Development of High Fidelity Flight Dynamics model for Rotary UAV using Deep Learning

A thesis submitted in fulfilment of the requirements

for the degree of Master of Technology

by

Praveen Kumar Ranjan

DEPARTMENT OF AEROSPACE ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

June 2019

Certificate

It is certified that the work contained in this thesis entitled "Development of High Fideli ty flight dynamics model for Rotary UAV using Deep Learning" by "Praveen Kumar Ranjan" has been carried out under my supervision and that it has not been submitted elsewhere for a degree.

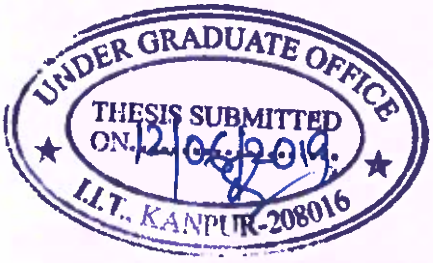
Dr. Abhishek

Associate Professor

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

June 2019



Statement of Thesis Preparation

I, **Praveen Kumar Ranjan**, declare that this thesis titled, **"Development of High Fidelity Flight Dynamics Model for Rotary UAV using Deep Learning"**, hereby submitted in partial fulfillment of the requirements for the degree of **B.tech-M.tech (dual)** and the work contained herein are my own. I further confirm that:

1. The "Thesis Guide" was referred to for preparing the thesis.
2. Specifications regarding thesis format have been closely followed.
3. The contents of the thesis have been organized based on the guidelines.
4. The thesis has been prepared without resorting to plagiarism.
5. All sources used have been cited appropriately.
6. The thesis has not been submitted elsewhere for a degree.

Name: Praveen Kumar Ranjan

Roll No.: 14807494

Department of Aerospace Engineering

June, 2019

Abstract

Name of the student: **Praveen Kumar Ranjan** Roll No: **14807494**

Degree for which submitted: **M.Tech.** Department: **Aerospace Engineering**

Thesis title: **Development of High Fidelity Flight Dynamics model for Rotary UAV using Deep Learning**

Thesis supervisor: **Dr. Abhishek**

Month and year of thesis submission: **June 2019**

Rotary UAVs or helicopters by their capability of vertically taking-off and landing in rough terrains have been proved useful in many applications such as surveillance, product delivery and mapping. There has been an increasing demand for autonomy in UAVs providing them with the ability to make decisions of their own. Such research requires accurate modelling of the physical system dynamics for testing new control and navigation algorithms. Flight dynamics model representing actual vehicle behaviour forms an integral part of Flight Simulators to provide proof of concept. Helicopter is a complex system and developing highly accurate physics-based model is a challenging task. Thereby, it arises a need to create models using System Identification techniques on real flight data.

This thesis focuses on the development of a high Fidelity flight dynamics model of the Rotary UAV 'Align Trex-700' providing a general roadmap for the development of such models for any vehicle. The whole identification process is divided into two models: 1) Baseline model and 2) Grey-box(Deep Learning) Model. The baseline model is a Linear Time Invariant State-Space model which has been derived using CIFER based on Frequency Domain System Identification Technique. Two acceleration correction models ReLU non-linear and a simple Linear network are trained to predict corrections in the baseline model. To train the model forward flight and hovering sweep data was collected over a couple of

hours and validated in the time domain. Further optimisation of the deep learning model parameters has been discussed to ensure fast and accurate convergence during training. The ReLU model significantly outperforms other models, capturing all the non-linear behaviour for a larger regime of forward flight.

Keywords: System Identification, Deep Learning, Flight Dynamics Model

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to Dr. Abhishek, who supervised my thesis and helped me to get direction in life. The Helicopter and VTOL Laboratory IIT Kanpur, has been thriving with world-class research because of his guidance and encouragement. I want to sincerely thank him for exposing me to an excellent problem statement and encouraging me to work on it.

I would like to thank Sagar Setu who helped me with my thesis topic selection, giving me insights about practically using deep learning for simulations. I would like to thank B Sai. Tharun who helped me to quickly get accustomed with system identification concepts. I would also like to thank Amey Loya whose knowledge in machine learning helped me to clarify my doubts about my problem statement. I would also like to express my gratitude to lab staff at Helicopter and VTOL Laboratory, IIT Kanpur, who assisted me with the conduction of experiments and flying of vehicle.

Words fail me to express my deepest appreciation to Sainyam, Kaushik, Rajat, Koushal, Dev, Abhiroop and all my friends who are a constant source of support and creative inspiration. I also thank them for making my stay at IIT Kanpur memorable and worthwhile.

Most importantly, none of this would have been possible if not for the love and support from my parents and siblings. I would like to sincerely thank my mother for being a constant source of my inspiration and commitment towards hard work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Simulation brings the engineer into a special and intimate relationship with the system he or she is modelling and the helicopter is a classic example."

— G. D. Padfield

1.1 Objective and Motivation

The global Unmanned Aerial System market is projected to reach $21.47 billion by 2021. In India, this market has expanded exponentially in the last five years and is expected to grow to $885.7 million by 2021. UAVs have been useful in many sectors of the industry such as agriculture, railways, mining, construction, product delivery, security and surveillance. Helicopters or R-UAVs have the edge over fixed-wing UAVs because of their capability of hovering in the air and have been used to reach rough terrains where fixed-wing UAVs cannot reach.

Helicopters are complex system vibrating perpetually at high frequency, thereby having reduced signal-to-noise ratio in data measurements. They have higher order vehicle dynamics as compared to fixed wing vehicles because of the unsteady dynamics involved with the rotors. The aerodynamic forces acting on the vehicle and working state of the rotor changes dramatically when operating in different flight conditions such as hover, cruise, ascending and descending flight. A single linear model cannot represent vehicle behaviour accurately in all regimes. All these factors make modelling Helicopters a challenging job.

Physics-based models are too tedious to model, involving a lot of assumptions which causes the model to diverge for longer simulation time. System Identification uses experimental

input-output data collected from a plant to produce a mathematical representation of the system's dynamics. The Linear Time-invariant State-Space models produced by System Identification are unable to replicate nonlinear behaviour of the vehicle and have inadequate off-axis responses. Deep learning is part of a broader family of machine learning methods based on the layers used in artificial neural networks which have been widely used in classification, pattern analysis and feature extraction. Neural networks were inspired by synaptic structures in biological systems and can be used to create models to learn specific behaviour (Such a process is also known as unsupervised learning).

1.2 Literature Review

Modelling helicopters involves accurate prediction of the dynamic response of the rotor blade of the helicopter either by system identification or by modelling physics of the flight. However, the inflow dynamics involved with the rotor makes the development of Flight Dynamics model of the helicopter a challenging task than fixed wings. The developement of the Flight Dynamics model could be physics-based or using data drived methods. Dr J Gordon Leishman used simple Blade Element momentum theory and laws of aeromechanics for prediction of helicopter preformance [14]. In 1996 GD Padfield published a book [18] giving a roadmap for helicopter trim stability analysis and simulation modelling. The NASA Minimum-Complexity Helicopter Simulation Math Model uses first-order flapping equations along with model matching and estimation procedures, which enabled the modelling of helicopters from primary data sources such as flight manuals [10].

"Comprehensive Identification from Frequency Responses" (CIFER) is a statistical tool developed by NASA which has been used for frequency domain System Identification of full-scale rotorcrafts. Tichler et. Al demonstrated the frequency domain flight testing techniques on U.S. OH-58D [8] and BO-15 [26] helicopters. The comprehensive work on frequency domain system identification of large scale helicopter in can be found in a book by Tischler [16] . Frequency domain technique was later shown on small rotorcraft Yamaha R-50 to develop complete dynamic model both hover and cruise flight conditions. Mettler et. al [17] describes the development of a parameterized model for a small-scale unmanned helicopter (Yamaha R50 with 10 ft rotor diameter) and its identification using a frequency domain identification technique. Subodh Bhandari et al.[5] developed a 12-degree-of-freedom flight dynamics model for a small-scale unmanned helicopter in both hovering and forward flight. The frequency domain identification technique using CIFER has been fully discussed for small helicopter in M.Tech thesis by B. S. Tharun[25] and V.

Praneeth[27]. La Civita et al.[12] proposed a first-principles-based non-linear model and a frequency domain technique to fit the unknown parameters from flight data to control maneuvers .

Linear models have been extensively successful for rotorcraft identification capturing vital essence of vehicle dynamics about an operating which could be used for control system development using numerical analysis and design technique available for linear systems. However, a single linear model cannot represent the vehicle at all flight regime. ARMAX and NARMAX are another popular models for identification of discrete-time system based on auto regression. S. Setu et. Al. [23] used a non-linear and jerk prediction model for modelling dynamics of a model helicopter. S. Hashimot et. Al. [9] used ARMAX model for identification and development of large scale autonomous unmanned aerial vehicle. Suresh et. Al. [24] used a feed-forward neural network to identify dynamics of a helicopter from flight data. Mahendra K Samal et. Al. [21] carried out on-line and off-line identification both for coupled and uncoupled dynamics of UAV helicopter. These Neural network models are posed as black-box modelling problem, that do not try to understand the underlying physical charactersitics of the helicopter. P Abbeel et. Al.[3] used improved acceleration based parametrization and lagged error training to learn dynamics of small helicopter. Apprenticeship learning algorithms were applied to learn vehicle dynamics during specific manoeuvres, to obtain a reward function for optimal control of trajectories such as inverted flight, autorotation in [1], [2]. A Punjani et. Al. [19] further used a hybrid model consisting of regression-based baseline model and a deep learning network with Relu hidden layer for development of flight dynamics model of a helicopter. S. Setu [22] used combination of deep learning and physics based baseline model to identifiy dynamics of a helicopter to be used in a HILS (Hardware In-The Loop) simulation setup. This thesis is a part of S. Setu's work performing data-driven System Identification using a Frequency based SISO baseline model and a deep learning model for correcting the output of the baseline models to develop a high fidelity FDM for an Align Trex-700 for a larger regime of forward flight.

1.3 Description of Helicopter Flight Dynamics problem.

The following section gives a brief introduction about the six degree of freedom Flight Dynamics Model of a helicopter. The helicopter is free to simultaneously rotate and translate. Figure 1.1 shows the body frame fixed (x, y, z body axes) to the center of gravity of the helicopter. The principal variables along the body axis include translation velocities u, v,

w; body angular rates p, q, r; and the Euler angle ϕ, θ, ψ. The tilting of the rotor disc is represented by β_{1s} and β_{1c}.

The helicopter is controlled by changing the blade pitch angle and the tail pitch angle. The main rotor blades are connected via pitch link to the swashplate. The swashplate tilts and moves up-down to provide roll-pitch and heave motion respectively. The primary aim of the tail rotor is to balance the torque due to main rotor and maintain a constant heading direction. The tail rotor collective can also be changed to provide yaw motion.

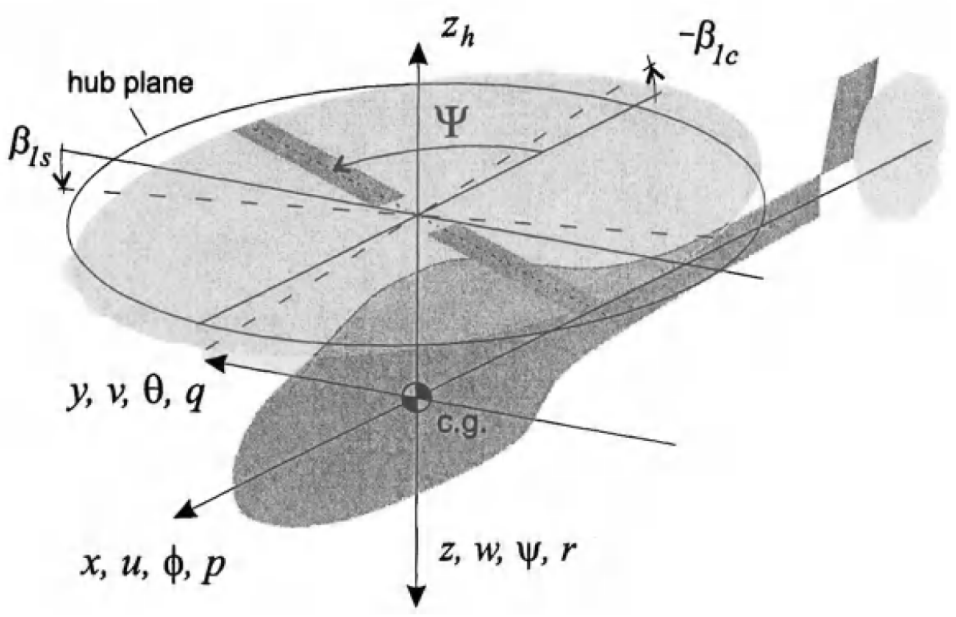


FIGURE 1.1: Helicopter with all frames of references

Our model helicopter uses three servos to tilt the swashplate and a single servo to control yaw pitch angle. These servos are controlled using the stick inputs from the transmitters. The four control inputs are Lateral cyclic, Longitudinal cyclic, Collective and yaw pedal providing roll, pitch, heave and yaw motion respectively. The following inputs and output data can be recorded easily by placing an onboard sensors:

- Inputs
  - Lateral Cyclic (δ_{lat})
  - Longitudinal Cyclic (δ_{lon})
  - Collective (δ_{col})

  - Pedal input (δ_{ped})

- Outputs
  - Attitude angles : $Roll(\phi), Pitch(\theta), Yaw(\psi)$
  - Angular Body Rates : $Rollrate(p), Pitchrate(q), Yawrate(r)$
  - Body Velocities : u, v, w.
  - Body Acceleraton: a_x, a_y, a_z.

The challenge is to derive a high fidelity model using the above set of data. It is necessary to understand that the time series data history of the inputs and the output can also be utilized to attain high accuracy in simulations.

1.4 System Identification

System identification refers to the extraction of mathematical models of the dynamical system using input-output data [15]. The model structure can be broadly classified into 1) White-box models 2) Black-Box Models 3) Grey-Box Models. The White-box models are derived using fundamental physical laws. Another common approach can be data-driven algorithms estimating parameters, depending on whether we make prior assumptions about the model or not. Grey-box modelling requires making assumptions about model structure based on some peculiarities and relating trends about the data collected, also known as sem-physical modelling. Most of the nonlinear modelling involves no assumptions about model structure, these type of models are known as Black-Box.

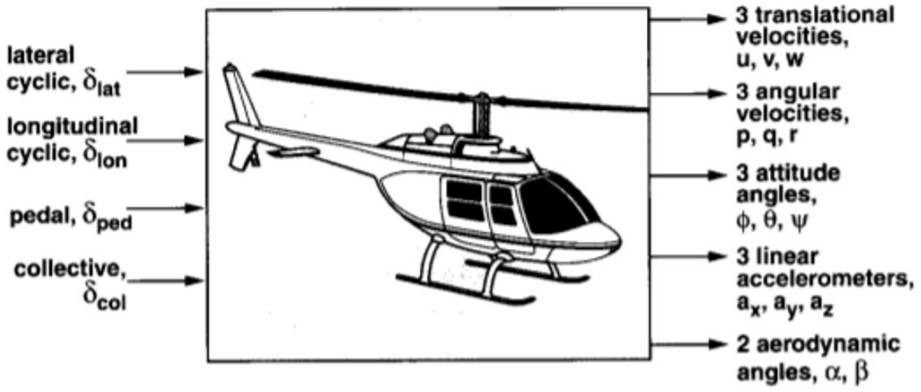


FIGURE 1.2: Helicopter input and output parameters [17]

Model complexity represents the size or flexibility of the model. The main aim of the identification process is to reduce memory and time complexity while increasing model accuracy. These models can be Linear, Nonlinear, hybrid, non-parametric differing in characteristics to be identified and mainly depend on techniques of identification used. Thereby the whole identification process can be seen as triangular optimisation problem among model complexity, information contents in data and sufficient validation. Neural Networks are mostly posed as black-box identification problem that does not contribute to the understanding of the internal dynamics in of the system. Figure 1.2 gives a better prespective to the inputs and outputs of a rotary UAV.

1.5 Deep Learning

Artificial Neural Networks (ANN's) are computing systems inspired by the biological nervous system. Each node of the ANN is made up of artificial neurons similar to biological neurons in the brain. A single Artificial neuron with adjustable synaptic weights and bias try to replicate its natural counterpart, thereby producing some output based on one or more inputs received. When these ANN's have multiple layers between input and output nodes, they are known as Deep Neural Networks (DNN's). The DNN's with a larger number of weights can be trained using similar learning algorithms as used in ANN's.

The motivation behind choosing a Deep Neural network is the Universal Approximation theorem which states '*An arbitrary continuous function, defined on [0,1] can be arbitrary well uniformly approximated by a multilayer feed-forward neural network with one hidden layer (that contains only finite number of neurons) using neurons with arbitrary activation functions in the hidden layer and a linear neuron in the output layer.*'[7]. It was also later shown that the universal approximation nature of the multilayer feed-forward network is not because of the choice of activation function but because of the network structure [11].

1.5.1 Activation functions

In Artificial Neural Networks, Activation function defines the output on the basis of inputs to the node. Activation functions can also be called Transfer functions. The different activation functions differ with each other in terms of behaviour, range, differentiability. Step, Linear, Sigmoid, Rectifier are some commonly used activation functions in Neural Networks. Rectified Linear Unit (ReLU) activation function allows only the positive part of the argument and is given by equation 1.1. ReLU activation hidden layer is efficient in

finding a non-linear mapping between input and output data. ReLU activation is one-sided with fewer vanishing gradient problems and allows faster and efficient training as compared to other activation function.

$$f(x) = x^+ = max(0, x) \tag{1.1}$$

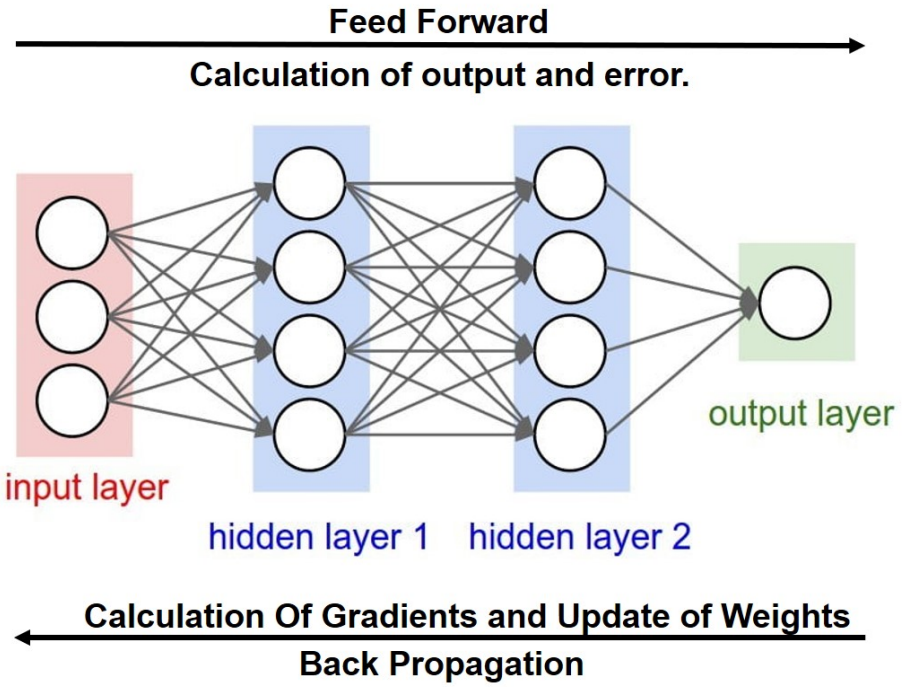


FIGURE 1.3: Training of Neural Networks

1.5.2 Training of Neural Networks

Learning can be supervised with data labels performing classification or Unsupervised finding trends in data. All the neural networks take input data to feed forward and calculated the output. This output is then utilized to calculate the error, which is back propagated through the layers to calculate gradients and update weight. The training algorithm ensures that the weights are updated in such a way that the error starts to decrease as the training progresses to reach a small value closer to zero. Figure 1.3 represents the schematic of the training process of Neural Networks giving an overview of the direction of input and error information flow. Some of the topics important while training NN are explained in brief below.

1.5.2.1 Cost function

Also known as loss function, gives us an estimate of the accuracy in making predictions for the given set of weights and biases in the neural network. The cost function has its own behaviour and properties (gradients), which assists in updating model parameters conditionally to increase accuracy.

1.5.2.2 Gradient Descent

It is an optimization algorithm used to minimize the cost function to update weights in direction of the steepest descent. Figure 1.4 shows us the direction of the weight update to minimize the cost function. Equation 1.2 represents the weight updation where E, w and η is the error in prediction, NN weights and learning rate respectively.

$$w^{new} = w^{old} - \eta \frac{\partial E}{\partial w} \tag{1.2}$$

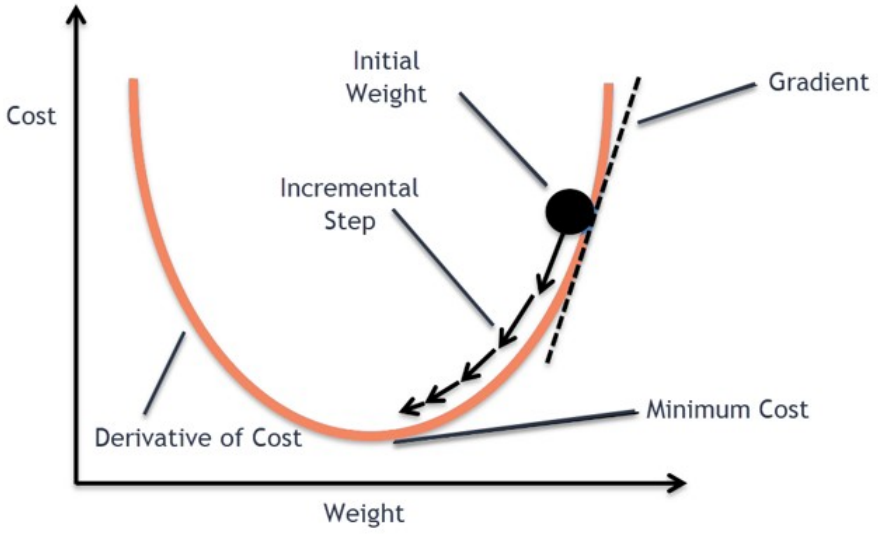


FIGURE 1.4: Weight update in Gradient Descent Algorithm

1.5.2.3 Learning Rate

Learning rate controls the resolution of weight update with respect to gradient of loss function. High learning rate causes the learning process to diverge as large update in

weights may skip our solution minima. Low learning rate slowly updates the weights in direction of decreasing error.However, it may drastically increase the training time. Figure 1.5 represents the variation of training error at different learning rate as the training progresses.

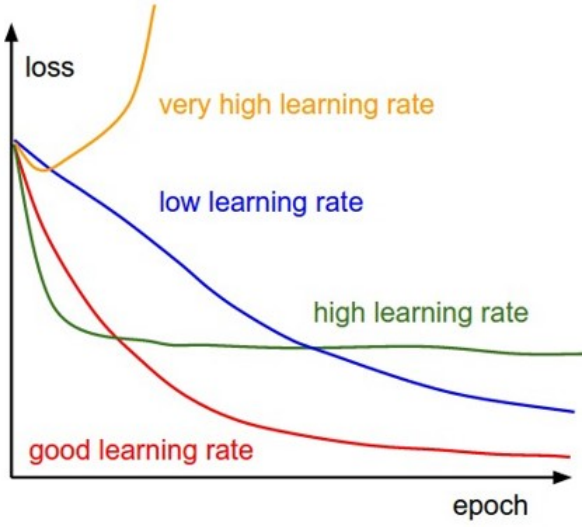


FIGURE 1.5: Comparision of Training error at various Learning rate

1.5.2.4 Momentum

Momemtum is used to accelerate the learning process and overcome the problem arising from error surfaces with step sides and shallow valley floor. It increases the learning rate to cross the plateau regions in error surfaces at faster rate and not get caught in the local minima.

The formulation of feed-forward multilayer artificial neural networks, gradient descent training algorithm and other optimisation parameters can be found out in the book by L.D. Behra [13]. A detailed survey of deep learning method can be found in a review article by A. Carrio et. Al. [6] giving a broad application survey of Deep Learning in the field of UAV. The review discusses the use of Deep Learning in the areas of vehicle identification, control, navigation and environment perception.

1.6 Organisation of thesis

The thesis discusses the problem of modelling small scale helicopters with high accuracy based on existing work in literature. The development of a high fidelity model is based on a hybrid approach, using a grey box model which has a combination of a baseline model and neural network. The baseline model used in the thesis is derived using Frequency Domain System Identification, producing Linear Time-Invariant Single Input Single output models. The baseline model obtained is then integrated with a Deep Learning Model. The thesis gives an analysis of the Deep learning structure optimization for deriving the model with the best performance.

Chapter 1 gives a brief introduction to modelling small scale helicopters. The chapter further discusses the problem statement and provides a brief introduction to concepts of System modelling and Deep Learning.

Chapter 2 describes the UAV test bed along with data-preprocessing laying down the guidelines for the collection of data for training and identification process.

Chapter 3 describes the identification of Single input single output time invariant models using Frequency domain based system identification techniques and provide time domain verification of the above-said model.

Chapter 4 describes the development of Deep learning model for correcting the accelerations of the baseline model. Further, the outputs of the baseline and deep learning models are compared to demonstrate the accuracy of the hybrid model obtained.

Chapter 2

Introduction to the Experiment and UAV Test Bed

2.1 Align TREX 700 Helicopter

Trex 700 Align is commercially available small UAV Helicopter. It has main rotor diameter of 158.2 cm and tail rotor diameter of 28.1 cm. The vehicle has a flying weight of 5.2 kg and has no fly bar in its rotor configuration. The swashplate tilt is controlled by a combination of three servos whose mixing is directly related to the blade pitch angle. The tail rotor pitch controlled by another servo placed near the tail. These four servos can be controlled using four stick inputs of the transmitter. Other vehicle details are mentioned in Table 2.1

FIGURE 2.1: Align Trex 700 side view [4]

FIGURE 2.2: Align Trex 700 front view [4]

Model Name	T-Rex 700L Dominator TOP
Main Rotor Diameter	1582 mm
Tail Rotor Diameter	281 mm
Vehicle Dimensions	1350 mm X 360 mm X 208 mm
Weight (without motor)	3310 g
Motor Drive Gear	13T
Main Drive Gear	110T
Autorotation Tail Drive gear	104T
Tail Drive Gear	22T
Drive Gear Ratio	8.46:1:4.73

TABLE 2.1: Align Trex 700 specifications

2.2 Flight Instrumentation

Earlier, the experiments were conducted in an indoor environment at IIT Kanpur for the frequency domain System Identification. Indoor tests used Viacom camera sensors for constructing vehicle position, velocity, acceleration and orientation data. The data recorded had high signal-to-noise ratio, however only hover data could be recorded in the indoor environment due to the limitation of space. While recording data in an outdoor environment, there are additional noises added to the vehicle due to winds. The onboard sensors have low signal-to-noise ratio due to vibrations due to the main rotor. Therefore it becomes difficult to record data in an open environment and can also be influenced by the skills of the flight pilot.

Trex 700 is hard mounted with PX4 2 also known as Cube autopilot board and GPS. This board is an upgrade to the earlier version with increased processing power and with an isolated and dampened IMU. All helicopter actuators, ESC and telemetry are connected

to this board. PX4 is a widely used autopilot consisting of various inbuilt sensors such as accelerometers, gyroscopes and magnetometers and also provides features to incorporate external sensors such as GPS too. A ground station was set up to work in coordination with the onboard GPS for Real-time kinematic positioning of the vehicle to provide the position data to centimeter level accuracy. The autopilot board and GPS have been mounted to an offset to the center of gravity. The whole experimental setup on the UAV can be seen in the figure 2.4

FIGURE 2.3: Pixhawk Cube

The following list mentions the technical specifications of the Cube autopilot board[20]:

- Processor
  - 32-bit ARM Cortex M4 core with FPU
  - 168 Mhz/256 KB RAM/2 MB Flash
  - 32-bit failsafe co-processor
- Sensors
  - Three redundant IMUs (accels, gyros and compass)
  - InvenSense MPU9250, ICM20948 and/or ICM20648 as first and third IMU (accel and gyro)
  - ST Micro L3GD20+LSM303D or InvenSense ICM2076xx as backup IMU (accel and gyro)
  - Two redundant MS5611 barometers
- Power

  - Redundant power supply with automatic failover
  - Servo rail high-power (7 V) and high-current ready
  - All peripheral outputs over-current protected, all inputs ESD protected

- Interfaces
  - 14x PWM servo outputs (8 from IO, 6 from FMU)
  - S.Bus servo output
  - R/C inputs for CPPM, Spektrum / DSM and S.Bus
  - Analogue / PWM RSSI input
  - 5x general purpose serial ports, 2 with full flow control
  - 2x I2C ports
  - SPI port (un-buffered, for short cables only not recommended for use)
  - 2x CAN Bus interface
  - 3x Analogue inputs (3.3V and 6.6V)
  - High-powered piezo buzzer driver (on expansion board)
  - High-power RGB LED (I2C driver compatible connected externally only)
  - Safety switch / LED
  - Optional carrier board for Intel Edison

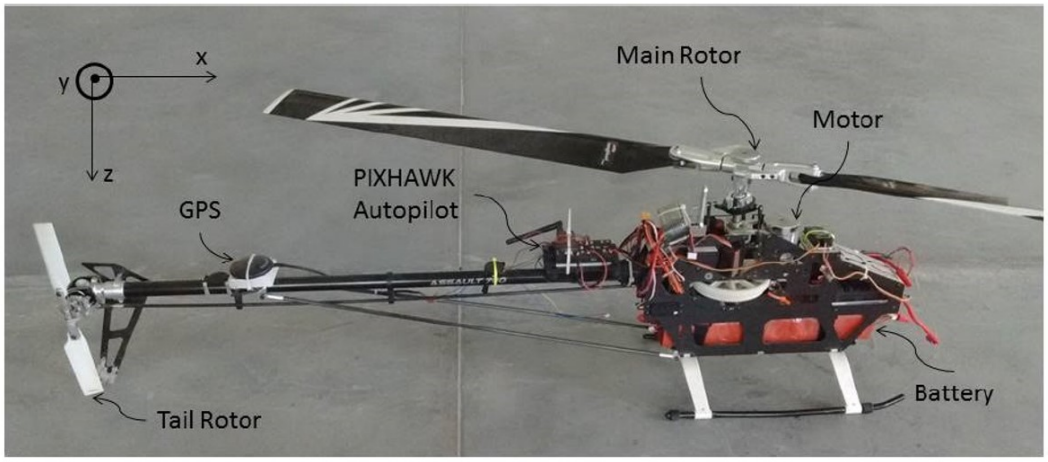


FIGURE 2.4: UAV components

2.3 Flight Experiment

Accurate model identification requires good data with sufficient input and output excitation. Measurement accuracy primarily depends on instrumentation and quality of the flight depends mostly on the flight experiments used for data collection. The Frequency Sweep excitation in hover has been used for frequency domain model identification, and the model has been validated using hover doublet excitation. Apart from this free flight data for over one hour has been collected for identification and validation of Deep Learning model.

Frequency sweep excitation is a heuristic input which has been extensively used in frequency domain identification of rotorcrafts which does not require any prior knowledge of system dynamics. Sweep excitations prove optimal for our case providing maximum information content for minimum manoeuvre time. The hover frequency sweep is 90-120 seconds long while the forward flight sweeps could only be collected for 10-15 seconds due to the limitation of space and piloting ability. It is important to understand that a particular dynamics can only be identified if it is present in data.

2.4 Data Preprocessing

The accelerometer values were too noisy. A low pass filter was applied to recover good acceleration data. The delay added by the use of a low pass filter was compensated for correcting the unknown dynamics added. The abrupt outliners were also removed from the data. One of set of procesed input-output data is plotted in figure 2.6 and 2.5.

$$a_x^{cg} = a_{xm} + (q^2 + r^2)x_{cg} + (pq - \dot{r})y_{cg} - (pr + \dot{q})z_{cg} \tag{2.1}$$

$$a_y^{cg} = a_{ym} - (pq + \dot{r})x_{cg} + (p^2 + r^2)y_{cg} - (qp - \dot{p})z_{cg} \tag{2.2}$$

$$a_z^{cg} = a_{zm} - (pr - \dot{q})x_{cg} - (qr - \dot{p})y_{cg} - (p^2 + q^2)z_{cg} \tag{2.3}$$

$$u_{cg} = u_m + ry_{cg} - qz_{cg} \tag{2.4}$$

$$v_{cg} = v_m + pz_{cg} - rx_{cg} \tag{2.5}$$

$$w_{cg} = w_m + qx_{cg} - py_{cg} \tag{2.6}$$

where $a_x^{cg}, a_y^{cg}, a_z^{cg}$ are the accelerations at center of gravity, a_{xm}, a_{ym}, a_{zm} are the accelerations measured by the IMU, u_{cg}, v_{cg}, w_{cg} are the calculated translational velocities at center

of gravity, u_m, v_m, w_m are the velocities measured by the GPS, x_{cg}, y_{cg}, z_{cg} represent the position of IMU and GPS with respect to center of gravity in the body-fixed frames. The required translations velocities and accelerations were then calculated from the measured data using equations 2.7, 2.8, 2.9.

$$\dot{u} = -qw + rv - gsin\theta + a_x^{cg} \tag{2.7}$$

$$\dot{v} = -ru + pw + gsin\theta cos\phi + a_y^{cg} \tag{2.8}$$

$$\dot{w} = -pv + qu + gsin\theta cos\phi + a_z^{cg} \tag{2.9}$$

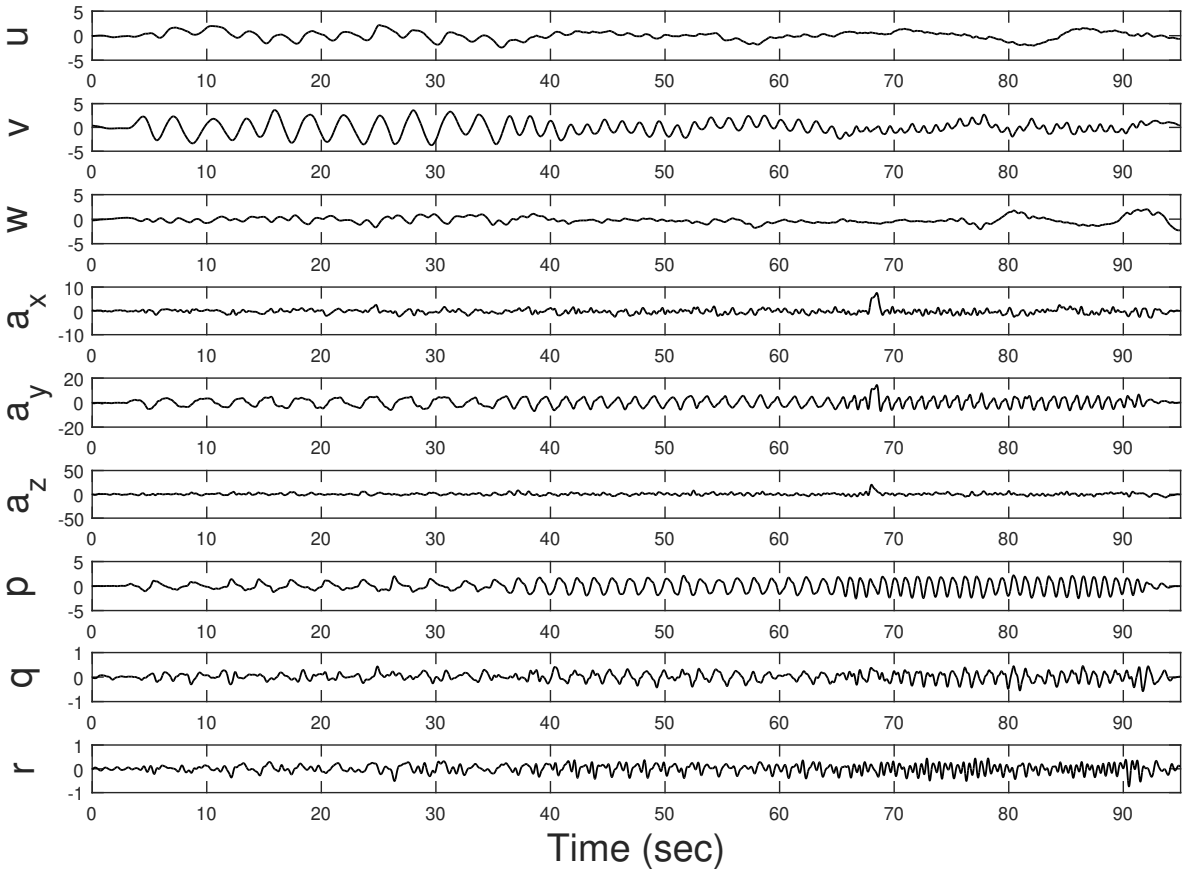


FIGURE 2.5: Processed Ouput data

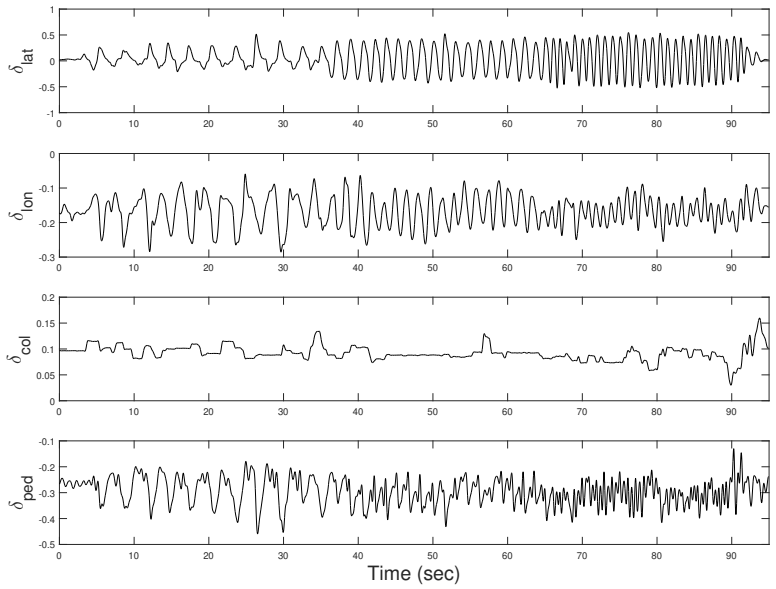


FIGURE 2.6: Processed Input Data

The following experiements were conducted and data was recorded:

- Lateral sweep for hovering and forward flight.
- Longitudinal sweep for hovering and forward flight.
- Yaw sweep for hovering and forward flight.
- Heave sweep for hovering and forward flight.
- Lateral doublet for hovering and forward flight.
- Longitudinal doublet for hovering and forward flight.
- Yaw doublet for hovering and forward flight.
- Heave doublet for hovering and forward flight.

The following inputs and outputs were recorded for determining the frequency responses. The stick inputs recorded are PWM values ranging from -1 to 1 with the extreme values representing extreme positions of stick. Body velocities, acceleration, attitude rate and attitude have units $meter/second$, $meter/sec^2$, $radian/seconds$ and $radians$ respectively.

Chapter 3

Development of the Baseline Model

3.1 Introduction

System Identification refers to the process of extracting a mathematical or dynamical model from measured data. System identification Modelling can be Parametric, which uses model structure assumptions prior to modelling. Non-parametric modelling does not assume the structure of the model and is similar to a black-box modelling. System Identification gives us dynamic models that can be used for many applications such as stability and control system analysis, pilot simulation and analysis of handling quality.

The rotary vehicle has a nonlinear behaviour due to nonlinear kinematics and dynamics associated with rigid body motion, gyroscopic forces, and nonlinear aerodynamics effects. But for a specified operating point, the nonlinearity present is small and can be considered insignificant. So, at a particular operating position, the rotorcraft dynamics can be accurately approximated by a linear model. This particular operating position is the trim conditions when the vehicle is in equilibrium.

In frequency domain we are able to convert large number time domain data to a small number of frequency domain data samples. This method is based on power spectrum of the input-output data which identifies the amount of signal in a particular bandwidth of frequency. The frequency domain identification process can focus only on the required frequency bandwidth which helps us to easily segregate the high frequency noises in the data. Another reason for choosing frequency domain identification is the conversion of differential equation to algebraic equations which makes calculations easier.

Frequency domain approach uses frequency responses between input-output data to model a system. Frequency response is primarily referred to the steady-state response of a time-invariant system when excited using constant sine wave input, which results in a harmonic response with the same frequency of excitation, a phase shift and a magnitude amplification. For a nonlinear system, the Frequency response is the best linear model for input-output, which provides signification information about the system. Frequency domain methods are computationally effective as they require fewer data points for identification of models. The roadmap for System Identification using frequency responses is shown in the figure. 3.1. The models identified in this chapter are used as baseline model for the Deep Learning models.

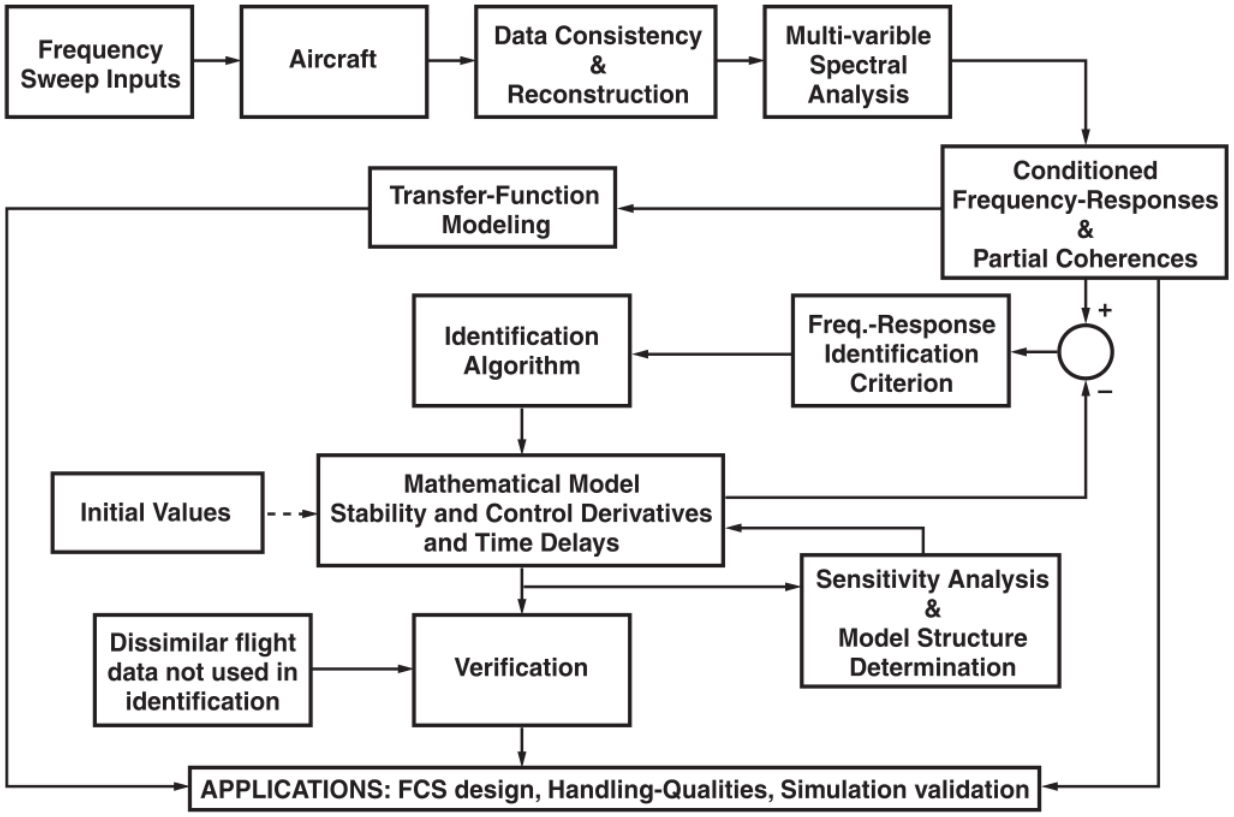


FIGURE 3.1: Roadmap to Frequency Domain System Identification

Frequency sweeps consist of information of broader spectrum of frequency and is preferred over other excitation methods for frequency domain system identification. The data measured is checked for consistency and corrected for errors such as measurement errors, center of gravity offset etc. The next step is to perform multivariable spectral analysis of the data to produce condition frequency responses. The model structure, conditioned

frequency responses and initial guess parameters are used to find the model parameters through various algorithms.

3.2 Overview of CIFER Software

CIFER (Comprehensive Identification from Frequency responses) is a tool developed by NASA for comprehensive analysis of aircraft and component dynamics. Successful identifications have been carried out by researchers on fixed wing as well as rotary vehicles including XV-15, Bell-214ST, BO-105, AH-64, UH-60, V-22, AV-8 Harrier, and OH-58D using CIFER. CIFER is a tool integratesd to provide better and structured user access for systm identification methodology. CIFER has six core programs taking care of different blocks of roadmap to frequency domain system identification. These programs are for conditioning the data and performing FFTs (FRESPID), multi-input conditioning (MISOSA), window combination (COMPOSITE), transfer-function model identification (NAVFIT), state-space model identification (DERIVID), and time domain verification (VERIFY).

Fast Fourier transform, Discrete Fourier transform and Chirp-Z transform are different domain transformation methods used in aircraft System Identification. CIFER uses CZT which is the most reliable and accurate method that provides high flexibility in the selection of sample rates and frequency resolution.

3.3 Coherence Function

Coherence function is defined at each frequency point and indicates how much the output spectrum is linearly attributable to the input spectrum. It shows us the linear correlation between output and input in terms of a numeric entity. Coherence value of more than 0.6 is regarded as suitable for giving a consistent quality. The value of Coherence function allows us to judge our experimental data initially and thereby is further used for the identification process. Coherence value of 1 represents a perfect linear relationship between intput and output entity.

The input and output data was uploaded to the CIFER database and checked for coherence values. Some of the important on-axis responses showing good coherence are shown in the figures 3.2. Responses with coherence value greater than 0.6 are appropriate for system identification.

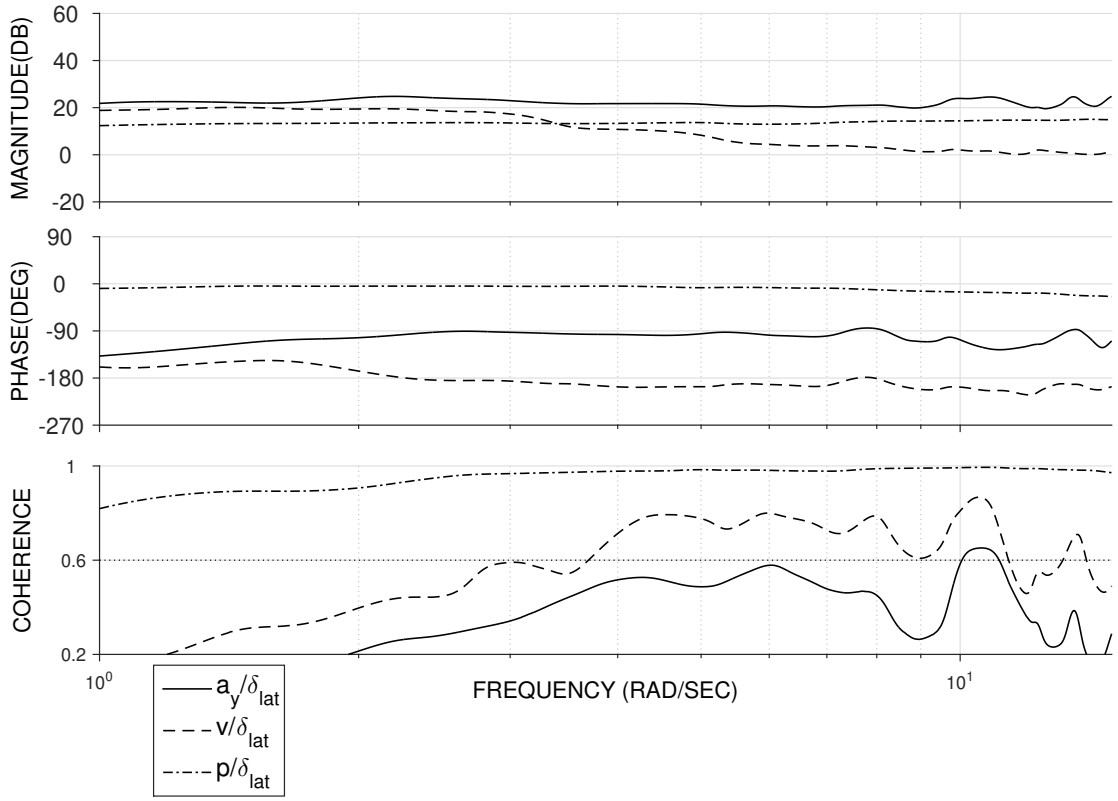


FIGURE 3.2: Lateral Sweep conditioned Frequency Responses

3.4 Complete Parametrized model

The parameterized state-space model has a general form :

$$M\dot{x} = Fx + Gu(t - \tau) \tag{3.1}$$

$$y = H_0 x + H_1 \dot{x} \tag{3.2}$$

with state vector

$$x = [u, v, w, p, q, r, \phi, \theta]^T \tag{3.3}$$

with measurement vector

$$y = [u, v, w, p, q, r, ax, ay, az]^T \tag{3.4}$$

The input vector is

$$u = [u_{lat}, u_{lon}, u_{col}, u_{ped}]^T \tag{3.5}$$

The system matrix F contains the stability derivatives, the input matrix G contains the input derivatives, and the M matrix is an identity matrix in this case. y is the measurement vector representing relation between the state vector and the measured data. The resulting general six dof state-space model is shown in Eq: 3.6. This equation represents a general six degree of freedom state-space equation with large number of state and control derivatives. The task of the identification process is to eliminate some of these unimportant derivatives and finding values of remaining derivatives depending on the flight data. Some of these derivatives can be intutive to eliminate. However, a proper sensitivity analysis of the identified matrices helps us to know about the parameters to be removed.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r & 0 & -g \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r & g & 0 \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r & 0 & 0 \\ L_u & L_v & L_w & L_p & L_q & L_r & 0 & 0 \\ M_u & M_v & M_w & M_p & M_q & M_r & 0 & 0 \\ N_u & N_v & N_w & N_p & N_q & N_r & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_{lat}} & X_{\delta_{lon}} & X_{\delta_{ped}} & X_{\delta_{col}} \\ Y_{\delta_{lat}} & Y_{\delta_{lon}} & Y_{\delta_{ped}} & Y_{\delta_{col}} \\ Z_{\delta_{lat}} & Z_{\delta_{lon}} & Z_{\delta_{ped}} & Z_{\delta_{col}} \\ L_{\delta_{lat}} & L_{\delta_{lon}} & L_{\delta_{ped}} & L_{\delta_{col}} \\ M_{\delta_{lat}} & M_{\delta_{lon}} & M_{\delta_{ped}} & M_{\delta_{col}} \\ N_{\delta_{lat}} & N_{\delta_{lon}} & N_{\delta_{ped}} & N_{\delta_{col}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{lat} \\ \delta_{lon} \\ \delta_{ped} \\ \delta_{col} \end{bmatrix} \tag{3.6}$$

3.5 Identification Process

The frequency responses corresponding to good coherence are selected for the identification process. The model and frequency responses are fed to state-space identification tool DERIVID in CIFER. The model is selected and the parameters are initialised. Model convergence with accurately identified parameters depends on Transfer function cost and parameter sensitivity analysis.

- Transfer function Cost(J). The solution of identification is to determine the values of coefficients of state and control derivative matrices that produce a frequency response closely matching the frequency responses obtained from flight data. Transfer function cost gives numerical way of determining out model convergence. Its value depend on the error between the identified model and flight data frequency response.

$$J_{avg} \leq 100 \tag{3.7}$$

- *Cramér − Rao* bound and insensitivity. The *Cramér − Rao* inequality provides the fundamental basis for the theoretical accuracy analysis. This inequality establishes the *Cramér − Rao* bounds CR_i as the minimum expected standard deviation σ_i in the parameter estimate θ_i that would be obtained from many repeated maneuvers.

An average Cost less than 100 and individual cost less than 200 is appropiate for successful model identification. The identification might diverge if initial guesses are too far away from the converging values. These values are carefully selected after studying the parameter values for other similar UAV and can also be derived analytically.

Sensitivity analysis of the identified parameters gives us the cramer rao percentage, insensitivity and parameter corelations. Refinements in the model can be made by studying the Sensitivity results and eliminating parameters. Insensitivities below 10 % and Cramer-Roa bound percentage below 20% identifies the important model parameters. The converged identified model has been verified using time domain state-space identification tool VERIFY in CIFER. This program can also be used to calculate biases and reference shifts in our measured data.

3.6 SISO Model Structure

System identification is an iterative and cumbersome process. The main aim of the identification process is identifying the coefficients of state and control derivative matrices. The process of identification starts with initial guess of model structure and coefficients. Some parameter coefficients are fixed while some remain free in the identification. Sensitivity analysis of the parameters gives us an idea about eliminating the parameters in the model structure. The process is repeated until the desired cost function is reached.

Deriving a higher degree model directly from measured data is difficult. SISO models also give us initial parameter values for extraction of the complete higher order state-space model. Four SISO models for each individual control input has been identified. The underlying sense in the grouping of the state variables in four different models comes from the physics involved. The δ_{Lat} is related with roll motion, So the lateral model is a function of lateral state variables $\{u, q, \phi\}$ and control input δ_{Lat}. Similarly, other models are functions of their primary control input and on-axis variable. The following four model structure were selected for the identification process.

- Lateral Model

$$\begin{Bmatrix} \dot{v} \\ \dot{p} \\ \dot{\phi} \end{Bmatrix} = \begin{bmatrix} Y_v & Y_p & g \\ L_v & L_p & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} v \\ p \\ \phi \end{Bmatrix} + \begin{bmatrix} Y_{\delta_{lat}} \\ L_{\delta_{lat}} \\ 0 \end{bmatrix} \left\{ \delta_{lat} \right\} \tag{3.8}$$

- Longitudinal Model

$$\begin{Bmatrix} \dot{u} \\ \dot{q} \\ \dot{\theta} \end{Bmatrix} = \begin{bmatrix} X_u & X_q & -g \\ M_u & M_q & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{Bmatrix} u \\ q \\ \theta \end{Bmatrix} + \begin{bmatrix} X_{\delta_{lon}} \\ M_{\delta_{lon}} \\ 0 \end{bmatrix} \left\{ \delta_{lon} \right\} \tag{3.9}$$

- Heave Model

$$\dot{w} = Z_w w + Z_{\delta_{col}} \delta_{col} \tag{3.10}$$

- Yaw Model

$$\dot{r} = N_r r + N_{\delta_{ped}} \delta_{ped} \tag{3.11}$$

3.7 SISO Identification Results

3.7.1 Parameters Identified

Parameter	Value	CR-%	Insensitivity-%
X_u	-0.99	13.34	4.772
M_q	-15.69	15.21	1.387
X_{lon}	5.17	18.95	6.346
M_{lon}	95.06	14.11	1.340
Y_v	-1.180	13.88	4.227
L_p	-32.31	16.05	1.389
Y_{lat}	-13.44	9.98	2.939
L_{lat}	192.3	15.55	1.360
N_r	2.342	19.60	9.476
N_{ped}	-80.54	3.871	1.871
Z_w	-1.250	11.08	5.204
Z_{col}	-79.02	3.040	1.428

TABLE 3.1: Identified SISO model parameters for hover condition

3.7.2 Selected Frequency responses for SISO Identification

Good coherence data is important for detrmination of state space derivatives. The following table shows the frequency range for important frequency responses required for model Identification.

	δ_{lat}	δ_{lon}	δ_{col}	δ_{ped}	Transfer function costs
p	1.90-20	-	-	-	30.80
q	-	1.13-11.31	-	-	77.98
r	-	-	-	7.4-15	26.95
a_x	-	1.32-6.06	-	-	67.09
a_y	1.50-5.64	-	-	-	68.46
a_z	-	-	1.92-10.28	-	32.83
u	-	1.32-6.06	-	-	67.09
v	1.70-5.47	-	-	-	75.66
w	-	-	1.83-9.94	-	35.00
Average Transfer function cost					53.54

TABLE 3.2: Frequency responses and frequency ranges (in rad/sec) selected for the SISO identification process and its corresponding transfer function costs after identification

3.7.3 Time domain Verification

The identified SISO models were verified in time domain using doublets. Lateral, Longitudinal, Heave, Yaw doublets were used to test the identified Lateral, Longitudinal, Heave and Yaw SISO models respectively. All of the models were accurately identified and matched with the identified SISO models. All the parameters identified have $cramér - rao$ bound percentage less than 20% and insensitivity percentage less than 10%. The identified transfer function cost has been reduced below 100 for all the transfer functions. Figure 3.3 shows the time domain verification of the four SISO model. Simulation time of 5 seconds is selected for the validation. If larger simulation time is selected the simulation may diverge. The models appear to perform well for the doublet input with predicted output values following similar pattern as flight data. Section 3.6.4 shows the comparison of the frequency responses of the identified SISO models and the flight data.

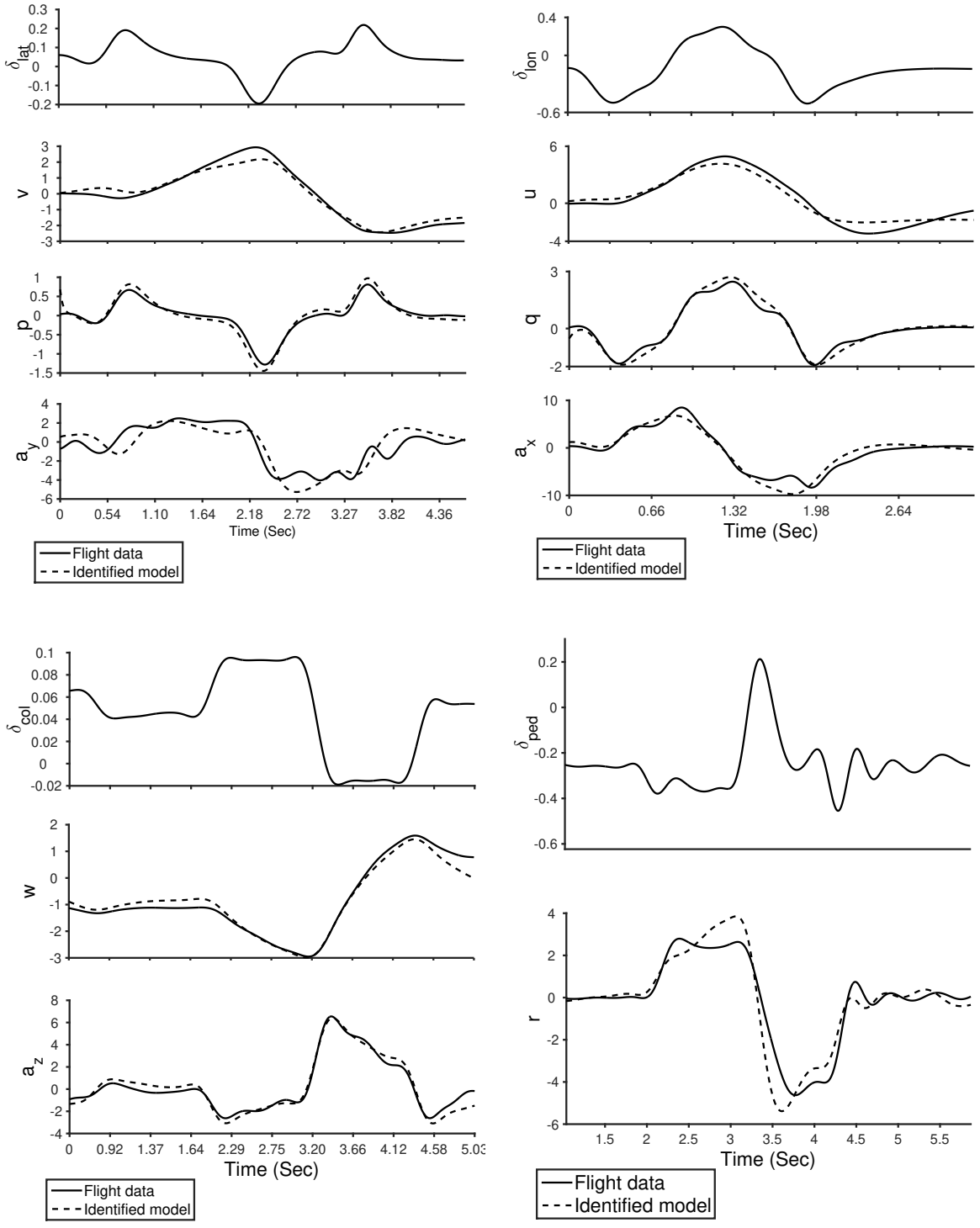
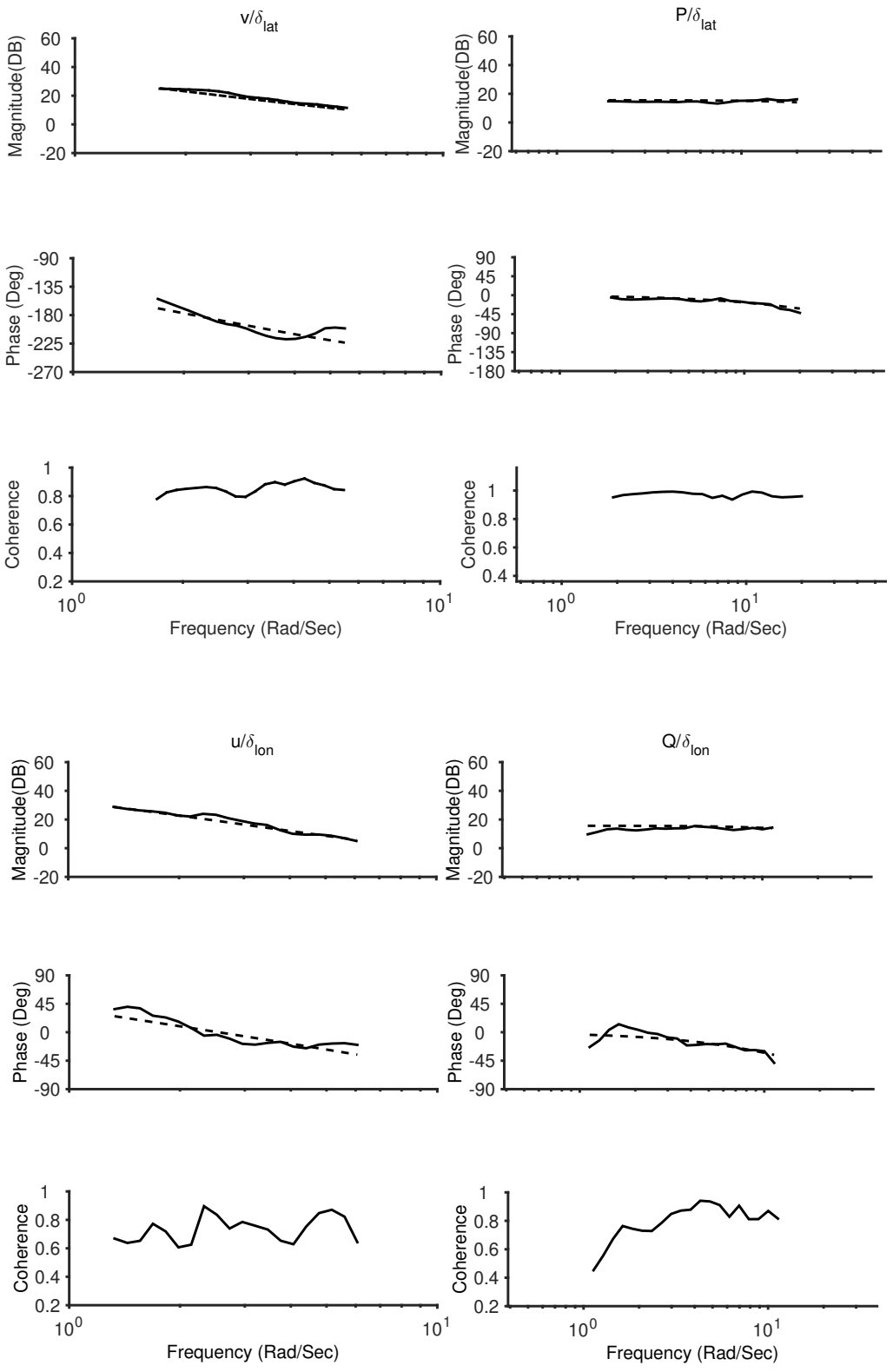


FIGURE 3.3: Roll, Pitch, Heave, Yaw SISO verification results.

3.7.4 Frequency Response



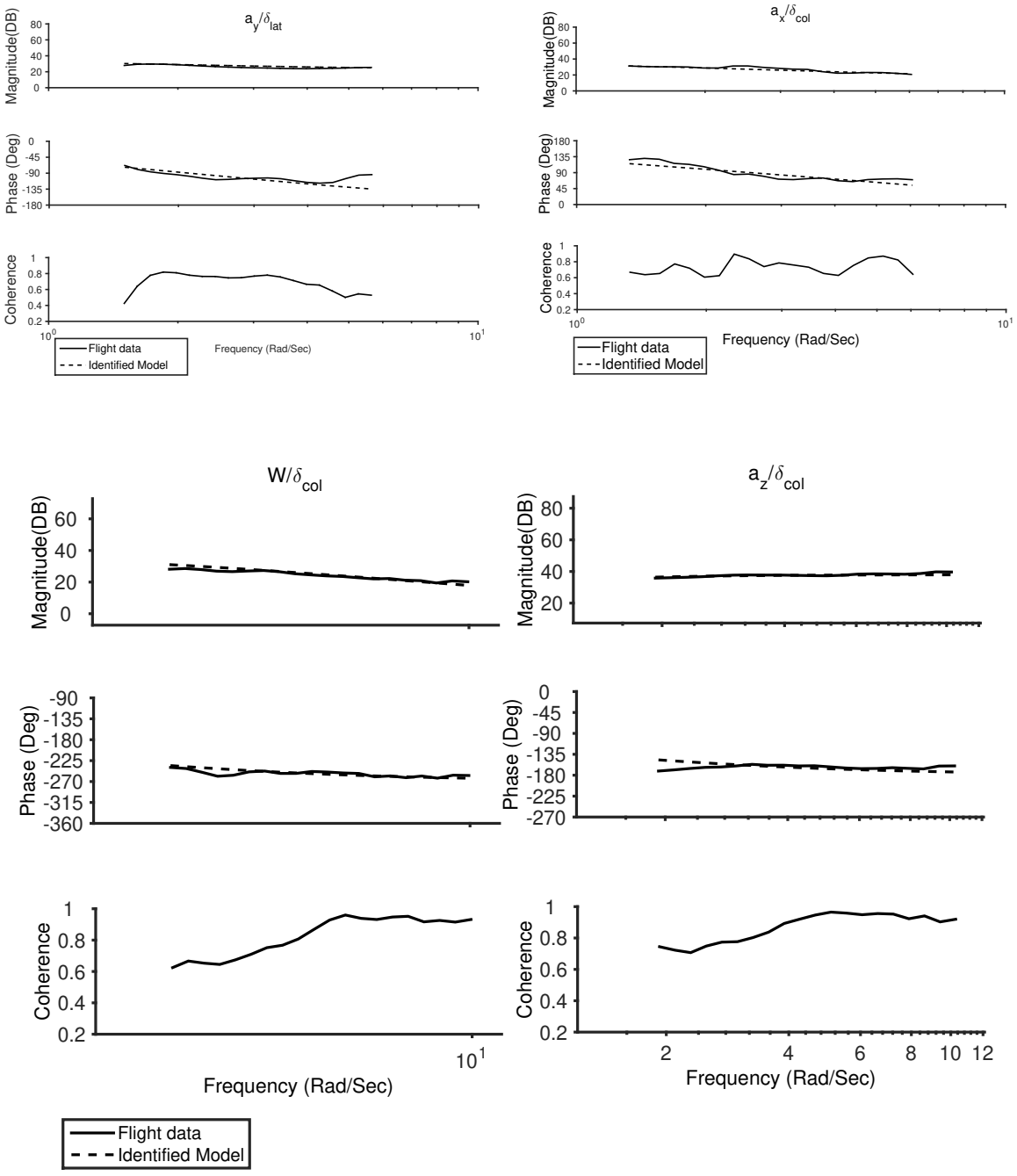


FIGURE 3.4: Frequency Response Identification Bode Plots.

Chapter 4

Development of High Fidelity Deep Learning Model

4.1 Introduction

Hover sweep data was used for the frequency domain System identification process to obtain LTI SISO models. Time domain verification of SISO models shows good accuracy for on-axis outputs. These models do not account for any non-linear behaviour and represent only a narrow regime of flight. Poor off-axis response and higher inaccuracies make these models diverge for larger simulation time due to the accumulation of error. These models help us to understand the behaviour of the vehicle which can be used for control system design; however, they are not ideally suited for high-fidelity flight dynamics simulation.

The linear system fails to capture specific properties of a dynamical system such as inertia. The accuracy of a model can be increased by increasing the number of state variables. However, for a helicopter, it is difficult to keep a record of more state variable due to the higher order dynamics involved. There arises a need to utilize the time history record of the state variable to increase accuracy. [3] expresses the system dynamics of the helicopter as a Markov decision process model based on time history of states and control inputs to find the latent hidden states in the system.

Deep learning models have been used extensively used for identification of helicopter dynamics mostly posed as black-box modelling problem without making much sense to the physics involved. P. Abeel et. Al.[19] applied high dimensional regression using a Neural

Network model comprising of Relu hidden units in order to construct a global nonlinear model of the helicopter that does not require annotation or alignment of trajectories, has few tuning parameters and that captures dynamics throughout the entire set of flight regimes over which data is collected. Sagar Setu [22] here at IIT, Kanpur utilized similar Deep Learning technique using a hybrid physics baseline model and an optimized Deep Learning model to identify the system for controller design.

The basic idea is to project system dynamics in higher dimensional space to find common trends which can be stitched with the SISO models obtained in chapter-3 using Deep Learning. Physics-based First principle modelling and System identification techniques are tedious and involve a lot of assumptions. Deep Learning technique allows engineers to quickly identify the system accurately with lesser flight data for simulation studies.

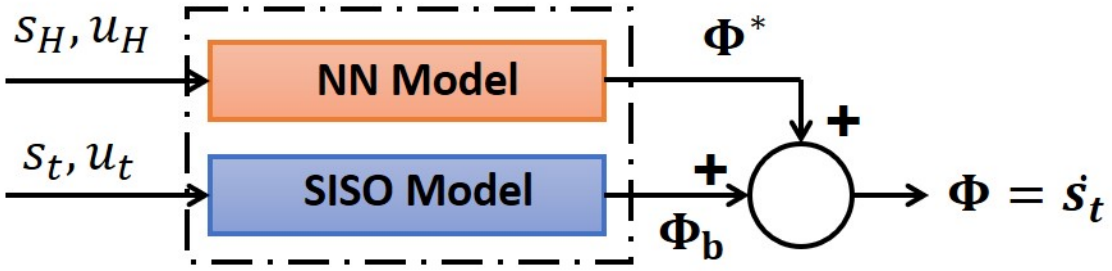


FIGURE 4.1: Deep Learning plant

4.2 The Underlying Principle

The SISO models obtained in chapter 3 were derived using hover sweep data, which represents on a narrow regime of flight. These models have good accuracy for a hovering flight but seem to diverge for an aggressively manoeuvring flight. This baseline model takes $s_t = \{u, v, w, p, q, r\}$ and $u_t = \{\delta_{lat}, \delta_{lon}, \delta_{col}, \delta_{ped}\}$ at time t as input to give $\Delta s = \Phi_b(s_t, u_t)$.The error in the output of the baseline model is given by $\Phi^* = \Phi - \Phi_b$ (where Φ is the real output).

The helicopter is a higher order system than the one represented by s_t, and the model $\Phi_b(s_t, u_t)$ is a Markov chain of this lower order approximation. In such a setting, the actual model Φ^*, and consequently the augmentation Φ_e, will depend on not only s_t and u_t, but also on a certain minimum length of history of s and u through which the dynamics hidden states could be captured. Assuming that the system is smooth, Taken's theorem suggests

that it would be possible to find an approximation of Φ^* given a sufficiently long history of s and u [19]. So it is possible to find a mapping between the inputs $p_h = \{s_h, u_h\}$and the output $\Phi_e(p_h)$. The workflow of the deep learning identification process has been shown in figure 4.1.

We exploit the above principle to find a Neural Network configuration to map p_h to Φ_e The hidden layer of our network uses Rectified linear units(Relu) as activation function, which has been used extensively to find nonlinear mappings. The input to our Deep learning model is $p_H = \{s_H, u_H\}, s_H$ being the time history of states $s_t, s_{t-1}, ..., s_{t-k}$ and u_H a history of inputs $u_t, u_{t-1}, ..., u_{t-k}$, where both s_H and u_H have history upto time Δt_{k+1}. The output of the deep learning model gives us the corrections in the baseline model Φ^* at any particular time.

4.3 Network Description

Taken's Theorem states that for a large class of nonlinear dynamical systems with d-dimensional state space, only 2d+1 previous measurements of a single system output are required to effectively reconstruct the system state[19, 22]. Our State-Space model has six state variable and four control inputs. So by virtue of taken's theorem, we need state history of minimum thirteen previous time steps to reconstruct states at any instance of time. The flight data was captured at a sampling rate of 200 Hz. The following networks take state and control input data history of previous 20 time steps(0.1 seconds) for the reconstruction of state at the current instance. The entire identification process has been uncoupled by training six neural networks with one output each corresponding to six state variables. Hence each of the NN model has input vector$\{p_H(s_H, u_H)\}$ of length 200 with one dimensional output vector$\{\Phi^*\}$. These models are run in parallel for time domain identification of the model. The following two models with the different internal structure are chosen to correct the accelerations given by the baseline model.

4.3.1 ReLU acceleration Correction model

The internal structure comprises of three-layer neural network. The first layer is a fully connected linear input layer with weights and biases as W_1 and b_1, followed by a nonlinear hidden layer consisting of Rectified Linear units (ReLU) F_{relu}. The last layer is also a fully connected linear output layer with weights and biases W_2 and b_2 respectively. Equations 4.1 and 4.2 represents the mathematical form of the network structure chosen. 50 units in

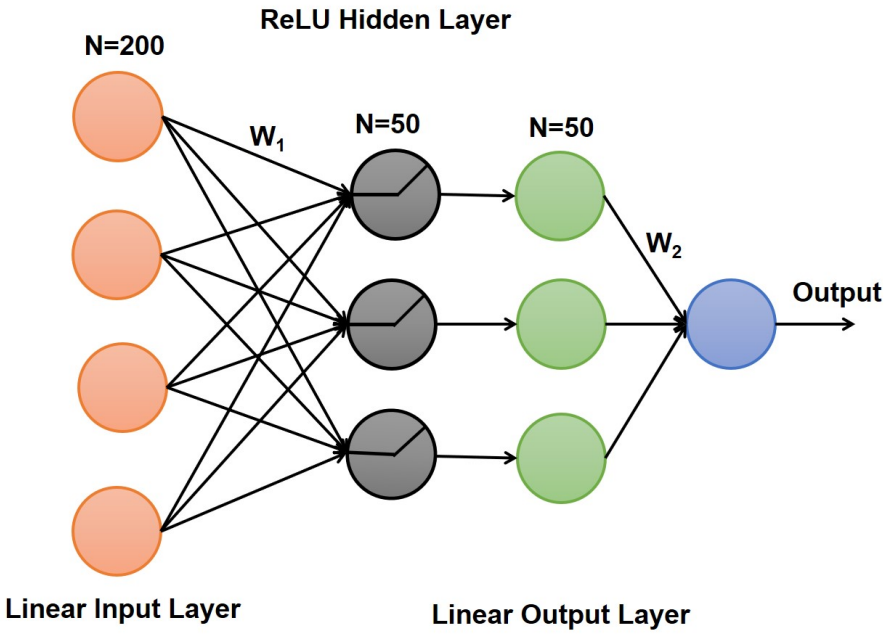


FIGURE 4.2: ReLU acceleration correction Model

ReLU hidden layer were able to provide good accuracy which prevented the model diverge for larger simulation time.

$$\Phi_e = W_2 F_{relu}(W_1 p_H + b_1) + b_2 \tag{4.1}$$

$$F_{relu}(.) = max(0, .) \tag{4.2}$$

4.3.2 Linear acceleration correction model

This model is consists of two fully connected linear layer. This model looks similar to the ReLU model without the hidden layer. The input layer consists of 200 nodes followed by 50 nodes in the first linear layer. The final output layer consists of a single node. Both of the above-mentioned models have the same number of weights to be learnt. The two sets of weight (between the input layer and the intermediate layer) and (the intermediate layer and the output layer) has dimensions 200×50 and 50×1 respectively. Equation 4.3 represents mathematical formulation of the linear acceleration correction model.

$$\Phi_e = W_2(W_1 p_H + b_1) + b_2 \tag{4.3}$$

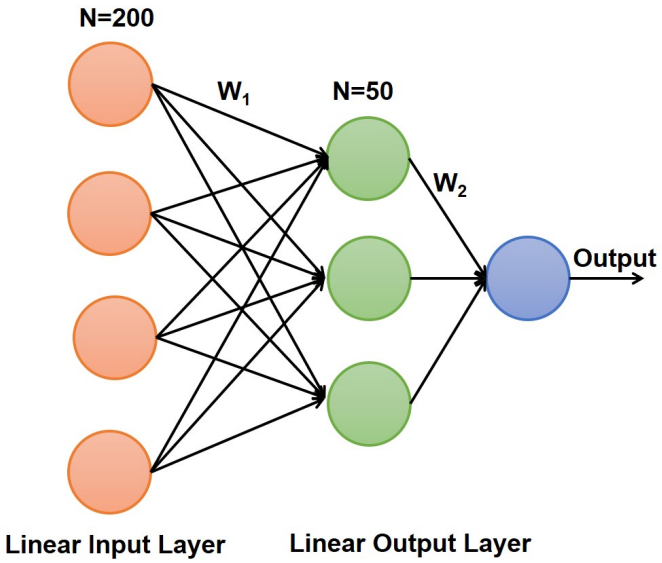


FIGURE 4.3: Linear acceleration correction Model

4.4 Training and Optimization

One hour of Forward flight roll, pitch, heave and yaw sweep data corresponding to 7×10^5 data point is used for training of the neural network. The training data should have high signal-to-noise ratio as noisy data may cause the model to diverge. It is advised to remove any abrupt outliners in order to bound the training data between finite range to prevent the model from diverging as the simulation progresses. The model is validated using freestyle forward flight data. The training and the validation datasets have been kept constant for all the models trained to compare the performance of different models.

The model is trained using Stochastic Gradient Descent (SGD) algorithm using the mean square(MSE) as the error criterion. Mini-batch of 50 data points is selected randomly from the training dataset to perform mini batch SGD with learning rate = 10^{-6} and momentum = 0.9 in order to accelerate the learning process. An open source deep learning library PyTorch is used for the training of the neural network. Eight core i-5 8th gen CPU took 1.5-3 hrs for each NN model to train going over 20,000 epochs of the entire training dataset.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.4}$$

The training error has been shown in figures 4.4,4.5 where the error over the training batch converges to a finite value. The mean square error is represented in the equations 4.4, where Y_i, $\hat{Y}_i$ represents the desired output and the model output respectively. The output

of the baseline model is used at every time step to train the model by subtracting it from the real output.

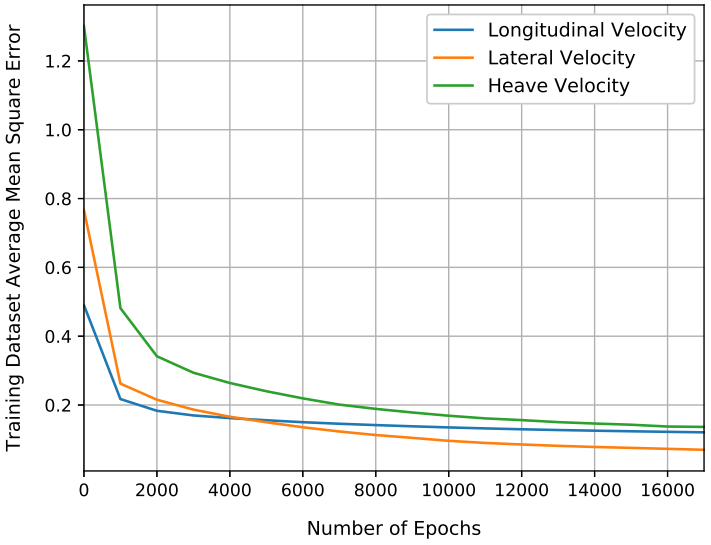


FIGURE 4.4: Translational Velocity Training Error

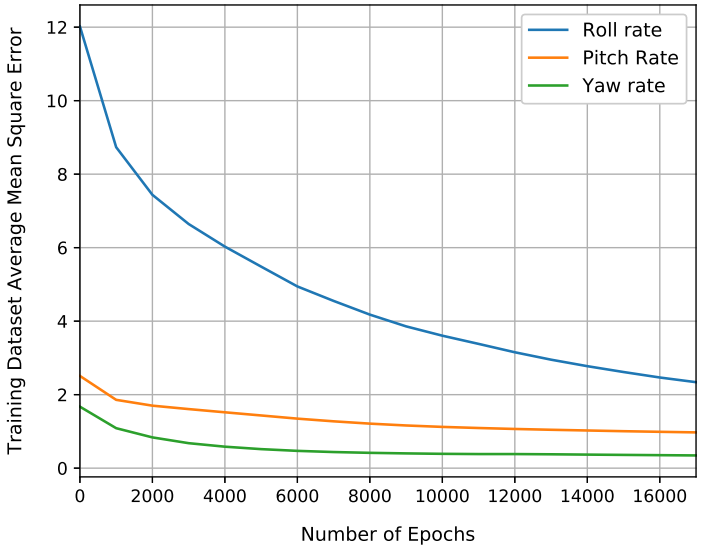


FIGURE 4.5: Angular Velocity Training Error

4.5 Time-Domain Verification

Free flight data is used to validate the above derived models. Equations 4.5-4.10 represent the assembeled equations for the time domain verification. The output of the Deep Learning and baseline model is represented by Φ^* and Φ_b respectively.

$$\dot{u}_M = \Phi^*_u(p_H) + X_u u_t - g\theta_t + X_{lon}\delta_{lon} \tag{4.5}$$

$$\dot{v}_M = \Phi^*_v(p_H) + Y_v v_t + g\phi_t + Y_{lat}\delta_{lat} \tag{4.6}$$

$$\dot{w}_M = \Phi^*_w(p_H) + Z_w w_t + Z_{col}\delta_{col} \tag{4.7}$$

$$\dot{p}_M = \Phi^*_p(p_H) + L_p p_t + L_{\delta_{lat}}\delta_{lat} \tag{4.8}$$

$$\dot{q}_M = \Phi^*_q(p_H) + M_q q_t + M_{\delta_{lon}}\delta_{lon} \tag{4.9}$$

$$\dot{r}_M = \Phi^*_r(p_H) + N_r p_t + N_{\delta_{ped}}\delta_{ped} \tag{4.10}$$

While performing the simulation, the linear and angular velocities at subsequent time steps can be obtained by using Euler integration 4.11-4.16.

$$u_{t+1} = u_t + (\dot{u}_M + \dot{u}_D)\Delta t \tag{4.11}$$

$$v_{t+1} = v_t + (\dot{v}_M + \dot{v}_D)\Delta t \tag{4.12}$$

$$w_{t+1} = w_t + (\dot{w}_M + \dot{w}_D)\Delta t \tag{4.13}$$

$$p_{t+1} = p_t + (\dot{p}_M)\Delta t \tag{4.14}$$

$$q_{t+1} = q_t + (\dot{q}_M)\Delta t \tag{4.15}$$

$$q_{t+1} = q_t + (\dot{q}_M)\Delta t \tag{4.16}$$

The subscript (D) represent apparent translational acceleration due to rotation of frame of reference given by eq 4.17. The sampling rate of the model is same as the sampling rate of the data used for training($\Delta t = 0.005$ seconds).

$$\{\dot{u_D}, \dot{v_D}, \dot{w_D}\} = -\omega_b \times V_b \tag{4.17}$$

Figures 4.6 and 4.7 show the time domain verification of the derived models. The ReLU acceleration correction model significantly outperforms other models not diverging for larger simulation time.

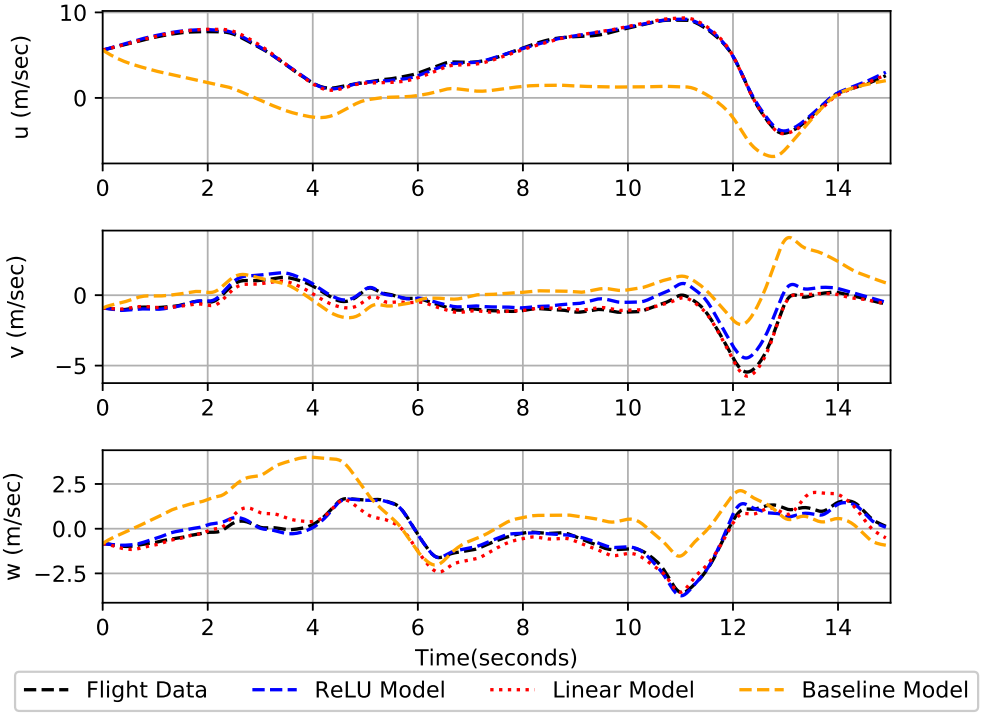


FIGURE 4.6: Translational Velocity Time Domain verificationl

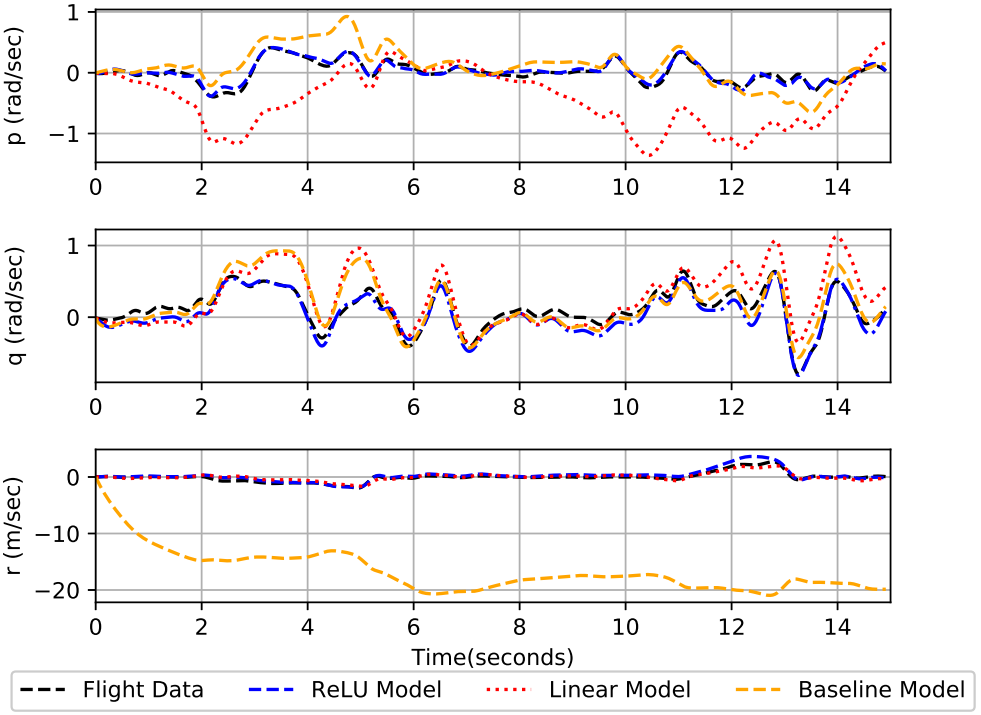


FIGURE 4.7: Rotational Velocity Time Domain verification

4.6 Model Validation

In this section, the derived models are compared to prove the performance. Figure 4.10 compares the simulated values with the measured values of the baseline as well the ReLU model for all the translational and angular velocities. The baseline model has points scattered all over showing less accuracy and diverging nature of the baseline model. The outputs of the ReLU model are bounded between a small and finite value of error. Note that the error bound range is different for the six ReLU networks. Higher error bound range implies less accuracy. Therefore the Lateral, heave and yaw angular velocity ReLU models are less accurate as compare to other ReLU models.

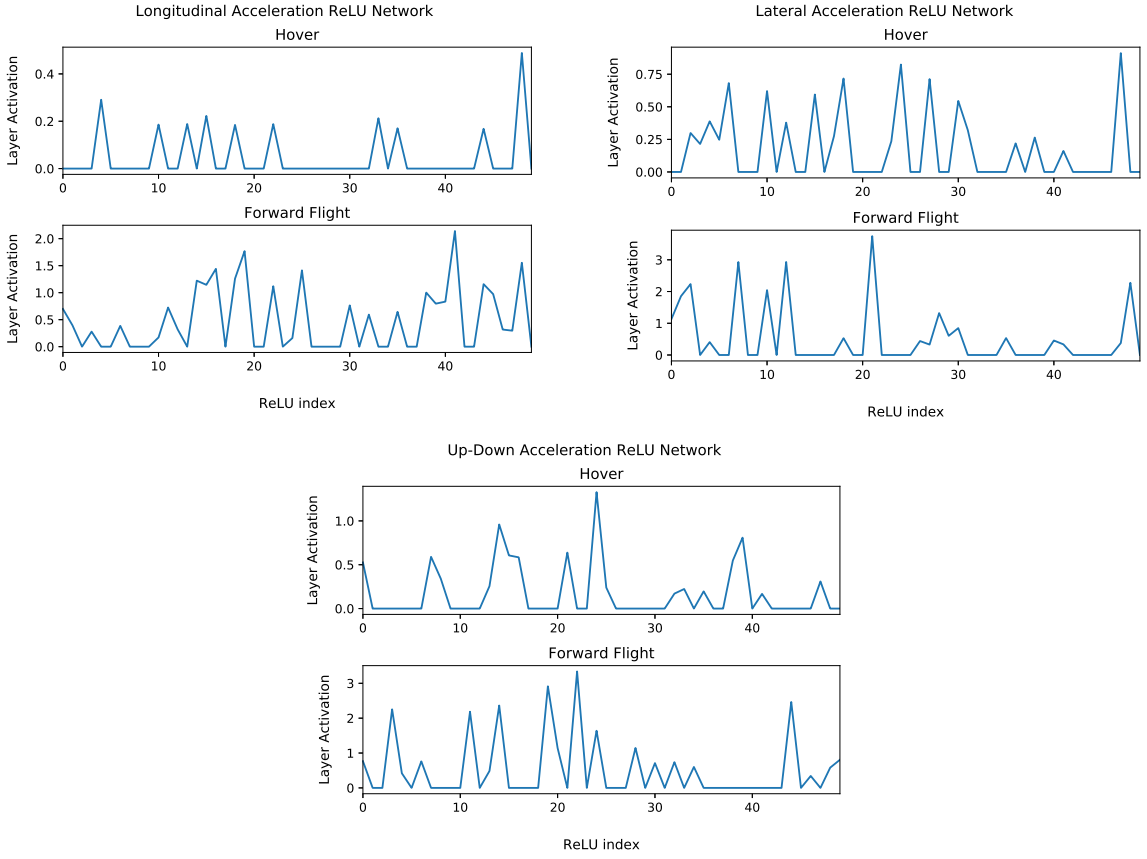


FIGURE 4.8: Comparison of ReLU layer activation at different flight conditions.

The figure 4.8 shows different nodes activated for different flight conditions, which depicts the successful working of the hidden layer thereby finding an effective nonlinear mapping

between the input and output data. To ensure that the non-linear layer is actually functional, the output of the ReLU layer for different flight conditions were compared. For the ReLU layer to contribute to the nature of Φ_e, the following two conditions should be met:

1. Not all outputs of the ReLU layer should be activated, i.e., some of the elements of the output should be zero.
2. The same elements in the output of the ReLU layer shouldn't be always activated for all flight conditions.

Further, the performance of ReLU and Linear acceleration model is shown in figure 4.9. The ReLU model has less output error as compared to the linear correction model. Note that the difference between output error for ReLU and Linear acceleration correction model is less for angular velocity as compared to translation velocity. This clearly depicts that the Linear Correction model performs well for angular velocity prediction as compared to translation velocity. The poor performance of the linear correction model is due to the incapability of the model to account for non-linear behaviour. The ReLU hidden layer activates different nodes at different flight conditions, thereby trying to capture a larger regime of flight.

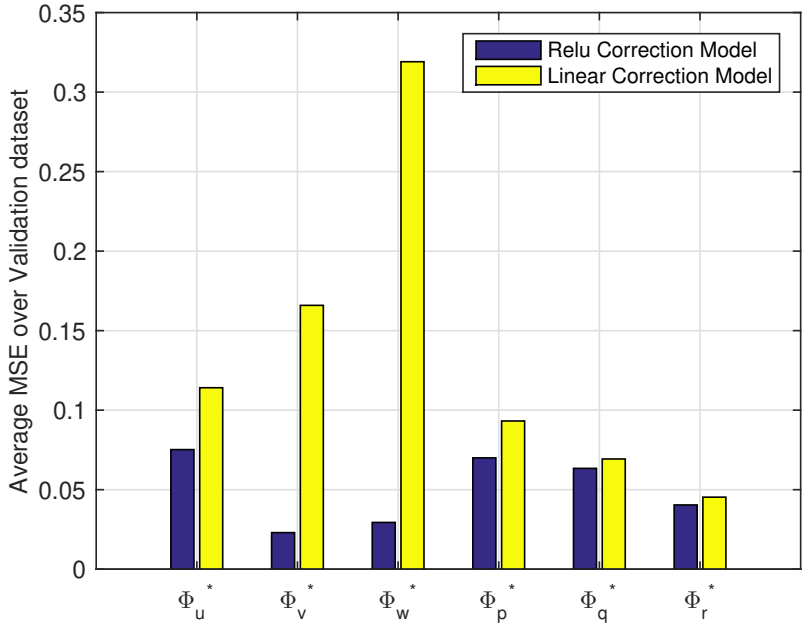


FIGURE 4.9: Batch Mean square error over validation dataset

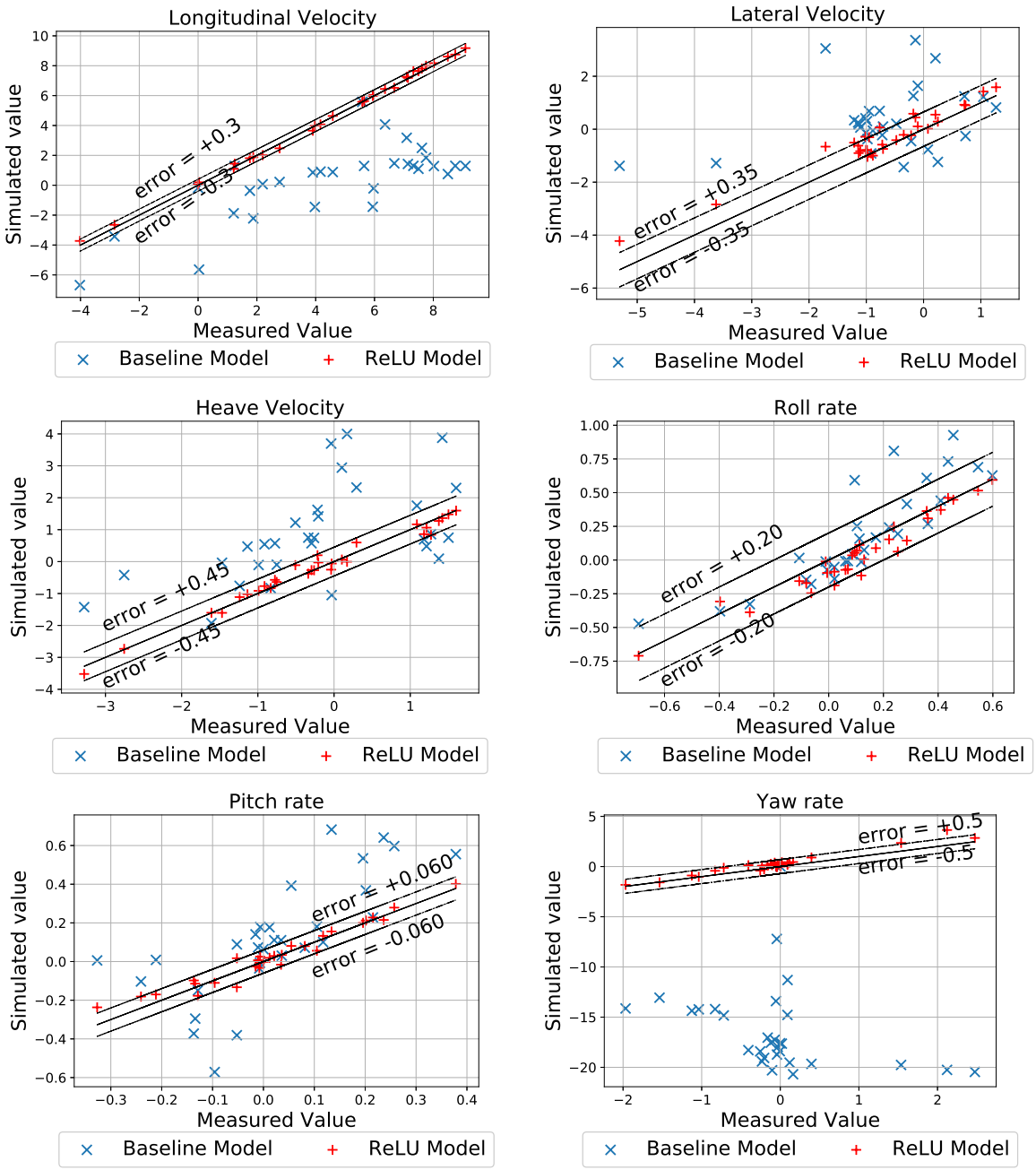


FIGURE 4.10: Comparison of absolute error in prediction of Translational and angular velocities for the baseline model and ReLU model.

4.7 Results and Discussion

Four SISO models were successfully identified using Frequency Domain System Identification corresponding to Lateral, Longitudinal, Heave and Yaw motion. These models are

linear in nature working around an operating condition (The trim conditions in hover and forward flight). These models have good performance when validated against hover doublets and sweep data making them useful control system analysis. However, these models start to diverge as simulation time increases making them unfit for flight dynamic simulations. All the identified parameters of the SISO models have $cramér-rao$ bound percentage less than 20% and insensitivity percentage less than 10%. The identified transfer function costs have been reduced below 100 after convergence.

The baseline model is important for training the Deep Learning model and simulation of flight physics. Two sets of Six neural networks representing translational and angular accelerations $\{\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}\}$, 1)ReLU hidden layer 2)Linear hidden layer; were trained for providing correction to baseline model. These networks have similar model structure; 200 units in the input layer, 50 units in the hidden layer and a single output node to keep memory complexity same for both the model to compare performance. Two hours of aggressive forward flight data is used to train each of the networks which took 3-4 hrs to converge to a small finite error. The activation of the hidden ReLU layer is compared for different flight conditions which show that different nodes are activated for different flight conditions.

Both of the above neural networks are validated against aggressive freestyle forward flight data and the ReLU network is found to outperform the linear correction models as well as the baseline model by bounding the acceleration outputs of the flight dynamics model between small finite error value. Comparison between the output on validation dataset shows that the ReLU network has less average mean square error over the batch than the linear hidden model. The ReLU hidden layer acts as a decision gateway with different nodes activated for different flight condition acting as a universal function approximator for mapping corrections in the baseline model to input, output time series data. The time series data of the input and outputs have been utilised which act as input to the deep learning model.

4.8 Future Work

The accuracy of the accelerations predicted by the ReLU network can be further increased by choosing a larger network structure. A. Punjani et. Al. [19] shows that the average error on training dataset decreases with an increase in units in the hidden layer. However, the error on validation dataset decreases slowly with increase in units in the hidden layer

to reach a point where the error does not change much with the number of hidden nodes. Therefore the model can also be horizontally expanded to check for accuracy.

The deep learning model has a large number of simulation processes, which could be parallelized to reduce computational time. This computational time is an essential benchmark for flight dynamics models of a Simulator. This flight dynamics model can be integrated into a simulator to conduct Software-in-the-loop (SITL) and Hardware-in-the-loop(HITL) simulations for virtually testing control and navigation algorithms on the vehicle.

More Flight data should be collected covering all types of flight conditions including manoeuvres to train networks that can accurately predict the broader regime of the forward flight. The Deep Learning technique can be applied on other vehicles such as fixed wings, tilt-rotors and compound helicopters to quickly develop their flight dynamics model.

Bibliography

[1] Abbeel, P., Coates, A., and Ng, A. Y. (2010). Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639.

[2] Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pages 1–8.

[3] Abbeel, P., Ganapathi, V., and Ng, A. Y. (2006). Learning vehicular dynamics, with application to modeling helicopters. In *Advances in Neural Information Processing Systems*, pages 1–8.

[4] Align Corp. Ltd. (2017). Trex 700L Top Instruction Manual.

[5] Bhandari, S. and Colgren, R. (2015). High-order dynamics models of a small uav helicopter using analytical and parameter identification techniques. *Journal of the American Helicopter Society*, 60(2):1–10.

[6] Carrio, A., Sampedro, C., Rodriguez-Ramos, A., and Campoy, P. (2017). A review of deep learning methods and applications for unmanned aerial vehicles. *Journal of Sensors*, 2017.

[7] Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48.

[8] Ham, J. A., Gardner, C. K., and Tischler, M. B. (1995). Flight-testing and frequency-domain analysis for rotorcraft handling qualities. *Journal of the American Helicopter Society*, 40(2):28–38.

[9] Hashimoto, S., Ogawa, T., Adachi, S., Tan, A., and Miyamori, G. (2000). System identification experiments on a large-scale unmanned helicopter for autonomous flight.

In *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No. 00CH37162)*, pages 850–855. IEEE.

[10] Heffley, R. K. and Mnich, M. A. (1988). Minimum-complexity helicopter simulation math model.

[11] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257.

[12] La Civita, M., Messner, W. C., and Kanade, T. (2002). Modeling of small-scale helicopters with integrated first-principles and system-identification techniques. In *AHS International, 58 th Annual Forum Proceedings-*, volume 2, pages 2505–2516.

[13] Laxmidhar Behera, I. K. (2009). *Intelligent Systems and Control: Principles and Applications*. Oxford University Press.

[14] Leishman, J. G. (2000). *Principles of Helicopter Aerodynamics*. Cambridge University Press.

[15] Ljung, L. (2008). Perspectives on system identification. *IFAC Proceedings Volumes*, 41(2):7172–7184.

[16] Mark B. Tischle, R. K. R. (2006). *Aircraft and rotorcraft system identification*. American Institute of Aeronautics and Astronautics, Inc.

[17] Mettler, B., Kanade, T., and Tischler, M. B. (2000). *System identification modeling of a model-scale helicopter*. Carnegie Mellon University, The Robotics Institute.

[18] Padfield, G. D. (1996). *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. Blackwell Publishing.

[19] Punjani, A. and Abbeel, P. (2015). Deep learning helicopter dynamics models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230.

[20] Px4 Development Team (2018). Cube(Pixhawk 2). `https://docs.px4.io/en/flight_controller/pixhawk-2.html`. [Online].

[21] Samal, M. K., Anavatti, S., and Garratt, M. (2008). Neural network based system identification for autonomous flight of an eagle helicopter. *IFAC Proceedings Volumes*, 41(2):7421–7426.

[22] Setu, S. (2018). *Modeling, hardware-in-the-loop simulation and flight controller development for rotary wing UAVs From first flight to autonomous autorotation*. PhD thesis, Indian Institute of Technology, Kanpur.

[23] Setu, S., Abhishek, A., and Venkatesan, C. (2019). Time-domain system identification of helicopters using nonlinear acceleration and jerk prediction model. *Journal of Aircraft*, pages 1–9.

[24] Suresh, S., Kumar, M. V., Omkar, S., Mani, V., and Sampath, P. (2002). Neural networks based identification of helicopter dynamics using flight data. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.*, volume 1, pages 10–14. IEEE.

[25] Tharun, B. (2017). Identification of hybrid model for small ruav using frequency domain method. M.tech thesis, Indian Institute of Technology Kanpur.

[26] Tischler, M. B. and Cauffman, M. G. (1992). Frequency-response method for rotorcraft system identification: Flight applications to bo 105 coupled rotor/fuselage dynamics. *Journal of the American Helicopter Society*, 37(3):3–17.

[27] Vayalali, P. (2016). Frequency domain based system identification of a small-sized ruav. M.tech thesis, Indian Institute of Technology Kanpur.