may be a simple regression model of prediction features based on prediction altitudes. If this were to be true, then the relative error should either decrease or remain constant with decreasing altitude. This is because aircraft speed will converge to a single value near touchdown. However, the increasing relative error indicates that the neural network model is learning from the aircraft behavior to improve the prediction from a simple regression model. The RMSE of CAS and TAS are almost similar. This is expected because TAS is CAS corrected for compressibility and temperature. The compressibility and temperature are not significant factors near touchdown so the two should have similar recordings. The small difference is due to the difference in the min-max range of each feature at prediction altitudes Figure 6.10.

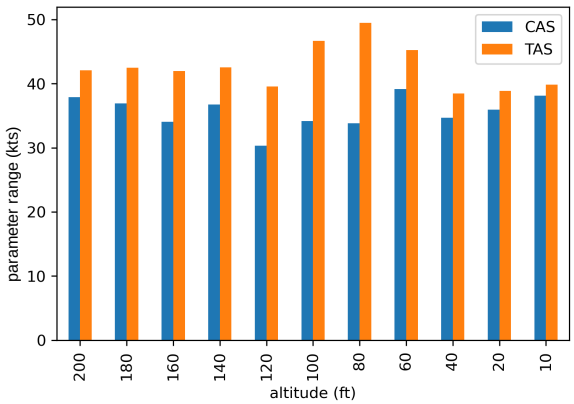

Figure 6.10: Minimum and maximum range of each prediction features across prediction altitudes

As mentioned, the neural network model is observed to be more than a simple regression model of aircraft speed over altitude. So a polynomial regression model of TAS was also generated with HAT as input. Only the TAS was created because the neural network prediction error is lower. The polynomial regression model was generated by increasing the polynomial order until the error converges to a minimum. For the TAS model this occurred with the sixth-order polynomial. This means for the data collected for this study, the nominal TAS behavior near touchdown is in the order of six. This training process is shown in Figure 6.11.

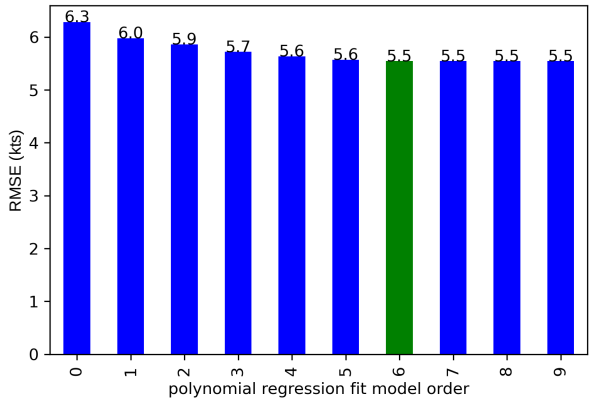

Figure 6.11: Polynomial prediction model of order six selected (green) based on the improvements in RMSE with increasing order

The RMSE and relative error of the polynomial prediction model are shown in Figure 6.12. The result shows that the RMSE is constant between 200 ft to 10 ft HAT at approximately 5.5 kts. This is close to double that of the neural network prediction at 200 ft HAT and close to 1.4 times that of the neural network prediction at 10 ft HAT. This is an indication that the neural network model is learning the aircraft dynamic. The difference in relative error between neural network and polynomial regression model predictions as well. The polynomial regression model also gives more information about how the distribution of the final approach behavior. The RMSE error tells that unlike what is expected, the aircraft speed does not necessarily converge to a single value with decreasing altitude. In fact, for altitudes below 200 ft, the distribution of speed does not change. However, that

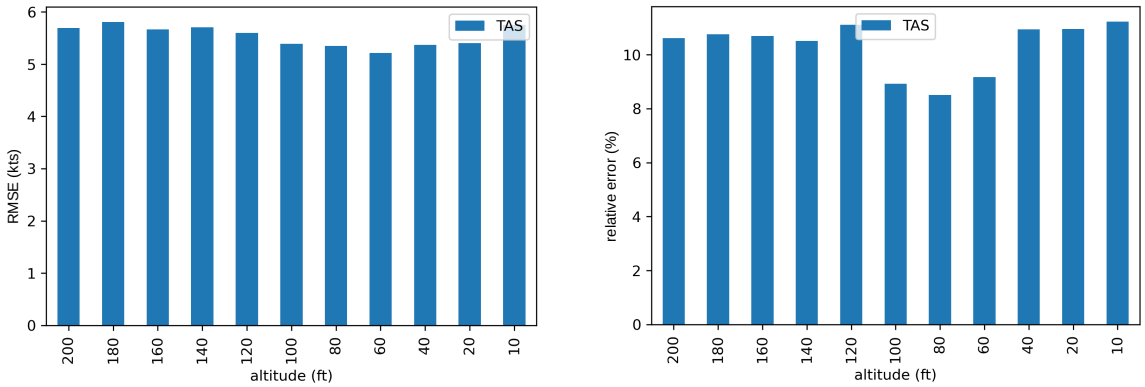

Figure 6.12: Polynomial regression model test prediction RMSE (left) and relative (right) error distributions

does not mean the min-max range of the speed is constant. A short drop in relative error at 80 ft HAT is observed and this indicates the increase in the min-max range at this altitude. The potential cause of this will be discussed in the future section.

The prediction errors of IVV and PTCH prediction are shown in Figure 6.13. The

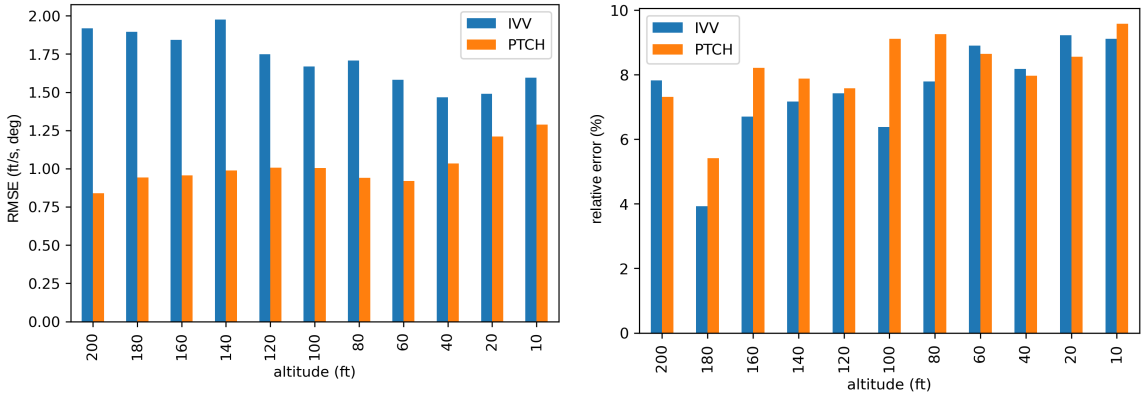

Figure 6.13: Test prediction RMSE (left) and relative (right) error distributions for IVV and PTCH

prediction results show that for IVV model the RMSE decreases with decreasing altitude. The RMSE ranges from 2 ft/s to 1.5 ft/s. This behavior aligns with the previous statement that the prediction should improve with decreasing altitude because the IVV converges to zero at touchdown. This is an indication that the neural network model may be a simple

regression model, again because the model is not capable of capturing the details of the cyclic behaviors. However, the RMSE distribution of PTCH much closely resembles that of the TAS. The PTCH prediction error ranges between 0.8 deg to 1.25 deg. This indicates that the PTCH is more predictable than IVV.

Interestingly the relative error distribution of IVV and PTCH are very similar. This is expected because the trajectories of the two features follow a similar pattern. This means for both features the prediction model is making a regression line through the center of the distribution at each altitude. So why are the RMSE distributions different? One explanation is related to the magnitude of the IVV and PTCH records. The IVV recordings are much higher than that of PTCH. The problem is the change in PTCH just before touchdown is close to the IVV RMSE. So due to the dynamics of the aircraft motion and the units, the PTCH prediction may seem much more accurate that IVV. However, that doesn't mean the neural network is completely unable to capture the dynamics of PTCH better than a polynomial regression model.

The polynomial regression model error distributions, shown in Figure 6.14, show that the neural network model does have a slightly lower error distribution for PTCH than that of the polynomial regression. However, the improvements are small, indicating that the

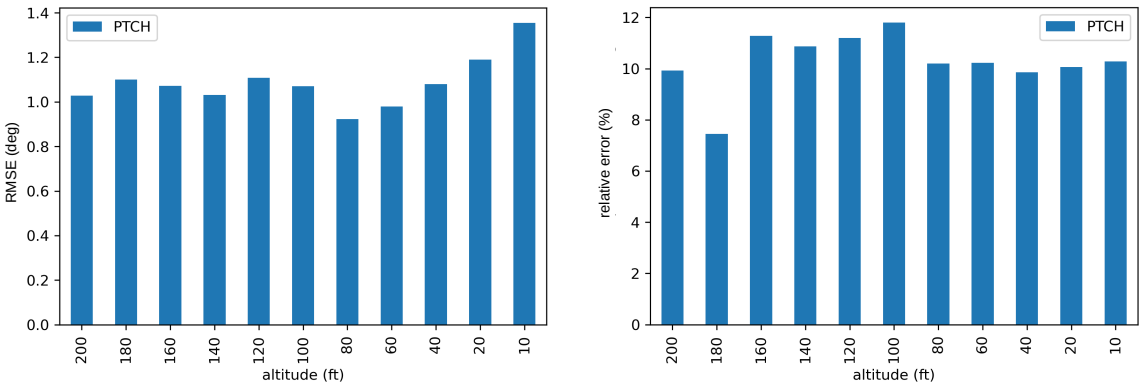

Figure 6.14: Polynomial regression model of PTCH RMSE (left) and relative (right) error distributions

major problem of cyclic behavior must be investigated to reduce the error. The relative

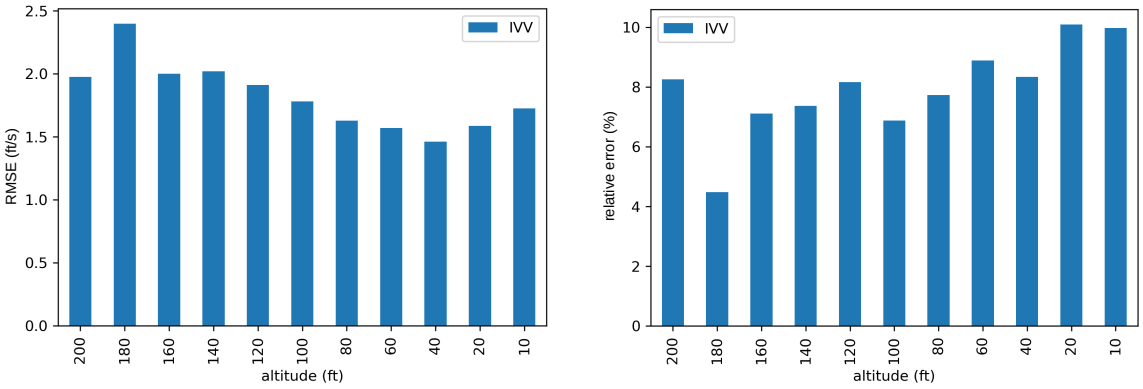

Figure 6.15: Polynomial regression model of IVV RMSE (left) and relative (right) error distributions

error for PTCH is also higher for the polynomial regression model, which again points to the need to investigate a neural network model that can capture cyclic system behavior. Unfortunately for the IVV prediction, the error with the neural network model is almost identical to the error with polynomial regression as shown in Figure 6.15.

6.4 Prediction Model From A Different Perspective

So far the focus has been on testing two prediction models trained with a single data configuration. Both of these were compared against a baseline prediction model created with polynomial regression of predicted states based on HAT. Then a question came up, how does the prediction result change with different data configurations? Before answering this question with the HGA, two models were created to observe any improvements by changing the hyperparameters. The first is a prediction model with training data indexed by time. The resolution of this data is set as one Hz. The second is a prediction model without data pooling. The mean pooling window size of the original model was 2. Then one more model was created to see if flights flown on autopilot may impact the predictability of aircraft states. As technology improves the reliance on autopilot is also increasing. A prediction model may provide pilots with knowledge in an environment where the risk

may be higher to maintain autopilot. Note the same set of train, validation, and test data sets were used to train the models and generate the results.

The prediction model for CAS and TAS trained with data indexed by time has a similar error distribution as the model trained with data indexed by altitude. The RMSE and relative error distributions are shown in Figure 6.16. However, the increase in RMSE is

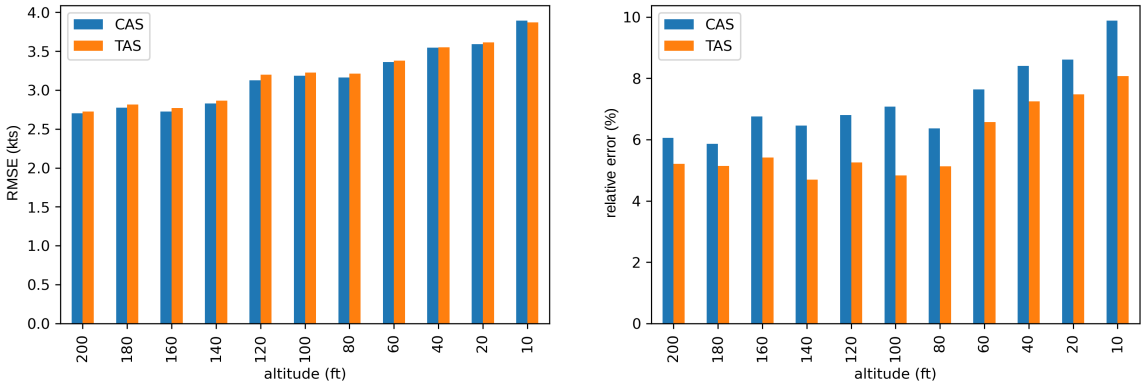

Figure 6.16: Neural network model of CAS and TAS, trained based on time index, test prediction RMSE (left) and relative (right) error distributions

much more linear with decreasing HAT. This indicates a potential correlation between aircraft states at certain altitudes above 300 ft to the aircraft states below 200 ft HAT. This behavior can also be observed from the relative error distribution plot. The relative error of the new model is relatively constant, at about 5 kts, until 80 ft HAT. Below 80 ft HAT the relative error increases linearly for both CAS and TAS. Constant accuracy above 80 ft HAT shows that the flight data between 300 ft and 1000 ft HAT have a higher correlation to the aircraft states above 80 ft than the aircraft state below 80 ft HAT. This result again points to a unique behavior across multiple flights at 80 ft HAT that will be discussed in the future section.

This time the RMSE and relative error distributions of the model trained without data pooling are shown in Figure 6.17. The error distributions are almost identical to that of the original model discussed in Figure 6.9. There are multiple factors why the error distributions may be identical. The first is that a window size of 2 may not be the correct size

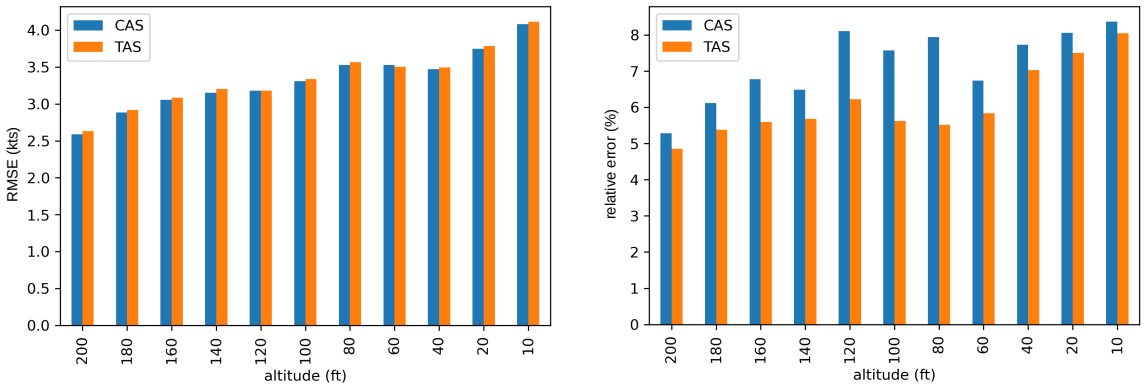

Figure 6.17: Neural network model of CAS and TAS, trained without averaging, test prediction RMSE (left) and relative (right) error distributions

for reducing the noise in the data. The second is that the noise in the flight data cannot be simply reduced by taking the average from a few data points before and after. The last is that the neural network prediction model in the end is still a regression model, which means the predictions are the central values based on the input data configuration. However, the different mean pooling window sizes may improve the results and this will be further investigated in section 6.6.

The test result for the neural network model trained with autopilot flights showed better results. For this experimental environment, the autopilot may be defined in four different ways. The first is when the autopilot was on during the entire final approach. The second is when the autopilot was on above 300 ft HAT. The third is when the autopilot was on below 300 ft HAT. The last is when the autopilot did not turn on at all during the final approach. The 300 ft threshold is set based on the lowest altitude that the training data is used. To increase the number of available for training, the flight segments above or below 300 ft are considered autopilot as long as there was any recording of autopilot switching on. Based on this definition, 2880 flights had autopilot above 300 ft HAT and the data set representing the second autopilot condition was used to train the model. The RMSE and relative error distributions of the model trained and tested with autopilot flights are shown in Figure 6.18. As expected, the autopilot flights were more predictable. Besides

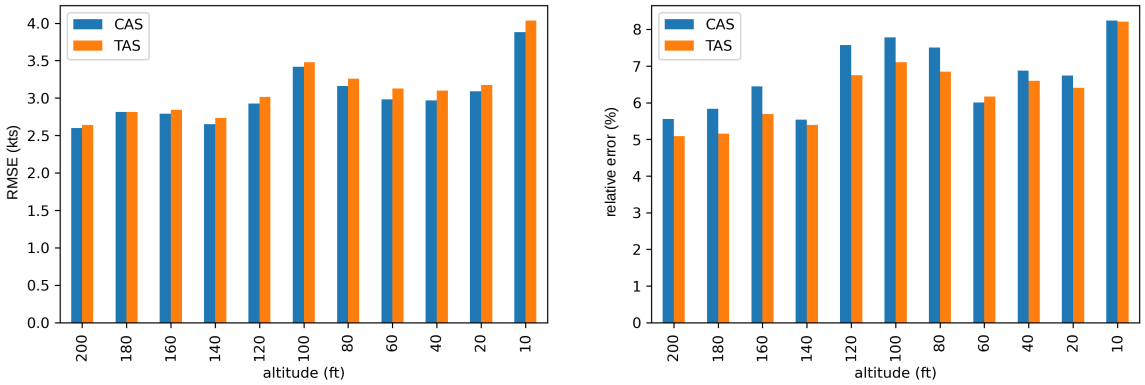

Figure 6.18: RMSE (left) and relative (right) error distributions of CAS and TAS neural network model trained and tested with autopilot flights

prediction at 100 ft and 10 ft HAT, the RMSE and relative error both decreased slightly. As a reminder, the definition of autopilot was set very loose. However, the small improvement is an indication that the prediction model has better synergy with the autopilot system.

So how do the IVV and PTCH prediction results change with different training data configurations? The Figure 6.19 shows the error distributions of IVV and PTCH prediction trained with data indexed by time. The results are almost identical to the model trained

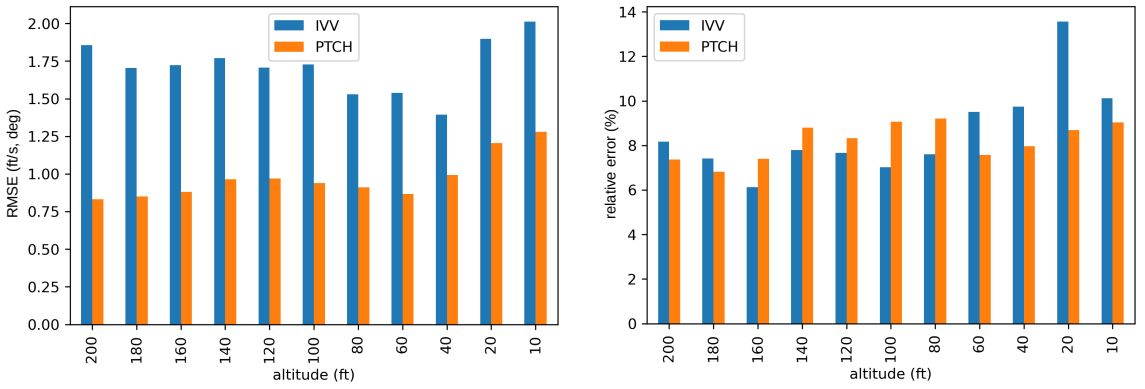

Figure 6.19: Neural network model of IVV and PTCH, trained with data indexed by time, test prediction RMSE (left) and relative (right) error distributions

with data indexed by altitude. So not a significant correlation between final approach states above 300 ft HAT and prediction states below 200 ft HAT can be made. In fact, slightly

higher errors in the IVV predictions are observed indicating that the IVV prediction model is behaving like a polynomial regression model of IVV based on altitude.

A similar conclusion can be made with IVV and PTCH prediction models trained with data without mean pooling. The error distributions can be found in Figure 6.20. For this

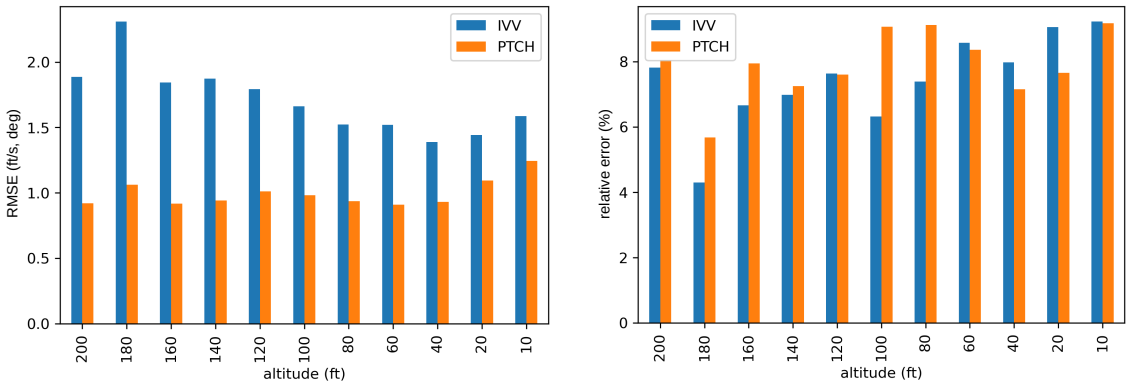

Figure 6.20: Neural network model of IVV and PTCH, trained without averaging, test prediction RMSE (left) and relative (right) error distributions

model, the IVV prediction error has the same distribution as the original model with mean pooling. Interestingly the PTCH prediction error distribution is even more similar to the polynomial regression error distribution compared to the original model error distribution. The mean pooling was thought to be a potential cause for the model to behave like a polynomial regression model. But this result shows that the pooling was not the issue and the issue is with the model's lack of ability to consider cyclic behaviors.

The RMSE and relative error distributions of the IVV and PTCH model trained and tested with autopilot flights are shown in Figure 6.21. Unfortunately, this also did not see many improvements in prediction accuracy. This is expected because the cyclic trajectories of the IVV and PTCH parameters were visible for fights that landed with autopilot as well.

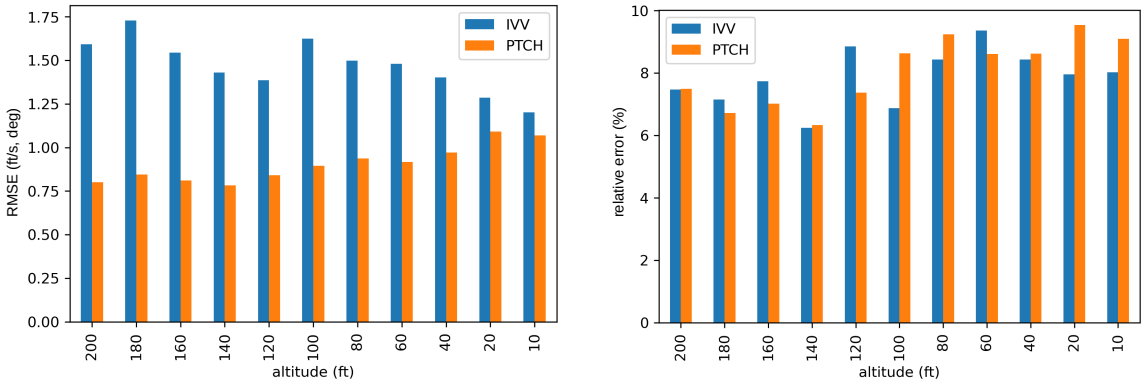

Figure 6.21: RMSE (left) and relative (right) error distributions of IVV and PTCH neural network model trained and tested with autopilot flights

6.5 Clustered Model Evaluation

Another consideration to improve the prediction result was to cluster similar flights and train the model separately for each cluster. The idea was that since the prediction model is in truth a more complex regression model, the prediction accuracy should improve if the flights being predicted have similar trajectories. Yes, according to this logic there needs to be a method of clustering the data and the accuracy of this model will also contribute to the overall prediction accuracy. Regardless of this risk, the potential benefit of training prediction models with clustered data was investigated.

Multiple data clustering techniques were introduced in subsubsection 2.2.1. Two different clustering methods were considered for this study. The first method is one introduced by the author [165] where the braking ground roll data was clustered to analyze the braking performance. This technique uses Dynamic Time Warping (DTW) to measure the similarity of flight trajectories. The similarity of trajectories is measured with the dynamic programming method. Here the step-wise difference for every pair of trajectories is measured. Each step of the two trajectories is connected such that each step minimizes the measured difference between the two trajectories. An illustration of this process is shown in Figure 6.22 Once the similarities of each pair of trajectories are measured, they are clus-

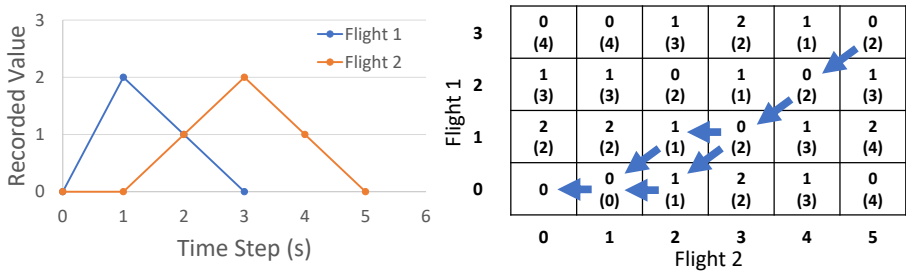

Figure 6.22: Step wise difference matrix between trajectories of different size and the difference minimizing path shown with blue errors [165]

tered using machine learning. Another method of clustering the flights is by converting each final approach trajectory into the frequency domain. The frequency data is then used to cluster the flights with machine learning.

The benefit of the first method is that it clusters flights directly based on the trajectory flown. This includes the altitude at each point of the flights, how long were the flights, when were the controls applied, and more. However, this process does cost a lot of computation power, especially with the increasing number of data sizes to calculate the similarity measures between all flights. The second method doesn't have the issue with cost because converting each flight into the frequency domain is not expensive to compute. Also, in the frequency domain, cyclic behaviors may be easily identified and grouped. The issue is that the clustered flights may have similar flight patterns but not necessarily the same. For example, two sine waves with a phase shift of 90 degrees will have the same representation in the frequency domain. However, as observed previously, the neural network will simply form a regression line that goes through the center of the data distribution since at certain points in time the two sine waves will diverge. The second technique was selected based on the limitation of computational power. So all of the final approach data were converted to the frequency domain with the Fourier transform. Then the frequency domain data was clustered with the agglomerative technique based on the success of this technique in the past study [165].

The amplitude, frequency, HAT, DWPT, IVV, and TAS plots of each cluster are shown

in Figure 6.23 to Figure 6.26 The results show that the HAT of each cluster have very

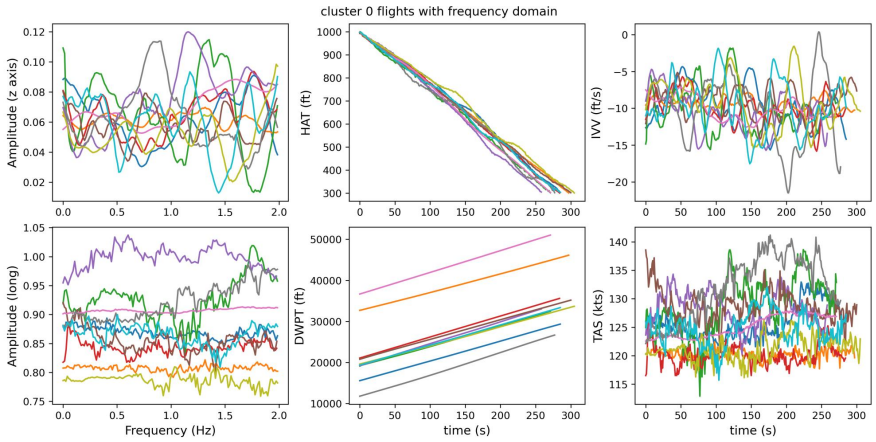

Figure 6.23: Group 1 of the final approach data clustered in frequency domain

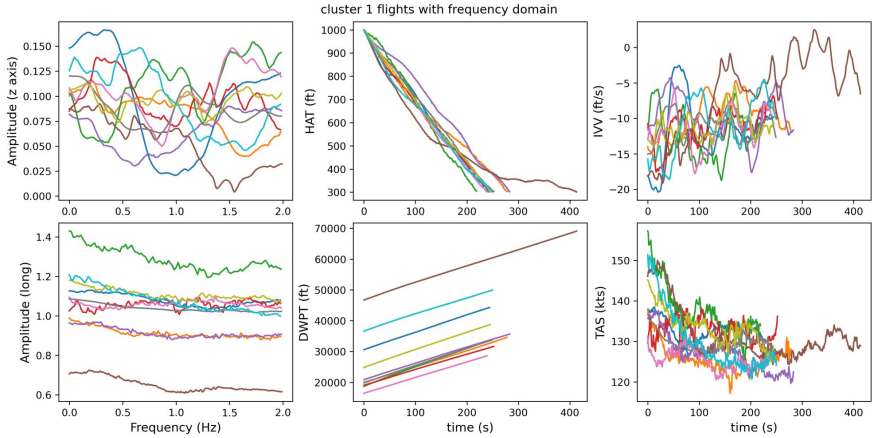

Figure 6.24: Group 2 of the final approach data clustered in frequency domain

similar trajectories. Also, not only are IVV and TAS trajectories similar for each cluster, the magnitude that these trajectories represent in each cluster is different.

Since the number of flights available for training is limited, the cluster with the largest number of flights was selected to train the prediction model. The CAS and TAS prediction error distributions based on models trained with clustered data are shown in Figure 6.27. The results did not show improvement in prediction accuracy with data clustering. In fact the RMSE and relative errors plots show that the errors are consistent with altitude. This is the behavior observed with polynomial regression error and it is not preferred. The same

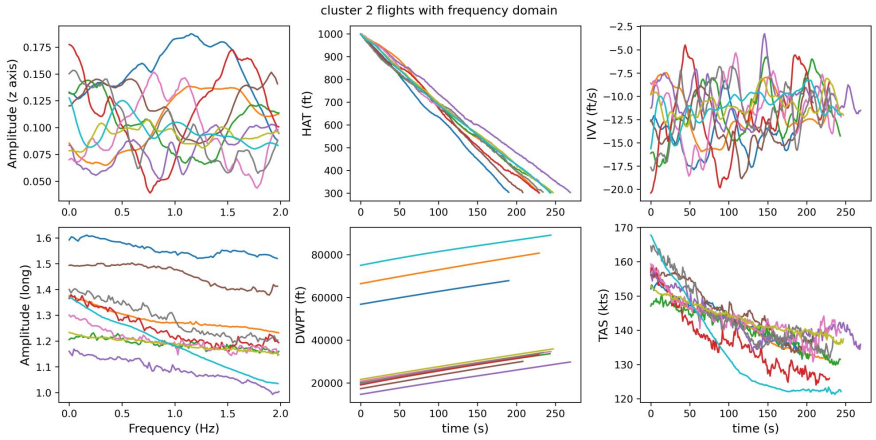

Figure 6.25: Group 3 of the final approach data clustered in frequency domain

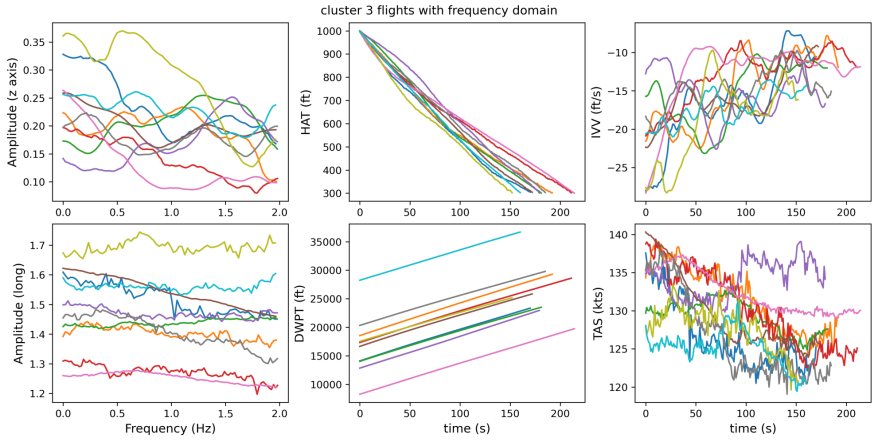

Figure 6.26: Group 4 of the final approach data clustered in frequency domain

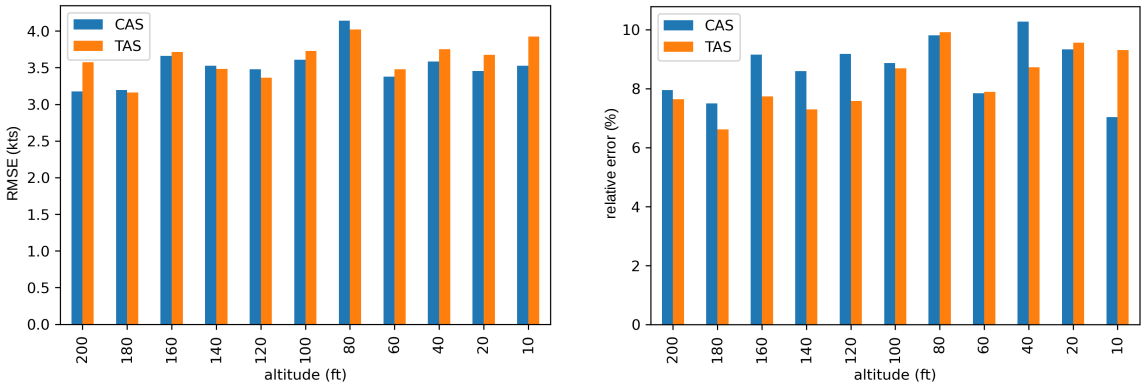

Figure 6.27: Neural network model for CAS and TAS, trained with clustered data, test prediction RMSE (left) and relative (right) error distributions

result is observed with the IVV and PTCH prediction models trained with clustered data. So based on this experiment, data clustering did not help with improving prediction model

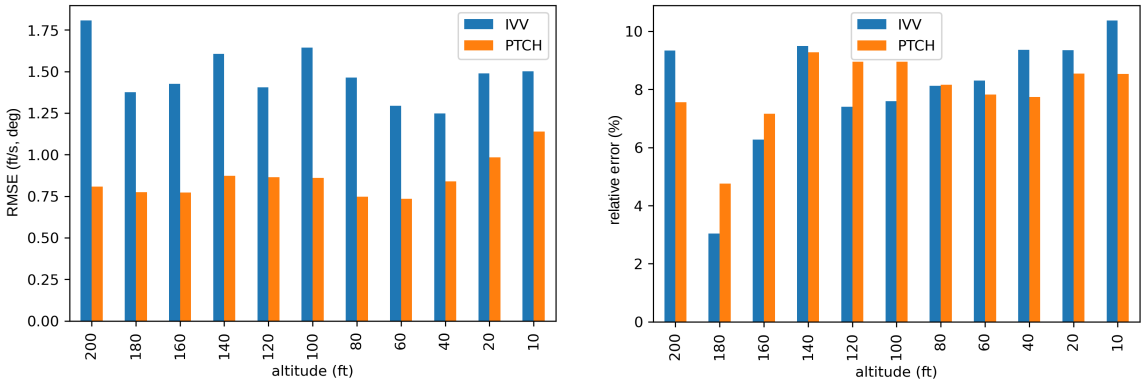

Figure 6.28: Neural network model for IVV and PTCH, trained with clustered data, test prediction RMSE (left) and relative (right) error distributions

accuracy. However, there are a few considerations to be made. The first is the number of data samples available for training. After clustering the data, the largest cluster had 1883 flights. This is a relatively small set of data to train a neural network with multiple layers. The second is the clustering techniques to be tested. As mentioned earlier, the clustered result may vary significantly with the clustering method. Different methods should be tested until the best algorithm to train the clustered data model.

6.6 Input Optimized Model Evaluation

Hyperparameter optimization was performed by using HGA to optimize the data and model parameters. The parameters optimized are discussed in section 5.3. This time, instead of generating separate prediction models for longitudinal and vertical states, a single model that predict all the features was made. The features predicted are IVV, PTCH, and TAS. IVV and PTCH are included to see if prediction results can improve by modifying the data and model parameters. TAS is included instead of CAS because the prediction accuracy for TAS is higher than that of CAS.

The ideal configuration of the prediction model is as shown in Table 6.2. For the data

Table 6.2: hyperparameter optimized configuration for the prediction model

No.	Data Parameters	Optimized Value (unit)
1	input index	altitude (ft)
2	min altitude for training data	300 (ft)
3	max altitude for training data	1070 (ft)
5	resolution: altitude	10 (ft)
6	mean pooling size for training data	0
7	normalization	min max
No.	Model Parameters	Optimized Value (unit)
8	CNN activation function	tanh
9	CNN encoder	True
10	CNN filter	32
11	number of prediction layers	1
8	LSTM activation function	relu
12	prediction model	LSTM
13	dropout	11 (%)
14	mean pooling size for CNN output	0
15	training batch size	8

parameters, the optimized structure was almost identical to the one used to generate the results for the initial studies. A noticeable change is that the ideal resolution is identified to be 10 ft instead of 20 ft. Another change is the max altitude for training data and this is found to be 1070 ft. This is close to the 1000 ft used to generate the initial model. However, the next ideal configurations have a max altitude higher than 1070 ft (between 1070 and 1300) so a larger input window is observed to improve the prediction results. The optimal model configuration is found to be different from the initial configuration as well. The optimized model configuration is found to have one CNN and one LSTM neural network layers for prediction. Having one CNN layer to filter the input data to generate different convolution data set match with the findings from the past study [117], which used hyperband to optimize the model. However, this time only one LSTM layer was selected, instead of two, for the ideal configuration. The final score for the optimal configuration is RMSE of 2.445%. The percentage sign is added to the RMSE score because the input data was normalized with the min-max range of each feature. Since RMSE keeps the unit of the original data, the final score may be considered as a percentage from the max range of 1.

The HGA was successful in improving the prediction accuracy for the TAS feature. The comparison of hyperparameter optimized model RMSE vs best TAS initial model RMSE, in Figure 6.17, is shown in Figure 6.29. The improvement is across the full prediction

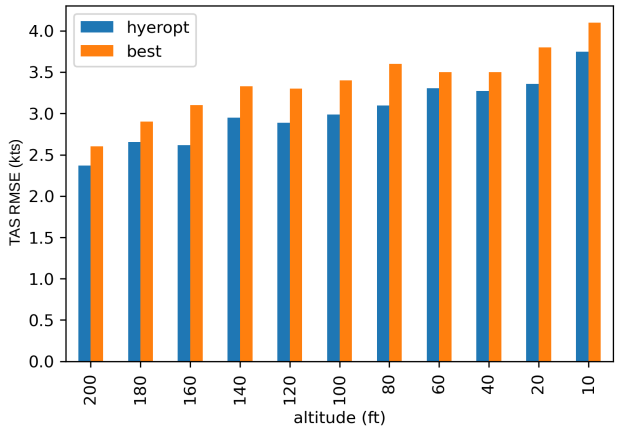

Figure 6.29: Hyerparameter optimized neural network model for TAS, test prediction RMSE distribution

altitudes indicating that the optimized algorithm was able to interpret the dynamics of the aircraft TAS better. The improvement from the hyperparameter optimization is small, limited to about 0.3 RMSE.

The HGA optimized result for IVV and PTCH are shown in Figure 6.30. For the

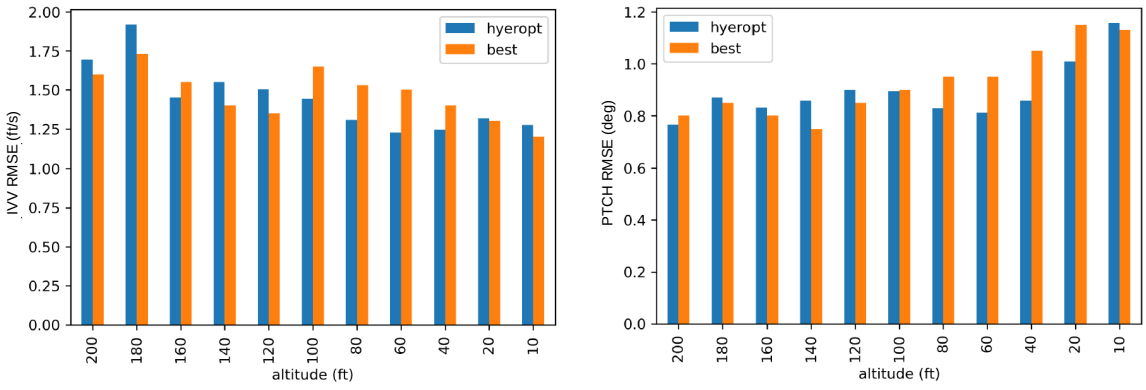

Figure 6.30: Hyperparameter optimized neural network model for IVV and PTCH, test prediction RMSE distributions

IVV prediction, there was no improvement through hyperparameter optimization. In fact,

the prediction closer to touchdown has lower accuracy for the hyperparameter-optimized model. This is possible because the objective function is wrapped into a single value, so the loss in the IVV was hidden behind the improvements in the prediction improvement of other features. This result is another indicator that IVV model is not much more sophisticated than a regular polynomial regression model. And with the current set of neural network model selections, improving the prediction result may be difficult. A similar conclusion can be made with the PTCH prediction. Although the hyperparameter optimized model did improve the prediction below 100 ft HAT, the improvement is only 0.1 degrees. Overall, it is difficult to conclude that the hyperparameter optimization was fruitful for IVV and PTCH features prediction.

So how does the optimized prediction model compares against its peer? Two different studies on prediction aircraft ground speed can be compared. The study by Puranik et al. [44] developed a prediction model with a prediction range of 300 ft (which is equivalent to approximately 18 seconds) and prediction RMSE of 2.98 kts. The study by Kang et al. [42] developed a prediction model with a prediction range of 3 seconds (which is equivalent to approximately 50 ft) and prediction RMSE of 1.71 kts. The optimized prediction model RMSE is on par with the one by Puranik et al. and higher than the one by Kang et al. However, the optimized prediction model predicts multiple points and multiple parameters. Both prediction range and multiple parameter prediction are identified to be crucial for increasing awareness of overrun precursors. So the results are promising to address the research objective of this study.

Training time and computational time are also important factors to consider when developing a neural network model with multiple layers. For training the prediction model the TensorFlow platform was used with GPU enabled. The Cuda toolkit was used to speed up the model training and the GPU used for training was GTX 2060. Based on this computational power the time it took to train a prediction model ranged from 2 to 10 minutes. Note a patience iteration of 10 was used which means the model training stopped early

when no improvement in the objective function was observed for over 10 iterations. So it may be difficult to train a prediction model with an entire fleet of flight data. However, since clustering similar flights showed potential for improving the prediction accuracy, it is encouraged that the flights be clustered in detail before training. Clustering criteria may include airframe, runway, autopilot, trajectory, weather, and more. When it comes to predicting the flight states the computational time for predicting 829 test flights was 0.53 seconds. So deploying the prediction model in flight is feasible.

6.7 Prediction Model Sensitivity

The downside of a neural network model is that it is considered a black box because the exact function for calculating the output is unknown. Since a neural network is considered a black box, it is important to carefully evaluate the limits of the model. In this section, the limits of the model will be evaluated under two conditions. First is the model sensitivity to large maneuvers, such as flare. Based on the model prediction results explored above, the prediction model accuracy reduces with increasing prediction range. This is expected because with an increased prediction range, the likelihood for external factors to influence the current trajectory increases. Pilot control is an external factor that the prediction model cannot account for due to the unique behavior exercised by the pilots each time they fly an aircraft. Ideally, with a robust prediction model, the prediction accuracy is robust to the pilot controls applied after the prediction has been made. One of the largest maneuvers performed by the pilot during the final approach is flare. So the prediction accuracy sensitivity to flare is evaluated.

The second condition to evaluate the limits of the prediction model is wind, more specifically large wind. Just like pilot control, the wind is an external factor that the prediction model could not account for when the prediction has been made. This is because the exact wind speed cannot be predicted ahead of the touchdown. So the sensitivity of the prediction model to various wind conditions will be evaluated to identify the limits to the prediction

model. The reason why only these two conditions are considered for sensitivity analysis is that pilot control and wind speed are data available in FOQA. This does not mean these two are the only external factors that can reduce prediction accuracy. Another external factor that cannot be evaluated in this study is the weather. Future studies should try to integrate other sources of data to evaluate the model performance at a deeper level. But external factors are not the only source of error for the prediction model. As shown in Figure 6.6, data structure intrinsic features may impact prediction accuracy. The same goes for the model's intrinsic features. Since the focus of this study is on utilizing FOQA data to improve runway overrun, the prediction model sensitivity will be evaluated based on what is provided by the FOQA data.

6.7.1 Significant Parameter Identification

Evaluating the limits of a neural network model starts with understanding the features of significance. The exact function of the neural network black box may be difficult to identify. However, the first-order terms that make up the black box function may be identified from a permutation of importance. The key idea behind permutation importance is that the permutation feature decrease in a model score when the values in a single feature are randomly shuffled [166]. This means the drop in model score represents the model's dependence on that feature. This also means if the model score does not change even after shuffling a feature, then the model is independent of that feature. An important consideration to be made is that the permutation importance depends on the model's accuracy. So a feature with low importance in a bad model may be very significant in a good model. So the permutation importance is measured with the hyperparameter-optimized prediction model. And even so, the actual measurement of the significance will not be discussed in detail unless a feature has dominating significance. The permutation importance will mainly be used to identify the main features that affect the predictability of the aircraft states and whether they are the expected features.

The permutation importance plot for the hyperparameter optimized prediction model is shown in Figure 6.31. This plot shows the importance of each model feature by the mean

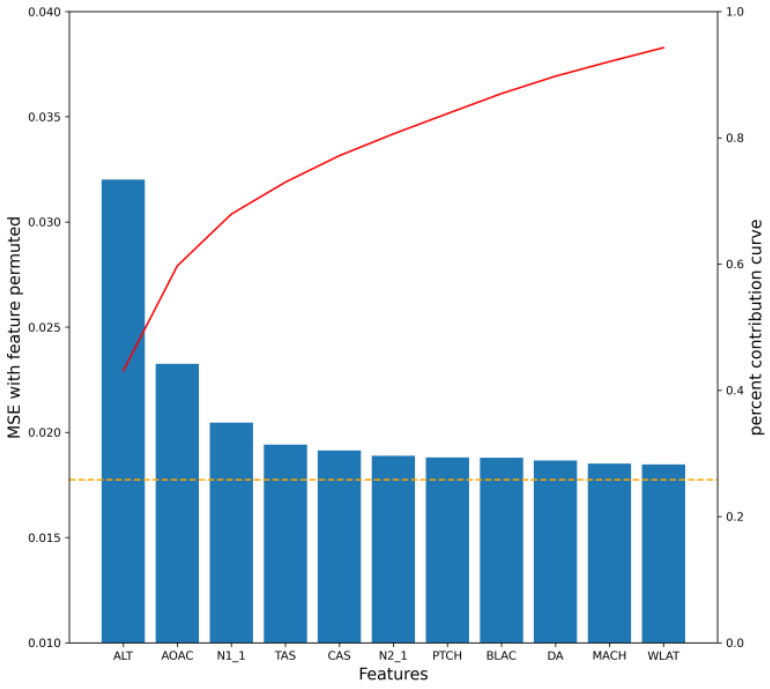

Figure 6.31: Permutation importance plot of the top 11 model features for the hyperparameter optimized prediction model, the cumulative percent contribution curve (red) of the model features based on their importance and the prediction MSE (yellow) at which the permuted feature would have no impact

squared error of the prediction result. The higher the MSE represents a higher significance because the model error will increase when the given feature values are randomly shuffled. The yellow dashed line indicates the threshold at which the feature would have no importance to the model if the prediction MSE is on the dashed line. The red curve indicates the cumulative sum of the importance from the most important feature to the least. This is calculated by dividing the importance value past the yellow dashed line, for each feature, by the total sum of the importance values past the yellow dashed line. Note, to generate the permutation importance, the hyperparameter optimized model was retrained without the ALT parameter. Or else the result would significantly skew with altitude-related parameters being the dominating feature.

The result indicates that the altitude and the angle of attack trajectories have a contri-

bution of over 60% to the model result. This is an expected outcome because altitude is the main indicator identified for this study to represent the final approach progress. So shuffling this feature would be completely mixing up the final approach trajectory, which would make the input data meaningless. This is also why the polynomial regression model was generated only with altitude as input and still managed to predict aircraft states decently well. TAS and PTCH are two other prediction parameters that are identified as significant to the model. This aligns with how these features impact the final approach and overrun. TAS is a feature that is managed from the start of the final approach to ensure that the aircraft is at a safe energy level. Ideally, the TAS converges to a lower value at touchdown, which means depending on the TAS above 300 ft HAT, the change in TAS below 300 ft HAT may be known. The PTCH is a parameter that is also related to TAS because near touchdown aircraft is pitched up to reduce speed while increasing lift. The same goes for AOAC, which is the second most important feature. So the aircraft pitch above 300 ft HAT will have some correlation with the aircraft states near touchdown. $N1_2$ and $N2_1$ are the other two features with top importance to the prediction model. Engine parameters have a direct relationship to TAS and in some sense, these features represent the pilot control since they are directly controlled by the PLA. So the engine parameters should also have importance over the prediction model trained.

6.7.2 Flare

Based on the permutation importance analysis, the prediction model is behaving as expected. The first-order features indicate that the model is learning the dynamics of the aircraft. So how robust is the trained model to large maneuvers such as flare? Evaluating the prediction model sensitivity to a flare would mean that the flare maneuver should first be identified. Flare is the final pitch-up maneuver to help the aircraft smoothly land on the runway. To evaluate the model sensitivity to flare, the flare altitude for each flight must first be identified. For this study, the flare index is found by backstepping each final approach

trajectory from the touchdown.

To start, the first point before touchdown where the aircraft pitch crosses zero degrees is identified. This is because observation of the flight data shows that the aircraft that the data represents approach the runway with the nose pitched down. Since the aircraft has to touchdown in the pitch-up configuration, the point at which the pitch cross zero degrees should occur during the final pitch-up maneuver. From this point, the pitch rate trajectory is backstepped until the first time it crosses zero. This is the point where the pilot initiated the final pitch-up maneuver and the pitch rate has turned positive. Finally, the elevator trajectory is backstepped until the first the difference in elevator crosses zero. It is assumed that this is the point when the pilot initiated the final approach by changing the aircraft controls. An illustration of the process for finding the flare initiation point is found in Figure 6.32. Based on this process the distribution of HAT and TAS at which the flare is

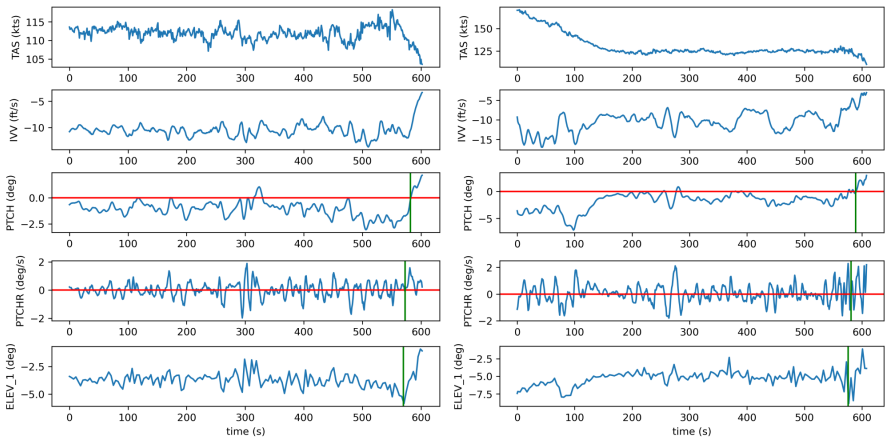

Figure 6.32: Flare initiation identification process, where the red lines indicate zero pitch and pitch rate and the green lines indicate the points at which pitch of zero, pitch rate of zero, and difference in elevator control of zero are crossed.

initialized can be plotted as shown in Figure 6.33. From this result, the mean of the flare initiation altitude is 80 ftHAT and TAS is 125 kts. This means the prediction accuracy above and below 80 ft can be compared to analyze the prediction model sensitivity to flare.

A larger view of the distribution of flight counts of flare initiation altitude in 20 ft intervals is shown in Figure 6.34. As shown, most of the flights initiated touchdown from

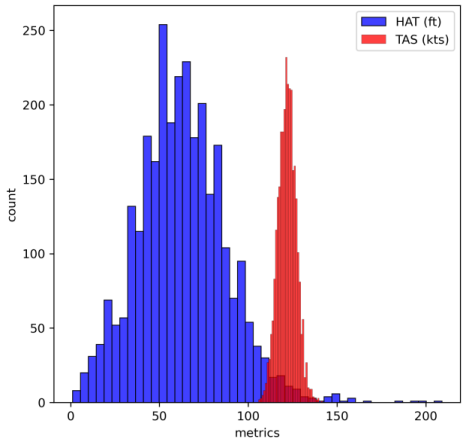

Figure 6.33: Distribution of HAT and TAS at flare initiation time

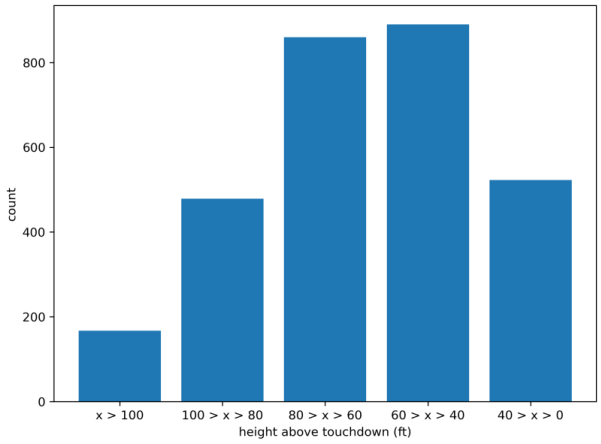

Figure 6.34: Distribution of flare initiation altitude in 20 ft interval

80 ft HAT. The Figure 6.35 shows the RMSE of the IVV predictions classified by the flare initiation intervals shown in Figure 6.34. As shown the blue and orange bars represent prediction RMSE calculated from data collected before (above) or after (below) the start of flare. The x-axis of the bar plot shows the set of flights that the start of flare altitude falls in. As shown, a consistent trend is observed for flights where the flare occurred below 100 ft HAT. For these flights, the prediction was better after the start of flare. This is expected because after a flare, the vertical speed is stabilized and the aircraft is ready for touchdown. So the aircraft is less likely to swerve up and down, making it easier to predict. The opposite is true for the flights where flare occurred before 100 ft HAT. For these flights, the pilots may have had to continue adjusting the vertical speed because the aircraft were far above

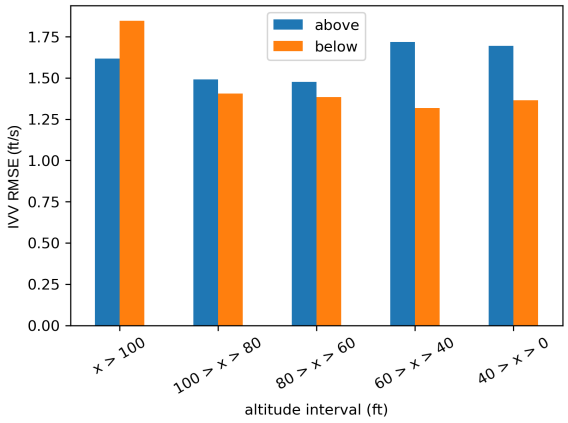

Figure 6.35: Prediction RMSE for IVV based on predictions above or below the start of flare and based on flare initiation altitudes

touchdown. Because of this, the error may be higher for predictions after the flare even though the final pitch-up maneuver was initiated.

The opposite behavior is observed for the TAS and PTCH RMSE as shown in Figure 6.36 For both metrics, the predictions after the start of flare had a higher error. The

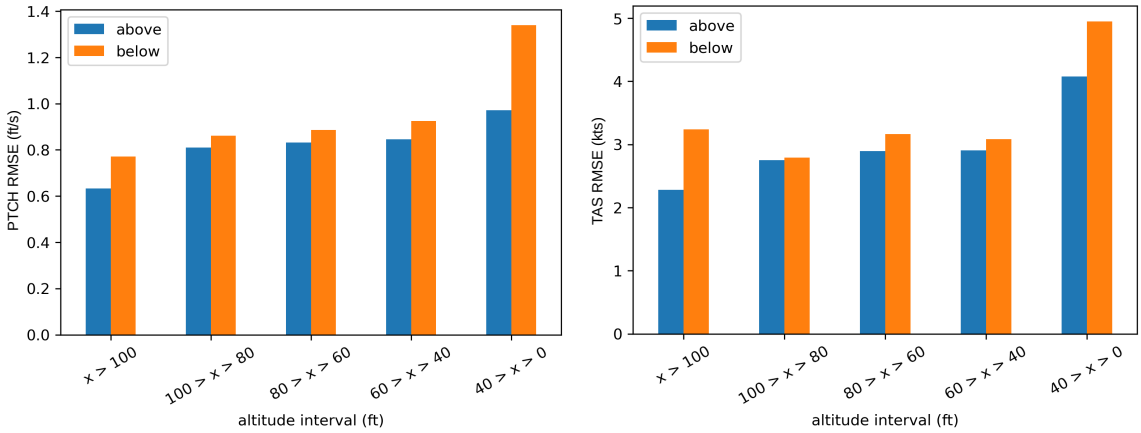

Figure 6.36: Prediction RMSE for TAS and PTCH parameters based on predictions above or below the start of flare and based on flare initiation altitudes

start of flare indicates an increase in lift and drag. While the increase in lift results in IVV converging to zero, the increase in drag simply reduces TAS. Compared to the final pitch-up maneuver, the adjustments made during the rest of the final approach are small. And depending on the pilot and the situation, the level of decrease in TAS will change. So

predictions after the start of flare may be harder with TAS. The same is true for the PTCH metric. The aircraft PTCH is a direct fall out of the level of flare performed. Depending on the landing condition and the pilots' experience, the desired PTCH to reduce TAS and increase lift would change. So PTCH prediction would also be harder after the start of flare. Overall the results show that the prediction accuracy does vary with flare. So creating a model that does consider control trajectories may improve the prediction results. This step is not considered for this study due to the lack of flights with no errors in recording throughout the entire final approach.

6.7.3 Wind

Evaluating the sensitivity of the prediction model to wind is started by identifying good and poor wind conditions. A Survey of a pilot wind shear guide by FAA [167] showed that microbursts may change the horizontal speed between 10 to 100 kts with a median of 40 kts. The aviation organizations also record large changes in wind direction for 45 degrees or more in less than 15 minutes, with wind speeds over 10 kts. These findings point to two wind conditions to be considered for evaluating prediction model sensitivity. Only 4 flights satisfied the horizontal speed of larger than 40 kts and only 60 flights satisfied the change in wind direction of 45 degrees in less than 15 minutes. Since the purpose of this study is to evaluate the model sensitivity to the wind conditions, these conditions were relaxed to increase more flights. 199 flights were identified to have horizontal speeds larger than 20 kts at one point in the final approach. 197 flights were identified to have changes in direction of larger than 10 degrees within 15 minutes and with higher than 10 kts of speed for the duration. In terms of the test data the number of flights with normal wind, strong longitudinal wind, large wind direction change, and both wind conditions are 381, 31, 17, and 67.

Once the flights have been labeled with one of the four wind conditions, the RMSE metric for each predicted parameter was measured. The calculated results for the IVV

predictions are shown in Figure 6.37. The IVV prediction RMSE shows a consistent dis-

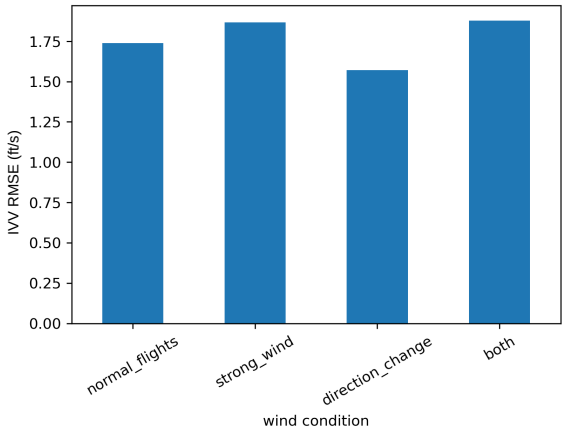

Figure 6.37: Prediction RMSE for IVV based on predictions different wind speed conditions

tribution for various wind conditions. In fact, the error is higher for flights with large wind direction changes. An important factor to remember is that although flights with strong longitudinal wind and large wind direction change conditions are identified, the threshold for these constraints was relaxed. This means the contribution from these conditions may be small. Especially wind direction change conditions. Because unless there is a large crosswind or change in the crosswind, the wind direction alone does not contribute much to the longitudinal dynamics of the aircraft. This is why the prediction accuracy of flights under direction change is similar to that of flights with the normal wind. However, this is not the case for the strong wind because the longitudinal wind was used specifically to label flights with this condition. A strong wind will directly impact the lift which directly impacts IVV. As expected the prediction error of strong wind flights is higher than that of normal wind flights. In fact, the prediction error of strong wind flights is almost identical to the at of flights with both strong wind and large direction change. This may be because direction change has no impact on the prediction accuracy of IVV metric so both conditions are mainly driven by the strong wind condition.

The wind conditions are observed to have a larger impact on the PTCH and TAS metrics. The sensitivity of PTCH and TAS metrics prediction error to wind condition is are

shown in Figure 6.38. Both strong longitudinal wind and large wind direction change have

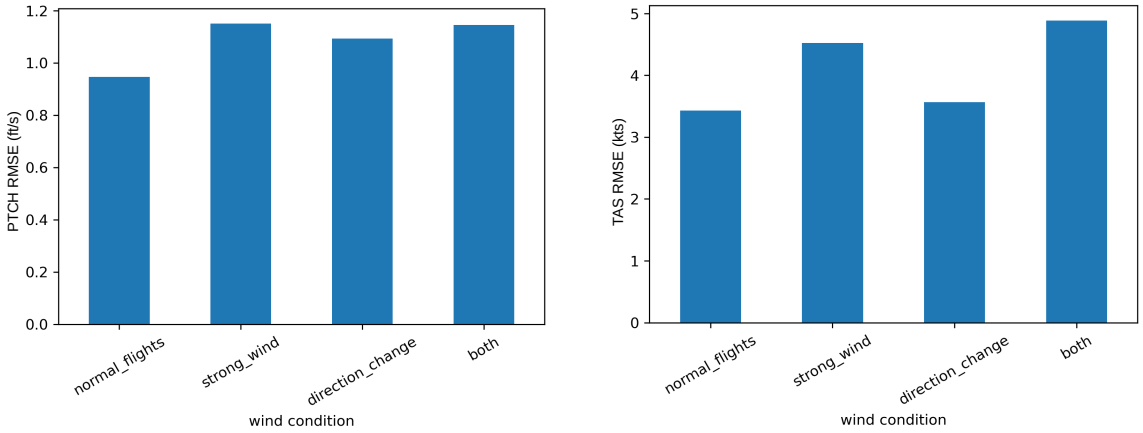

Figure 6.38: Prediction RMSE for PTCH and TAS parameters based on predictions different wind speed conditions

a large impact on the prediction accuracy of the PTCH metric. The effect of strong longitudinal wind on PTCH is identical to the effect on IVV feature. However, PTCH is also influenced by wind direction change and an explanation of this behavior is the coupling effect of aircraft attitude and attitude rate terms. The dynamical equation of aircraft pitch rate, assuming symmetry in the xz plane, is shown in Equation 6.1.

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{6.1}$$

Here $\dot{\theta}$ is the pitch rate in the earth frame, q is the pitch rate in the body frame, ϕ is the roll, and r is the yaw rate in the body frame. So a change in wind direction that would change r or ϕ would affect PTCH. The decrease in prediction accuracy, when both strong wind and large wind direction change occurred, is the largest. Similar behavior is observed for the TAS feature. However, the effect of wind direction change has less of an impact on TAS than PTCH. The decrease in prediction accuracy is mainly driven by the longitudinal wind which directly impacts TAS. Again the largest reduction in prediction accuracy is observed when both strong wind and large wind direction changes are observed. Overall, because the prediction metrics are mainly represented by the longitudinal dynamics, the predic-

tion model is sensitive to longitudinal wind. When a large longitudinal wind is observed, there is a drop in the prediction accuracy. A potential explanation is that the aircraft states closely associated with runway overrun are associated with the longitudinal dynamics of the aircraft. So the longitudinal wind speed should be considered when using the prediction model to identify potential risks in flight.

The findings and contributions from this chapter can be summarized as follows:

1. HGA was able to optimize both data and model parameters for the flight state prediction model

2. The neural network prediction model was able to learn the behaviors of airspeed features near touchdown from the final approach data

3. The neural network prediction model performed similarly to a polynomial regression model for vertical speed and pitch angle prediction

4. Since the prediction model is a complex regression model, the prediction result may improve by grouping similar flights (such as autopilot)

5. The flight features with the top contribution to airspeed, vertical speed, and pitch angle near touchdown are altitude, angle of attack, engine core rpm, and airspeeds

6. Sensitivity analysis showed that flare and wind have a small contribution to prediction accuracy

CHAPTER 7
DEVELOPMENT OF DYNAMICS MODEL FOR FINAL APPROACH SIMULATION

A simulation model was proposed as another attempt to increase the pilots' awareness to risk during the final approach. Simulation models developed based on the physics of the aircraft already exists and are in use for training pilots. These simulation environments help enhance the pilots' attitude, skill, and knowledge, also known as ASK, of flight risk without flying a real aircraft. So what about after the flight? What if pilots can get feedback on past flights and modify them to enhance their knowledge of risks during the final approach? And what if this feedback can help pilots gain insight into how external factors, such as strong wind, affect aircraft dynamics? It would tie back to the global initiative GAPPRE, which mentioned a need to enhance energy management during the final approach. This would require the pilots' understanding of aircraft interaction with external factors. This requirement can be satisfied with an aircraft dynamics model that accurately models the aircraft's behavior during the final approach. The problem is modeling such behavior requires a complex model that can incorporate the coupling effect between various controls. These coupling effects are small compared to the primary effects that each control has on the aircraft dynamics. So the data used to train this complex model also needs to represent these coupling effects. The complexity of the dynamics model and the level of detail in the data required makes this a large design space problem. In this chapter, a framework is proposed to create an aircraft dynamics model that can simulate the aircraft motion during the final approach accurately while considering the wind effect. A summary of the framework is illustrated in Figure 7.1.

The framework is designed to create an aircraft dynamics model that can simulate the final approach trajectory given the initial states and control trajectories. The model is trained

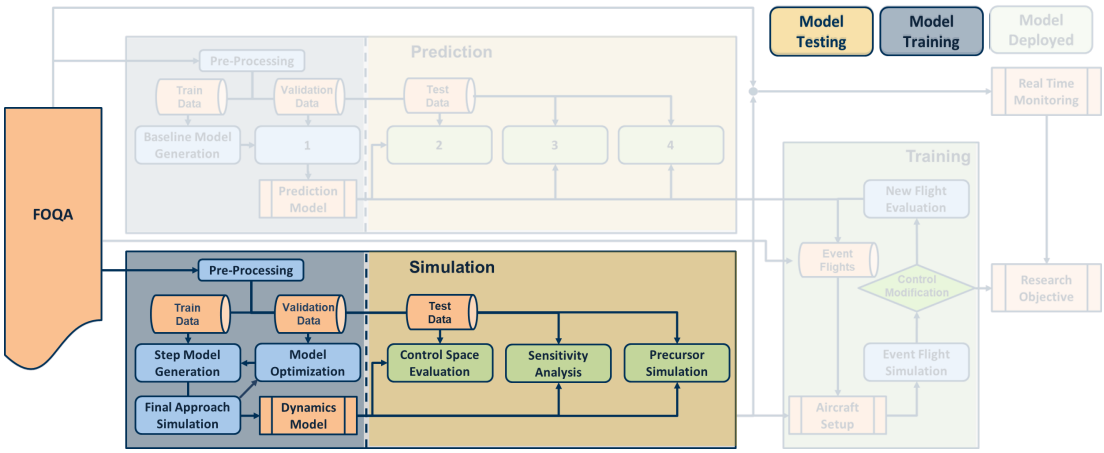

Figure 7.1: Dynamics model development process overview

to predict aircraft states (x) at $t+1$ given $x(t)$ and $u(t)$, where u is the aircraft control. This process is repeated until the control is no longer provided. So the model is trained based on flight data recorded at each time step. However, the simulation is performed over the entire final approach trajectory. An illustration of this framework is shown in Figure 7.2.

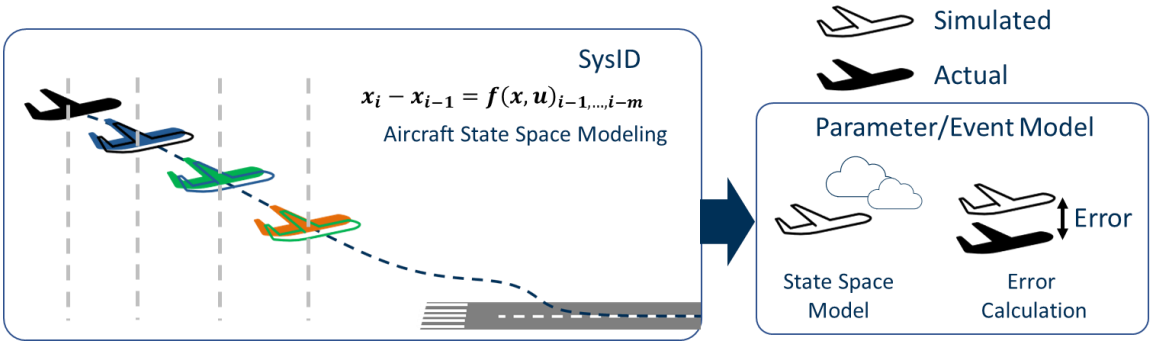

Figure 7.2: Dynamics model architecture

7.1 Data Pre-Processing

A similar data preprocessing step is taken for the prediction model generation. For the dynamics model training, flights with missing steps were kept. However, the flights were split at the points with missing recordings. This is because the dynamics model is trained

in an open loop, with each data step used as input for estimating the data at the next time step. So, the full final approach process for a given flight is not necessary as long as the aggregate data represents the dynamics at the majority of the final approach. Based on this logic there were a total of 4128 flights available for training and testing the prediction model. When these flights were divided into time steps, over 100,000 data steps were collected to train the dynamics model. The counts of control features recorded for the entire final approach data are shown in Figure 7.3. The abbreviations used for the flight control metrics are shown in Table 7.1. This data distribution gives a summary of how the aircraft

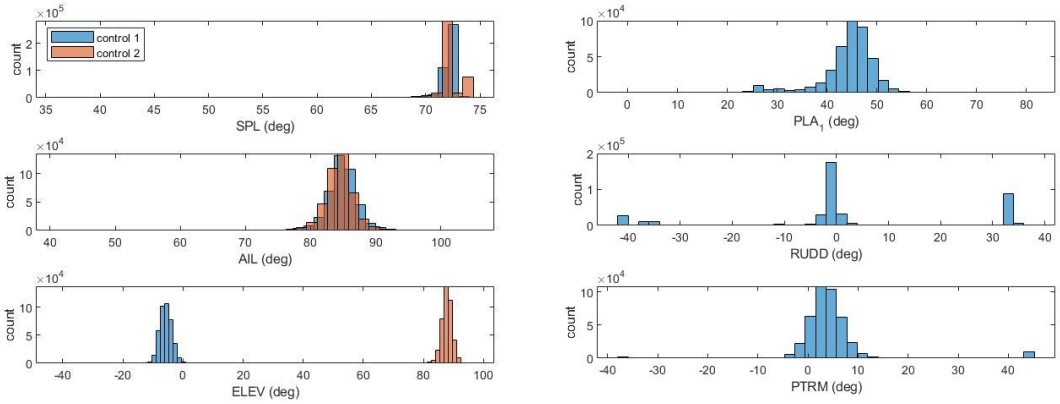

Figure 7.3: Aircraft control parameters recorded for all of the final approach data. Here controls 1 and 2 represent two different surfaces for each control.

maneuvered touchdown. The first and second control for the spoiler and aileron represents the left and right control surfaces. For both, the mean of the left and right controls are aligned, indicating that most of the flights were able to stabilize by the time they initiated the final approach. The same can be implied from the rudder deflection where a large peak in the recording is observed at zero degrees deflection. Just based on this observation, it is assumed that the dynamics model generated can only be accurate for longitudinal dynamics and not lateral dynamics. So this study will emphasize simulating the final approach for the longitudinal motion. The elevator control had two surfaces that represents the elevator (blue) and the pitch trim tab (orange) of the aircraft. The concentration of the elevator recording at the negative indicates that the aircraft was trying to pitch down for most of

Table 7.1: List of all of the flight metrics, used for dynamic model training, and their abbreviations

No.	Abbreviations	Definitions	$units$
1	SPL	Spoiler deflection angle	deg
2	AIL	Aileron deflection angle	deg
3	ELEV	Elevator deflection angle	deg
4	PLA	Power lever angle	deg
5	RUDD	Rudder deflection angle	deg
6	PTRM	Pitch trim angle	deg
7	HAT	Height above touchdown	ft
8	DWPT	Distance to touchdown	ft
9	GS	Ground speed	ft/s
10	u	longitudinal speed (body axis)	ft/s
11	w	vertical speed (body axis)	ft/s
12	uacc	longitudinal acceleration (body axis)	ft/s^2
13	wacc	vertical acceleration (body axis)	ft/s^2
14	ROLL	Roll attitude	deg
15	PTCH	Pitch attitude	deg
16	DA	Drift angle	deg
17	ROLLR	Roll rate	deg/s
18	PTCHR	Pitch rate	deg/s
19	YAWR	Drift rate	deg/s
20	WLAT	Lateral wind speed	kts
21	WLONG	Longitudinal wind speed	kts

the flights, which is expected for a data set that represents the final approach. Most of the flights were flying with low thrust based on the low power lever deflection angle observed, which is also expected for a data set that represents the final approach.

For this study, each control trajectory was scanned to identify the points where missing steps or shifted readings occurred. Each flight data was spliced at these points and so that the segments of flight with understandable recordings were kept. Once all of the flight data was cleaned, they were modified to create a dataset that is easier for the neural network model to interpret. To do so, longitudinal acceleration data was augmented by taking the step difference of the Ground Speed (GS) and IVV parameters. This was necessary because most of the acceleration parameters had shifted readings or missing steps. The reason why GS and IVV parameters are trusted was that the integral (trapezoidal sum) of these parameters, from start to finish of the final approach, were less than 1% different from the change in position and altitude metrics. However, the first and second integrals of the

acceleration parameters, from start to finish of the final approach, did not match the two velocity and position metrics. The wind speed parameters were augmented so that the longitudinal and lateral wind speeds were calculated from wind speed and wind direction recordings. This was so that the sin and cos relationship between wind speed and aircraft heading does not have to be considered, and the complexity of the dynamics model may be reduced. The fuel quantity metric was augmented by taking the cumulative sum of the fuel flow metric. This was because the fuel quantity metric cannot be trusted due to the significant noise from the sloshing effect. So instead the initial aircraft weight for each flight was calculated by taking the average fuel quantity two steps before and after the initial weight. The calculated initial weight is then added to the cumulative sum of the fuel flow at each time step. Finally, the aircraft velocity and acceleration states were transformed into a body-fixed reference frame. The transformation was based on the Euler angle rotation matrix shown in Equation 7.1.

$$R_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7.1a}$$

$$R_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \tag{7.1b}$$

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \tag{7.1c}$$

$$V_B = R_x(\phi)R_y(\theta)R_z(\psi)V_E \tag{7.1d}$$

$$\tag{7.1e}$$

Aircraft velocity and acceleration states in the earth frame were transformed into the body

frame based on the improvements observed from experiments.

The modified flight data is then indexed so that the data structure matches the neural network requirement. The input and output data structure is illustrated in Figure 7.4. Here n is the input data window length and n+1 to n+p is the output data window length. The input states and controls are used to predict the output states at the next time stamp. Then the output states were used as the next input states and this combined with the next input controls was used to predict the next output states. Each set of input, output states, and controls was considered as a single sample. This process of sampling and collecting the data was repeated with all the flights for the entire final approach. The data structures

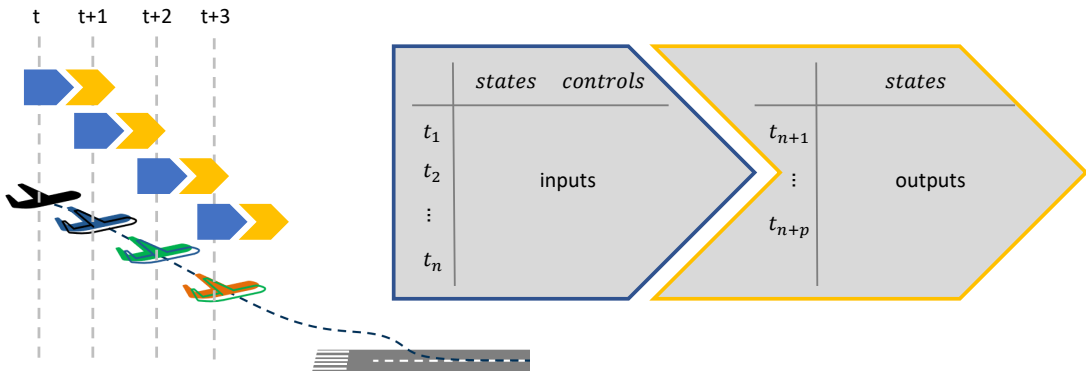

Figure 7.4: Data structure for the dynamics model input (blue) and output (yellow)

were modified with the following data parameters. For this study, the input and output data window lengths were varied between 1 to 10 and 1 to 5 seconds. Input data window length was varied up to 10 seconds to ensure that there was enough data representing the low-frequency response of an aircraft. Output data window length was varied up to 5 seconds to test and see if the extra data can improve how the model learns the dynamics from the flight data. The resolution varied between 0.5 and 2 Hz. The higher resolution was not considered since the highest frequency that all the features recorded was 2 Hz. The last data parameter was normalization. Just like the prediction model, the data were normalized either not normalized, normalized by the min-max range of each feature, or normalized by

calculating and converting each value to their corresponding z-score.

For the simulation, the following neural network models were considered: RNN, GRU, LSTM, and NNARX. When testing each of these models some of the model parameters were modified to identify the ideal model configuration to represent aircraft dynamics. The activation function was selected between linear, relu, tanh, and sigmoid functions. The activation function was varied for each layer instead of sharing the same function. The number of hidden units for each model layer varied between 10 and 30. The larger the number of hidden units, the more complex the model can be. However, a higher number of hidden units was not used experimentation showed a small change in the prediction accuracy. The training batch size varied between 8, 16, 32, and 64. For RNN, GRU, and LSTM models the number of prediction layers varied between one to four. drop out percent between each layer varied between 0 to 50%. These parameters were not modified for the NNARX model because this model is a two-layer model with the first layer being the hidden units and the second layer being the output.

This study used the same set of data as the prediction model to develop the simulation model. Out of the 5376 flights, a total of 4045 flight segments were collected after splicing the data at the points where missing steps and shifted readings in control features were observed. Note each flight segment had to be longer than 15-time steps so that when the max number of input and output window lengths were selected, the segment was long enough. From these flight segments, 169764 samples were collected with the aforementioned processes, at the resolution of 1 Hz. Since each sample represents one-time step, the number of samples would multiply with the resolution frequency. For the dynamics model training, 90 % of the data was used for training, 5 % for validating, and the last 5 % for testing the model. All of the results discussed in this study are generated based on the 5 % of the test data set.

7.2 Base Model Structure

Base simulation models were created with each of the neural network models selected. RNN and GNN were both selected because the impact of diminishing or exploding feedback on the model has not been evaluated. Since for each sample the input data only contains a short segment of the flight, the gates are not expected to have a significant impact on the convergence of the dynamic model. This is especially true the shorter the input window length is. And for the base simulation models, the input window length of five was selected. A longer input window length was not selected to simplify the base model since the goal of this step was to evaluate various neural network models. The output window length of one was selected so that only the next time step is predicted. The resolution was set at 1 Hz and min-max normalization was selected so that all the recordings normalized to one. For both RNN and GNN two-layer models were used to match the model depth to NNARX. Another reason why two-layer models were used is that, as discussed in subsubsection 2.2.2, the complexity of a general aircraft equation of motion is of second order. Considering that a simple two-layer Feed Forward (FF) network with linear activation function can represent up to second-order terms, two-layer RNN and GNN were considered complex enough to model the motion of an aircraft. The dropout of 25% was selected since this is the center of the dropout range considered. For both RNN and GNN, the tanh activation function was selected to match with the default NNARX activation function. The number of hidden units was set at 200. Finally, the training batch size of 64 was selected to speed up the training. Matlab deep learning toolbox was used for developing and testing the neural network models.

All of the simulation data would range between 1000 ft and 5 ft HAT. Because the ROD of each flight is different and some of the flights are spliced, the length of each flight varies. The objective function used for designing the simulation model was the RMSE of the predicted and actual states of all the features. The ideal model is selected based on

step-wise prediction accuracy. Because the entire simulated trajectory is expected to be closer to the actual trajectory with higher step-wise prediction accuracy.

7.3 Initial Model Evaluation

The trained dynamic models were used to simulate the aircraft states from the start of the final approach to the touchdown with test data. The accuracy of different models was compared mainly with two metrics. The first is the MSE of the state estimation results and this is the metric used to train the neural network models. The second is the correlation coefficient R. R is calculated with test data after the model training and it is not used in the training process. Both of the metrics are calculated with the residual of prediction in each step. Note that the flight data is normalized by feature so the magnitude of the features does not contribute to the performance comparison.

So based on this information the NNARX model was trained and results were generated as shown in Figure 7.5. Two sample flight trajectories simulated with the trained NNARX model are shown in Figure 7.6. The plot to the left of Figure 7.5 shows the actual vs

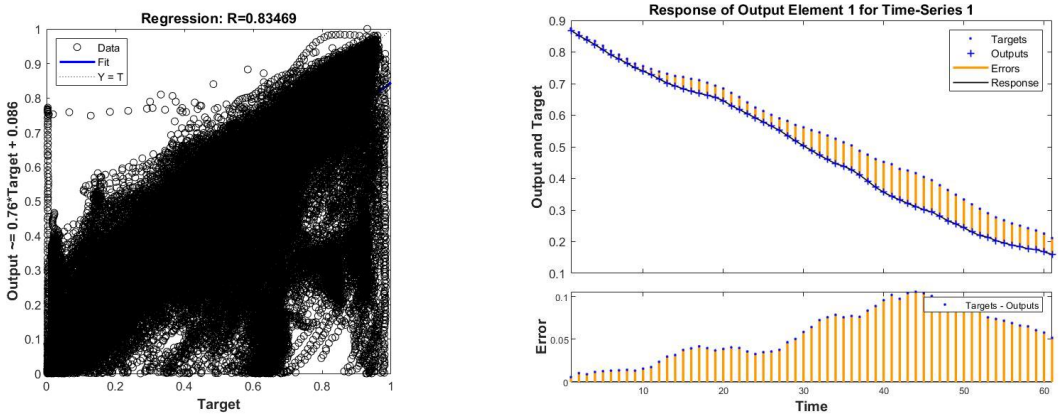

Figure 7.5: Summary plots of the NNARX based simulation prediction results

predicted plot of the step-wise prediction of all the steps in the test data. Each point on the plot represents the mean of all the features for one step. As shown, the distribution of data in the scatter plot is very scattered. This indicates that the model was trained to the

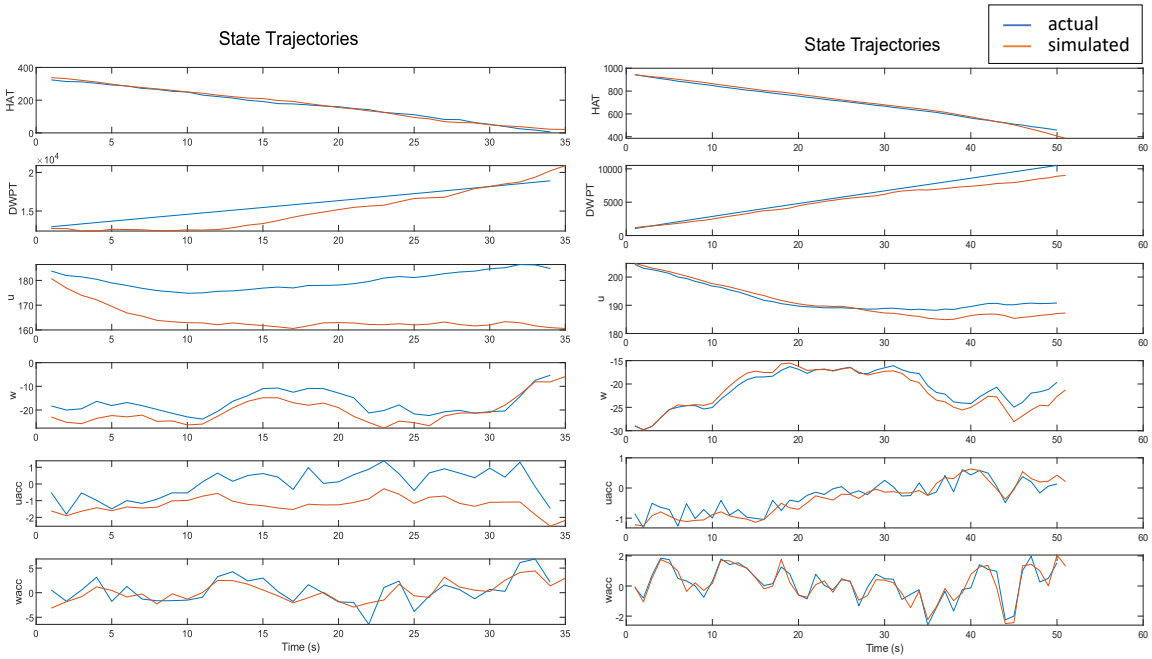

Figure 7.6: Trajectory plots of the NNARX based simulation prediction results

best of its ability but the model was not sophisticated enough to represent the data that it is trained with. The corresponding R-value is 0.8347 add the model MSE is 0.00034 at with the best epoch. The plot to the right shows the simulated response trajectory (blue cross) for one sample flight from the start to the end of the final approach. The black dots represent the target states and the yellow vertical line represents the residual of predicted vs actual states. Essentially the plot is a time series response of predicted response and target response of a flight. What this plot shows is that the simulation error starts out small but increases over time, which is expected behavior. Two example flights shown in Figure 7.6 show the trajectories of six states: HAT, DWPT, u, w, uacc, and wacc. The two sample flights indicate that the model is capable of simulating short trajectories but over time the residual does accumulate. The accumulating residual does sometimes result in a significant error as shown in Figure 7.7. These plots are examples of exploding feedback errors for the NNARX model. In essence, the NNARX model is a stepwise feed-forward model with a hidden layer and an output layer. The model lacks the ability to model nonlinearity in

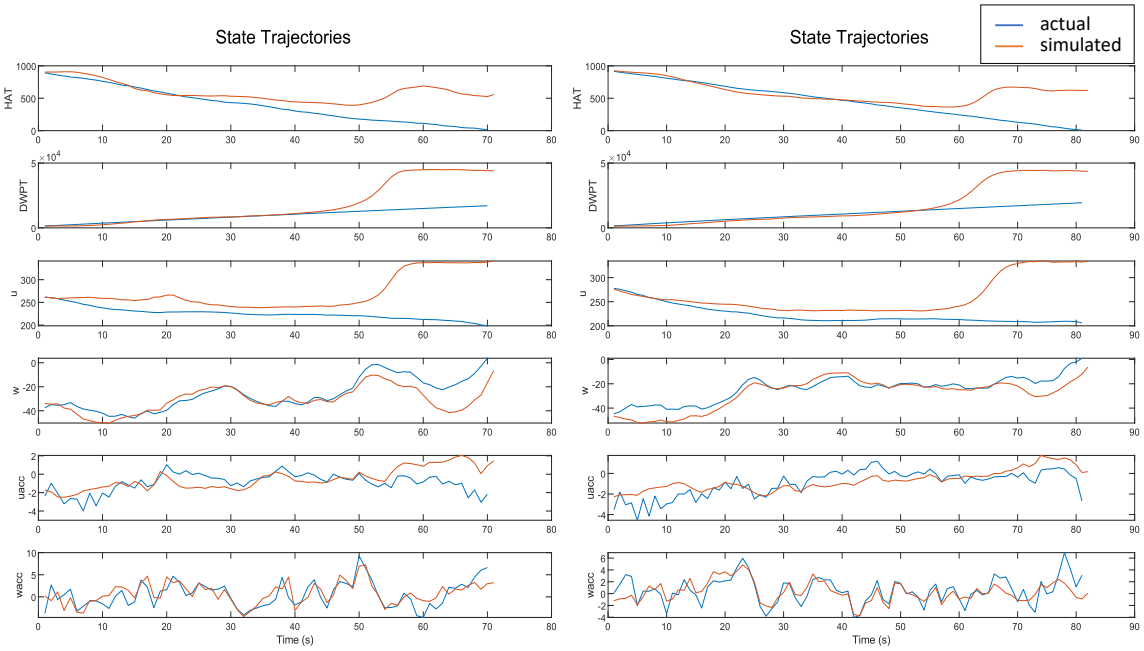

Figure 7.7: Summary plots of the NNARX based simulation prediction results

the aircraft dynamics and fails after a small deviation from the states represented by the data. The plots also indicate the NNARX model's lack of robustness to external factors that would deviate the trajectories from the expected path. Based on this information the NNARX is not the best selection since the goal of this research is to refly the flights to learn the risk so the consistent result is needed and the model cannot generate consistent results.

The next model trained was a RNN based model. This model has a slightly different architecture than NNARX model. It adds a feedback loop to the hidden layer of the NNARX model instead of the entire two-layer network. Because of this RNN is much simpler and cannot represent the second or higher-order terms. The simulation performance of the RNN model is shown in Figure 7.8. The trajectories of two test flights are shown in Figure 7.9. The left plot of Figure 7.8 indicates that the RNN is insufficient for modeling the dynamics of an aircraft. The correlation coefficient value is 0.714 with the best model MSE of 0.0176. The first sign is the vertical boundaries to the left and right of the scatter plot. These are due to the leveling of the data at the two extremes of the hyperbolic tangent

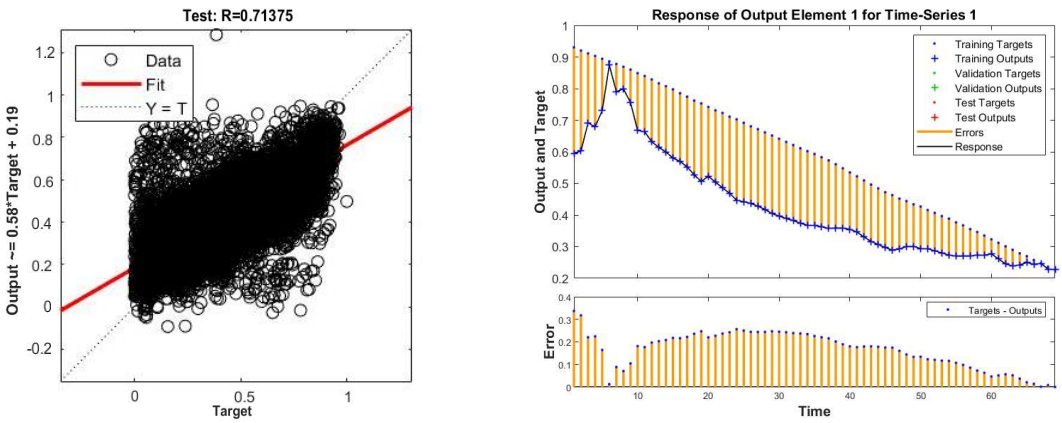

Figure 7.8: Summary plots of the RNN based simulation prediction results

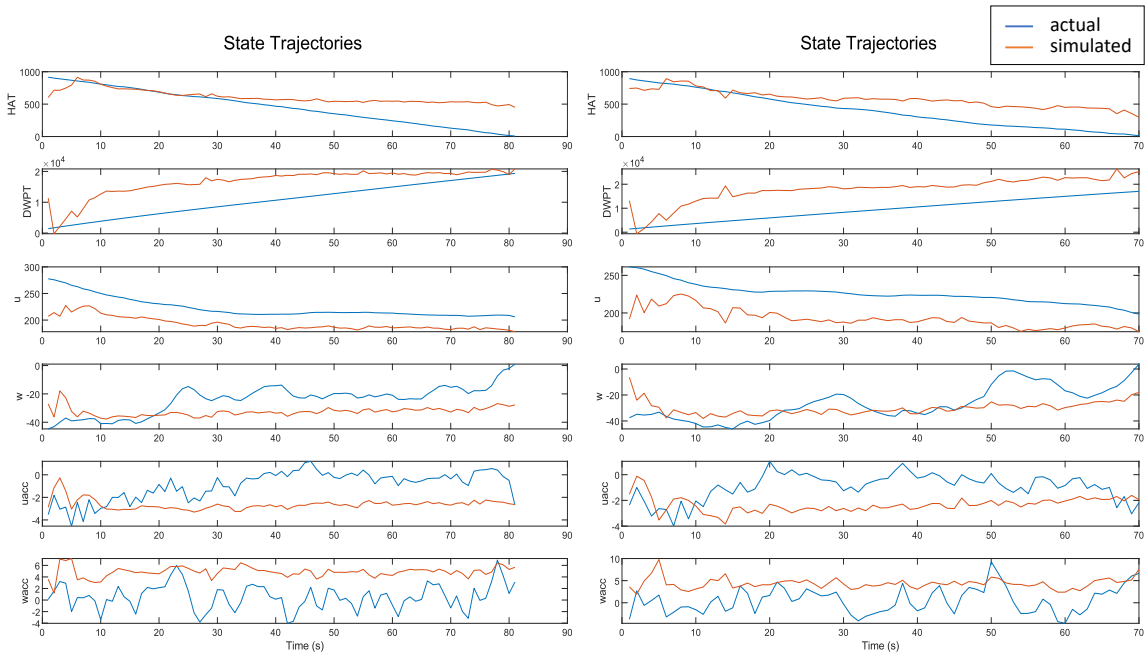

Figure 7.9: Two sample plots of the RNN based simulation prediction results

activation function. This behavior indicates that the model training process either did not have enough data points, iterations, or model complexity so the weights and biases could not improve to fit the data. The second sign is the consistent spread of the data points above and below the best fit line. If the scattered plots formed a shape or curve then it is an indication that a higher-order term is required to fit the data. The consistent spread is a sign that the model is fitted to the data. So an alternative model is required to fit the data better.

The RNN model's lack of complexity is also shown in the actual and predicted response by time plot, to the right. The actual and predicted outputs over time plot shows that the model is consistently generating outputs far from the target, just like a data point on the scatter plot that is far away from the best fit line. The simulated trajectories of the RNN model shown in Figure 7.9 speak the same story. In fact, the right sample trajectories represent the actual and predicted response by time plot.

So far NNARX and RNN models were trained and tested and found that these models are not complex enough for an aircraft system. GNN are trained to determine if they are sufficient for modeling flight behavior and the first model trained is the LSTM based model. The performance results are shown in Figure 7.10 and the sample simulated trajectories are shown in Figure 7.11. The performance metrics show a significant improvement from the

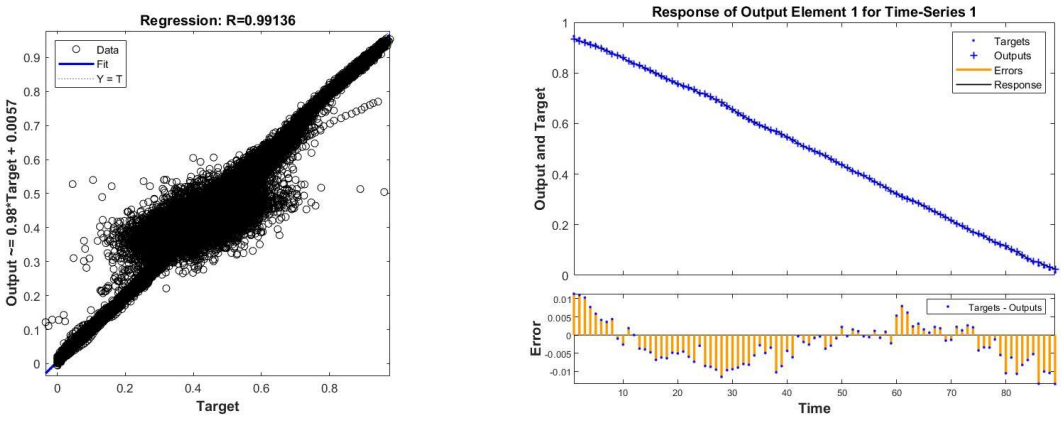

Figure 7.10: Summary plots of the LSTM based simulation prediction results

past two models. The correlation coefficient is 0.9914 with the best model MSE of 0.00029. The scatter plot of the actual vs predicted plot shows a significant improvement where most of the samples lie exactly on the best fit line. There are some dots scattered at the center of the plot with a fit line that seems to be a few degrees off from the best fit line. This indicates that either the selected model lacks complexity or the data does not sufficiently represent a switch in the flight condition. The actual and predicted outputs over time plot shows that for the sample illustrated there is no deviation in the residual and the simulation

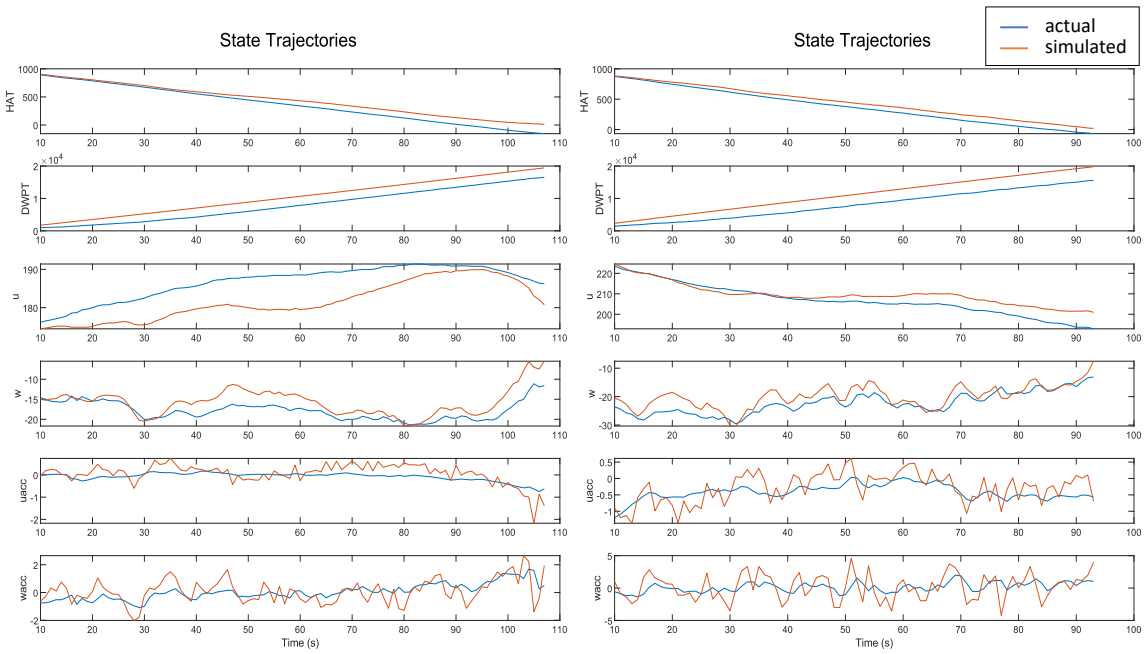

Figure 7.11: Two sample plots of the LSTM based simulation prediction results

stays accurate for the full final approach. Two test sample trajectories shown in Figure 7.11 also indicate that the model stays accurate for most of the final approach. However, there are some deviations in the longitudinal velocity parameter that may be causing the few data points in the target vs predicted scatter plot to diverge from the best fit line. The simulated longitudinal acceleration trajectories, in blue, also do not track the changes in the actual trajectories. The conversion between the earth and to the body reference frame may have resulted in this change and this issue will be addressed later.

Just like the improvements seen in the LSTM model, the same improvements are also seen in the GRU model. The performance results of GRU are shown in Figure 7.12 and the two test sample trajectories are shown in Figure 7.13. The correlation coefficient and best model MSE for the GRU model are 0.9911 and 0.00031. The exact same results are observed for the GRU model as the LSTM model. The actual to predicted scatter plot shows most of the data points lying on the best fit line. And just like the LSTM model, there are some samples in the scatter plot that generated their own fit line. The two test sample

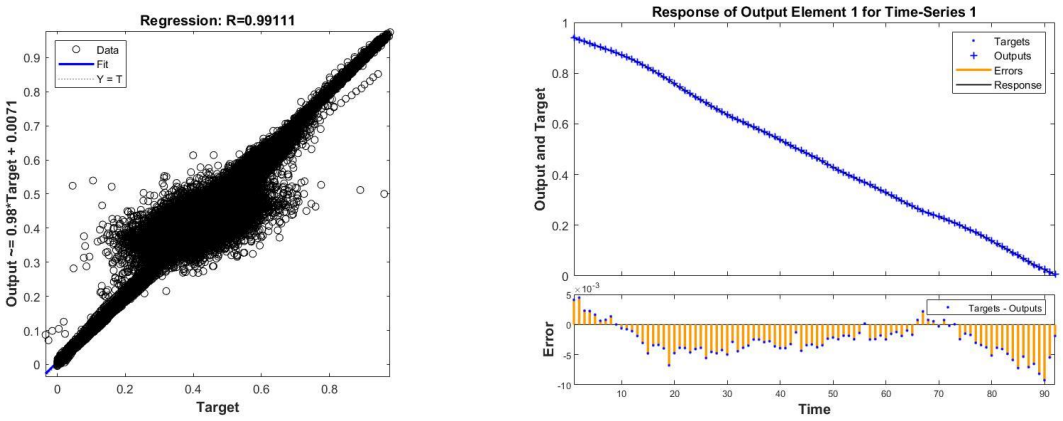

Figure 7.12: Summary plots of the GRU based simulation prediction results

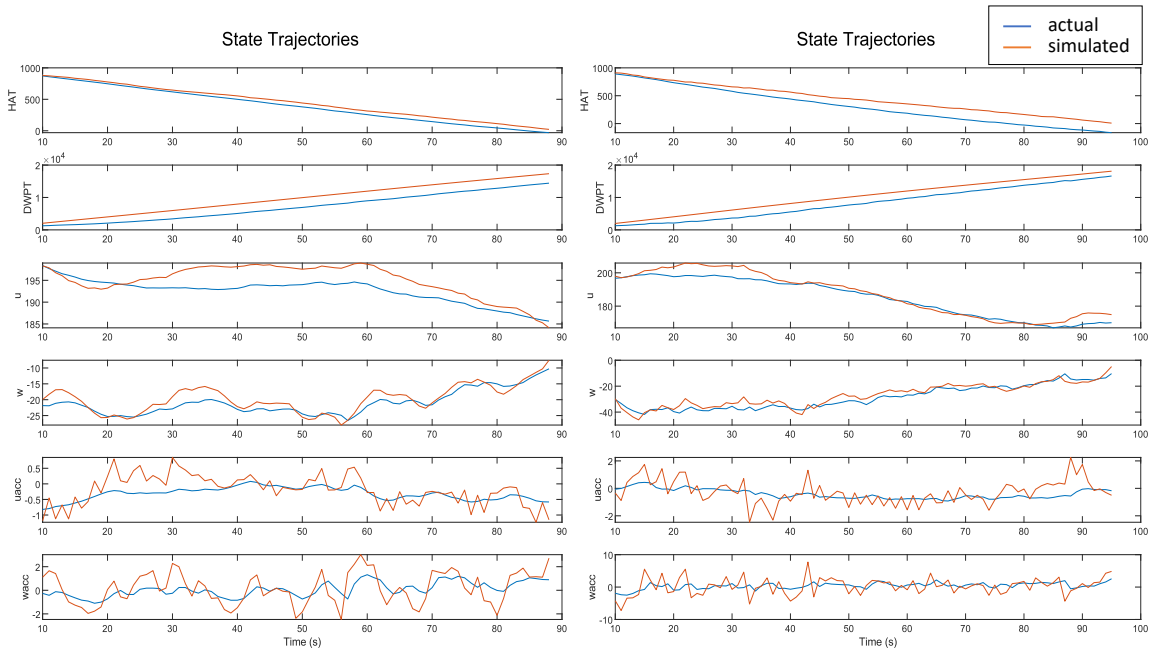

Figure 7.13: Two sample plots of the GRU based simulation prediction results

trajectories show that again these samples are most likely associated with deviations in the longitudinal velocity features.

Overall the GNN are sufficient in modeling the flight behaviors. However, some additional iteration is required to address the deviations with the longitudinal velocity and acceleration features. To do so a new LSTM model was created this time trained with flight data in the earth frame. The LSTM model was selected for the next iteration of training

since it had a slight edge in terms of the two performance metrics compared to the GRU model. The result of the new LSTM models is shown in Figure 7.14 and Figure 7.15. The

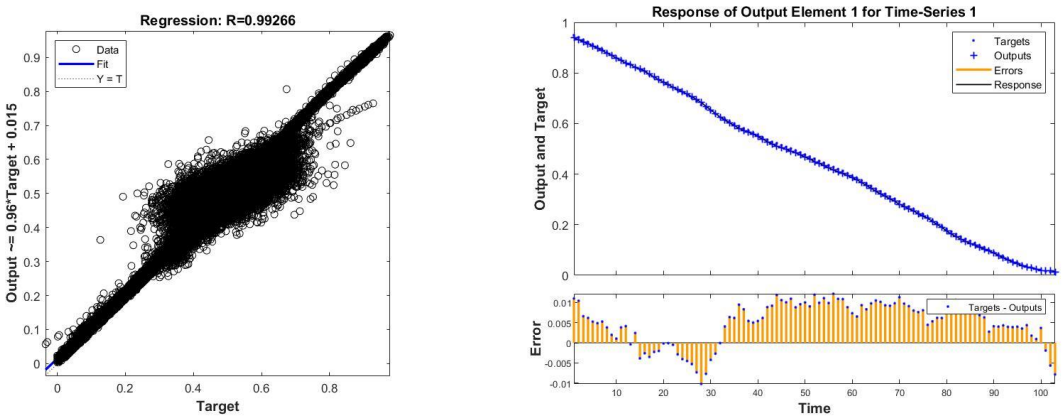

Figure 7.14: Summary plots of the LSTM based simulation prediction results trained with earth frame data

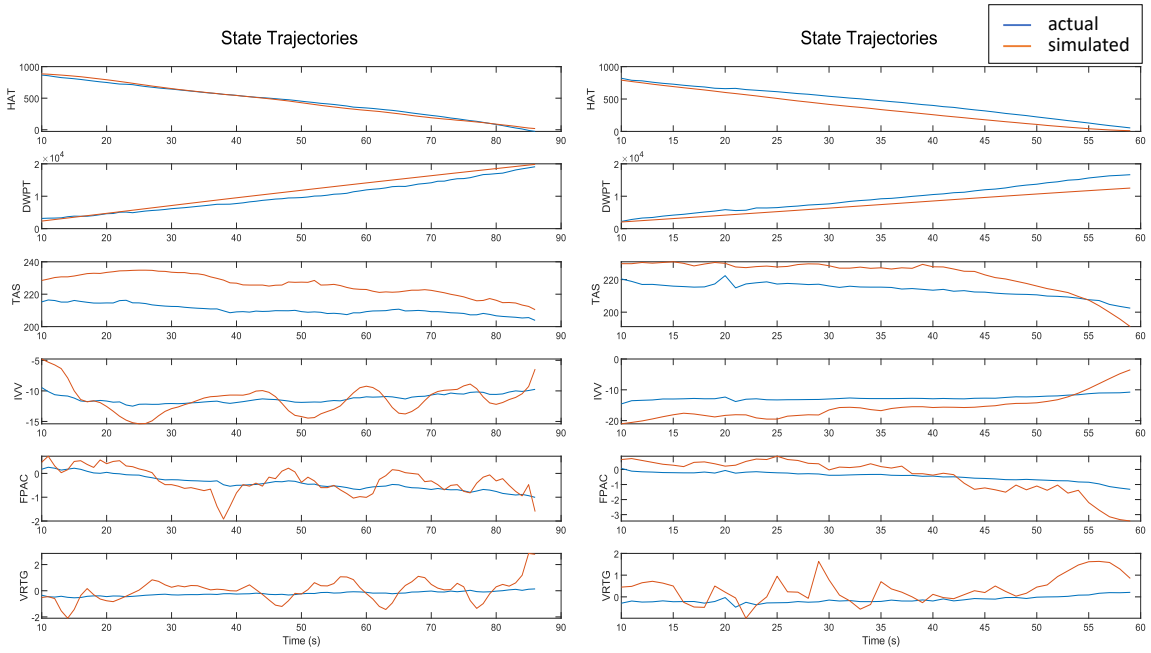

Figure 7.15: Two sample plots of the LSTM based simulation prediction results trained with earth frame data

correlation coefficient and MSE of the new model are 0.9927 and 0.00028. The correlation coefficient increased slightly and the MSE used for training improved significantly. The

increase in correlation coefficient is from the difference in scatter points near the middle of the actual vs predicted plot. Unlike the model trained with body frame data, the scattered samples of the model trained with earth frame data created a much more structured scatter of points that represents a block of points limited by the activation function, similar to the left plot in Figure 7.8. Even, though the two performance metrics show an increase in simulation model accuracy, the improvement is not clear from the sample trajectory plots Figure 7.15. With the earth frame data, the model was only able to follow the general trend. Small deviations in the TAS, IVV, FPAC, and VRTG are not captured by the simulation model. The result is still good considering that none of the test flights diverged significantly from the actual flight paths. Also, because the generic trend is followed, deviation in the scattered samples from the best fit line is limited by the fluctuations in the TAS, IVV, FPAC, and VRTG. This may have resulted in a more structured center scatter of data. The goal of this study is to increase awareness of runway overrun risks. This goal is achieved by increasing pilots' awareness of overrun precursors, which are often defined by velocity and position metrics. The precursors may be exceeded in a very short time and following the general trend is not enough to incorporate these short precursor triggers. So, the simulation in the body frame using the LSTM model is identified as the best configuration for simulating the final approach trajectories.

7.4 Input Optimized Model Evaluation

Hyperparameter optimization was performed using HGA to optimize the data and model parameters. The parameters optimized are discussed in section 5.3. The optimized configuration has some significant differences from the initial model configuration. The input window length for the optimized configuration is 10, this is double the initial model configuration. Also, the output window length increased from 1 to 3. As expected, increasing the input window length did improve the model performance since longer data was required to capture the low-frequency dynamics. However, the result shows that the improvement is

Table 7.2: hyperparameter optimized configuration for the simulation model

No.	Data Parameters	Optimized Value (unit)
1	input window length	10 (s)
2	output window length	3 (s)
3	resolution	2 (Hz)
4	normalization	min max
No.	Model Parameters	Optimized Value (unit)
5	activation function	tanh
6	number of prediction layers	1
7	number of hidden units	500
8	prediction model	LSTM
9	dropout	0.2
10	training batch size	128

minimal. This is an indication that the neural network based dynamic model can capture the low-frequency behavior of an aircraft. This also means that with the cyclic trajectories observed in the final approach, control is only applied for a short time and the low-frequency dynamic is not significant. The optimized resolution is identified to be 2 Hz. One reason why 2 Hz is a more optimal solution is that the model is making a shorter estimation in the aircraft states. Since estimation accuracy does vary with estimation length, the MSE of the model will improve by increasing the resolution. Under the model parameters, the optimal number of hidden units is 500. This is over double the number of hidden units used for the initial model. The number of hidden units represents complexity of the model. The more complex the model, the higher order terms it can model. As a sanity check, a LSTM model with 700 hidden units was trained, and found that the accuracy was reduced. The test sample trajectories showed that the simulated trajectory was almost flat for all the features. The downfall of the complex model is that they require a larger number of iterations and data to train. It is possible that a hidden unit of 500 is close to the ideal configuration and a larger number of hidden units is not necessary.

The performance plots for the optimized LSTM model are shown in Figure 7.16 with test sample trajectories shown in Figure 7.17. The correlation coefficient and the training MSE are 0.9934 and 0.00026. Both metrics have seen small improvements from the initial

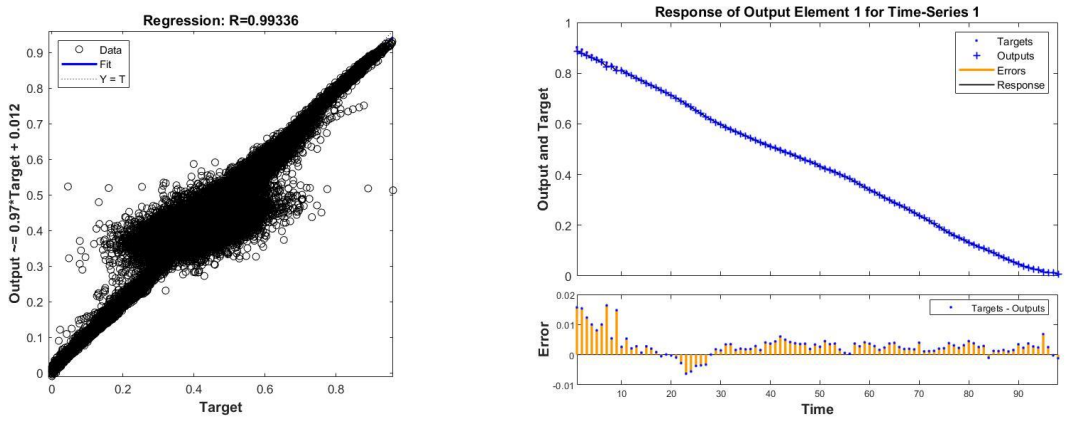

Figure 7.16: Summary plots of the LSTM based simulation prediction results trained with hyperparameter optimized configuration

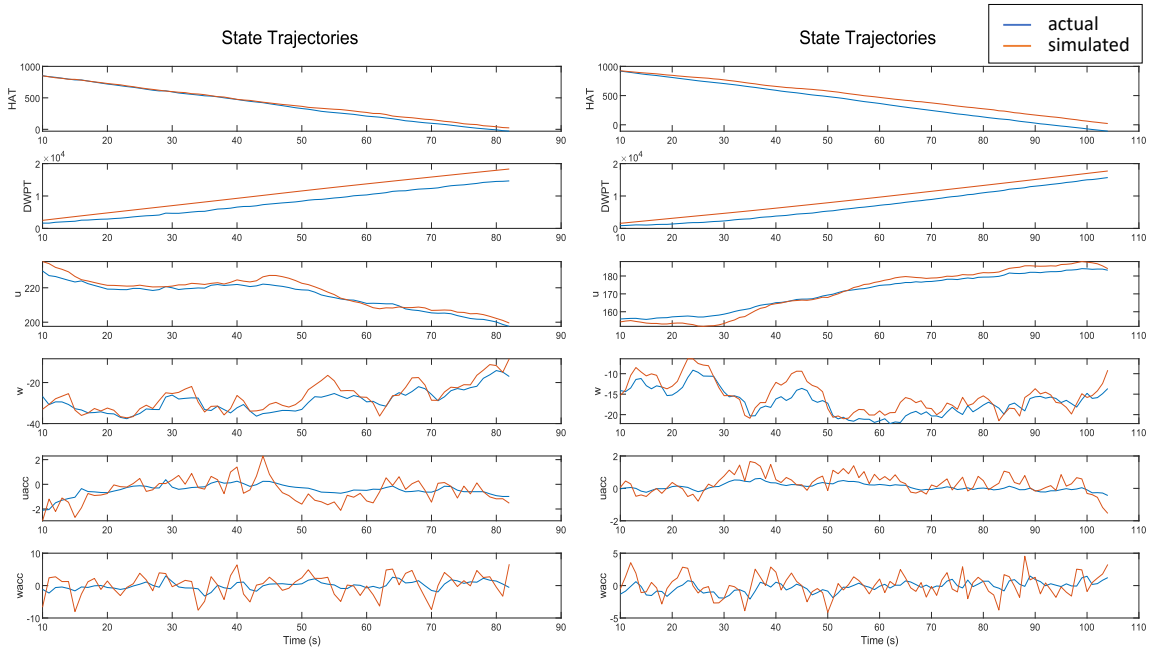

Figure 7.17: Two sample plots of the LSTM based simulation prediction results trained with hyperparameter optimized configuration

model. As discussed above these improvements may be associated with increased data and model complexity. The most significant improvement is observed from feature u, the longitudinal speed in the body frame. Now the simulated trajectory much more closely follows the actual path.

It is noted that the change in the performance metrics is very small. For the training

MSE the metric was reduced by 16%. For the correlation coefficient, the metric increased by 0.23%. The 16% in training MSE may seem big but consider that this is only an improvement in step-wise estimation. The cumulative error in the full trajectory is not necessarily improved. What this signifies is that the single layer LSTM model is sophisticated enough to simulate an aircraft's final approach with high accuracy for short flights. To improve the simulation results further, a new neural network model may need to be developed or an improved method of preprocessing the data is needed. Also, an alternative objective function that measures the performance based on the full trajectory instead of each step may need to be developed.

The simulation accuracy of the model developed in this study is compared with that of its peer. A study by Roudbari and Saghafi [96] tried to develop a neural network based aircraft dynamics model using test flight data. This study focused on modeling the forces and moments of the aircraft system. The vertical and longitudinal acceleration RMSE of this model are 2.75 and 2.87 ft/s^2. The vertical and longitudinal acceleration RMSE of this model are 0.3224 and 1.2052 ft/s^2. This shows that optimizing both the data and model parameters to generate the aircraft dynamics model may provide significant benefits to the model's accuracy.

Training time and computational time are also important factors to consider when developing a simulation model as well. For the simulation model, Matlab environment was used to train and simulate the test flights. Again GPU was enabled for training and GTX 2060 was the GPU used. Based on this computational power the time it took to train a prediction model was approximately 3 minutes. Note a patience iteration of 10 was used which means the model training stopped early when no improvement in the objective function was observed for over 10 iterations. Just like the prediction model training, it may be difficult to train a simulation model with an entire fleet flight data. However, the sensitivity analysis in the next section will show how that balance of the training data is important for having consistent model accuracy. So the flights with extreme controls may be more

heavily weighted when selecting the flights to train the simulation model. When it comes to simulating the final approach, the average computational time for each flight is 0.46 seconds. So deploying the simulation model in flight is feasible.

7.5 Dynamic Model Performance

Just like the prediction model, the simulation model's robustness to external inputs must also be carefully evaluated. In this chapter, the limits to the model will be evaluated under three conditions. The first is the model sensitivity to large maneuvers such as flares. Based on the initial model evaluation, it is clear that with extreme inputs and large feedback errors the neural network model simulation will diverge from the actual. This is expected since the aircraft control is not adjusted during the simulation even though the estimated states do not resemble the actual states. Utilizing the trained simulation model for increasing awareness to overrun requires a model that is robust to noise and feedback errors. Noise in recordings of control and aircraft states would have a direct impact on the estimated future states. As the simulation model accumulates the error over time, the simulation error would increase significantly. This is especially true for the aircraft configuration that is less represented in the flight data. For example, the aircraft configuration during a flare is expected to be underrepresented by the final approach data compared to the aircraft configuration during the rest of the final approach. So the simulation model sensitivity to flare will be evaluated. Another similar but different condition is a flight in severe wind. Distribution of the flights with large wind speeds or large wind direction changes shows an imbalance between normal flights and severe wind flights. However, the wind is a feature that directly impacts measurements such as true airspeed, which are known to have a direct correlation to runway overrun precursors. So observing the simulation model's sensitivity to severe wind will also provide guidance on how robust the model is.

Besides sensitivity analysis, it is also important to determine the control space that with high model accuracy. When simulating a flight, large control inputs with low representa-

tion in the data may be used. When these controls are constantly used the probability of resulting in an incorrect simulated trajectory will increase. And even if these controls are used for a very short period of time, a large deviation in the prediction from the actual results in feedback error would propagate. So it is crucial to understand the control space and where the accuracy is slow

These analyses will be done by observing the step-wise estimation RMSE for a pair of features. For the flare sensitivity analysis, the two features are HAT and ELEV since the final pitch-up maneuver is mainly controlled by elevator deflection and flare altitude is measured by HAT. For the wind sensitivity analysis, the two features are WLAT and WLONG, which are the wind speeds in the aircraft body's longitudinal and lateral directions. Finally, for the control space analysis, four different features are paired with the ELEV feature. The first three are the aircraft attitudes ROLL, PTCH, and DA. Aircraft attitude states are observed because the aircraft equations of motion for the pitching moment include coupling terms between attitude and attitude rates. Then the PLA feature is paired with ELEV since the relationship between thrust and elevator control is observed when analyzing the longitudinal dynamics of an aircraft. A contour plot that shows the model accuracy across various attitude and control input configurations will be used to identify the boundary of controls and states at which the model can be trusted. Then the same pair of features will be used to create a map of the distribution of data points. Since the model performance of a neural network is sensitive to the number of data, these maps will help identify the potential cause of low accuracy at various points on the map.

The focus of section 7.5 is on the ELEV feature because of its significance on the longitudinal dynamics of the aircraft. The longitudinal dynamics is important because multiple features associated with it are used in the definitions of runway overrun precursors, for example, airspeed and altitude at threshold crossing. So the distribution of ELEV feature is shown in Figure 7.18 The mean of this distribution is at -6 degrees and the majority of recordings are between -10 and 0 degrees. This is an expected distribution since the aircraft

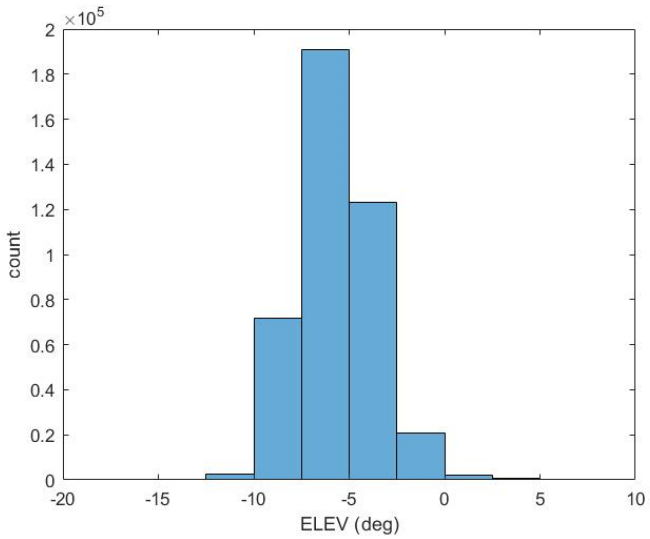

Figure 7.18: The distribution of ELEV feature for all of the flights

is pitched down during the final approach. And typically the aircraft need the elevator deviation to maintain this attitude. Interestingly only on a few occasions, the aircraft elevator deflected above 0 degrees indicating that for the given aircraft type it is not necessary to have a large elevator deflection for the final pitch-up maneuver.

7.5.1 Flare

To understand the effect of flare on the simulation model accuracy, the overall trend in the model accuracy needs to be evaluated. The overall trend is evaluated by observing the distribution of average RMSE across different levels of ELEV before and after the flare. The result is shown in Figure 7.19. This result is generated by taking the average stepwise RMSE at each set of data categorized by the following:

1. whether the HAT is above the start of flare for the given flight
2. whether the ELEV is below -7.5 degrees, between -7.5 and -5 degrees, between -5 and -2.5 degrees, between -2.5 and 0 degrees

The ELEV category of above 0 degrees is not considered because the number of data collected for this category is too few so the result is not representative of general model perfor-

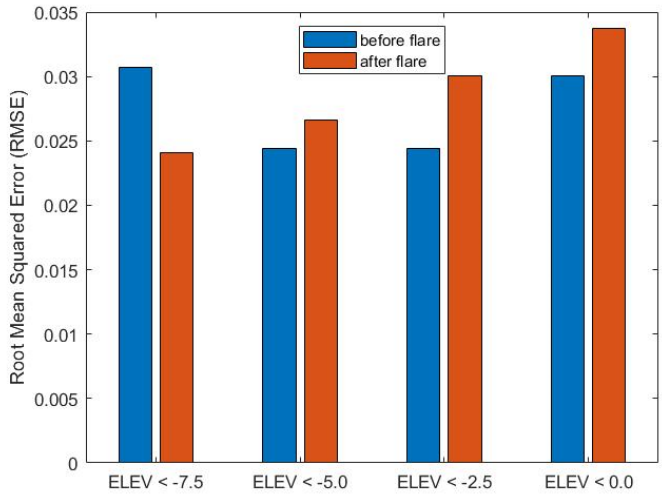

Figure 7.19: The error distribution based on the start of flare and elevator deflection

mance. The result shows an interesting trend where independent of whether the simulation is performed before or after the flare, the error increases with decreasing ELEV. This is expected since the data distribution is significantly less for ELEV between -2.5 and 0 degrees. The same should be true for ELEV below -7.5, instead, this behavior is only seen for simulation performed before the flare. The RMSE distribution after the start of flare shows that the accuracy decreases with increasing ELEV. Based on the ELEV trajectories shown in Figure 6.32, it is likely that the ELEV below -7.5 degrees category represents the start of flare while ELEV between -2.5 and 0 degrees represents the end of flare. From this logic, a potential explanation is that as a flare is initiated the aircraft dynamics are hard to estimate. This is expected since near the end of flare multiple external factors contribute to the aircraft dynamics, for example, the ground effect. Considering that the RMSE metric keeps the original unit and all the features were normalized, in terms of the min-max ELEV distribution the error increase is approximately 1%. However, relatively the RMSE at ELEV between -2.5 and 0 is approximately 30% higher than that of RMSE at ELEV below -7.5. Overall, before the flare, the simulation error distribution varies with the distribution of data points. After the flare, the simulation error distribution varies with the flare process. So ideally more data should be collected for ELEV at two extremes, which would

help reduce the decrease in model accuracy.

The data distribution and model performance map between HAT and ELEV features can shed some light on this observation. These two results are generated as mentioned in section 7.5 and shown in Figure 7.20. The results are generated with test data only. The plot

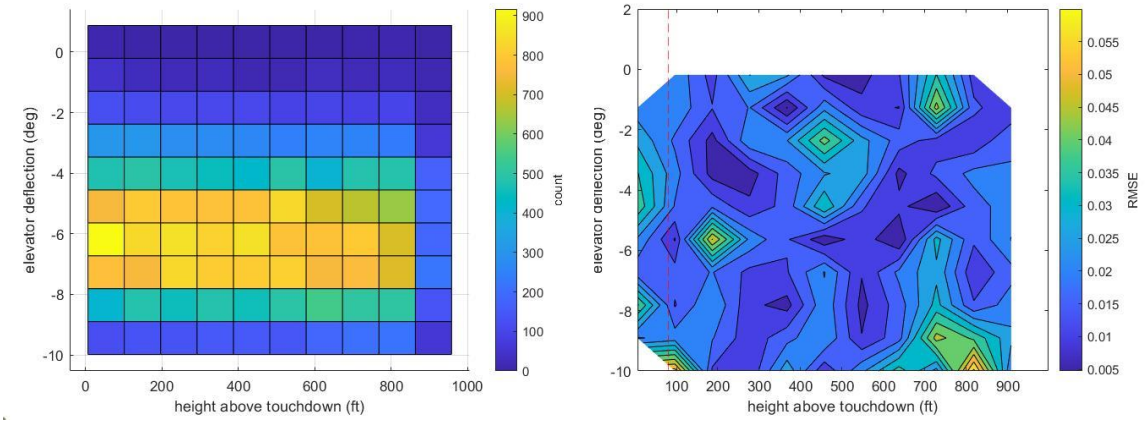

Figure 7.20: The data distribution and model performance map based on ELEV and HAT features

to the left supports the data distribution of ELEV in Figure 7.18 and it also shows that the data distribution is independent of HAT. This is expected because all aircraft will decrease in altitude from 1000 to 0 ft during the final approach. The significantly low data count between 900 and 1000 ft HAT is because the input window length is 10s. The plot to the right is the RMSE map of HAT and ELEV. The red vertical dashed line in this plot is at 80 ft HAT which is the median of the start of flare. The map shows that the model accuracy is fairly consistent across most of the data range except for a few locations. The first is below 80 ft HAT where the RMSE is above 0.02 for the entire region. This is another indicator that the model performance is low after the flare. The second is at the bottom right corner of the map with HAT above 700 and ELEV below 8 degrees. This is the region that represents flights at the start of flare where the pilot is pitching down to reduce altitude. The reduced model accuracy is expected due to the rarity of such data. The last locations are a few points in the middle of the map where the model is not well-trained. The diamond shape around these points indicates a possibility of few data points in that local region where the model

lacks indicated from the sudden jump in RMSE gradient. Overall the model performance is reliable during most of the final approach and in the demonstration chapter, the reduced accuracy after flare should be accounted for.

7.5.2 Wind

To understand the effect of wind on simulation model accuracy, the distribution of RMSE across different levels of WLAT and WLONG is evaluated. The categories of WLAT and WLONG considered are as follows:

1. WLONG below 0 kts, between 0 and 5 kts, between 5 and 10 kts, between 10 and 15 kts, and above 15 kts

2. WLAT above 10 kts, between 10 and 0 kts, between 0 and -10 kts, and below -10 kts

The RMSE distribution is shown in Figure 7.21. In this figure, the four extreme scenarios

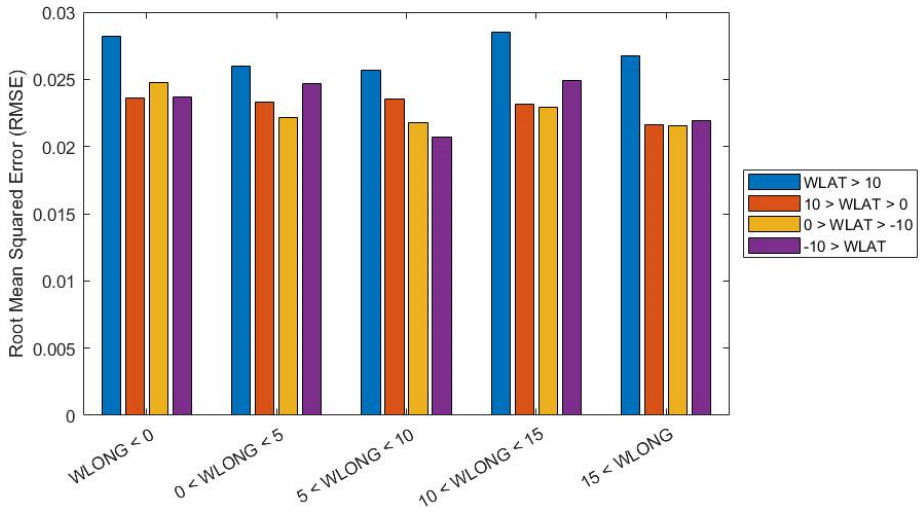

Figure 7.21: The error distribution based on large wind speed or wind direction change

where the data is least recorded are WLONG below 0 degrees or WLONG above 15 degrees while WLAT is above 10 degrees or below -10 degrees. The first expectation is that the RMSE distribution for these categories is high, which makes the relatively higher RMSE

for the WLAT class understandable. However, the RMSE for the WLAT class is in some cases lower or on par with other classes. Although unexpected, this is the ideal accuracy distribution where the model accuracy is independent or less sensitive to the data frequency. Overall the RMSE ranged between 0.02 and 0.025 for all the classes but WLAT above 10 kts. And even for this category the error is only 0.5% higher in the min-max range so the model performance may be considered as fairly independent of the wind speed.

The data distribution and RMSE maps in terms of WLONG and WLAT features are generated in Figure 7.22. The data distribution plot shows a single point with high data

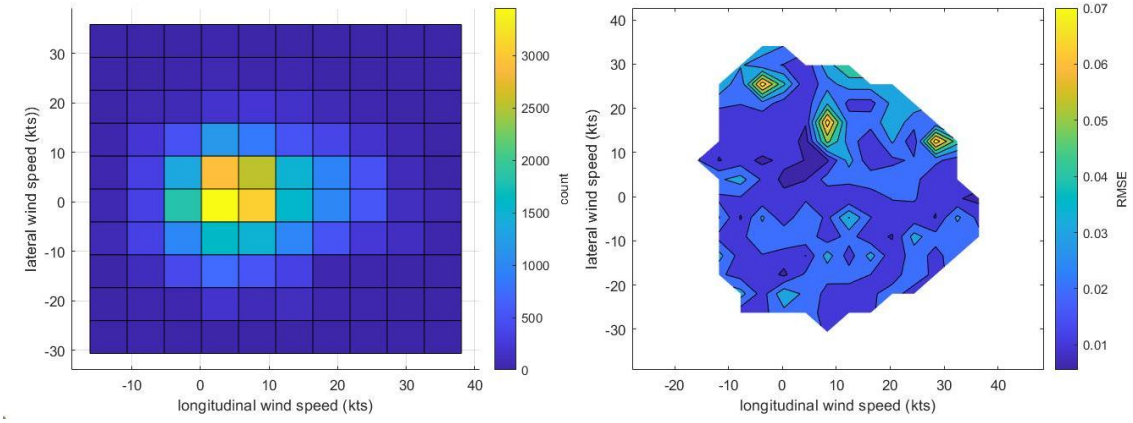

Figure 7.22: The data distribution and model performance map based on WLAT and WLONG features

concentration where WLAT is close to 0 kts and WLONG is close to 5 kts. So as previously mentioned, it is likely for model accuracy to decrease with deviation in wind speed at this point. However, the RMSE map shows that the model accuracy is fairly consistent which shows that the optimization algorithm has done well in improving the model performance at a wide data range. Unfortunately, there are three points on the map, WLAT above 10 kts, where the RMSE is high. The randomness of these points shows that the reason may be associated with how the mode is trained. Based on all of the analysis so far, improving the model performance will require identifying a model that better represents the dynamics, based on Figure 7.16. Besides this, the current model is able to consistently model estimate aircraft state. So to improve the result will require a different technique to preprocess the

data so that data factors not considered in this study, such as noise, can be minimized to better represent the aircraft dynamics.

7.5.3 Feasible Control Space

The data distribution map between ELEV, ROLL, PTCH, DA, and PLA features are shown in Figure 7.23 and Figure 7.24. The data distribution shows that for ROLL and YAW fea-

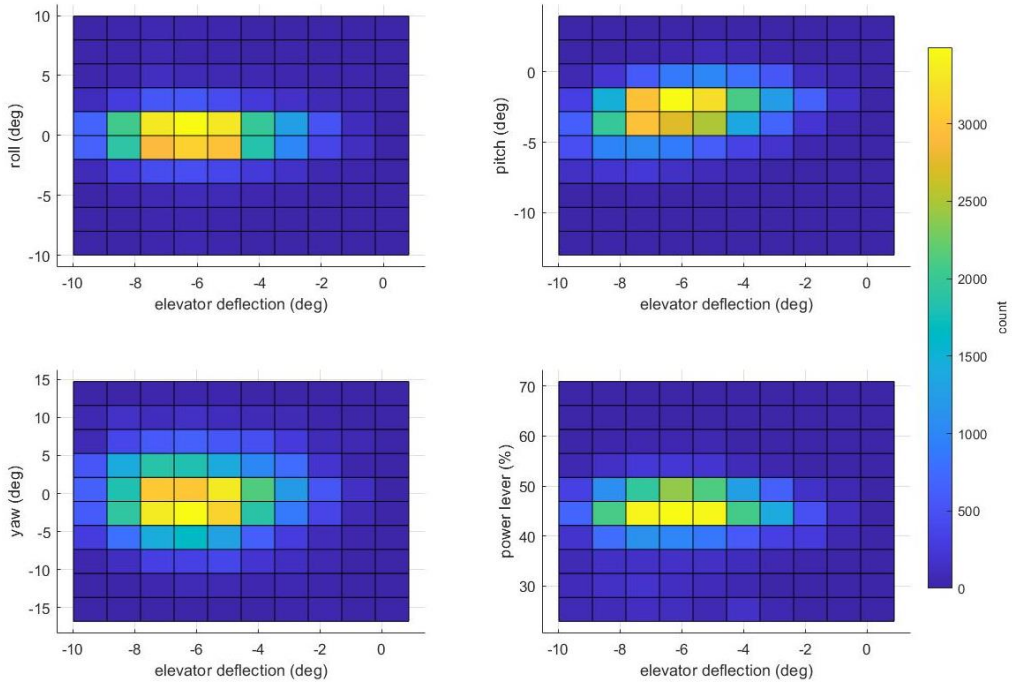

Figure 7.23: The data distribution between ELEV, ROLL, PTCH, DA, and PLA features

tures the center of the distribution is at 0 degrees and the record count reduces significantly with increasing distance from this center. This is expected since the standard procedure for the final approach is to align the aircraft to the runway. During crosswind, the aircraft may add some DA to account for the wind, which is why the distribution of DA is wider than ROLL distribution. The center of PTCH distribution is at -3 degrees with most of the data between -10 and 0 degrees. This is expected as well since during the final approach the aircraft is pitched down until flare. Note that the distribution map also shows a positive linear relationship between ELEV and PTCH which is expected since ELEV controls PTCH. For

the PLA distribution, the center is at 45 % with a concentration between 40 to 50 %. This is expected since during the final approach the engines are set at idle and increased before touchdown. So the model performance is expected to decrease beyond these concentrated data regions.

Indeed the model performance map shown in Figure 7.24 indicates that higher RMSE in state estimation are found on the regions beyond the data concentrated area. However,

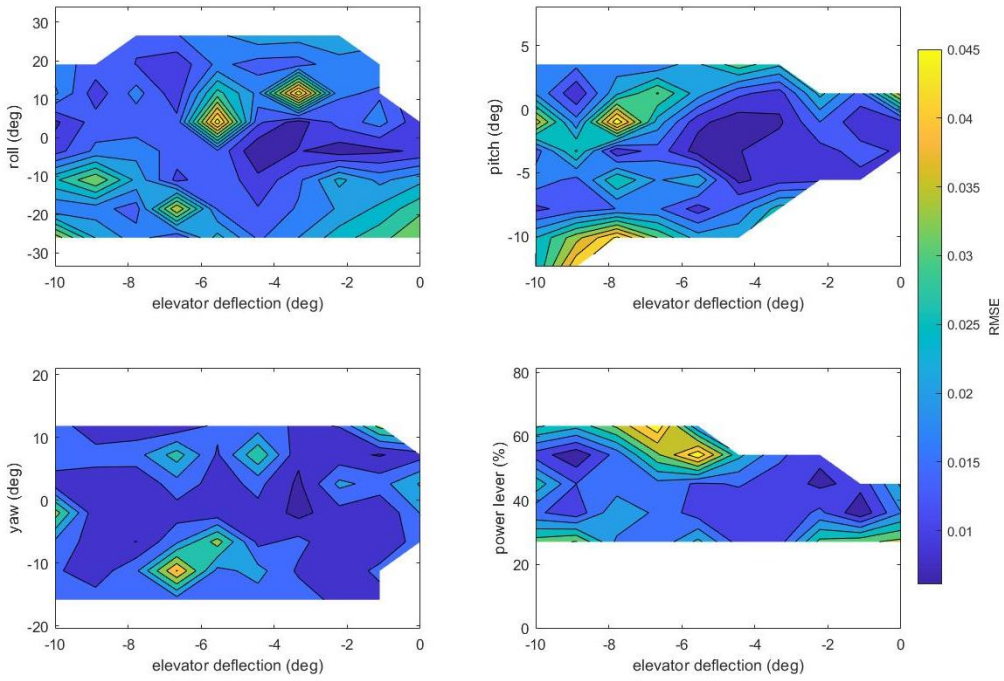

Figure 7.24: The model performance map based on ELEV, ROLL, PTCH, DA, and PLA features

no significant correlation between data concentration and model performance is observed for the relationship between ELEV, ROLL, YAW, and PLA. Since the results show that the simulation model is also fairly accurate in the data less concentrated regions. The exception is with ELEV and PTCH results where the accuracy is specifically low at the data concentrated region. It should be mentioned again that during the final approach the ROLL and DA stay low. So a large ROLL or DA in this data set is actually a small ROLL or DA for aircraft dynamics. This means the consistent result in ROLL and DA is likely due to a small change in the aircraft motion for the data used in this study. Overall, the model

accuracy is low for most data ranges represented in this study and Figure 7.17 shows that with the current model the flight trajectories do not diverge significantly.

The findings and contributions from this chapter can be summarized as follows:

1. HGA was able to optimize both data and model parameters for the flight state simulation model

2. LSTM-based neural network simulation model was the most accurate model without exploding feedback error

3. The simulation model was sensitive to both flare maneuvers and various wind conditions

4. the control space for the simulation model was narrow due to consistent landing configuration during the final approach and the accuracy was mostly consistent across the control space as well

CHAPTER 8
DEMONSTRATION OF THE PROPOSED MODEL FOR IMPROVING FINAL APPROACH

Now the two most important questions remain.

1. What does accurate prediction and simulation model mean in terms of runway overrun

2. How can the current models be used to increase awareness of the risk of overrun during the final approach

The first question may be addressed by associating the model accuracy results with runway overrun metrics. That is when certain metrics are predicted or simulated to provide the risk of overrun, the result must consider the error in prediction. So the error in prediction and simulation needs to be quantified by a metric that defines overrun. Once the two model results are mapped to an overrun metric, they need to be used to train the pilots to improve the three aspects of safety, ASK. As identified from the literature survey, runway overrun is frequently associated with precursors. So this chapter will discuss how prediction and simulation models can be used to identify trigger precursors in flight to notify pilots of increasing or decreasing the risk of overrun. Mainly, the simulation model will be used to show when the current flight triggered overrun precursors and what can be done to mitigate this. This will be combined with the prediction model at the end of the simulation where the accuracy is identified to be lower. The prediction model will identify more overrun precursors that the current flight will trigger if the current flight continues. This way the risk of overrun could be identified early while reducing the concern with poor simulation for the last stretch in the final approach where the model accuracy is lower. An illustration of how the prediction and simulation models combine to increase pilot awareness of the

risk of overrun is shown in Figure 8.1.

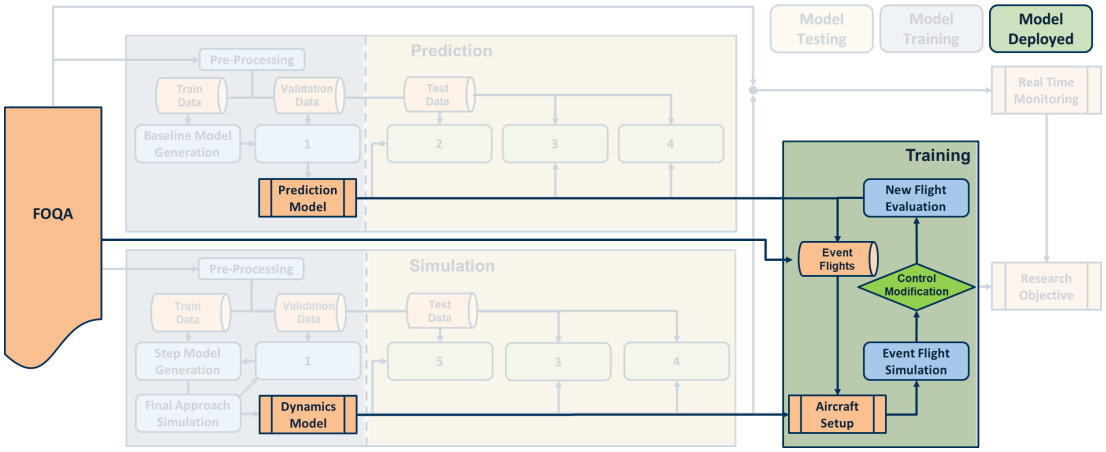

Figure 8.1: Overview of how the prediction and simulation models can be used for pilot training

8.1 Interpreting the Prediction and Simulation Results Based on Metrics Associated to Overrun

The main benefit of the prediction model is that the aircraft states near touchdown can be predicted at an altitude early enough for pilots to respond. The neural network based prediction has shown the potential to learn the overall trend in aggregate flights but also the difference in trajectories for individual flights based on the final approach data. Nonetheless, the neural network based prediction model still observed some prediction error and the error would increase with increasing prediction range. When only looking at the prediction results, they could be used to calculate the estimated required runway distance. They could also be used to identify the precursors associated with runway overrun that would function as an alarm in flight. With the error, both of these use cases require an output that shows a range of outcomes due to the error in prediction. The same should be considered with the simulation model. The feedback error from each step could propagate to a large error near the end of the simulation. The probability of simulation error may be calculated from the error in the test flights. In both scenarios, there is a need to map the errors to a metric that

can interpret to increase or decrease the probability of overrun. Since overrun is defined as crossing the other end of the runway, the errors can be mapped to the required landing runway distance. The exact calculation of the required landing runway distance requires a lot of data including, runway surface condition and aircraft brake effectiveness. With these data, the required landing distance can be calculated based on the energy carried by aircraft at touchdown vs the energy the aircraft brakes can dissipate. Since these data are not available only the ratio of required landing distance will be calculated to better interpret the prediction and simulation results.

A study by FSF [6] has mapped various aircraft conditions to the increase in the required runway distance. This study mentioned that with 5 % increase in final approach speed results in a 10 % increase in landing distance. This means if the prediction error of airspeed is 5 % and the prediction model is used to calculate the risk to overrun based on the required landing distance, then the addition of 10 % of the required landing distance should be accounted for. Considering that the final approach speed is measured at 50 ft height above touchdown, which is the desired altitude at threshold crossing, this means the final approach speed ranges from 89 kts to 138 kts, with a difference of approximately 49 kts, which would round to 50 kts. Since 5 % of the final approach speed range is 2.5 kts and this corresponds to a 10 % increase in required runway distance, the hyperparameter optimized TAS prediction RMSE of 4 kts at 10 ft HAT corresponds to 16 % additional required runway distance at 10 ft. With this, all of the aircraft states could be matched to a runway distance multiplier to create a uniform metric that would show the model performance.

For the simulation model, since the full final approach trajectory is generated the flight may be associated with runway overrun metric at multiple points during the final approach. AC 91-79A [28] illustrates this at three altitude windows (1000 ft, 500 ft, and threshold crossing) to satisfy stabilized approach. Satisfying stabilized approach is a good way of evaluating the flight risk to overrun due to the high correlation between stabilized approach and runway overrun. At the three altitude windows, the flight paths' proximity to the

stabilized approach glide path angle, vertical speed, and velocity at threshold crossing can be evaluated. However, this proximity metric does not directly map to the required runway distance. So just like the prediction model the aircraft speed at 50 ft HAT in the simulation can be used. For the test data used in this study, the simulation RMSE at 50 ft HAT is 0.0057. This is a value normalized by the min-max range which indicates that the average simulation error at 50 ft is 0.57 %. This corresponds to 1.4 kts average RMSE, which means to account for the simulation error, if the required landing distance is calculated from the simulation result, the required landing distance should be increased by 5.6 %. The simulation result brings an extra dimension to map the model result to the required landing distance. For example, the study by FSF also showed that the required landing distance increases with increased altitude at the threshold. The study mentioned that the required landing distance will either be added by 1000 ft or multiplied by 1.3. This information may be used to trace back an aircraft state that can also be mapped to the required landing distance. Assuming that all the flights are flying close to CDFA the glide slope during the final approach should be approximately 3 degrees. This means the glide slope should also be around 3 degrees at 300 ft HAT, where the prediction is made, and at threshold crossing. Deviations in the glide slope directly translate to the changing altitude. That means the error in glide slope translates to the error in altitude, which can be mapped to the required runway multiplier. Based on the test data the average glide slope ratio of predicted over actual, during the final approach, is 0.55 degrees. Assuming the ideal scenario where the glide slope is 3 degrees and the altitude at threshold crossing is 50 ft, then the error of 0.55 degrees will result in an increase in threshold crossing of 5.66 ft. This is equivalent to a 100 ft increase or 0.03 increase in the multiplier for the required landing distance.

Note that all of the calculations made above are based on a study that assumed no thrust reverser applied during landing, which is not the case for most of the test flights used in this data. Also, the required landing distance multipliers identified were not designed for the purpose of evaluating prediction and simulation model performance during the final

approach. Nonetheless, it is used to provide a glimpse of the significance of errors in the two models so that the prediction results can be understood in terms of flight safety. This brings up the need in the future to statistically find the multipliers that map the required landing distance to landing performance indicators. The findings may be used directly to evaluate model performance but also used to augment new metrics that rate the landing performance with a single value. An example of a such metric is discussed in a study by the author [117].

8.2 Evaluating Potential Risk with Overrun Precursors

Now that the results from prediction and simulation models are evaluated, the method of identifying risk through overrun precursor is introduced. Out of the many overrun precursors, high approach speed is selected due to its contribution to many other precursors such as high energy and stabilization. According to AC 120-82 [12] the high approach speed is defined as $CAS > V_{REF} + x\ (kts)$. The exact threshold velocity changes with altitude. For this study, the high approach speed is defined based on the 95th percentile of all the airspeed recordings. The 95th percentile is selected because all of the flights landed safely so none of the test flights are problem flights. So only the 95th percentile is observed to select out the flights with abnormally high speed even though they landed safely. The distribution of all CAS recordings from the available flight data is shown in Figure 8.2. The 95th percentile CAS is 135 kts or 228 ft/s, which is used as the threshold for identifying flights with high approach speed.

For the simulation model, each flight that exceeds the high approach speed between 1000 ft HAT and 300 ft HAT is identified. The confusion matrix of the high approach speed will be used to evaluate how accurately the precursor flights can be identified. For the flights where the simulation model successfully identified the high approach speed, the time index at which the precursor is triggered is analyzed. The results are shown in Figure 8.3 The confusion matrix to the right shows that the simulation model is accurate

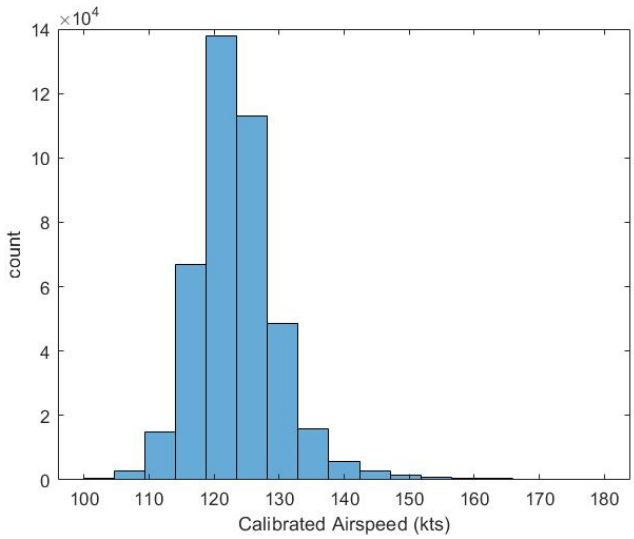

Figure 8.2: CAS distribution for all the data

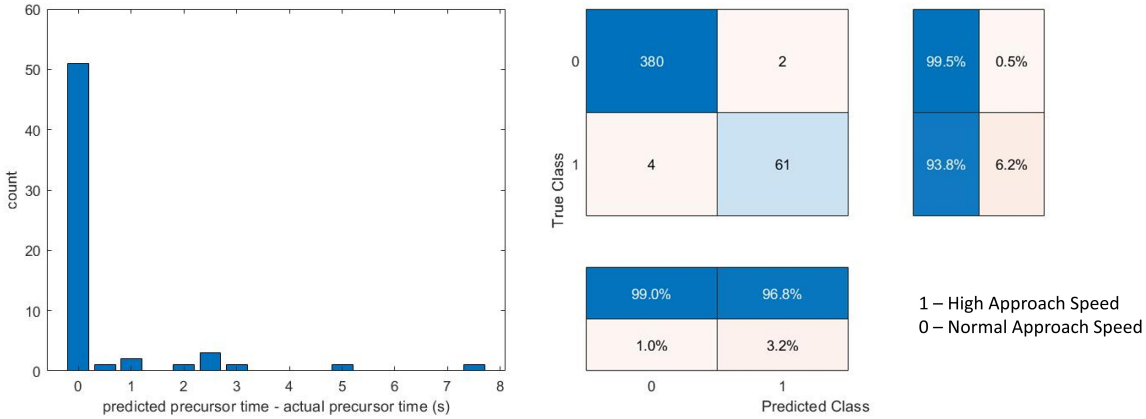

Figure 8.3: confusion matrix of high approach speed (right) and the time difference between simulated high approach speed trigger and actual high approach speed trigger (left)

enough to maintain precision and recall over 93 %. The f1 score for the high approach speed precursor is 96.56 % which is high as well. The simulated and actual high-speed trigger time difference distribution plot shows that for 83.61% of the time the precursor triggered at the same time for both simulated and actual fights. What this shows is that in this simulated environment, the pilot is able to fly a final approach path while experiencing a similar experience to real flights in terms of high-speed approach triggers.

For the prediction model, each flight that exceeds the high approach speed at any prediction altitude point is identified. Note with the prediction model, the risk of evaluating

runway overrun based on precursor is that the prediction output is not sequential and previous observations show that the prediction model acts like a regression model. However, the airspeed measurements are often very noisy with constant ups and downs that would trigger the precursor for a very short period of time. So the confusion matrix would show a very high False negative, which is detrimental to safety. To account for this issue the airspeed threshold has been relaxed by the RMSE at each prediction altitude so that a bracket of airspeed is used as the trigger for high approach speed. This would reduce the false negative but increase the false positive significantly. However, when it comes to safety, a false positive is preferred over a false negative. That doesn't mean the false positive outcomes are acceptable since they numb the pilots' response to the prediction and simulation results. The result based on this change is shown in Figure 8.4. The result shows that the predic-

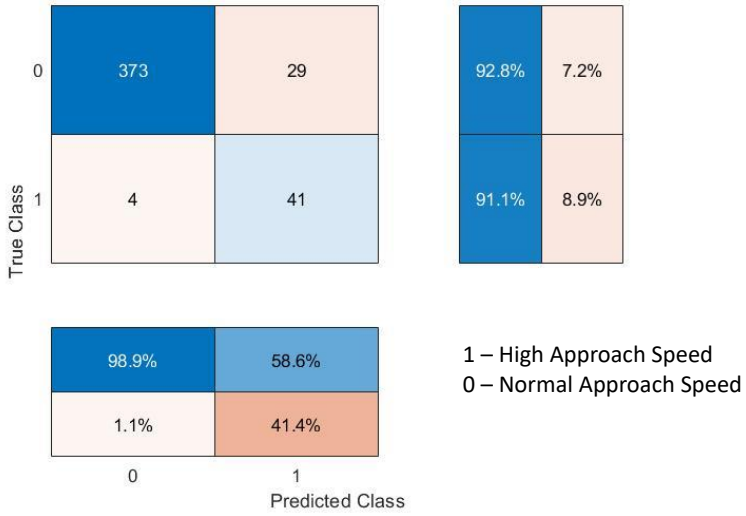

Figure 8.4: confusion matrix of high approach speed based on results below 200 ft HAT

tion model is capable of predicting high approach speed with a recall of over 91 %. The precision is very low, at 58.6 % but as discussed earlier the precision is less of a concern when it comes to safety. Overall what the result shows is that the prediction model has the potential to evaluate overrun risk based on precursors. However, the nature of prediction makes it difficult to predict the fine changes in the trajectory which makes precursor-based risk analysis less suited for the prediction model.

8.3 Improving Control Trajectory to Minimize Potential Risk

So far the analysis has shown that neural networks can learn the aircraft behavior just before touchdown to predict flight states early enough for pilots to adjust the flight. The neural network model can accurately simulate the final approach and use precursors to alert potential risk for overrun. The goal of this study is to improve flight safety by increasing awareness of runway excursion precursors. A demonstration of how the prediction and simulation model can be used together to improve flight safety is shown in this chapter. This demonstration is generated from the test data used in this study.

This demonstration will present a two-step process. In the first step, the pilot is flying an aircraft in a training environment. The flight results in a high approach speed trigger below 200 ft HAT. Then the imaginary scenario starts where the pilot is alerted by the prediction model of the potential to trigger high approach speed. This imaginary scenario is followed by a training flight where the control trajectories of another flight with similar states in the first 20 seconds of the final approach are used to adjust the current flight to avoid a high approach speed trigger. Two flights with similar states in the first 20 seconds of the final approach are found by calculating the RMSE of the states of two flights that did and did not trigger high approach speed below 200 ft HAT. Then the control trajectories of the flight that did trigger the precursor were replaced by the flight that did not.

The results of this demonstration can be found in Figure 8.5. The plots to the left show the original flight from the actual data between 1000 ft and touchdown. The top plot is for height above touchdown (HAT) while the bottom plot is for the longitudinal velocity in body axis (u). The red dashed line is the prediction index at which the high approach speed was triggered. For this original flight, the prediction was unable to identify high approach speed at 200 ft HAT. This is because the prediction model is a complex regression model of the aircraft states. The rarity of flights with increased aircraft speed below 200 ft HAT makes it difficult for the model to predict. As mentioned in the last section, for each

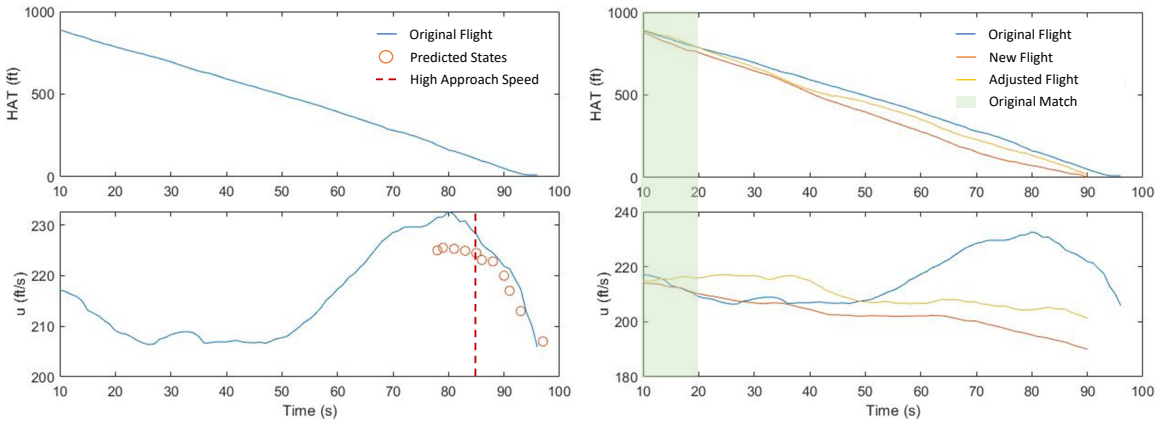

Figure 8.5: Demonstration of how the prediction and simulation models can be used in a training environment to get in-flight and post-flight feedback to reduce the risk of overrun

prediction index the high approach speed threshold was relaxed by the RMSE of that index. So at the 120 ft HAT the prediction model was able to predict a high-speed approach.

Now that the pilot is notified in flight about the precursor, the pilot is able to study the final approach after the flight. The plots to the right show the original flight, the peer flight (adjusted flight), and the new flight after adjusting the control trajectories on the original flight. The peer flight in this case is the other flight with similar states in the first 20 seconds of the final approach but did not trigger high approach speed below 200 ft HAT. The green block shows the segment of flight where the original and peer flights match. This is also the segment where the new flight is using the original control trajectories. The new control trajectories only kick in after the green block. This is why the red and blue line has very similar trajectories early on but they diverge from the 20 s mark. What the plots show is the new control trajectory that is able to touchdown without triggering high approach speed. An important note is that the new flight was not generated with a path planning algorithm. So the end states may not match the desired states. For example, the aircraft may successfully touchdown by the end of the trajectory because all of the trajectories used has an end state of 0 ftHAT. However, the aircraft will most likely fail to fly the same distance which means the end states of the DWPT parameter will not match. This is not considered a problem because the distance to the touchdown may be adjusted by delaying

the descent trajectories. In reality, this would cause conflict with operational constraints. To restrict the scope of this study, operational constraints are not considered. The results are generated specifically to increase awareness of runway overrun in flight and learn from a peer flight that successfully touchdown without triggering high approach speed.

The illustration of original trajectories and peer trajectories are shown in Figure 8.6. In this figure, the blue lines are the new trajectories. As shown the new acceleration tra-

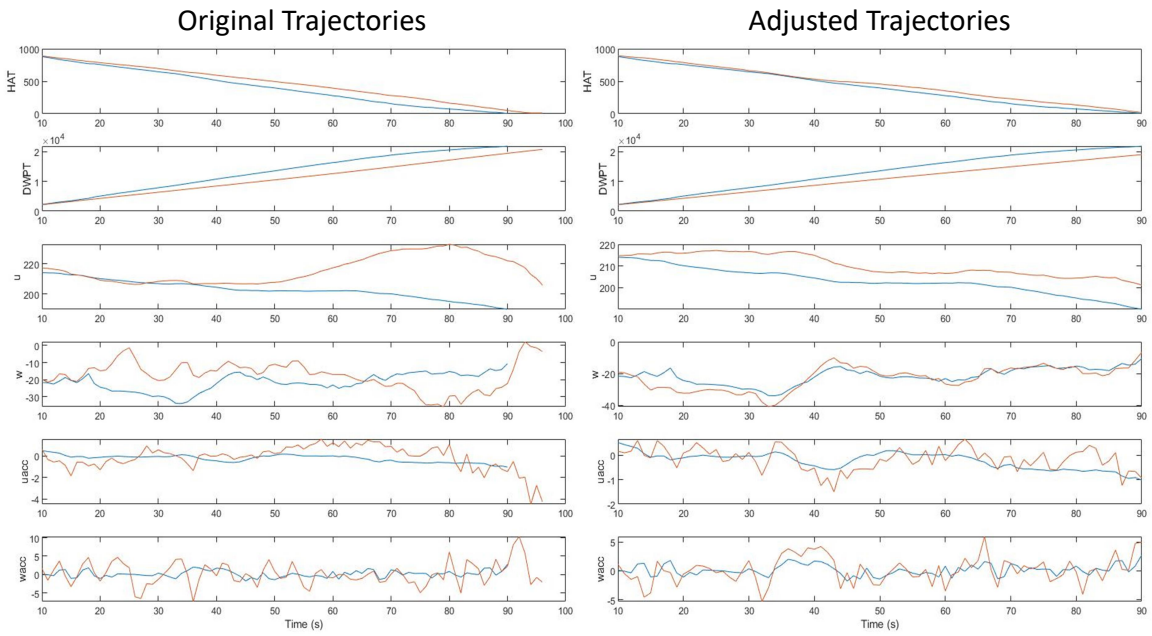

Figure 8.6: Demonstration of how the prediction and simulation models can be used in a training environment to get in flight and post-flight feedback to reduce the risk of overrun

jectories closely follow the peer acceleration trajectories, since they are direct fall out of the controls. The velocity trajectories show some deviation in the first 20 seconds but eventually follow the paths of the peer flight. Finally, as discussed above the new flight successfully touched down at the end of the flight but failed to fly the same distance as the original flight.

The findings and contributions from this chapter can be summarized as follows:

1. The prediction and simulation errors may be mapped to the required landing distance, a metric that is easily interpreted

2. The prediction results had a lot of false positives when converted to a high-speed approach precursor

3. The simulation results were accurate for simulating flights that triggered high-speed approach

4. The demonstration showed the potential for the prediction model to be used to adjust the flight behavior near touchdown to reduce exposure to high-speed approach

5. The demonstration showed the potential for the simulation model to give immediate feedback on the current flight behavior that resulted in a high-speed approach

6. The demonstration showed the potential for the simulation model to immediately compare the current flight behavior to its peer with a lower risk profile post-flight

CHAPTER 9

CONCLUSION

Flight accident data shows that runway overrun is the accident type that occurs the most frequently. Runway overrun is defined by the aircraft crossing the opposite end of the runway during take-off or landing. Recent improvements in sensors and the increased number of flights allowed for the development of data-centric tools that would prevent runway overrun. One such tool is the FOQA program. FOQA program takes the aggregate flight data and identifies trends in flight by creating thresholds for different flight events. Some of these thresholds are identified as precursors to runway overrun. An example includes high approach speed, which is when the aircraft CAS exceeds the predetermined threshold for given altitudes. By observing the distribution of speed at various points in flight, the nominal flight behavior can be identified and used as a reference for future flights. More recent studies combined FOQA data and artificial intelligence to better understand runway overrun. Some studies used machine learning to develop clustering models through unsupervised learning to label the flights with unique characteristics. This study investigated methods to improve runway overrun by creating supervised machine-learning models that predict and simulate aircraft states near touchdown. The objective of this study can be summarized as follows:

Research Objective

Develop a methodology that increases pilots' awareness of the risk of runway overrun by predicting and simulating flight states that define a stable approach

To meet this objective, it is hypothesized that model-based methodology can provide a means of runway overrun risk detection based on the flight data collected in flight. This methodology consists of three components: prediction model development, simulation

model development, and flight analysis based on these two models. An illustration of the three components is shown in Figure 9.1. The key to this framework is on increasing

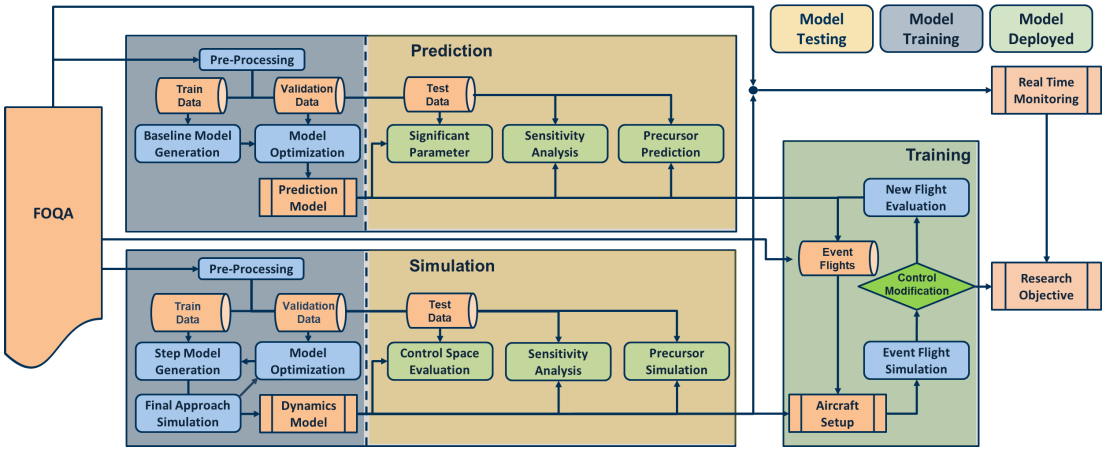

Figure 9.1: An illustration of the proposed methodology and each component

the pilots' awareness of runway overruns by exposing plots to overrun precursors. With the prediction model the overrun precursors are predicted, in flight, during final approach based on the flight states near touchdown. This model is trained as a sequence to multi point model where the final approach data is used to predict aircraft states at various altitudes. Altitude is selected as the prediction index because it represents the landing progress during the final approach. Then the simulation model is used, post-fligth, to simulate peer flights flown in a similar condition to compare and learn. This model is initially trained in open loop such that each time stamp in the flight data is a separate sample. During testing the model is tested in close loop such that the estimated aircraft states at time t are used to estimate the aircraft states at time t+1. The best models are selected from the test results to perform flight analysis. The flight analysis is started by mapping the prediction and simulation outputs to the required landing distance. Then how the two models can be used to increase awareness of runway overrun was demonstrated from test flight data.

The focus of research question 1 was on developing a prediction model that can be used in flight to identify runway overrun risk. The first research question stated: The survey of past studies showed that predicting aircraft states is indeed possible and this can be achieved

Research Question 1

Is it possible to accurately predict runway excursion events and the metrics that define them far enough into the future for the pilots to respond

with various models: sequence-to-point, sequence-to-sequence, and point-to-point models. The real question was whether the prediction can be made far enough and be robust to external factors such as adverse wind and flare. Also, can this model be generated from FOQA data collected during the regular flight? Maximizing prediction range was one of the focuses of the model training process because pilots are unable to respond to alerts too close to a touchdown. Finding a method to identify the optimal configuration with far enough prediction range and accuracy was another focus of the model training process. The prediction accuracy, prediction range, and optimization are mentioned in research sub-questions 1.1, 2, and 3. The optimized model was used to identify the significant parameters for predicting flight states near touchdown. Significant parameters are important because they can be used to identify new precursors related to overrun, minimize the model size by retraining the model with only the significant features, and validate that the model is behaving as expected. The significant parameter identification is mentioned in research sub-question 1.4. Once the optimized model has been identified, the model's robustness to wind was evaluated because during the final approach the aircraft is exposed to ground effect and wind shear that will affect flight behavior. The optimized model robustness to flare was also analyzed because the magnitude and timing of flare vary significantly for each flight. Analysis of model sensitivity to wind and flare is mentioned in research sub-question 1.5 and 6.

The focus of research question 2 was on developing a simulation model that accurately models the flight behavior during the final approach. The second research question stated: Just like the first research question, the feasibility has already been established from past studies but the real question is can this model be generated from FOQA data collected in regular flight? How to create the simulation model from the FOQA data is mentioned in re-

Research Question 2

Is it possible to generate six degrees of freedom high fidelity model of an aircraft from flight data obtained in flight to simulate event and nonevent approach and landing for runway excursion accidents.

search sub-questions 2.1 and 2. And similar to the prediction model, the model's robustness to wind and flare was also evaluated. These were especially important for the simulation model because an accurate simulation would require the modeling of coupling effect during adverse wind and large maneuvers. The question on simulation model sensitivity to flare and wind is mentioned in research sub-question 2.3. In addition, the control space for the model was also evaluated. The concern with any machine learning model is that the training data size must be large to develop an accurate model. The scarcity of the model at the control extremes brought up the question of how reliable is the final model at these control spaces. The question on simulation model control space evaluation is mentioned in research sub-question 2.4

After the prediction and simulation models were developed the training component took these two models to answer the main research objective. As mentioned above the prediction and simulation outputs were mapped to the required landing distance metric. The two-mode outputs were also mapped to high approach speed, an overrun precursor, to see how accurately can overrun precursors be modeled. During the demonstration, the two models were used in a two-step approach to improve awareness of overrun risks. With the demonstration, the potential for the proposed prediction and simulation models to increase awareness of overrun risk was shown.

9.1 Findings and Recommendations

The methodology developed from this research has proven successful in predicting an overrun precursor and simulating the flight that triggered an overrun precursor. This was made possible by optimizing not only the model structure but also the data structure. Hypothesis

1.1 was focused on improving prediction accuracy and prediction range by optimizing the model and data parameters. The same was true for the simulation model where hypothesis 2.1 discussed whether the simulation model accuracy can improve by optimizing the model and data parameters. To optimize both models a HGA was developed by combining GA and QIGA. The HGA successfully optimized both models, which were described by experiments 1.1 and 2.1. The optimized models had the maximum prediction model accuracy, maximum simulation model accuracy, and enough prediction range for pilots to respond to the predictions.

The optimized prediction model performed better than most of the existing models and showed that the neural network machine learning models can learn flight behaviors from FOQA data. A table that compares the model prediction range and accuracy between this study and past studies is shown in Table 9.1. One important factor to point out is that

Table 9.1: Results comparison of machine learning prediction models for aircraft speed

Measure	This study	Puranik et al. [44]	Tong et al. [40, 45]	Diallo [41]	Kang et al. [42]
RMSE (kts)	2.4–3.7	2.98	0.0065	3.5	1.71
Prediction Time (s)	6 – 18 s (100 – 300 ft)	18 (300 ft)	1	Retrospective	3 (50 ft)
Prediction Type	Multi-Point	Single-Point	Single-Point	Sequential	Sequential

for this study the prediction was on the TAS of the aircraft while for the other studies the prediction was on GS. So the direct comparison of RMSE may be difficult. However, the relatively low RMSE and long prediction range show that experiment 1.1 was a success. Based on the results it can be concluded that the prediction accuracy and range do depend on the model and data parameters. Also, the prediction model developed in this study was capable of predicting three states instead of one, like the other models compared. Note, the prediction results for the other states showed that the model was unable to create a complex regression model of the flight state across different altitudes for these states. So alternate methods of pre-processing the data and prediction models may need to be considered to improve the prediction results for PTCH and IVV features. The reason why PTCH and IVV features are hard to predict is because of their cyclic nature. So an alternative metric

that provides similar information to PTCH and IVV with less cyclic motion may also be considered for prediction. Some recommendations when training and using the prediction model are as follows:

- As mentioned above, the prediction model should be trained with features other than PTCH or IVV, such as AOAC, for predicting vertical motion
- The predicted results should be used directly, without mapping to an overrun precursor. This is because the prediction model is a regression model of the predicted states so the prediction results can accurately represent the trend but not the minor details of a trajectory.

The simulation model also successfully generated full final approach trajectories with accuracy on par with a past study that generated a simulation model from test flight data. A table that compares the simulation accuracy between these two models is shown in Table 9.2 The results compared the RMSE of the longitudinal and vertical acceleration on

Table 9.2: Results comparison of machine learning simulation models of final approach

	This study		Roudbari and Saghafi [96]	
Measure	longitudinal accel.	vertical accel.	longitudinal accel.	vertical accel.
RMSE (ft/s^2)	0.3224	1.2052	2.7514	2.87

the body axis and the simulation model created from this study was more accurate. This showed that experiment 2.1 was a success. Some recommendations when training and using the simulation model are as follows:

- It is recommended that the simulation model is used for shorter flight phases, such as the final approach so that the error in each simulation step does not compound too much.
- It is also recommended that balanced data is collected across the entire control space so that the full simulation trajectory can be trusted.

- When training the simulation model it would be best to consider weather and runway data as well. Runway type and weather conditions have a significant impact on the landing process and incorporating these factors into the simulation will be important.

The optimized prediction and simulation results were used to perform experiments for the rest of the sub-research questions. For the prediction model, the significant parameters to the predicted flight states near touchdown were identified using the permutation of importance. This experiment, experiment 1.2, was performed based on hypothesis 1.2, which was proved by comparing the significant parameters to the parameters that are known to have a significant impact on overrun. The significant parameters were identified to be the altitude, angle of attack, core rpm, and air speeds. The other significant features identified are correlated features such as fan rpm, pitch angle, and body longitudinal acceleration. The identified significant features were expected features that have a direct relation to the predicted features. Based on the identified significant features, new precursors for overrun may be generated to evaluate runway overrun. After finding the significant features, the prediction model sensitivity analysis, experiment 1.3 and 4, were performed based on hypothesis 1.3 and 4. The key point in hypotheses 1.3 and 4 was that the prediction model accuracy varied with the number of data available to train the model. So the prediction model was expected to perform poorly when there are large winds and maneuvers because there is a lack of data representing those configurations. The outcome of the prediction model sensitivity analysis showed that indeed the model was sensitive to flare and wind. And some correlation between the number of data and prediction accuracy was observed. So the conclusion was training data should incorporate the full data range that the model will be used to predict. More importantly, the results of sensitivity analysis showed that ideally the prediction model can be divided by the flight environment because the prediction model is a complex regression model. A regression model is expected to improve prediction results with a smaller data range it represents. This is proved when the prediction accuracy was observed to increase when predicting only the autopilot flights.

For the simulation model, the sensitivity analysis focused first on identifying the control space of the model based on hypothesis 2.4. The analysis result based on experiment 2.4 showed that the model was accurate for most of the control range represented by the test data. Most of the regions of control space with lower accuracy had a lesser number of data to train that control space. Therefore, having sufficient data was critical to developing an accurate simulation model. There were a few regions of control space with relatively higher data counts but still had low model accuracy. A potential cause includes a need for a different training method to ensure that the model well represents the aircraft dynamics. The model sensitivity to wind and flare was evaluated with the test data as well. Since adverse wind and flare were both measured by the control features, model sensitivity to wind and flare was also expected to be related to the training data size, as mentioned in hypotheses 2.2 and 3. The results from experiments 2.2 and 3 showed that the model was definitely sensitive to flare because the error distributions across different levels of elevator deflection were different before and after the flare. Just like for hypothesis 2.4, the sensitivity to wind and flare ties back to the lack of data on the extreme wind or elevator control configuration with fewer data to train. Note, the model sensitivity to wind showed that the model accuracy was fairly consistent with wind speed for the control space other than the extremes. This indicates that the neural network models are capable of learning aircraft dynamics with the wind effect.

The list of all of the main contributions from this work is shown below.

1. The HGA has been developed by combining GA and QIGA

2. A framework for developing a simulation model and a prediction model with data and model structures optimized was introduced and tested

3. The flight features with a top contribution to airspeed, vertical speed, and pitch angle near touchdown are identified to be the altitude, angle of attack, engine core rpm, and air speeds

4. Used sensitivity analysis to prove the robustness of the developed prediction and simulation model against flare and adverse wind

5. Performed control space analysis of the simulation model to show that the model accuracy is consistent across the entire control space

6. Introduced how prediction and simulation output errors can be mapped to required landing distance metric to understand the output in terms of runway overrun

7. Demonstrated how prediction and simulations may be used to increase pilots' awareness to high-speed approach, an overrun precursor

9.2 Future Works

Recommendations for future work can be divided between problem formulation and model development. Below is the list of future works in terms of model development.

- The results have shown the importance of having enough data that is clean with small variations in the data. State estimation techniques that reduce noise, such as the Kalman filter, should be considered to improve the data so the prediction and simulation errors are not affected by the noise.

- Another method of reducing variation in the data is by clustering similar flights. Clustering flight based on the final approach trajectories and generating separate models for each cluster can potentially improve model accuracy.

- In terms of the prediction model, the state estimation technique will also be necessary when indexing the data by altitude so that the size of each step is equal. The flight data is recorded in time. When data is indexed by altitude, let's say every 10 ft, it is likely that there is a $\pm$ 5 ft for each step. Since the frequency of the sensor cannot be changed, the state estimation technique may resolve this variation.

- For the training process, alternate objective functions that emphasize the more significant features or altitudes may improve both models. This includes weighted RMSE based on the significance of the features being estimated. So an iterative process of training a model, identifying the significant features of the model, and then training a new model with RMSE weighted based on the significance may improve the overall accuracy of the simulation model.

Below is the list of future works in terms of problem formulation.

- This study utilized existing precursors and precursor definitions to evaluate the overrun risk. While these precursors have shown effectiveness in representing the aggregate flight trend, the definitions use few features with a single threshold. However, aircraft dynamics are nonlinear and have coupling effects. New precursors based on the significant parameters identified in this study should be identified to better understand the causes of runway overrun. When the new overrun precursors are identified, a new prediction model may need to be developed with different prediction features.

- The effectiveness of the proposed methodology in predicting precursors to other accidents should also be considered. Runway overrun alone has multiple precursors with high correlation. Using the predicted flight states to simulate these precursors and finding the final approach trajectories that can minimize the exposure to these precursors should be considered.

- The simulation model should also be improved to model the aircraft dynamics for the entire flight and not just the final approach. This is necessary for simulating and learning the flight behaviors of other accidents unrelated to the final approach.

- The simulation model may also be used for online system identification for fault identification. Anomaly is flight behavior that may be detected in flight to notify damages to the aircraft.

- Data is key creating an accurate neural network. The simulation model could also be used to collect more data used for training the prediction and simulation model.

REFERENCES

[1] International Civil Aviation Organization, *Effects of novel coronavirus (covid-19) on civil aviation: Economic impact analysis*, https://www.icao.int/sustainability/Documents/COVID-19/ICAO_Coronavirus_Econ_Impact.pdf, Retrieved: 9/2022, 2022.

[2] ——, *Economic development dec 2020: Air transport monthly monitor*, https://www.icao.int/sustainability/Documents/MonthlyMonitor-2020/MonthlyMonitor_Dec_2020.pdf, Retrieved: 3/2021, 2021.

[3] ——, *Economic development dec 2022: Air transport monthly monitor*, https://www.icao.int/sustainability/Documents/MonthlyMonitor-2022/MonthlyMonitor_Apr_2022.pdf, Retrieved: 9/2022, 2022.

[4] International Air Transport Association, *Safety report 2021*, https://www.iata.org/en/publications/safety-report/, Retrieved: 9/2022, 2021.

[5] *Introduction to ROPS*, RGS WG/2, Airbus, Cairo Egypt: AIRBUS Operations S.A.S, May 2015.

[6] Flight Safety Foundation, *Reducing the risk of runway excursions*, https://flightsafety.org/files/RERR/fsf-runway-excursions-report.pdf, Retrieved: 3/2021, 2009.

[7] ——, "Fsf rsi briefing note: Pilot braking action report," FSF, alexandria, VA, USA, Tech. Rep., 2009.

[8] National Transport Safety Board, *Aircraft accident report: Runway overrun and collision southwest airlines flight 1248*, https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR0706.pdf, Retrieved: 3/2021, 2007.

[9] ——, *Aircraft accident report: Runway overrun during landing american airlines flight 1420*, https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR0102.pdf, Retrieved: 3/2021, 2001.

[10] ——, *Aircraft accident report: Runway excursion during landing delta airlines flight 1086*, https://www.ntsb.gov/investigations/AccidentReports/Reports/AAR1602.pdf, Retrieved: 3/2021, 2016.

[11] Federal Aviation Administration, *National runway safety plan : 2018 - 2020*, https://www.faa.gov/airports/runway_safety/publications/media/NRSP-RSC-47.pdf, Retrieved: 3/2021, 2021.

[12] ——, *Flight operational quality assurance*, https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_120-82.pdf, Retrieved: 3/2021, 2004.

[13] F. Bati, F. Lee, Y. Li, and N. Scialli, "Runway safety metric: Weighting scheme," *Federal Aviation Administration*, Aug. 2017.

[14] G. Konrad *et al.*, "Development of a predictive runway overrun awareness and alerting system," Aviation Electronics Europe, Munich Germany: AEE, Jun. 2018.

[15] A. Jacob, R. Lignee, and F. Villaume, "The runway overrun prevention system," Airbus, Blagnac Cedex France, Tech. Rep. Safety First 08, 2009.

[16] M. Jenkins and R. F. J. Aaron, "Reducing runway landing overruns," *Aero Magazine*, no. 47, pp. 15–19, 2012.

[17] Flight Safety Foundation, Eurocontrol, "Global action plan for the prevention of runway excursions part 1 - recommendations," Tech. Rep. 1, Jan. 2021.

[18] International Civil Aviation Organization, *Accident statistics*, https://www.icao.int/safety/iStars/Pages/Accident-Statistics.aspx, Accessed: 09/2022, 2022.

[19] P. Hart *et al.*, "Effectiveness of a computer-based educational program on nurses' knowledge, attitude, and skill level related to evidence-based practice," *Worldviews on Evidence-Based Nursing*, vol. 5, no. 2, pp. 75–84, 2008. eprint: https://sigmapubs.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1741-6787.2008.00123.x.

[20] A. A. Bakarman, "Attitude, skill, and knowledge:(ask) a new model for design education," *Proceedings of the Canadian Engineering Education Association (CEEA)*, Jul. 2005.

[21] R. Masoodi, F. Alhani, J. Moghadassi, and M. Ghorbani, "The effect of family-centered empowerment model on skill, attitude , and knowledge of multiple sclerosis caregivers," *Journal of Birjand University of Medical Sciences*, vol. 17, no. 2, 2010. eprint: http://journal.bums.ac.ir/article-1-659-en.pdf.

[22] International Air Transportation Association, *Unstable approaches: Risk mitigation policies, procedures, and best practices*, 2nd ed., IATA, Montreal, Quebec, Canada, 2016.

[23] European Aviation Safety Agency, *Review of accident precursors for runway excursions*, 1st ed., EASA, Cologne, Germany, May 2018.

[24] Interanational Air Transportation Association, "Runway safety accident analysis report: 2010-2014," IATA, Montreal, Canada, Tech. Rep., 2015.

[25] Federal Aviation Administration, *Airplane Flying Handbook (FAA-H-8083-3B)*. Skyhorse Publishing Inc., 2016, ch. 8, pp. 2–36.

[26] A. Campbell, P. Zaal, J. Schroeder, and S. Shah, "Development of possible go around criteria for transport aircraft," 3198, Session: Reliability and Safety, American Institute of Aeronautics and Astronautics, Atlanta, Georgia, USA: AIAA, Jun. 2018.

[27] Federal Aviation Administration, *Continuous descent final approach*, https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_120-108.pdf, Retrieved: 4/2021, 2011.

[28] ——, *Mitigating the risks of a runway overrun upon landing*, https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_91-79A_Chg_2.pdf, Retrieved: 4/2021, 2018.

[29] Airbus, "Flight operations briefing notes, landing techniques, bounce recovery - rejected landing," Blagnac, Cedex, France, Tech. Rep. 2, May 2005.

[30] P. Wu, M. Voskuijl, and L. Veldhuis, "An approach to estimate aircraft touchdown attitudes and control inputs," *Aerospace Science and Technology*, vol. 71, pp. 201–213, 2017.

[31] Airbus, "Flight operations briefing notes, landing techniques, croswind landings," Blagnac, Cedex, France, Tech. Rep. 3, Mar. 2008.

[32] G. Van Es, "Braking capabilities on flooded runways: Flight test results obtained with a business jet," in *AIAA Flight Testing Conference*, Denver, Colorado, USA, 2017, p. 3651.

[33] Federal Aviation Administration, *Takeoff performance data for operations on contaminated runways*, https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_25-31.pdf, Retrieved: 4/2021, 2015.

[34] T. J. Yager, "Factors influencing aircraft ground handling performance," NASA Langley Research Center, Hampton VA, USA, Technical Memorandum 19830019708, Jun. 1983.

[35] A. Trujillo, "Airline transport pilot preferences for predictive information," National Aeronautics and Space Administration, Technical Memorendum NASA-TM-4702, Feb. 1996.

[36] ——, "Pilot mental workload with predictive system status information," in *Proceedings Fourth Annual Symposium on Human Interaction with Complex Systems*, 1998, pp. 73–80.

[37] S. Jacobson, "Aircraft loss of control causal factors and mitigation challenges," in *AIAA Guidance, navigation, and control conference*, Toronto, Ontario, Canada, 2010, pp. 8007–8022.

[38] J. R. Merkt, "Flight energy management training: Promoting safety and efficiency," 2013.

[39] T. Puranik, H. Jimenez, and D. Mavris, "Energy based metrics for safety analysis of general aviation operations," in *Journal of Aircraft*, vol. 54, 2017, pp. 2285–2297.

[40] C. Tong, X. Yin, S. Wang, and Z. Zheng, "A novel deep learning method for aircraft landing speed prediction based on cloud-based sensor data," *Future Generation Computer Systems*, vol. 88, pp. 552–558, 2018.

[41] O. N. Diallo, "A predictive aircraft landing speed model using neural network," in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, 2012, pp. 3D2-1-3D2–10.

[42] Z. Kang *et al.*, "A deep sequence-to-sequence method for aircraft landing speed prediction based on qar data," in *Web Information Systems Engineering – WISE 2020*, vol. 12343, 2020, pp. 516–530, ISBN: 978-3-030-62008-0.

[43] J. Post, "Identification and analysis of veer-off risk factors in accidents/incidents," Netherlands Aerospace Center NLR, Tech. Rep. D3.4, 2015.

[44] T. G. Puranik, N. Rodriguez, and D. N. Mavris, "Towards online prediction of safety-critical landing metrics in aviation using supervised machine learning," *Transportation Research Part C: Emerging Technologies*, vol. 120, p. 102 819, 2020.

[45] C. Tong *et al.*, "An innovative deep architecture for aircraft hard landing prediction based on time-series sensor data," *Applied Soft Computing*, vol. 73, pp. 344–349, 2018.

[46] H. Zhang and T. Zhu, "Aircraft hard landing prediction using lstm neural network," in *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*, ser. ISCSIC '18, Stockholm, Sweden: Association for Computing Machinery, 2018, ISBN: 9781450366281.

[47] G. Jarry, D. Delahaye, and E. Feron, "Approach and landing aircraft on board parameters estimation with lstm networks," in *AIDA-AT 2020 1st Conference on Artificial Intelligence and Data Analytics in Air Transportation*, Singapore, 2020, ISBN: 978-1-7281-5381-0.

[48] Insightful Corporation and JetBlue AIrways, "Applicaation of insightful corporation(r)'s data mining algorithms to foqa data at jetblue airways: A technology

demonstration in partnership with the federal aviation administration an the global aviation information network (gain)," Tech. Rep., 2005.

[49] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: Algorithm and aviation safety case study," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10, Washington, DC, USA: Association for Computing Machinery, 2010, pp. 47–56, ISBN: 9781450300551.

[50] H. Lv, J. Yu, and T. Zhu, "A novel method of overrun risk measurement and assessment using large scale qar data," in *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2018, pp. 213–220.

[51] S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03, Washington, D.C.: Association for Computing Machinery, 2003, pp. 29–38, ISBN: 1581137370.

[52] L. Li, S. Das, R. Hansman, R. Palacios, and A. Srivastava, "Analysis of flight data using clustering techniques for detecting abnormal operations," *Journal of Aerospace Information Systems*, vol. 12, pp. 1–12, Sep. 2015.

[53] S. Budalakoti, A. N. Srivastava, and M. E. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 1, pp. 101–113, 2009.

[54] L. Li, M. Gariel, R. J. Hansman, and R. Palacios, "Anomaly detection in onboard-recorded flight data using cluster analysis," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, pp. 1–31.

[55] A. Nanduri and L. Sherry, "Anomaly detection in aircraft data using recurrent neural networks (rnn)," in *2016 Integrated Communications Navigation and Surveillance (ICNS)*, 2016, pp. 5C2-1-5C2–8.

[56] L. Basora, X. Olive, and T. Dubot, "Recent advances in anomaly detection methods applied to aviation," *Aerospace*, vol. 6, no. 11, 2019.

[57] C. Hu, S. Zhou, Y. Xie, and W. Chang, "The study on hard landing prediction model with optimized parameter svm method," in *2016 35th Chinese Control Conference (CCC)*, 2016, pp. 4283–4287.

[58] X. Qiao, W. Chang, S. Zhou, and X. Lu, "A prediction model of hard landing based on rbf neural network with k-means clustering algorithm," in *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2016, pp. 462–465.

[59] D. Martínez *et al.*, "Forecasting unstable approaches with boosting frameworks and lstm networks," in *9th SESAR Innovation Days*, Dec. 2019.

[60] H. Lee *et al.*, "Critical parameter identification for safety events in commercial aviation using machine learning," *Aerospace*, vol. 7, no. 6, 2020.

[61] G. Van Es, "Hydroplaning of modern aircraft tires," National Aerospace Laboratory (NLR), Montreal, Canada, Tech. Rep. NLR-TP-2001-242, May 2001.

[62] L. Fu and P. Li, "The research survey of system identification method," in *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, Hangzhou, China, 2013, pp. 397–401.

[63] J. Valasek and W. Chen, "Observer/kalman filter identification for online system identification of aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 347–353, 2003.

[64] G. Alcalay, C. Seren, G. Hardier, M. Delporte, and P. Goupil, "An adaptive extended kalman filter for monitoring and estimating key aircraft flight parameters," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 620–627, 2018, 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018.

[65] B. Woo, O. Park, S. Kim, J. Suk, and Y. Kim, "Real-time aircraft upset detection and prevention based on extended kalman filter," *Journal of the Korean Society for Aeronautical and Space Sciences*, vol. 49, no. 9, pp. 724–733, 2017.

[66] A. Costalago Meruelo, D. M. Simpson, S. M. Veres, and P. L. Newland, "Improved system identification using artificial neural networks and analysis of individual differences in responses of an identified neuron," *Neural Networks*, vol. 75, pp. 56–65, 2016.

[67] Y. Wang, "A new concept using lstm neural networks for dynamic system identification," in *2017 American Control Conference (ACC)*, Seatle, WA, USA, 2017, pp. 5324–5329.

[68] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, May 1994.

[69] C.-T. Sun, "Rule-base structure identification in an adaptive-network-based fuzzy inference system," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 1, pp. 64–73, 1994.

[70] A. Tewari and M.-U. Macdonald, "Knowledge-based parameter identification of tsk fuzzy models," *Applied Soft Computing*, vol. 10, no. 2, pp. 481–489, 2010.

[71] C. Li, J. Zhou, X. Xiang, Q. Li, and X. An, "T-s fuzzy model identification based on a novel fuzzy c-regression model clustering algorithm," *Eng. Appl. Artif. Intell.*, vol. 22, no. 4–5, pp. 646–653, Jun. 2009.

[72] E. A. Morelli and V. Klein, "Application of system identification to aircraft at nasa langley research center," vol. 42, no. 1, pp. 12–24, 2005.

[73] A. Dorobantu, A. Murch, B. Mettler, and G. Balas, "System identification for small, low-cost, fixed-wing unmanned aircraft," vol. 50, no. 4, pp. 1117–1130, 2013.

[74] G. Chowdhary, W. Debusk, and E. Johnson, "Real time system identification of a small multiengine aircraft with structural damage," in *AIAA Infotech@Aerospace 2010*, Atlanta, Georgia, USA, 2010, pp. 1–6.

[75] S. A. Bagherzadeh, "Nonlinear aircraft system identification using artificial neural networks enhanced by empirical mode decomposition," *Aerospace Science and Technology*, vol. 75, pp. 155–171, 2018.

[76] E. A. Morelli, "Low-order equivalent system identification for the tu-144ll supersonic transport aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 354–362, 2003.

[77] D. J. Linse and R. F. Stengel, "Identification of aerodynamic coefficients using computational neural networks," vol. 16, no. 6, pp. 1018–1025, 1993.

[78] J. Bauer and D. Andrisani, "Estimating short-period dynamics using an extended kalman filter," in *Orbital Debris Conference: Technical Issues and Future Directions*, Baltimore, MD, USA, 1990.

[79] G.-g. Seo, Y. Kim, and S. Saderla, "Kalman-filter based online system identification of fixed-wing aircraft in upset condition," *Aerospace Science and Technology*, vol. 89, pp. 307–317, 2019.

[80] S. Amin, V. Gerhart, and E. Rodin, "System identification via artificial neural networks - applications to on-line aircraft parameter estimation," in *1997 World Aviation Congress*, Anaheim, CA, USA, 1997.

[81] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 422–431, 1995.

[82] R. Aykan, C. Hadjiyev, and F. Caliskan, "Aircraft icing detection, identification and reconfigurable control based on kalman filtering and neural networks," in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, San Francisco, California, USA, 2005.

[83] G. Van Es, "Braking capabilities on flooded runways: Flight test results obtained with a business jet," in *AIAA Flight Testing Conference*, Denver, Colorado, USA, 2017.

[84] M. K. Wahi, "Airplane brake-energy analysis and stopping performance simulation," *Journal of Aircraft*, vol. 19, no. 10, pp. 688–694, 1979.

[85] H. R. Pasindu, T. F. Fwa, and G. P. Ong, "Computation of aircraft braking distances," *Transportation Research Record*, vol. 2214, no. 1, pp. 126–135, 2011. eprint: https://doi.org/10.3141/2214-16.

[86] M. S. Selezneva, K. A. Neusypin, and A. Proletarsky, "Navigation complex with adaptive non-linear kalman filter for unmanned flight vehicle," *Metrology and Measurement Systems*, vol. 26, no. 3, pp. 541–550, 2019.

[87] K. Soal, Y. Govers, J. Bienert, and A. Bekker, "System identification and tracking using a statistical model and a kalman filter," *Mechanical Systems and Signal Processing*, vol. 133, p. 106 127, 2019.

[88] A. Stelmach, "Modeling of selected aircraft flight phases using data from flight data recorder," *Archives of Transport*, vol. 23, pp. 541–555, 4 2012.

[89] C. Lee and C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2000.

[90] G. Kouba, R. M. Botez, and N. Boely, "Fuzzy logic method use in f/a-18 aircraft model identification," *Journal of Aircraft*, vol. 47, no. 1, pp. 10–17, 2010.

[91] D. Furey, O. Rediniotis, C. Moore, N. Schaeffler, and D. Telionis, "Fuzzy logic and neural network system identification for high alpha delta wing maneuvers with deployable control surfaces," in *34th Aerospace Sciences Meeting and Exhibit*, Reno, NV, USA, 1996.

[92] F. Saghafi and A. Roudbari, "Modeling and identification of fighter aircraft nonlinear flight dynamics, by using fuzzy logic algorithm," in *Scientific Cooperations International Workshops on Engineering Branches*, Istanbul, Turkey, 2014.

[93] N. Wahid and M. F. Rahmat, "Pitch control system using lqr and fuzzy logic controller," in *2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, 2010, pp. 389–394.

[94] D. Gao, Z. Sun, and T. Du, "Dynamic surface control for hypersonic aircraft using fuzzy logic system," in *2007 IEEE International Conference on Automation and Logistics*, 2007, pp. 2314–2319.

[95] J. Harris, F. Arthurs, J. V. Henrickson, and J. Valasek, "Aircraft system identification using artificial neural networks with flight test data," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA, USA, 2016, pp. 679–688.

[96] A. Roudbari and F. Saghafi, "Intelligent modeling and identification of aircraft nonlinear flight," *Chinese Journal of Aeronautics*, vol. 27, no. 4, pp. 759–771, 2014.

[97] D. Wang, K. Lum, and G. Yang, "Parameter estimation of arx/narx model: A neural network based method," in *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, vol. 3, 2002, 1109–1113 vol.3.

[98] R. Kumar, S. Srivastava, J. R. Gupta, and A. Mohindru, "Comparative study of neural networks for dynamic nonlinear systems identification," *Soft Comput.*, vol. 23, no. 1, pp. 101–114, Jan. 2019.

[99] O. Ogunmolu, X. Gu, S. Jiang, and N. Gans, "Nonlinear systems identification using deep dynamic neural networks," in *American Control Conference, 2017*, 2016.

[100] T. Qi and L. Zhang, "Data driven nonlinear dynamical systems identification using multi-step cldnn," *American Institute of Physics Advances*, vol. 9, no. 085311, 8 2019.

[101] Z. Boussada, O. Curea, A. Remaci, H. Camblong, and N. M. Bellaaj, "A nonlinear autoregressive exogenous (narx) neural network model for the prediction of the daily direct solar radiation," *10th International Conference On Sustainable Energy and Environmental Protection*, vol. 11, no. 3, p. 620, 2018.

[102] S. Hochreiter and J. Schmidhuber, "Long short term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[103] K. Cho *et al.*, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1724–1734.

[104] International Air Transportation Association, *Guidance material for instructor and evaluator training*, 1st ed., IATA, Montreal, Quebec, Canada, 2018.

[105] J. Oehling and D. J. Barry, "Using machine learning methods in airline flight data monitoring to generate new operational safety knowledge from existing data," *Safety Science*, vol. 114, pp. 89–104, 2019.

[106] B. de Courville, "Go-around decision and maneuver: How to make it safer?" In *FSF 63rd annual IASS*, Milan, Italy, 2018.

[107] International Air Transportation Association, *Data report for evidence-based training*, 1st ed., IATA, Montreal, Quebec, Canada, 2014, ISBN: 978-92-9252-228-5.

[108] T. G. Puranik *et al.*, "Review of flight data driven metrics for aircraft braking performance assessment on runways," in *ATM Seminar 2021*, (unpublished), Virtual Event, 2021.

[109] M. Memarzadeh, B. Matthews, and T. Templin, "Multi-class anomaly detection in flight data using semi-supervised explainable deep learning model," in *AIAA SciTech 2021 Forum*, Virtual Event, 2021.

[110] E. V. Odisho, "Predicting pilot misperception of runway excursion risk through machine learning algorithms of recorded flight data," Ph.D. dissertation, Embry Riddle Aeronautical University, Daytona Beach, Florida, Feb. 2020.

[111] N. E. Daidzic, "Modeling and computation of the maximum braking energy speed for transport category airplanes," *Journal of Aviation Technolgoy and Engineering*, vol. 6, pp. 2–25, 2 2017.

[112] P. Castaigns and L. De-Baudus, *Control your speed... during descent, approach and landing*, Safety First 24, Airbus, 2017, pp. 6–25.

[113] T. Blajev and W. Curtis, *Go-around decision-making and execution project*, Flight Safety Foundation, 2017.

[114] J. Ackley, T. G. Puranik, and D. N. Mavris, "A supervised learning approach for safety event precursor identification in commercial aviation," in *AIAA Aviation 2020 Forum*, Virtual Event, 2020.

[115] Federal Aviation Administration, *Flight standards information management system 8900.1*, FAA, 2011, ch. 54.

[116] ——, *Flight standards information management system 8900.1*, FAA, 2018, ch. 3.

[117] H. Lee, T. G. Puranik, and D. N. Mavris, "Deep Spatio-Temporal Neural Networks for Risk Prediction and Decision Support in Aviation Operations," *Journal of Computing and Information Science in Engineering*, vol. 21, no. 4, Feb. 2021, 041013. eprint: https://asmedigitalcollection.asme.org/computingengineering/article-pdf/21/4/041013/6643914/jcise_21_4_041013.pdf.

[118] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6765–6816, Jan. 2017.

[119] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "Boa: The bayesian optimization algorithm," in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 1*, ser. GECCO'99, Orlando, Florida: Morgan Kaufmann Publishers Inc., 1999, pp. 525–532, ISBN: 1558606114.

[120] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'12, Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 2951–2959.

[121] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing neural networks using genetic algorithms," in *Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 379–384.

[122] Z. He, J. Zhou, H.-N. Dai, and H. Wang, "Gold price forecast based on lstm-cnn model," in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2019, pp. 1046–1053.

[123] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using cnn-lstm neural networks," *Energy*, vol. 182, pp. 72–81, 2019.

[124] F. U. M. Ullah, A. Ullah, I. U. Haq, S. Rho, and S. W. Baik, "Short-term prediction of residential power energy consumption via cnn and multi-layer bi-directional lstm networks," *IEEE Access*, vol. 8, pp. 123 369–123 380, 2020.

[125] X. Shi and D. Yeung, "Machine learning for spatiotemporal sequence forecasting: A survey," *ArXiv*, vol. abs/1808.06865, 2018.

[126] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single layer & multi-layer long short-term memory (lstm) model with intermediate variables for weather

forecasting," *Procedia Computer Science*, vol. 135, pp. 89–98, 2018, The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) Empowering Smart Technology in Digital Era for a Better Life.

[127] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *Proceedings of the 16th International Conference on Artificial Neural Networks - Volume Part I*, ser. ICANN'06, Athens, Greece: Springer-Verlag, 2006, pp. 632–640, ISBN: 3540386254.

[128] E. D. Harrison, "A methodology for predicting and mitigating loss of control incidents for general aviation aircraft," Ph.D. dissertation, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA, Dec. 2018.

[129] S. Chen, S. A. Billings, and P. M. Grant, "Non-linear system identification using neural networks," *International Journal of Control*, vol. 51, no. 6, pp. 1191–1214, 1990.

[130] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "Lstm-based flight trajectory prediction," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[131] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, "Modeling trajectories with recurrent neural networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3083–3090.

[132] S. Ayhan and H. Samet, "Aircraft trajectory prediction made easy with predictive analytics," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 21–30.

[133] X. Zhang and S. Mahadevan, "Bayesian neural networks for flight trajectory prediction and safety assessment," *Decision Support Systems*, vol. 131, p. 113 246, 2020.

[134] C. Che, H. Wang, Q. Fu, and X. Ni, "Combining multiple deep learning algorithms for prognostic and health management of aircraft," *Aerospace Science and Technology*, vol. 94, p. 105 423, 2019.

[135] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[136] W. Yin, K. Kann, M. Yu, and H. Schutze, "Comparative study of cnn and rnn for natural language processing," *ArXiv*, vol. 1702.01923, 2017.

[137] S. Indolia, A. K. Goswami, S. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network- a deep learning approach," *Procedia Computer*

Science, vol. 132, pp. 679–688, 2018, International Conference on Computational Intelligence and Data Science.

[138] J. Reason, "Safety Paradoxes and Safety Culture," *Injury Control and Safety Promotion*, vol. 7, no. 1, pp. 3–14, Mar. 2000, doi:10.1076/1566-0974(200003)7:1;1-V;FT003.

[139] S. Amin, V. Gerhart, and R. E., "System identification via artificial neural networks - applications to on-line aircraft parameter estimation," in *World Aviation Congress*, Anaheim, CA: AIAA, 1997.

[140] S. A. Bagherzadeh, "Nonlinear aircraft system identification using artificial neural networks enhanced by empirical mode decomposition," *Aerospace Science and Technology*, vol. 75, pp. 155–171, 2018.

[141] J. Harris, F. Arthurs, J. V. Henrickson, and J. Valasek, "Aircraft system identification using artificial neural networks with flight test data," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA: IEEE, 2016, pp. 679–688.

[142] K. Kirkpatrick, J. J. May, and J. Valasek, "Aircraft system identification using artificial neural network," in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, TX: AIAA, 2013.

[143] M. K. Samal, S. Anavatti, and M. Garratt, "Neural network based system identification for autonomous flight of an eagle helicopter," in *The International Federation of Automatic Control*, vol. 41, Seoul, Korea: 7421-7426, 2008.

[144] S. S. Shamsudin, "The development of neural network based system identification and adaptive flight control for an autonomous helicopter system," University of Canterbury, Christchurch, New Zealand, Aug. 2013.

[145] K. Zarzycki and M. Ławryńczuk, "Lstm and gru neural networks as models of dynamical processes used in predictive control: A comparison of models developed for two chemical reactors," *Sensors*, vol. 21, no. 16, 2021.

[146] W. Gu, K. P. Valavanis, M. J. Rutherford, and A. Rizzo, "A survey of artificial neural networks with model-based control techniques for flight control of unmanned aerial vehicles," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA: IEEE, 2019, pp. 362–371.

[147] S. Nikbakht, C. Anitescu, and T. Rabczuk, "Optimizing the neural network hyperparameters utilizing genetic algorithm," pp. 407–426, A 22 2021.

[148] S. Lee, J. Kim, H. Kang, D.-Y. Kang, and J. Park, "Genetic algorithm based deep learning neural network structure and hyperparameter optimization," *Applied Sciences*, vol. 11, no. 2, 2021.

[149] P. Hou, B. Zhao, O. Jolliet, J. Zhu, P. Wang, and M. Xu, "Rapid prediction of chemical ecotoxicity through genetic algorithm optimized neural network models," *ACS Sustainable Chemistry & Engineering*, vol. 8, no. 32, pp. 12 168–12 176, 2020.

[150] N. M. Aszemi and P. Dominic, "Hyperparameter optimization in convolutional neural network using genetic algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.

[151] C. Li *et al.*, *Genetic algorithm based hyper-parameters optimization for transfer convolutional neural network*, 2021.

[152] T. Hinz, N. Navarro-Guerrero, S. Magg, and S. Wermter, "Speeding up the hyperparameter optimization of deep convolutional neural networks," *International Journal of Computational Intelligence and Applications*, vol. 17, no. 02, Jun. 2018.

[153] H. Alibrahim and S. A. Ludwig, "Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021, pp. 1551–1559.

[154] Z. Tian and S. J. Fong, "Survey of meta-heuristic algorithms for deep learning training," 2016.

[155] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.

[156] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 61–66.

[157] Y. Dong and J. Zhang, "An improved hybrid quantum optimization algorithm for solving nonlinear equations," *Quantum Information Processing*, vol. 20, no. 4, Apr. 2021.

[158] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 2, 2000, 1354–1360 vol.2.

[159] J. King *et al.*, *Quantum-assisted genetic algorithm*, 2019.

[160] S. Mondal and A. Tsourdos, "Two-dimensional quantum genetic algorithm: Application to task allocation problem," *Sensors*, vol. 21, no. 4, 2021.

[161] E. P. Adorio and R. January, "Mvf - multivariate test functions library in c for unconstrained global optimization," 2005.

[162] A. Abs da Cruz, M. Vellasco, and M. Pacheco, "Quantum-inspired evolutionary algorithm for numerical optimization," in *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 2630–2637.

[163] C. Qin, Y. Liu, and J. Zheng, "A real-coded quantum-inspired evolutionary algorithm for global numerical optimization," in *2008 IEEE Conference on Cybernetics and Intelligent Systems*, 2008, pp. 1160–1164.

[164] H. Wang, J. Liu, J. Zhi, and C. Fu, "The improvement of quantum genetic algorithm and its application on function optimization," in *Mathematical Problems in Engineering*, 2013.

[165] H. Lee *et al.*, "Aircraft braking performance classification through braking rollout clustering," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, 2022.

[166] L. Breiman, "Random forest," *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

[167] Federal Aviation Administration, *Pilot windshear guide*, https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC00-54.pdf, Retrieved: 4/2021, 1988.

VITA

Hyunki Lee was born in Pyeong Taek, South Korea, on Feb 17, 1992. He attended elementary school in Cheng Du, China, and graduated from Cheng Du International School in May 2010. He received his Bachelor's degree in Aerospace Engineering from the Georgia Institute of Technology in December 2018. Then he enrolled in the Master's program the following January and joined the Aerospace Systems Design Laboratory (ASDL) at the Georgia Institute of Technology in January 2019. He earned his Master of Science degree in August 2020 and continued with the Doctor of Philosophy program at ASDL. During his time in ASDL, he worked on many projects including but not limited to hybrid electric propulsion modeling and analysis, civil aviation accident precursor identification, and braking effectiveness on contaminated runway analysis. His research interests include the conceptual design of hybrid propulsion aircraft, system model design, optimization, and flight risk analysis and prediction.