UAV Parameter Estimation with Gaussian Process Approximations

Madu Prasad Hemakumara

A thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

March 2013

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Madu Prasad Hemakumara

March 28, 2013

Abstract

Madu Prasad Hemakumara
The University of Sydney

Doctor of Philosophy (Research)
March 2013

UAV Parameter Estimation with Gaussian Process Approximations

Unmanned Aerial Vehicles (UAVs) provide an alternative to manned aircraft for risk associated missions and applications where sizing constraints require miniaturized flying platforms. UAVs are currently utilised in an array of applications ranging from civilian research to military battlegrounds. A part of the development process for UAVs includes constructing a flight model. This model can be used for modern flight controller design and to develop high fidelity flight simulators. Furthermore, it also has a role in analysing stability, control and handling qualities of the platform. Developing such a model involves estimating stability and control parameters from flight data. These map the platform's control inputs to its dynamic response. The modeling process is labor intensive and requires coarse approximations. Similarly, models constructed through flight tests are only applicable to a narrow flight envelope and classical system identification approaches require prior knowledge of the model structure, which, in some instances may only be partially known.

This thesis attempts to find a solution to these problems by introducing a new system identification method based on dependent Gaussian processes. The new method would allow for high fidelity non-linear flight dynamic models to be constructed through experimental data. The work is divided into two main components. The first part entails the development of an algorithm that captures cross coupling between input parameters, and learns the system stability and control derivatives. The algorithm also captures any dependencies embodied in the outputs. The second part focuses on reducing the heavy computational cost, which is a deterrent to learning the model from large test flight data sets. In addition, it explores the capabilities of the model to capture any non-stationary behavior in the aerodynamic coefficients. A modeling technique was developed that uses an additive sparse model to combine global and local Gaussian processes to learn a multi-output system. Having a combined approximation makes the model suitable for all regions of the flight envelope. In an attempt to capture the global properties, a new sampling method is introduced to gather information about the output correlations. Local properties were captured using a non-stationary covariance function with KD-trees for neighbourhood selection. This makes the model scalable to learn from high dimensional large-scale data sets.

The thesis provides both theoretical underpinnings and practical applications of this approach. The theory was tested in simulation on a highly coupled oblique wing aircraft and was demonstrated on a delta-wing UAV platform using real flight data. The results were compared against an alternative parametric model and demonstrated robustness, improved identification of coupling between flight modes, sound ability to provide uncertainty estimates, and potential to be applied to a broader flight envelope.

Acknowledgements

Looking back at the journey to the completion of this thesis, there were many who have continuously provided me with support and encouragement, without which this thesis would not have been possible.

First of all I am indebted to my supervisor Professor Salah Sukkarieh for presenting me with the opportunity to undertake this thesis project. Also, I am grateful for his guidance, patience, and suggestions in all phases of my research.

I would like to extend my sincere thanks to the technical advice I received from Arman Melkumyan, Tariq Abuhashim and Jason Gan throughout my degree.

Whilst at the Australian Centre for Field Robotics (ACFR) the practical experience I gained through many flight trials was invaluable. During these trials, I appreciate the help and advice received from our UAV team, especially Jeremy Randle, Steve Keep, Zhe Xu and Esa Attia on operating UAVs. Also, to my other friends at ACFR thanks for making my postgraduate years an enjoyable one. I am fortunate to have the opportunity to work with such a great group of colleagues.

I would also like to acknowledge the help and support received from Melody Chiou during the last six months of my thesis.

Finally, a special thank you to my parents, Dr. and Mrs. Hemakumara for their unwavering support and encouragement throughout my lifetime. If it were not for their encouragement and faith in me during my postgraduate years, this thesis would not be possible. Also, for my sister Mindya thanks for motivating me throughout my studies.

The Australian Research Council Centre of Excellence for Autonomous Systems Scholarship in Field Robotics and the Australian Centre for Field Robotics funded the research described in this thesis. This support is gratefully acknowledged.

Dedicated to my parents.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

ACFR	Australian Centre for Field Robotics
AGL	Above Ground Level
CFD	Computational Fluid Dynamics
CIFER	Comprehensive Identification from Frequency Responses
CG	Centre of Gravity
DGP	Dependent Gaussian Process
DGPS	Differential Global Positioning System
DOF	Degrees of Freedom
GP	Gaussian Process
GPS	Global Positioning System
LS	Least Squares
MI	Mutual Information
MIMO	Multi-Input Multi-Output
MPC	Model Predictive Control
NASA	National Aeronautics and Space Administration
NN	Neural Network
RBF	Radial Basis Function
RMS	Root Mean Square
SISO	Single-Input Single-Output
UAV	Unmanned Aerial Vehicle
VTUAV	Vertical Takeoff UAV

Greek Symbols

α	Angle of attack, deg
β	Angle of sideslip, deg
δ	Control deflection, deg
ρ	Air density, kg/m^3
ϕ	Bank angle, deg

θ	Pitch angle, deg
ψ	Heading angle, deg

Latin Variables

b	Wing span, m
$\mathbf{C}_a^i$	Air-path to inertial frame transformation matrix
C_D	Drag coefficient
C_X	Longitudinal force coefficient
C_Y	Lateral force coefficient
C_Z	Down force coefficient
C_l	Roll coefficient
C_m	Pitch coefficient
C_n	Yaw coefficient
D	Number of dimensions
g	Gravitational acceleration, m/s^2
m	Vehicle mass, kg
p	Roll rate, deg/s
q	Pitch rate, deg/s
r	Yaw rate, deg/s
S	Wing area, m^2
s	Time, s
u	Forward velocity component, m/s
v	Sideways velocity component, m/s
V	True airspeed, m/s
w	Downwards velocity component, m/s
z	Observation

Subscripts

0	Initial value
a	Aileron
b	Body
e	Elevator
i	Initial
r	Rudder

Chapter 1

Introduction

The aim of this thesis is to develop a non-parametric Unmanned Aerial Vehicle (UAV) system identification method to learn a comprehensive model of the aerodynamic parameters. The thesis also aims to demonstrate and test the algorithms using a real UAV.

As a robotic platform, UAVs provide an alternative to manned aircraft for risk associated missions, and for applications where sizing constraints require small scale flying platforms. They also help to reduce the cost and increase operational capabilities of aerial missions. Fixed wing UAVs are currently in use in a wide range of areas from civilian research to military applications. Civilian applications include exploration, surveying, and search and rescue missions. Military applications include reconnaissance, target tracking, and as a communication relay.

One of the first stages in building an aircraft is to construct a mathematical model to simulate the flight. These can be constructed from aerodynamic, inertial and structural characterisations of the aircraft's individual component elements such as the fuselage, wing and empennage. This helps to analyse stability, control and handling qualities of the platform. The models can also be used for modern flight controller design, which rely on feedback of the state vector. They characterise the applied forces and moments acting on the platform arising from aerodynamics and propulsion. The thrust forces and moments from propulsion can be obtained by performing ground tests. Thus, learning involves only knowing the forces and moments result from aerodynamic characteristics and control variables. The aerodynamic model is based on first principles such as using the finite wing theory, or by using empirical data of a similar platform. More complex models may involve wind-tunnel tests and numerical methods such as computational fluid dynamics. Estimates of the mass and inertia typically involve coarse approximations. The more complex structural models are based on NASTRAN [11] calculations. Thus, constructing a mathematical model for a flight is a modular systematic process and is prone to error. For UAVs there might also be significant modifications done to the platform that might cause the vehicle inertia characteristics to change on a frequent basis depending on the mission. Hence, it is not always practical to perform physical based modeling.

In order to learn a more comprehensive realistic model, flight tests can be conducted to learn the system dynamics from the imperfect observations or measurements collected. This is known as aircraft system identification. Here, test maneuvers are designed to excite the dynamic response modes of the aircraft so a dynamic model can be extracted from the logged flight data. This involves designing test maneuvers to excite the dynamic response modes and regressing the collected flight data into a hypothesized model [35]. To construct a complete model the vehicle parameters are identified for different speeds as well as different dynamic pressures.

UAVs also have a shorter development cycle compared to manned aircraft. For instance, a UAV could take 6 - 12 months to design and develop, as opposed to 5 - 10 years for a manned aircraft. This shorter time-frame usually precludes physical based modeling approach. As a result, the flight control system can be entirely dependent on the system identification flights made at the early stages of flight testing. This has been demonstrated in both fixed- and rotary-wing UAVs, namely Northrop-Grumman vertical takeoff UAV (VTUAV) [13], NASA's Pathfinder and Yamaha R-50 [76].

The aim of this thesis is to provide an understanding of complex aerodynamic models, and to define where these existing models can be improved. Also, the thesis aims to develop algorithms to address these issues and provide validation through demonstrating by learning a model for a real UAV.

1.1 Motivations

This thesis is motivated in two ways. First, it is to bridge the gap in learning non-traditional flight models with cross coupling between longitudinal and lateral flight modes. Second, it is the general problem of utilizing a non-parametric system identification method to learn aerodynamic models.

Airplane platforms with either delta wing configuration or an oblique wing have high level of cross coupling between its longitudinal and lateral dynamics. These conditions are required when designing a platform that has to travel at high speeds and longer ranges with minimal fuel consumption. The design of the platform with an increase in wing sweep will decrease the aerodynamic (lift induced) drag. This brings an operational advantage for the given set of mission requirements. Nevertheless, these platforms suffer when it comes to control. They lack directional stability and roll-pitch coupling. Thus, in order to design a control system for such platforms, a good understanding of the dynamics is required. It must also consider and learn the aforementioned factors of the coupled flight dynamics.

The existing methods in classical system identification require prior knowledge of the model structure, which in some instances may only be partially known. The goodness of this structure is dependent on how much time and effort was invested by the design engineer. A limited model structure will restrict its applicability to a narrow band of the flight envelope. This restriction can be alleviated by using a non-parametric system identification technique.

Hence, the motivation of this work is to address these issues by introducing a new system identification method based on dependent Gaussian processes.

1.2 Related Fields

This work is primarily drawn from both the robotics and aerospace communities. A brief overview of the main areas is provided here. A more in-depth review is given in the following chapters.

System Identification

System identification refers to the processes of learning a mathematical model for a system. It uses statistical methods from mathematics to construct models for dynamic systems from measured data. For UAVs or aircraft this is done by performing flight tests and using the observations collected to regress into a hypothesized model [35]. The models can describe the behavior of the system in either the time or frequency domain. There are two classes of problems here; one is when the system process can be derived from first principles such as using Newton's equations of motion and the other is when it is not possible to resort to theoretical derivations. This is due to practical reasons and the nature of increased complexity to model such a system. A perfect example for this is an aerodynamic model for an aircraft or a UAV which has to account for many factors that vary across the flight regime.

The approach when it comes to such problems is to start from external inputs to the system and measurements of its behavior to determine the relationship between them. The model can be developed by either using some knowledge of the system or by using no-prior information about the system [35]. If there is prior knowledge about the system then a model structure can be assumed and system identification becomes a parameter estimation problem. When it is not known, a model can be constructed by using advanced statistical techniques. This class of problems are classified as black box models. Primarily they consist of algorithms that come from the machine learning community.

The quality of the data collected has a major impact on the resultant model. This includes optimal experimental design to gather the most informative data to fit the models [25]. The experiments are designed by evaluating statistical criteria to maximize the information content. For aircraft and UAVs, this involves exciting the system modes such that the sensitivities of the model outputs to the parameters are high and the correlation among the parameters are low [79].

Machine Learning

Machine learning is a discipline concerned with developing algorithms that take empirical data as inputs and to use that information to make future predictions [5]. It is also used to recognise complex patterns and make decisions based on prior input data. In this work a machine learning approach is combined with the known equations of flight mechanics to estimate the aircraft state. Hence, the algorithms developed are only used to model a part of the system. This reduces the complexity of the model that has to be learned. In general, this class of problem is known as non-parametric techniques. Most of the existing algorithms developed have used techniques such as neural networks [42] and apprenticeship learning [12].

1.3 Main Contributions

The thesis has several contributions in both theoretical and practical applications in robotics. They have been presented in [26], [27] and [28]. The fundamental contributions of this thesis are as follows:

- The theoretical development of a new system identification method for aircraft. The system parameters are identified by modeling them as Dependent Gaussian Processes (DGPs). The non-parametric nature of the approach enables it to capture a wide range of dynamics. Therefore, prior knowledge of the model structure is not required. The ability to capture dependencies through cross coupling terms enables the models to be applicable to a broader flight envelope than existing methods saving significant costs on flight testing. In addition, it has the inherent capability to handle noise and biased data. The predictions from the model also come with uncertainty estimates which can be used in maneuver design for system identification and for flight controller design. Finally, it is not entirely a black-box prediction model as opposed to techniques such as neural networks.

- The work demonstrates the dependencies for which the method was explicitly chosen, its robustness to unmodeled disturbances and quantify improvements over an alternative least squares estimator. Also, an in-depth analysis to identify parameters for the highly coupled AD-1 oblique wing aircraft was presented. The analysis was performed using a high fidelity flight simulator.

- To address the increase in computational cost, a new sampling method was introduced for DGPs. This is based on a mutual information criterion that can sample features from data to describe output correlations. Hence, it requires fewer features to describe the data set.

- The model was also extended to capture any non-stationary properties in aerodynamics. To tackle this problem while retaining the desired properties of DGP, an additive sparse Gaussian process model was proposed. This combines both short and long length-scale phenomena for a multi-output GP.

- In simulation, a quantitative comparison of the performance was done with a classical system identification method. The models developed were then demonstrated with several examples using real flight test data from a UAV. The results were analysed and any improvements were noted. The experiments also identified implementation challenges.

1.4 Overview of the Thesis

This thesis is structured as follows:

Chapter 2 introduces the problem of system identification and provides a literature review on existing methods. It lays down the groundwork of flight dynamic modeling and system identification theory. The elements of system identification such as experiment design, data collection, model structure selection, parameter estimation and model validation is discussed. The general mathematical model form for a UAV is reviewed. It shows how the system identification can be reduced down to an aerodynamic parameter estimation problem. It reviews the theory related to parameter estimation and shows a classical technique for model regression. In addition, an introduction to non-parametric Bayesian regression and the basic theory behind Gaussian processes is presented.

Chapter 3 presents Gaussian process modeling of the flight dynamics. It gives a background on how GPs have been used in dynamic modeling. Then the chapter introduces how to incorporate Dependent Gaussian Processes (DGPs) to learn an aerodynamic model. It shows the advantage of using DGPs for modeling as opposed to GPs. It also discusses how this can be incorporated in the overall identification process. Lastly, the flight testing procedure for parameter identification is elaborated and model validation is performed in a comprehensive simulation. The simulator uses a complex aircraft model with a high level of parameter cross coupling to demonstrate the advantage of using the proposed algorithm. These results are then compared to a classical system identification method. They emphasise the benefit to be gained by not having to know an a prior model structure.

Chapter 4 develops an algorithm to make an approximation on the multi-output GP. It gives an overview of GP approximation techniques and presents the theory for a combined local and global approximation. In doing so it also introduces a sampling strategy for multi-output GPs based on a mutual information criterion. Additionally, simulations were done to demonstrate the performance of mutual information based sampling verses the well known entropy sampling. It shows the improvement in reduction in prediction error with less number of sample points using the proposed algorithm. The results are then compared for global GP, local GP and the combined approximation to illustrate the benefits of having a combined approximation. It shows why the two individual components are important. Finally, a discussion of the performance of the approximation compared to the full DGP model is provided.

Chapter 5 presents experimental results of algorithm testing with data from real UAV flight tests. The platform, instrumentation and data collection procedure is presented. The results show the model performance of the two algorithms described in Chapters 3 and 4. These were shown as two separate experiments. The first experiment is to test the DGP algorithm provided in Chapter 3 to model aerodynamic coefficients. It shows the measurands, the procedure for flight testing and model training, testing of the learned model and the experimental results of the estimated aerodynamic coefficients. In addition, details of how to estimate the aircraft states from the aerodynamic coefficients is provided along with results. The second experiment is done to test the model performance of using the approximate multi-output GP proposed in Chapter 4. Results of estimating several test flights are shown to test the robustness.

Chapter 6 concludes the thesis. A note on using the proposed models for control is given. Finally, it summarises the contributions made and provides suggestions for future research directions.

Chapter 2

Background

This chapter presents the background and theory that forms the technical basis of this thesis work. An overview of UAV system identification is presented followed by the technical background. This provides a context of the ideas and techniques presented in the subsequent chapters. The fundamentals and the basic theory behind UAV system identification are presented. Furthermore, the following subsections develop the underlying principles of Bayesian regression, in particular Gaussian processes which was explored as a possible solution to non-parametric UAV system identification.

2.1 UAV System Identification

A UAV can be considered as an input-output system (see Figure 2.1). The vehicle dynamics are excited by the control inputs, a traditional UAV has ailerons δ_a, elevator δ_e, rudder δ_r and throttle δ_T as inputs. The ailerons are used for roll control, elevator is used for pitch control, rudder is used for yaw control and throttle is to control the speed. The UAVs response to these inputs can be recorded by an onboard computer. The flight dynamics that are recorded and taken into consideration are: translational and angular velocities, Euler angles, aerodynamics angles and accelerations. The dynamic model then relates the control inputs of the vehicle to its response. In system

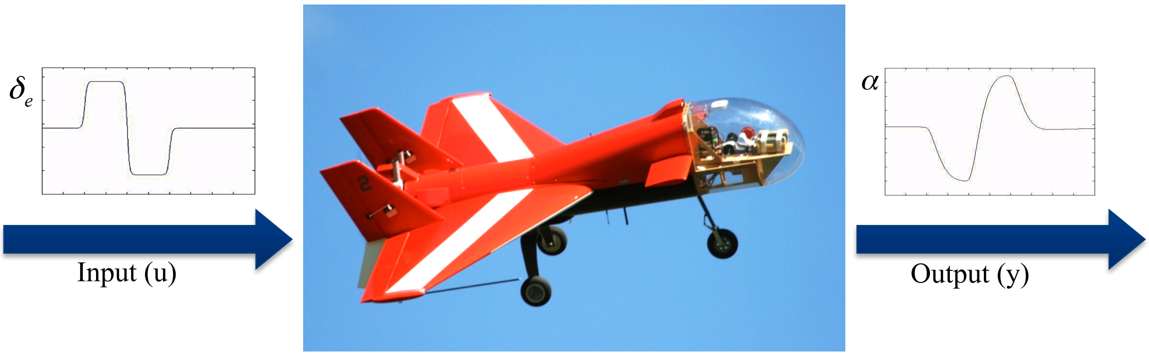


Figure 2.1: System identification overview.

identification, flight tests are conducted to collect observation data in-order to accurately model the flight dynamics. This involves designing test maneuvers to excite the dynamic response modes and regressing the collected flight data into a hypothesized model. To construct a complete model the vehicle parameters are identified for different speeds as well as different dynamic pressures. These models constructed can then be used for modern flight controller design, which rely on feedback of the state vector. They can also be used to develop high fidelity flight simulators. In addition, the system outputs can be used to analyse stability, control and handling qualities of the platform.

The parameters to be identified are also known as stability and control derivatives [1], which are measures of how particular forces and moments on an aircraft change as other parameters related to stability (rotational rates, angle of attack, etc) and control (control inputs) change. The first step in identifying parameters is to postulate the equations governing the flight dynamics and then design experiments to obtain measurements of those variables. The aircraft motion can be described by the Newton's second law of motion in translational and rotational forms. A detailed derivation of the dynamics is given in Section 2.3. A summary of the flight dynamics in vector form are

$$m\,\dot{\boldsymbol{V}}_b + \boldsymbol{\omega}_b \times m\,\boldsymbol{V}_b = \boldsymbol{F}_G + \boldsymbol{F}_T + \boldsymbol{F}_A(\,\boldsymbol{V}_b, \boldsymbol{\omega}_b, \boldsymbol{u}_b, \boldsymbol{\vartheta}) \tag{2.1.1}$$

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \boldsymbol{I}\boldsymbol{\omega}_b = \boldsymbol{M}_T + \boldsymbol{M}_A(\,\boldsymbol{V}_b, \boldsymbol{\omega}_b, \boldsymbol{u}_b, \boldsymbol{\vartheta}) \tag{2.1.2}$$

where m is the UAV mass, $\boldsymbol{V}$ and $\boldsymbol{\omega}$ are translational and angular velocity vector components, $\boldsymbol{u}$ is the control vector, subscript b represent they are components from the UAV body axis. The right side of the equations represent applied forces and moments. The forces result from gravity ($\boldsymbol{F}_G$), thrust ($\boldsymbol{F}_T$), and aerodynamics ($\boldsymbol{F}_A$). Applied moments are a result of thrust ($\boldsymbol{M}_T$) and aerodynamics ($\boldsymbol{M}_A$). The quantity $\boldsymbol{\vartheta}$ are parameters that specify aerodynamic characteristics of the UAV.

UAV system identification can be defined as the determination of unknown parameters $\boldsymbol{\vartheta}$ contained in the model structure from input and output measurements. For practical application this prior model structure is assumed to be known and the identification problem then reduces to parameter estimation. There are several steps to be taken; model postulation, experimental design, data compatibility analysis, collinearity diagnostics, model structure selection, parameter estimation and model validation [35]. These are necessary to maintain integrity and the relevance of the data for the hypothesised model structure. A block diagram of this process is given in Figure 2.2.

Prior Model Structure. This is the flight model structure based on prior knowledge about the UAVs flight dynamics. These can be obtained by an empirical model of a similar UAV. The prior knowledge about the model structure will influence the experiments to be designed to learn those parameters. By knowing the underlying structure it can be used to design test maneuvers to excite the required model parameters.

Experiment Design. This includes designing the flight test maneuvers, instrument selection and choosing the right flight conditions to conduct the experiment. The maneuvers influence the response of the UAV. They are designed to collect high signal to noise ratio data sets and to excite the set of parameters to be learned. The instrumentation is done to choose the correct sensors to make recordings of the input and output variables of the system. These can be logged into an on-board flight computer, which can be used for post processing. The monitored input variables include all the control inputs such as the throttle positions and control surface deflections. The outputs include translational and angular velocities, accelerations, Euler angles,

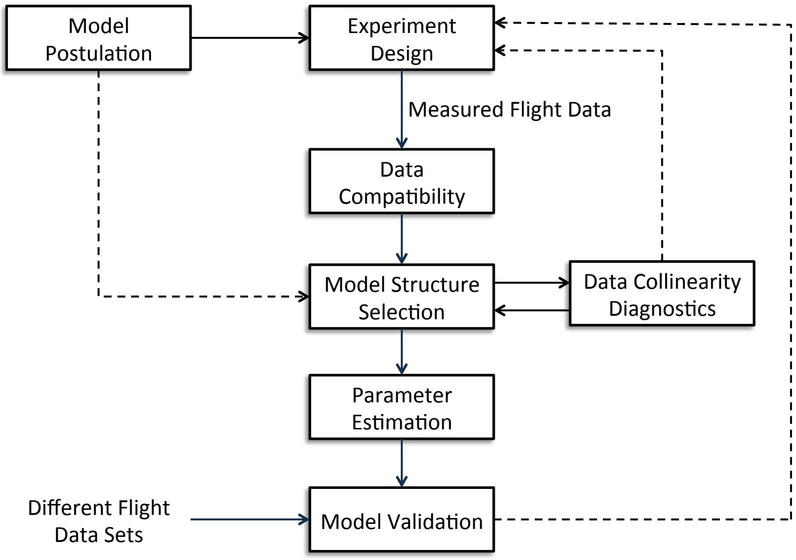


Figure 2.2: System identification block diagram.

and magnitude (true airspeed) and direction (angle of attack and angle of sideslip) of the air-relative velocity. In addition, to record the flight condition, the air density is also logged.

Data Compatibility. The data collected from an experiment may not always be correct. In many practical instances they may contain systematic errors or high level of noise. An analysis can be conducted to check whether the data collected for training is accurate. This involves reconstructing the aircraft state response with known rigid-body kinematic equations and comparing the reconstructed response with the measured responses. This is further discussed in the Section 5.2.1.

Model Structure Selection. With some prior knowledge about the model structure and the data collected the final form of the model structure can be determined. This might involve choosing an appropriate set of polynomial expansion terms. The model shall not be over parametrized but must fit all the data well. A more complex model can always fit the data better but may generalize poorly. Hence, the structure should be chosen such that it supports the estimation of all the unknown parameters associated with the model terms as well as giving the model good prediction capability.

Parameter Estimation. This is one of the most important steps in the identification process. Its where the estimation technique learns the parameters from the logged flight data. The most widely used technique here is to use a parametric technique such as the equation error or output error method. More about these various techniques are given in Section 2.2.

Collinearity Diagnostics. Data collinearity occurs when several regressors are said to be linearly dependent. In this case it is difficult to assign a unique value for each parameter to represent it's contribution. Hence, this would make the parameter identification problem ill-conditioned. In practice they might be almost linearly dependent but may not be a perfect fit, but identifying the slight changes and differentiating each is a difficult task. Operationally, when data collinearity exists the parameter estimation process will produce inaccurate parameter estimates with large variances or the estimation process itself may fail.

Model Validation. This is the last step in the system identification process. It tests if the identified model has good prediction capability for different flight data sets. The measured input data is applied to the model to generate the predicted responses. These are then compared with the measured values. The consistency in predictions across various maneuvers confirms the parameters identified were a good representation of the system.

2.2 Related Work

System identification techniques use time response or frequency response data from flight tests to produce flight models. These are accurate in the region of the collected data points. However, the performance degrades outside of the operational neighborhood. In addition, it should be noted that the frequency domain characterization can only be done by linearizing the dynamics.

There has been work done on developing models using both parametric and non-parametric approaches. Parametric approaches involve time domain techniques [29] such as ordinary least squares [35], maximum likelihood [49], equation-error [33] and output error [47] methods. There are also parametric approaches based on frequency domain methods [34, 50, 52, 77]. The advantage of these approaches is that one can incorporate prior knowledge about the structure of the parameterization into the estimation formulation. If this prior knowledge is correct, the generalization error of parametric estimation will outperform any non-parametric method. In addition, during the model construction process it is possible to observe the responsiveness of each parameter independently which is not possible using non-parametric techniques. Nevertheless, in order to regress the observations a model structure is required. It is often the case that designing a good parametric model structure may require significant commitment of time and effort by the design engineer. A limited model structure will restrict its applicability to a narrow band of the flight envelope. Therefore, a good physical understanding of the system being modeled is required. It might also be necessary to start with a good initial guess for the identified parameters. For new unconventional UAVs this might not work as it is not possible to make initial assumptions without knowing the platform dynamics. But once the model structure is known the parameters could be identified with ease using parametric techniques. The problem then reduces to determining the values for the unknown coefficients. Hence, it can also be thought of as a parameter estimation problem. Such restrictions can be alleviated by using a non-parametric system identification technique.

In [90] two autoregressive with exogenous input (ARX) models were identified and were used for control design. The models identified were to estimate pitch and roll angles and are dependent only on the elevator and aileron respectively. This is only applicable during steady flight conditions where the flight performed loitering maneuvers. Also, the identified model can only estimate two states of the flight hence it is not a complete representation of the system. In addition, it will not be applicable if there is cross coupling between input and output parameters. The proposed technique does not have these limitations.

There have been several instances where non-parametric methods were used to model aircraft. This is concerned with characterizing only the measured input-output behavior of the UAV dynamics and not its equations of motion. They can be modeled in time domain or frequency domain as well. In either case no assumptions about the model structure is required. In [2, 12] apprenticeship learning was applied to learn the model for a robotic helicopter which was designed to perform aerobatic maneuvers. The results demonstrated that it is able to learn a flight model along a complex flight trajectory. Nevertheless, the model is confined to operate only within that trained trajectory. Neural Networks (NN) [42] have also been applied to the problem of aircraft system identification. In [4] they were used to model wing bending moment, torsional loads and control surface hinge moments. In [10] they were used to drive the state error between a reference flight model and the actual aircraft to zero. However they have not been used to construct a complete flight model for an aircraft. Gaussian Processes (GPs) have been applied to the problem of system identification from training data in [24, 36]. The work in [36] demonstrated GPs and reinforcement learning could be used for system identification and control of a blimp. Here a GP model was used in conjunction with a non-linear dynamic model to learn the state errors. The approach is limited to identifying single independent outputs and hence can not capture dependencies between the identified parameters. The experiments were also conducted in a lab environment without wind disturbance.

The US Army and NASA Ames research centre jointly developed a modeling and software technique called the Comprehensive Identification from Frequency Responses (CIFER) [78] for system identification based on the non-parametric frequency response. It extracts a complete set of Multi-Input Multi-Output (MIMO) frequency responses, identifies state space models and calculates handling qualities to fully characterize the system. It has been used for both fixed wing and rotorcraft UAV programs. This includes the NASA's solar Pathfinder [76], the MQ-8B Fire Scout [20], the Shadow 200 fixed wing UAV [79], and SH-2G helicopter [80]. Not having to know a prior model structure has meant that it has been effective for problems of rotorcraft system identification.

2.3 Problem Formulation

An overview of aircraft dynamics and formulation of the problem is presented in this section. The continuous state and output equations for an aircraft can be expressed as

$$\dot{\boldsymbol{x}}(\tau) = \hat{\boldsymbol{\kappa}}[\boldsymbol{x}(\tau), \boldsymbol{\delta}(\tau), \boldsymbol{\varepsilon}(\tau), \tau] \tag{2.3.1}$$

$$\boldsymbol{y}(\tau) = \hat{\boldsymbol{\lambda}}[\boldsymbol{x}(\tau), \boldsymbol{\delta}(\tau), \tau]. \tag{2.3.2}$$

Here $\boldsymbol{x}$ denotes the state vector, $\boldsymbol{\delta}$ refers to the control input , system output is $\boldsymbol{y}$ and $\boldsymbol{\varepsilon}$ denotes zero mean system noise. The entire process varies from time τ_0 to time $\tau \geq \tau_0$ The global representation of the flight model is contained in $\hat{\boldsymbol{\kappa}}(\cdot)$ and $\hat{\boldsymbol{\lambda}}(\cdot)$. To model the aircraft it is assumed the platform is a rigid body such that the general motion can be described by Newton's second law.

2.3.1 Reference Frames

To develop a flight model it is necessary to first define the reference frames. The main reference frames required to describe the flight dynamics and system identification are shown in Figure 2.3.

- **Earth axes**: Fixed to the surface of the earth with positive x_e axis towards geographic north, y_e axis to the east, and z_e to the centre of the Earth. This reference frame is fixed with respect to the Earth and is assumed to be the inertial axes for system identification.

- **Body axes**: Fixed with respect the aircraft with the origin at the Centre of Gravity (CG) of the platform. The positive x_b points forward through the nose of the aircraft, y_b out the right wing and z_b is through the underside of the aircraft.

- **Wind axes**: This is also referred to as the air-path axes and is not fixed with respect to the body of the aircraft. It is aligned with the relative wind with

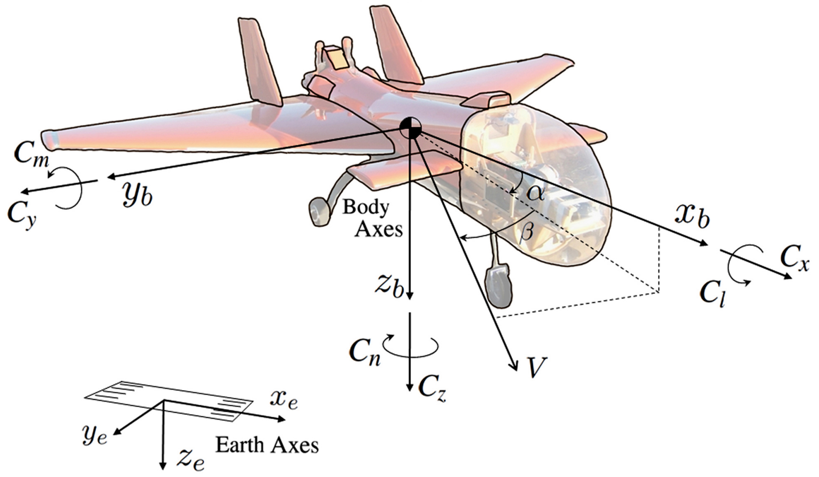


Figure 2.3: Frames of reference for a UAV, in body axes (x_b, y_b, z_b) and earth axes (x_e, y_e, z_e).

origin at the aircraft CG. The positive x_a axis is forward and aligned with the air-relative velocity vector $\bar{V}$ (see Figure 2.3), y_a axis is out right of the aircraft and z_a is through the underside.

2.3.2 Dynamics of the UAV

The forces and moments which act upon an aircraft determine its general motion. This can be described by using Newton's second law of motion in translation and rotational forms

$$\boldsymbol{F} = \frac{d}{dt}(m\,\boldsymbol{V}_b) \tag{2.3.3}$$

$$\boldsymbol{M} = \frac{d}{dt}(\boldsymbol{I}\boldsymbol{\omega}_b) \tag{2.3.4}$$

where $\boldsymbol{F}$ is the applied force, m is the mass, $\boldsymbol{V}_b$ is the translational velocity, $\boldsymbol{M}$ is the applied moment about the CG, $\boldsymbol{I}$ is the inertia matrix and $\boldsymbol{\omega}_b$ is the angular velocity. The vector components are $\boldsymbol{F} = [F_x, F_y, F_z]$, $\boldsymbol{M} = [M_x, M_y, M_z]$, $\boldsymbol{V}_b = [u, v, w]$ and $\boldsymbol{\omega}_b = [p, q, r]$. Also $m\,\boldsymbol{V}_b$ is the linear momentum and $\boldsymbol{I}\boldsymbol{\omega}_b$ is the angular momentum about the CG. Equations (2.3.3) and (2.3.4) describe the motion in vector form that is valid in the inertial frame but the individual quantities are generally described in terms of body axis components. To interpret the body axis results they are generally translated back to the inertial reference frame by performing coordinate transformations. The air-path to inertial transformation matrix is denoted by $\mathbf{C}_a^i$ and is made up of the standard rotational cosine matrices $\mathbf{L}_x$, $\mathbf{L}_y$ and $\mathbf{L}_z$

$$\mathbf{C}_a^i = \mathbf{L}_z(\psi)\mathbf{L}_y(\theta)\mathbf{L}_x(\phi) \tag{2.3.5}$$

where $\boldsymbol{\Psi}_b = [\phi, \theta, \psi]$ are the Euler angles. The inertia matrix $\boldsymbol{I}$ is constant in body axis. It is represented as

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{2.3.6}$$

For a UAV with symmetric rigid body relative to the $x_b z_b$ plane, the inertia matrix $\boldsymbol{I}$ is always symmetric. Therefore, $I_{xy} = I_{yx} = I_{yz} = I_{zy} = 0$ and the inertia matrix reduces to

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{zx} & 0 & I_{zz} \end{bmatrix} \tag{2.3.7}$$

where I_{xx}, I_{yy}, I_{zz} are the diagonal components and I_{xz} is the tensor component of the aircraft's symmetric inertia matrix. Body-axis component form of the equations (2.3.3) and (2.3.4) are given below. This accounts for the rate of change of the vector components and axis system rotation is given by

$$\boldsymbol{F} = m\,\dot{\boldsymbol{V}}_b + \boldsymbol{\omega}_b \times m\,\boldsymbol{V}_b \tag{2.3.8}$$

$$\boldsymbol{M} = \boldsymbol{I}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \boldsymbol{I}\boldsymbol{\omega}_b \tag{2.3.9}$$

where the vector components $\dot{\boldsymbol{V}}_b = [\dot{u}, \dot{v}, \dot{w}]$ are the translational accelerations and $\dot{\boldsymbol{\omega}}_b = [\dot{p}, \dot{q}, \dot{r}]$ are the angular accelerations in the body axes reference frame. With the assumption of rigid body and constant mass, the equations (2.3.3) and (2.3.4) can be represented in component form. The force components in body axis form are give by

$$F_x = m(\dot{u} + qw - rv) \tag{2.3.10}$$

$$F_y = m(\dot{v} + ru - pw) \tag{2.3.11}$$

$$F_z = m(\dot{w} + pv - qu) \tag{2.3.12}$$

Moments are slightly more involved based upon the aircraft's moments of inertia, the angular rates and angular accelerations

$$M_x = I_{xx}\dot{p} - I_{xz}(\dot{r} + pq) - (I_{yy} - I_{zz})qr \tag{2.3.13}$$

$$M_y = I_{yy}\dot{q} - I_{xz}(r^2 - p^2) - (I_{zz} - I_{xx})rp \tag{2.3.14}$$

$$M_z = I_{zz}\dot{r} - I_{xz}(\dot{p} - qr) - (I_{xx} - I_{yy})pq \tag{2.3.15}$$

Applied Forces and Moments

The applied forces and moments on the left hand side of equations (2.3.10) - (2.3.15) arise from the aerodynamics of the platform, gravity and propulsion. The primary system inputs are from the ailerons, elevator and rudder which control the roll, pitch and yaw of the aircraft respectively. Since the gravitational component acts through the CG and is assumed to be uniform, there is no gravity moment on the platform. Therefore, the above equations can be written as the body axis components of applied forces and moments

$$\boldsymbol{F}_G + \boldsymbol{F}_T + \boldsymbol{F}_A = m\,\dot{\boldsymbol{V}}_b + \boldsymbol{\omega}_b \times m\,\boldsymbol{V}_b \tag{2.3.16}$$

$$\boldsymbol{M}_T + \boldsymbol{M}_A = \boldsymbol{I}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \boldsymbol{I}\boldsymbol{\omega}_b \tag{2.3.17}$$

Applied forces in these equations come from gravity ($\boldsymbol{F}_G$), thrust ($\boldsymbol{F}_T$), and aerodynamics ($\boldsymbol{F}_A$). Applied moments are a result of thrust ($\boldsymbol{M}_T$) and aerodynamics ($\boldsymbol{M}_A$).

The contribution of gravity in vector form is

$$\boldsymbol{F}_G = \begin{bmatrix} -mg\sin(\theta) \\ mg\sin(\phi)\cos(\theta) \\ mg\cos(\phi)\cos(\theta) \end{bmatrix} \tag{2.3.18}$$

where, g denotes gravity. This takes into account the dependency of gravity components to aircraft orientation relative to the earth axis. The applied forces and moments due to thrust are modeled using the geometry of the installation and engine tests done on the ground. Engine thrust forces are described as

$$\boldsymbol{F}_T = T \begin{bmatrix} \Gamma_{r_x} \\ \Gamma_{r_y} \\ \Gamma_{r_z} \end{bmatrix} \tag{2.3.19}$$

where $\Gamma_{r_{(\cdot)}}$ are the thrust-line offset rotation constants and engine thrust is described by $T = f(\Omega, V)$, which is a function of propeller rotational rate (Ω) and airspeed (V). Engine thrust moments are given by

$$\boldsymbol{M}_T = \begin{bmatrix} -F_{T_y}\Gamma_{cg_z} + F_{T_z}\Gamma_{cg_y} \\ F_{T_x}\Gamma_{cg_z} + F_{T_z}\Gamma_{cg_x} \\ -F_{T_x}\Gamma_{cg_y} + F_{T_y}\Gamma_{cg_x} \end{bmatrix} \tag{2.3.20}$$

where $\Gamma_{cg_{(\cdot)}}$ are the thrust line offsets relative to aircraft CG.

The system identification problem then reduces to determining the aerodynamic forces ($\boldsymbol{F}_A$) and moments ($\boldsymbol{M}_A$). These can be expressed in terms of non-dimensional coefficients

$$\boldsymbol{F}_A = \bar{q}S \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix}, \quad \boldsymbol{M}_A = \bar{q}S \begin{bmatrix} bC_l \\ \bar{c}C_m \\ bC_n \end{bmatrix} \tag{2.3.21}$$

where S is the wing reference area, b is the wing span, $\bar{c}$ is the mean aerodynamic chord of the wing and $\bar{q} = (1/2)\rho V^2$ is the dynamic pressure, ρ is the air density and V is the airspeed. The non-dimensionalized force (C_X, C_Y, C_Z) and moment (C_l, C_m, C_n)

coefficients depend nonlinearly on the UAV translational and angular velocity components, angles of incidence of the air-relative velocity with respect to the aircraft body, control surface deflections, possibly their time derivatives and other non-dimensional quantities [35]. This dependence is usually characterized mathematically using parametric system identification methods as in [35, 79]. It is also important to point out that the average moment coefficient is zero.

Summary of the UAV Non-Linear Equations of Motion

The rigid body equations of motion in body-axes reference frame for a symmetric aircraft can be summarised as follows

$$m(\dot{u} + qw - rv) = -mg\sin(\theta) + F_{A_x} + F_{T_x} \tag{2.3.22}$$

$$m(\dot{v} + ru - pw) = mg\sin(\phi)\cos(\theta) + F_{A_y} + F_{T_y} \tag{2.3.23}$$

$$m(\dot{w} + pv - qu) = mg\cos(\phi)\cos(\theta) + F_{A_z} + F_{T_z} \tag{2.3.24}$$

$$I_{xx}\dot{p} - I_{xz}(\dot{r} + pq) - (I_{yy} - I_{zz})qr = M_{A_x} + M_{T_x} \tag{2.3.25}$$

$$I_{yy}\dot{q} - I_{xz}(r^2 - p^2) - (I_{zz} - I_{xx})rp = M_{A_y} + M_{T_y} \tag{2.3.26}$$

$$I_{zz}\dot{r} - I_{xz}(\dot{p} - qr) - (I_{xx} - I_{yy})pq = M_{A_z} + F_{T_z} \tag{2.3.27}$$

$$\dot{\phi} = p + q\sin(\phi)\tan(\theta) + r\cos(\phi)\tan(\theta) \tag{2.3.28}$$

$$\dot{\theta} = q\cos(\phi) - r\sin(\phi) \tag{2.3.29}$$

$$\dot{\psi} = q\sin(\phi)\sec(\theta) + r\cos(\phi)\sec(\theta) \tag{2.3.30}$$

There are nine coupled nonlinear first order differential equations which constitutes of nine variables with six aerodynamic and six propulsive force functions. The force and moment functions includes the pilot or the flight control system control inputs. The first three equations (2.3.22)-(2.3.24) define the velocity components, next three equations (2.3.25)-(2.3.27) define the rotation rate components and the final three equations (2.3.28)-(2.3.30) are to define the attitude rates $\dot{\mathbf{\Psi}}_b = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$ in the body-axes reference frame.

2.4 Aerodynamic Model

The non-dimensional aerodynamic coefficients expressed in (2.3.21) can be divided into longitudinal and lateral modes. Longitudinal modes being forward force (C_X), downward force (C_Z) and pitching moment (C_m). Lateral modes being sideways force (C_Y), roll moment (C_l) and yaw moment (C_n). Longitudinal coefficients are primarily dependent on longitudinal states and elevator inputs. Lateral coefficients are primarily dependent on lateral states and rudder and aileron control inputs. In general they are functionals of the state variables u, v, w, p, q, r or $V, \alpha, \beta, p, q, r$, and the control variables. Based on dimensional analysis the functional dependency can be represented as

$$C_i = C_i\left(\frac{V}{V_0}, \alpha, \beta, \frac{pb}{2V}, \frac{q\bar{c}}{2V}, \frac{rb}{2V}, \frac{\dot{\alpha}\bar{c}}{2V}, \frac{\dot{\beta}b}{2V}, \boldsymbol{\delta}\right) \tag{2.4.1}$$

for $i = X, Y, Z, l, m, n$, angle of attack α , side slip angle β, and $\boldsymbol{\delta}$ represents all the control inputs applied. For a conventional tailed aircraft the control inputs include elevator (δ_e), aileron (δ_a), rudder (δ_r) that changes during a maneuver as well as flap deflections (δ_f) and throttle (δ_T) inputs, which generally are constant throughout a maneuver. Other parameters are non-dimensionalized with respect to true velocity $V \equiv |\boldsymbol{V}| = \sqrt{u^2 + v^2 + w^2}$, wing mean geometric chord $\bar{c}$ and wing span b for dimensional consistency. The nondimensional quantities are

$$\hat{V} \equiv \frac{V}{V_0} \qquad \hat{p} \equiv \frac{pb}{2V} \qquad \hat{q} \equiv \frac{q\bar{c}}{2V} \qquad \hat{r} \equiv \frac{rb}{2V} \qquad \hat{\dot{\alpha}} \equiv \frac{\dot{\alpha}\bar{c}}{2V} \qquad \hat{\dot{\beta}} \equiv \frac{\dot{\beta}b}{2V}$$

The functional dependency shown in equation (2.4.1) is for an airplane in quasi-steady flow at low Mach number. The quasi-steady assumption presumes the flow reaches a steady state instantaneously, hence the dependence on the history of a variable can be neglected. For example, the angle of attack α will only affect $\boldsymbol{F}_A$ and $\boldsymbol{M}_A$ on the current value of $\alpha(t)$ and not its entire history. Given that there are also other functions which affect the aerodynamic force and moment, to model for the entire range of variables the modeling problem becomes more complex.

The functional dependency could further be extended to nondimensional stability and control derivatives. Stability derivatives are partial derivatives with respect to states and control derivatives are partial derivatives with respect to controls. The stability derivatives can further be divided into static stability, dynamic stability and derivatives associated with unsteady aerodynamics ($\dot{w}$, $\dot{\alpha}$). The static stability derivatives are associated with air-relative velocity quantities $(u, v, w, V, \alpha, \beta)$ and dynamic stability derivatives are associated with angular rates (p, q, r). The assumptions [35] made here are:

- The airspeed changes do not affect the aerodynamic coefficients for subsonic flight.
- The functional dependence of longitudinal and lateral coefficients on the state and control inputs are

$$C_a = C_a(\alpha, \beta, q, \boldsymbol{\delta}) \quad \text{for } a = X, Z, \text{or } m \tag{2.4.2}$$

$$C_a = C_a(\alpha, \beta, p, r, \boldsymbol{\delta}) \quad \text{for } a = Y, l, \text{or } n \tag{2.4.3}$$

- The aerodynamic coefficients constitute of static terms as well as nonlinear angle of attack and sideslip angle. They also include dynamic and control terms that are linear in p, q, r and $\boldsymbol{\delta}$. The other terms that could be included in the model are derivatives of the variables that depend nonlinearly on the angle of attack, sideslip and Mach number.

Under the aforementioned assumptions, the aerodynamic coefficients of the model can be written in longitudinal and lateral terms as

$$C_a = C_{a_0}(\alpha, \beta)_{q=\delta=0} + \bar{C}_{a_q} \frac{q\bar{c}}{2V_0} + C_{a_\delta}(\alpha)\delta \quad \text{for } a = X, Z, \text{or } m \tag{2.4.4}$$

$$C_a = C_{a_0}(\alpha, \beta)_{p=r=\delta=0} + C_{a_p}(\alpha)\frac{pb}{2V_0} + C_{a_r}(\alpha)\frac{rb}{2V_0} + C_{a_\delta}(\alpha)\delta \quad \text{for } a = Y, l, \text{or } n \tag{2.4.5}$$

where $\delta \in \boldsymbol{\delta}$. The first term represents the contribution from the initial condition when the control deflections are zero and they are known as the static components of the equations. The rest of the terms represent the dynamic stability derivatives, control derivatives and their dependence with respect to angle of attack. The equations (2.4.4) and (2.4.5) represent only a general representation of the aerodynamic forces and moments. These functional dependencies are obtained based on wind tunnel and other flight test experience [21].

Modeling these coefficients in equations (2.4.4) and (2.4.5) raises the question of what mathematical structure to use. If the model structure is too complicated for a limited amount of data, reduced accuracy for the estimated parameters can be expected. Hence, it is important to hypothesise a model structure that closely represents the dynamics of the system. This is normally done using previous experience or knowledge about the physical system to be modeled. However, the relationship between the model complexity and information in the measured data is not always intuitive. The functional dependency can be complicated, therefore a variety of experiments are performed to determine the characterization.

There exist numerical methods to determine the coefficients. These include panel methods, Computational Fluid Dynamics (CFD) [86]. In panel methods, the aircraft is diced into sections and approximations are made on the two-dimensional aerofoil sections; by summing up each of the components the total lift, drag and moment can be determined for the whole platform. In CFD, partial differential equations (typically the Navier-Stokes equations) are solved to model the motion of air about the platform. There also exist U.S. Air Force Stability and Control DATCOM [21] to determine dimensionless stability derivatives. The DATCOM is a collection of rules and opinions in the area of aerodynamic stability and control methods of fixed wing aircraft. These are based on experience and a complete set of stability and control derivative variables can be obtained by giving the airplane geometry and flight condition. However, such methods underperform in low angle of attack and in low rotational rates [35]. Therefore, experimental methods such as static tests, and steady and maneuvering flight tests have to be conducted.

2.4.1 Parameter Estimation

This consists of finding the values of unknown model parameters $\boldsymbol{\vartheta}$ for a hypothesized model structure based on noisy observation measurements $\boldsymbol{z}$. The estimator is a function of these observations that produces an estimate $\hat{\boldsymbol{\vartheta}}$ of unknown parameters $\boldsymbol{\vartheta}$. To make an estimation several conditions need to be satisfied, they are:

- A model structure with unknown parameters $\boldsymbol{\vartheta}$;
- Measurement of noisy observation measurements $\boldsymbol{z}$;
- Assumptions about the uncertainty in the noise measurements $\boldsymbol{\varepsilon}$ and model parameters $\boldsymbol{\vartheta}$.

For a model that is assumed to be linear the output $\boldsymbol{y}$ is given by

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\vartheta} \tag{2.4.6}$$

where matrix $\mathbf{X}$ is the system inputs which is formed with an initial hypothesis of (2.4.1). It should be noted that, in order for linear approximations to remain valid the system must operate over a restricted range of conditions, such as at a fixed altitude and a constant velocity. An example of whats contained in $\mathbf{X}$ is given below

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{V}_b & \boldsymbol{\omega}_b & \alpha & \beta & \dot{\alpha} & \dot{\beta} & \boldsymbol{\delta} \end{bmatrix} \tag{2.4.7}$$

where the state is composed of the body axis velocities $\boldsymbol{V}_b = [u, v, w]$, rotation rates $\boldsymbol{\omega}_b = [p, q, r]$, angle of attack α, side slip angle β, and control inputs $\boldsymbol{\delta} = [\delta_e, \delta_a, \delta_r]$. The observation equation is expressed as

$$\boldsymbol{z} = \mathbf{X}\boldsymbol{\vartheta} + \boldsymbol{\varepsilon} \tag{2.4.8}$$

where $\boldsymbol{z}$ are observations made and $\boldsymbol{\varepsilon}$ is the measurement noise. These observations can be obtained by re-arranging equations (2.3.16) - (2.3.21). They can be represented as

$$\boldsymbol{z}_F = \frac{1}{\bar{q}S}\left[\left(m\,\dot{\boldsymbol{V}}_b + \boldsymbol{\omega}_b \times m\,\boldsymbol{V}_b\right) - \boldsymbol{F}_G - \boldsymbol{F}_T\right] \tag{2.4.9}$$

$$\boldsymbol{z}_M = \frac{1}{\bar{q}S\zeta}\left[(\boldsymbol{I}\dot{\boldsymbol{\omega}}_b + \boldsymbol{\omega}_b \times \boldsymbol{I}\boldsymbol{\omega}_b) - \boldsymbol{M}_T\right] \tag{2.4.10}$$

for $\boldsymbol{z}_F = [C_X, C_Y, C_Z]$ and $\boldsymbol{z}_M = [C_l, C_m, C_n]$ where ζ is $\bar{c}$ for coefficient C_m and b for coefficients C_l and C_n.

For a model that is nonlinear in the parameters, the observation equation can be expressed as

$$\boldsymbol{z} = X(\boldsymbol{\vartheta}) + \boldsymbol{\varepsilon} \tag{2.4.11}$$

where the function $X(\boldsymbol{\vartheta})$ is assumed to be known. Now the optimal estimate $\hat{\boldsymbol{\vartheta}}$ can be obtained by solving (2.4.8) for problems that are linear in the parameters and by solving (2.4.11) for problems that are nonlinear in the parameters. Given this, the output estimate $\hat{\boldsymbol{y}}$ can be reconstructed based on the estimated model

$$\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\vartheta}} \tag{2.4.12}$$

The terms for the coefficient model in (2.4.8) can be found using various methods such as ordinary least squares [33], maximum likelihood [49], Bayesian estimation [67] and recursive least squares [35]. Here, two models for parameter estimation will be considered. One is a Bayesian estimator and the other is a least-squares estimator. Both these can model the uncertainties in the parameters and they are parametric estimators. The Bayesian model is shown to demonstrate the probabilistic representation of the problem and it provides the ground work for the models developed in the following chapters. In the Bayesian model, $\boldsymbol{\vartheta}$ is a vector of random variables with a probability density of $p(\boldsymbol{\vartheta})$ and system noise $\boldsymbol{\varepsilon}$ is a random vector with a probability distribution $p(\boldsymbol{\varepsilon})$. In the least squares model $\boldsymbol{\vartheta}$ is a vector of unknown constant parameters and $\boldsymbol{\varepsilon}$ is a random vector of measurement noise [5].

Bayesian Estimator

The Bayesian model follows from Bayesian estimation theory [5, 67]. It is assumed the probabilities of the errors are known for all possible values of observations $\boldsymbol{z}$ as well as the probability density $p(\boldsymbol{\vartheta})$. Hence, it is possible to obtain a conditional density function (or likelihood function) $p(\boldsymbol{\vartheta}|\boldsymbol{z})$ given the observation $\boldsymbol{z}$. It is related to the *a priori* probability densities by Bayes rules

$$p(\boldsymbol{\vartheta}|\boldsymbol{z}) = \frac{p(\boldsymbol{z}|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta})}{p(\boldsymbol{z})} \tag{2.4.13}$$

The estimator optimizers for $\hat{\boldsymbol{\vartheta}}$ by maximizing the conditional probability density $p(\boldsymbol{\vartheta}|\boldsymbol{z})$ and this varies with the distribution specified for the vectors $\boldsymbol{\vartheta}$ and $\boldsymbol{z}$. Given the distribution the most probable estimate can be obtained by solving the following equation

$$\hat{\boldsymbol{\vartheta}} = \max_{\boldsymbol{\vartheta}} p(\boldsymbol{\vartheta}|\boldsymbol{z}) \tag{2.4.14}$$

This is usually solved by minimizing a specified cost function $J(\boldsymbol{\vartheta})$. The probability density $p(\boldsymbol{z})$ is not dependent on $\boldsymbol{\vartheta}$ and therefore does not influence the cost function for parameter estimation.

Least-Squares Estimator

In least-squares, the sum of squared error is minimized between the estimated and the measured. This has no probability statements concerning parameters and measurement noise. However, if noise is assumed Gaussian then the least-squares model becomes the Fisher model, which is the maximum likelihood estimator [35].

The general form of the equation to be solved for total least squares in vector and matrix form is

$$\boldsymbol{z} = \mathbf{X}\boldsymbol{\vartheta} + \boldsymbol{\varepsilon} \tag{2.4.15}$$

where,

$$\boldsymbol{z} = \begin{bmatrix} z(1) & z(2) & \dots & z(M) \end{bmatrix}^T = M \times 1 \text{ vector;}$$
$$\boldsymbol{\vartheta} = \begin{bmatrix} \vartheta_0 & \vartheta_1 & \dots & \vartheta_D) \end{bmatrix}^T = D_p \times 1 \text{ vector of unknown parameters, } D_p = D + 1;$$
$$\mathbf{X} = \begin{bmatrix} 1 & \varkappa_1 & \dots & \varkappa_D \end{bmatrix}^T = M \times D_p \text{ matrix of vectors of ones and regressors;}$$
$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon(1) & \varepsilon(2) & \dots & \varepsilon(M) \end{bmatrix}^T = M \times 1 \text{ vector of measurement errors.}$$

The regressor vector contains the postulated model structure and the functions of the independent variables. As mentioned before, for the least squares model there is no associated probability on the model parameters $\boldsymbol{\vartheta}$ or the measurement noise $\boldsymbol{\varepsilon}$. Nevertheless, $\boldsymbol{\varepsilon}$ is assumed to be zero mean with a constant variance. This can be represented as follows

$$E(\boldsymbol{\varepsilon}) = \mathbf{0} \tag{2.4.16}$$
$$E(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T) = \sigma^2 \boldsymbol{I} \tag{2.4.17}$$

Now, the estimator for $\boldsymbol{\vartheta}$ can be determined by minimizing the sum of square error between the measurements and the model

$$J(\boldsymbol{\vartheta}) = \frac{1}{2}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\vartheta})^T(\boldsymbol{z} - \mathbf{X}\boldsymbol{\vartheta}) \tag{2.4.18}$$
$$\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\vartheta}} = -\mathbf{X}^T\boldsymbol{z} + \mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\vartheta}} = 0 \tag{2.4.19}$$

The cost function is given by $J(\boldsymbol{\vartheta})$ in (2.4.18) and by minimizing the error in the cost function by solving the equation (2.4.19) the parameter estimate $\hat{\boldsymbol{\vartheta}}$ can be obtained. The resultant is the ordinary least-squares estimator

$$\hat{\boldsymbol{\vartheta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{z} \tag{2.4.20}$$

Note that $(\mathbf{X}^T\mathbf{X})^{-1}$ is a normal matrix with orthogonal eigenvectors. Problems occur on the matrix $(\mathbf{X}^T\mathbf{X})^{-1}$ when the regressors that make up $\mathbf{X}$ have a high degree of correlation and not linearly independent. In this case it may have some eigenvalues close to zero.

Once the parameters have been estimated the measurement can be reconstructed as

$$\hat{\boldsymbol{z}} = \mathbf{X}\hat{\boldsymbol{\vartheta}} \tag{2.4.21}$$

where $\hat{\boldsymbol{z}}$ is the measurement estimate. The covariance matrix of the parameter estimate $\hat{\boldsymbol{\vartheta}}$ or estimation error $\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta}$ is

$$\begin{aligned} \text{cov}(\hat{\boldsymbol{\vartheta}}) &= E[(\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta})(\hat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta})^T] & (2.4.22) \\ &= E\left[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\boldsymbol{z} - \hat{\boldsymbol{y}})(\boldsymbol{z} - \hat{\boldsymbol{y}})^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\right] & (2.4.23) \end{aligned}$$

From the above matrix the Cramér-Rao bounds [46] can be determined which is the square root of the diagonal elements. Cramér-Rao bounds are particularly significant as they are a key indicator of parameter identifiability. It is a measure of the best possible precision attainable for $\boldsymbol{\vartheta}$ from the information available in the measured data $\boldsymbol{z}$. This is important since, if there is no activity in the measured response, there would not be any information in $\mathbf{X}$ that can be used to identify $\boldsymbol{\vartheta}$. Hence, the larger the response or the larger the signal to noise ratio the more information there is to identify the parameters. This can be determined by knowing the amount of information collected in the measured data. An alternative approach is to take the maximum likelihood. First the sensitivity of $z(i)$ to $\boldsymbol{\vartheta}$ needs to be determined. The sensitivity matrix $S(i)$ is defined as

$$S(i) = \left.\frac{\partial z(i)}{\partial \boldsymbol{\vartheta}}\right|_{\boldsymbol{\vartheta}=\hat{\boldsymbol{\vartheta}}} \tag{2.4.24}$$

where $i = 1, 2, \ldots, M$. The noise covariance matrix $\boldsymbol{R}$ is

$$\boldsymbol{R} = \frac{1}{M}\sum_{i=1}^{M}(z(i) - \hat{y}(i))^T(z(i) - \hat{y}(i)) \tag{2.4.25}$$

Combining equations (2.4.24) and (2.4.25) gives the Fisher information matrix $\boldsymbol{M}$.

This is a direct measure of the information contained in the data and is given by

$$\boldsymbol{M} = \sum_{i=1}^{M} S(i)^T \boldsymbol{R}^{-1} S(i) \tag{2.4.26}$$

The inverse of (2.4.26) is the dispersion matrix $\boldsymbol{D}$, consistent with the covariance matrix of $\boldsymbol{\vartheta}$, from which the Cramér-Rao lower bounds can be found. The dispersion matrix is

$$\boldsymbol{D} = \boldsymbol{M}^{-1} = \left[\sum_{i=1}^{M} S(i)^T \boldsymbol{R}^{-1} S(i) \right]^{-1} \tag{2.4.27}$$

This is a measure of the precision of the estimates and the diagonal terms of $\boldsymbol{D}$: d_{jj} are the Cramér-Rao lower bounds for $\boldsymbol{\vartheta}_j$ where $\boldsymbol{j} = 1, \ldots, D$ and D is the total number of regressors

$$\boldsymbol{\varphi}_j = \sqrt{d_{jj}} \tag{2.4.28}$$

It is a measure of the best possible precision attainable for $\boldsymbol{\vartheta}_j$ from the available information in the measurement data $\boldsymbol{z}$. If $\boldsymbol{\vartheta}_j$ is highly observable then $\boldsymbol{\varphi}_j$ will be small and $\boldsymbol{\vartheta}_j$ is poorly observable if $\boldsymbol{\varphi}_j$ is to be large.

One of the problems of least squares regression is that it lacks an associated measure of the quality of the prediction. That is, scalar predictions are provided at any test point without any measure of confidence or a full predictive distribution. Another is the problem of model structure determination which will be discussed in the following sub-section.

2.4.2 Model Structure Determination

For parametric models an assumption usually made is that the model structure of $\mathbf{X}$ is known. This model structure refers to the number and form of the model terms in the regression equation. If the structure is known then parameter estimation has to determine only the constant model parameter values with their standard error given the measured data.This can be done using the aforementioned estimation techniques. However, this is not the case in practice.

To develop a model it is important to reflect any available *a priori* knowledge such as platform performance results from wind-tunnel experiments and results from theoretical aerodynamic calculations. Likewise an'expert' analyst can specify candidate regressors that might affect the model. This procedure is called model structure determination. This ensures that parameter identification is not reduced by over modeling.

In aircraft system identification, for tradition symmetric fixed wing platform the estimation procedure is fairly straight forward and the parameters could be determined from small perturbation flight test maneuvers where the aerodynamic coefficients are linear at the given altitude and Mach number. This behaviour changes if the aircraft is not symmetric (such as an Oblique wing) or a delta wing configuration. In addition, it changes for rapid and large amplitude maneuvers as well as near the aircraft's flight envelope, which is the boundary of the design in terms of conditions such as airspeed, angle of attack, load factor or altitude within which the aircraft is aerodynamically stable. This would require modeling into flight regimes where there is nonlinear aerodynamic effects present. Thus it introduces a new problem of determining how complex the model should be.

In either case it might require the testing of different combinations of the model structure. In doing so it is a balance between two conflicting objectives. On one hand, it maybe required to have many regressors as possible to model nearly all the possible variations in the measured data. On the other hand, it is better to model few regressors as possible because the variance of the prediction $\hat{\boldsymbol{z}}$ increases with the number of regressors. Hence, this process might not always yield an optimal solution. If the model is over parameterized the model may have poor generalized performance, hence would make poor predictions at the test points. However, if the model is too simple, again the predictive performance drops. Hence, it is a trade off to obtain a model that can interpolate observations sufficiently while maintaining good generalized performance. In this work an alternative model is proposed that is based on a non-parametric technique which alleviates the problem of knowing the model structure and counters the problem of overfitting.

2.5 Non-Parametric Bayesian Regression

An alternative to the cost function based regression approach is to assume a noise model on the outputs. Such a model can be described by the following generative equation

$$y_i = f(\mathbf{x}_i, \mathbf{w}) + \varepsilon_i \tag{2.5.1}$$

where $f(\mathbf{x}_i, \mathbf{w})$ is the underlying function defined in terms of some inputs $\mathbf{x}_i$, parameters $\mathbf{w}$, and ε_i is an independently and identically distributed (i.i.d.) noise. This noise is generally assumed to be a Gaussian distribution with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Likelihood functions can then be used to develop a model. These are a class of machine learning algorithm referred to as Bayesian methods. A likelihood function can be used to reflect the information about the parameters contained in the data and the prior distribution which quantifies the information known about the parameters before observing the data. The likelihood function and the prior distribution can be combined to obtain a posterior predictive distribution over the test inputs. These distributions provide a useful method to quantify uncertainty for the predicted estimates. To illustrate this consider a member $\mathcal{B}_i$ with a prior probability $p(\mathcal{B}_i)$. On observing some data $\mathcal{D}$, the likelihood of hypothesis $\mathcal{B}_i$ is $p(\mathcal{D}|\mathcal{B}_i)$. Therefore, the posterior probability of $\mathcal{B}_i$ is given by

$$\text{posterior} \;\propto\; \text{prior} \times \;\text{likelihood} \tag{2.5.2}$$

$$p(\mathcal{B}_i|\mathcal{D}) \;\propto\; p(\mathcal{B}_i)p(\mathcal{D}|\mathcal{B}_i) \tag{2.5.3}$$

The proportionality can be turned into an equality by dividing the above equation with $p(\mathcal{D}) = \sum_i p(\mathcal{D}|\mathcal{B}_i)p(\mathcal{B}_i)$. This can also be interpreted as an integration where appropriate. It consists of the likelihood function multiplied by the prior density and is known as the marginal likelihood.

To obtain a prediction at a test point, rather than using only a set of parameters to make the prediction, Bayesian regression integrates over the entire posterior distribution. Therefore all the test points contribute to a prediction. As a consequence the model is powerful enough to capture the underlying function without having to tackle the parametrization issues. Furthermore, it gives the full predictive distribution at a test point which is useful as a measure of the model's confidence in its prediction.

In this thesis the focus is to tackle a regression problem hence only the Bayesian approach to the regression is discussed. In regression problems the goal is learn the mapping from some input space $\mathcal{X} \in \mathbb{R}^D$ of D-dimensional vectors to an output space $\mathcal{Y} \in \mathbb{R}$. The target observation are assumed to have arisen from some unknown function of the inputs, and have also been observed with some unknown noise. In particular the kernel-based Bayesian regression algorithm known as Gaussian Processes (GPs) is explored [45, 66, 88].

2.5.1 Multivariate Gaussian Distribution

It is a generalization of the one dimensional normal (or Gaussian) distribution to higher dimensions. The multivariate Gaussian distribution of a M-dimensional random variable $\boldsymbol{x} \in \mathbb{R}^M$ can be written in the following notation

$$p(\boldsymbol{x}; \boldsymbol{m}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m})\right) \tag{2.5.4}$$

with mean $\boldsymbol{m} \in \mathbb{R}^M$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{S}^M_{++}$. Also, note that $\mathbb{S}^M_{++}$ refers to the space of symmetric positive definite $M \times M$ matrices. The equation (2.5.4) can also be written as $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$.

2.5.2 Gaussian Processes

Gaussian processes are a probabilistic framework approach that presents a generalization of probability distribution to infinite-dimensional distributions. It defines a distribution over functions. In practice one will only ever need to work with finite dimensional objects. This can be achieved by the marginalization property of probability distributions. Hence, a Gaussian process can be a collection of random variables, any finite number of which have a joint Gaussian distribution [66]. It is a distribution over functions f, where f is a function mapping some input space $\mathcal{X}$ to an output space $\mathcal{Y}$.

$$f : \mathcal{X} \to \mathcal{Y} \tag{2.5.5}$$

Let this distribution be $p(\mathbf{f})$ and $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_M))$ be an M-dimensional vector of function values evaluated at M points where $\mathbf{x} = [x_1, \ldots, x_D]$ is the input vector of dimension D. Likewise, note that vector $\mathbf{f}$ is a random variable. Given this, $p(\mathbf{f})$ is a Gaussian process if for any finite subset $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\} \subset \mathcal{X}$, the marginal distribution over that finite subset $p(\mathbf{f})$ has a multivariate Gaussian distribution. Such a stochastic process is known as a Gaussian process.

In this framework, there is a training set $\mathcal{D}$ of M observations, $\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, M\}$ where y refers a scalar output. The column vector of all input cases M can be collected in the $D \times M$ *design matrix* $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$. The observation values are collected in the vector $\boldsymbol{Y} = [y_1, \ldots, y_M]$. Hence, the training set can be written as $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{Y})$. The output values of this setting are assumed to be real values. In this regression problem, GPs are used to learn the relationship between the inputs and output observations. Since, a Gaussian process is defined as a collection of random variables it implies a marginalization property [66]. Therefore, inferences can then be made using the training data set by conditioning the distribution of the observations given the inputs.

GPs are parametrized by a mean function, $m(\boldsymbol{X})$ and a covariance function $\mathbf{K}$ where $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ and $\boldsymbol{X}_* = [\mathbf{x}_{*1}, \ldots, \mathbf{x}_{*M_*}]$ are the training and test sets respectively. In here, $\mathbf{x}_* = [x_{*1}, \ldots, x_{*D}]$ is the D dimensional test input vector and M_* is the number test cases.

$$p(f(\boldsymbol{X}), f(\boldsymbol{X}_*)) = \mathcal{N}(\mathbf{m}, \mathbf{K}) \tag{2.5.6}$$

where,

$$\mathbf{m} = \begin{bmatrix} m(\boldsymbol{X}) \\ m(\boldsymbol{X}_*) \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} K(\boldsymbol{X}, \boldsymbol{X}) & K(\boldsymbol{X}, \boldsymbol{X}_*) \\ K(\mathbf{X}_*, \boldsymbol{X}) & K(\boldsymbol{X}_*, \boldsymbol{X}_*) \end{bmatrix}$$

and $K(\cdot, \cdot)$ is the covariance between two arbitrary functions of input pairs. Each observation y_i of a Gaussian process is dependent on an underlying function $f(\mathbf{x}_i)$ and Gaussian i.i.d. noise $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$

$$y_i = f(\mathbf{x}_i) + \varepsilon_i \tag{2.5.7}$$

In accordance with Gaussian process regression, a zero-mean Gaussian prior distribution is placed over $\mathbf{f}$. This can be represented as

$$\mathbf{f} | \boldsymbol{X}, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}) \tag{2.5.8}$$

with a $\mathbf{0}$ mean vector and $\mathbf{K}$ is an $M \times M$ covariance matrix. Elements of $\mathbf{K}$ are equal to $K(\boldsymbol{X}, \boldsymbol{X}_*)$ and $K(\cdot, \cdot)$ is a positive definite covariance matrix parametrised by hyperparameters $\boldsymbol{\theta}$.

Gaussian Process Predictions

Given the observations and the covariance function, predictions can be made using the GP model. Consider a case where the observations are noise free, the input set is given by $\{(\mathbf{x}_i, f_i) | i = 1, \ldots, M\}$, test set is given by $\boldsymbol{X}_*$ with M_* test points and the associated latent variables are denoted by $\mathbf{f}_*$. Under the GP framework the marginal distribution over any set of input points belonging to $\mathcal{X}$ have a joint multivariate Gaussian distribution. Therefore, the joint distribution of $\mathbf{f}$ and $\mathbf{f}_*$ is also a zero mean multivariate Gaussian. It is found by augmenting equation (2.5.8) with the new latent variables $\mathbf{f}_*$. The resultant is

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \Bigg| \boldsymbol{X}, \boldsymbol{\theta} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\boldsymbol{X}, \boldsymbol{X}) & K(\boldsymbol{X}, \boldsymbol{X}_*) \\ K(\boldsymbol{X}_*, \boldsymbol{X}) & K(\boldsymbol{X}_*, \boldsymbol{X}_*) \end{bmatrix}\right) \tag{2.5.9}$$

In practical application, it is realistic to observe the underlying function values themselves. It is only the noisy versions of the function values accessible. The observed output can then be represented as, $\boldsymbol{Y} = f(\boldsymbol{X}) + \varepsilon$ where ε is some Gaussian noise with variance σ^2. Given the Gaussian noise assumption on the observed function, the joint distribution over the targets $\boldsymbol{Y}$ and the test input predictions $\mathbf{f}_*$ can be written as

$$\begin{bmatrix} \boldsymbol{Y} \\ \mathbf{f}_* \end{bmatrix} \Bigg| \boldsymbol{X}, \boldsymbol{\theta} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\mathbf{I} & K(\boldsymbol{X},\boldsymbol{X}_*) \\ K(\boldsymbol{X}_*,\boldsymbol{X}) & K(\boldsymbol{X}_*,\boldsymbol{X}_*) \end{bmatrix}\right) \tag{2.5.10}$$

By conditioning the above joint Gaussian prior distribution on the observations $\boldsymbol{Y}$ (see A.2 in [66]) gives

$$\mathbf{f}_* | \, \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2 \sim \mathcal{N}(m(\boldsymbol{X}_*) v(\boldsymbol{X}_*)) \tag{2.5.11}$$

where the mean $m(\boldsymbol{X}_*)$ and variance $v(\boldsymbol{X}_*)$ of the predictions are

$$m(\boldsymbol{X}_*) = K(\boldsymbol{X}_*,\boldsymbol{X})[K(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\mathbf{I}]^{-1}\boldsymbol{Y} \tag{2.5.12}$$

$$v(\boldsymbol{X}_*) = K(\boldsymbol{X}_*,\boldsymbol{X}_*) + \sigma^2 - K(\boldsymbol{X}_*,\boldsymbol{X})[K(\boldsymbol{X},\boldsymbol{X}) + \sigma^2\mathbf{I}]^{-1}K(\boldsymbol{X},\boldsymbol{X}_*) \tag{2.5.13}$$

These are the key predictive equations for Gaussian process regression. They can be used to calculate the predictive distribution for any set of test points $\{\boldsymbol{X}_*\}$. It can also be noticed that the mean prediction in equation (2.5.12) is a linear combination of the observations $\boldsymbol{Y}$. Therefore, it can be interpreted as a linear predictor.

It is also important to introduce the marginal likelihood $p(\boldsymbol{Y}|\boldsymbol{X})$. This is the integral of the likelihood times the prior which can be represented as

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \int p(\boldsymbol{Y}|\mathbf{f}, \boldsymbol{X}) p(\mathbf{f}|\boldsymbol{X}) d\mathbf{f} \tag{2.5.14}$$

In this equation, the likelihood function is marginalized over the function values $\mathbf{f}$. The log Gaussian prior $\mathbf{f}|\boldsymbol{X} \sim \mathcal{N}(\mathbf{0}, K(\boldsymbol{X},\boldsymbol{X}))$ is given by

$$\log p(\mathbf{f}|\boldsymbol{X}) = -\frac{1}{2}\mathbf{f}^T K(\boldsymbol{X},\boldsymbol{X})^{-1}\mathbf{f} - \frac{1}{2}\log|K(\boldsymbol{X},\boldsymbol{X})| - \frac{M}{2}\log 2\pi \tag{2.5.15}$$

and the likelihood is $\boldsymbol{Y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I})$. The marginal likelihood can then be obtained by integrating equation (2.5.14)

$$\log(p(\boldsymbol{Y}|\boldsymbol{X})) = -\frac{1}{2}\boldsymbol{Y}^T(K(\boldsymbol{X},\boldsymbol{X})+\sigma^2\mathbf{I})^{-1}\boldsymbol{Y} - \frac{1}{2}\log|K(\boldsymbol{X},\boldsymbol{X})+\sigma^2\mathbf{I}| - \frac{M}{2}\log 2\pi \tag{2.5.16}$$

The three terms in the marginal likelihood equation (2.5.16) have their individual roles. The first term $\frac{1}{2}\boldsymbol{Y}^T(K(\boldsymbol{X},\boldsymbol{X})+\sigma^2\mathbf{I})^{-1}\boldsymbol{Y}$ relates to data fit of the observations. The second term $\frac{1}{2}\log|K(\boldsymbol{X},\boldsymbol{X})+\sigma^2\mathbf{I}|$ is the complexity penalty depending only on the covariance function and the inputs. The final term $\frac{M}{2}\log 2\pi$ is a normalization constant.

Similar to Bayesian methods, GP models quantify uncertainty in the predictions. This uncertainty is low when the predictive distribution is tightly packed around a single value of the training data. On the other hand, if the predictive distribution is spread widely across the training data then the model has a high uncertainty or it is less confident about what to expect given the test input. This can been observed in the example given in Figure 2.4. Here a GP prediction of an underlying function is given with the associated uncertainty highlighted in gray. Furthermore, a GP model is non-parametric, hence it does not have to deal with determining structure of the model as for the parametric methods. It also provides a natural framework for introducing kernels into a regression framework. This gives the flexibility to design kernel functions to model the underlying properties of the data. This is further explored in the following sub-section.

Covariance Functions

The covariance function specifies the covariance between pairs of random variables. A covariance function must be a positive semi-definite function. There is a large range of covariance functions available to model different properties of the underlying data. These properties include smoothness, periodicity and stationarity of the function predicted by the GP. From a supervised learning perspective the notion of similarity between training data and the test input points is essential. Hence, the assumption is that if the test point is close to $\boldsymbol{X}$ then it is likely to have similar target values $\boldsymbol{Y}$.

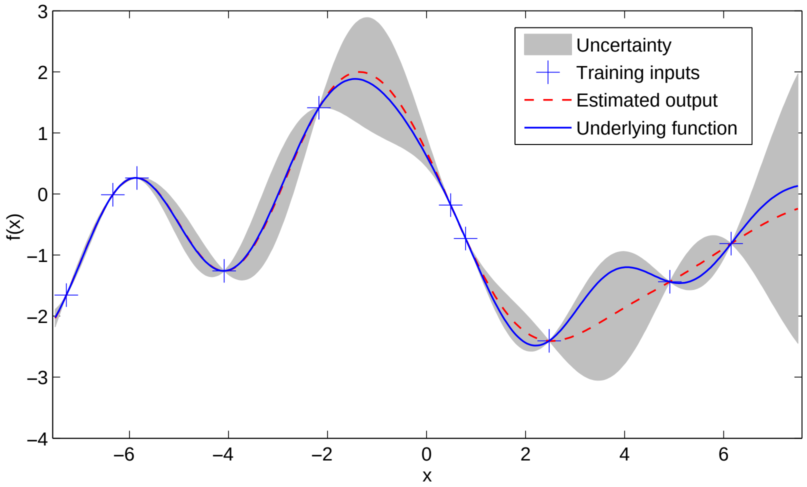


Figure 2.4: Gaussian process prediction of the underlying function $f(\boldsymbol{X})$ (solid blue) given the training inputs (blue crosses). The estimated prediction is given by the dashed red line and the shaded region represent 1σ uncertainty for the model prediction.

Therefore training points near a test point are more informative to make predictions.

There are two main classes of covariance functions: stationary and non-stationary. A stationary covariance function is a function of $\mathbf{x} - \mathbf{x}_*$ and is invariant to translations in the input space. Such a class is known as Radial Basis Functions (RBFs) when the real function value only depends on the distance from the origin [66]. In the case of a Euclidean space and the distance is isotropic, covariance $K(\cdot,\cdot)$ is now only a function of $|\mathbf{x} - \mathbf{x}_*|$. This can be shown for two arbitrary functions of input pairs $\mathbf{x}$ and $\mathbf{x}_*$ by

$$\operatorname{cov}(f(\mathbf{x}), f(\mathbf{x}_*)) = K(\mathbf{x}, \mathbf{x}_*). \tag{2.5.17}$$

The covariance for a Gaussian process also includes the observation noise term, hence the above equation can be modified to include the noise term and can be represented as

$$\operatorname{cov}(\mathbf{x}, \mathbf{x}) = K(\mathbf{x}, \mathbf{x}) + \sigma^2\mathbf{I} \tag{2.5.18}$$

where $\sigma^2\mathbf{I}$ is the associated noise added to the covariance. The most common choice

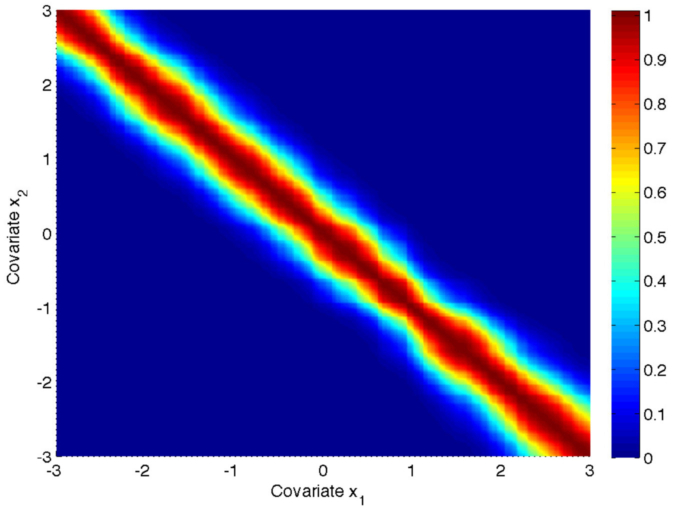


Figure 2.5: Covariance matrix for squared-exponential covariance function. The warm colours (red) represent high covariance and cool colours (blue) represent low covariance.

for a stationary isotropic covariance function is the squared-exponential

$$K(\mathbf{x}_a, \mathbf{x}_b|\boldsymbol{\Theta}) = \sigma^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}_a - \mathbf{x}_b)^2\right) \tag{2.5.19}$$

where $\boldsymbol{\Theta}$ are the hyperparameters σ and l that conditioned the covariance matrix. σ is known as the signal variance and l is the length-scale which encodes the smoothness properties of the underlying function. A short length-scale means the error grows faster when away from the training data. On the other hand, a longer length-scale means the function is slowly varying with a higher degree of noise. The optimal values for the hyperparameters can be found by maximizing the log marginal likelihood in equation (2.5.16). Note that this covariance can only model a single output dimension. A visual representation of the squared-exponential covariance function is given in Figure 2.5. The warmer the colours (red) represent high covariance, hence along the diagonal the elements are highly correlated and it reduces when the covariates are further apart.

As opposed to RBFs non-stationary covariance functions can model $\mathbf{x} - \mathbf{x}_*$ when it is variant to translation in the input space. Further details on this type of a covariance function will be presented in Section 4.4.2. For more details on Gaussian process theory and for details about various kernel functions please refer to [66].

2.6 Summary

This chapter presented the background and the underlying theory behind aircraft system identification. The complex framework for system identification is explained and the mathematical formulation of the problem is given. It showed how the system identification can be reduced down to an aerodynamic parameter estimation problem. The current issues related to parameter identification was described from which the motivations for this thesis were derived. The fundamentals in non-parametric Bayesian regression was presented. This provides the theoretical background to view the problem in a probabilistic sense. Furthermore, it gives a context of the ideas and techniques presented in the subsequent chapters. Finally, the basic theory behind Gaussian process regression was presented.

Chapter 3

Parameter Identification using Gaussian Processes

3.1 Introduction

Unmanned aerial vehicle parameter estimation involves performing flight test maneuvers and then extracting the estimate values from the logged flight data. These can also be obtained by various existing computational methods and wind-tunnel testing [35]. However, there are several reasons why it is important to determine aerodynamic parameters from real test flights. Firstly they give a better understanding of theoretical predictions and wind-tunnel testing, secondly the requirement of a high integrity model for modern flight control systems, lastly for simulators to give a more accurate representation of the system throughout the whole flight envelope. After collecting the flight data there are numerous techniques for parameter estimation. As mentioned in the previous chapter, there are two main system identification approaches. One using parametric and the other using non-parametric approaches. To use non-parametric approaches an observation model is required in order to regress. The goodness of this structure is dependent on how much time and effort is invested in the regression process (see Section 2.4.2). A limited model structure will limit its applicability to a narrow flight envelop. This could be avoided by using a non-parametric technique. Nevertheless, the existing methods can only estimate limited number of states in the system [10] or can only operate within the trained flight trajectory [12]. A recent machine learning technique Gaussian processes have been applied to the problem of non-linear system identification in [23, 38], in [24] they were applied to learn imitative whole body motions in a humanoid robot and in [36] they were used to learn the residual error between the non-linear model and the ground truth of an autonomous blimp. However, these models were limited to identifying a single output system. Therefore they do not capture dependencies between the identified parameters. This limits its applicability as most highly dynamic systems can entail coupled dynamics within its system outputs (see Section 2.4).

These limitations could be resolved by learning multiple outputs while capturing dependencies. In addition, linear approximations such as the one that is presented in Section 2.4.1 remains valid only if the system operates over a restricted range of conditions, such as at a fixed altitude and a constant velocity. These restrictions can be relaxed with the proposed non-linear flight dynamic model. What is explored here is a supervised learning algorithm. It is a form of inductive learning method that learns a functional mapping from a set of training data of inputs and outputs. This summary can then be used for predicting a new sample test point when required. The chapter examines the use of Gaussian processes for supervised learning of aircraft flight dynamics. In particular we show how to identify system parameters using Dependent Gaussian Processes (DGPs) [7, 8] to learn a multi-output flight model. The non-parametric nature of the approach enables it to capture a wide range of dynamics. Therefore, *a prior* knowledge of the model structure is not required. The ability to capture dependencies through cross coupling terms enable the models to be applicable to a broader flight envelope than existing methods saving significant costs on flight testing. In addition, it has the inherent capability to handle noise and biased data. The predictions from the model also come with uncertainty estimates which can be used in maneuver design for system identification and for flight controller design. Finally, it is not entirely a black-box prediction model as opposed to techniques such as neural networks.

This chapter demonstrates the dependencies for which the method was explicitly chosen, robustness to unmodeled disturbances and quantifies improvements over an alternative least squares estimator. In addition, an in-depth analysis to identify the parameters for the highly coupled AD-1 oblique wing aircraft in simulation is presented.

3.2 Related Work

Gaussian processes has previously been addressed in the geostatistical literature under the name kriging [48]. This has primarily been used for interpolation of geostastical data. Kriging in its early stages was mainly concerned with low-dimensional problems and did not consider a probabilistic formulation. Gaussian processes then emerged in the statistical community to define prior over functions and have been applied in [59] for one dimensional curve fitting problems.

In the machine learning community, the interest in Gaussian processes arose through Neural Networks (NN). In practice NN had some inherent problems, these include the decisions that need to be made as to what activation functions, learning rate and what architecture to use. There also was not a principle framework to address these issues [66]. By utilizing a probabilistic framework in a Gaussian process these issues were tackled while addressing the problem of overfitting. This was shown in the work by Neal in [55] where under certain conditions Bayesian neural networks converge to Gaussian processes. The work was then extended to use in the context of machine learning in [64, 89].

An interesting development of this research was the ability to handle multiple outputs. This is particularly useful for modeling dynamic systems where the system has multiple outputs that occur simultaneously. However, due to the difficulty of maintaining positive definiteness, the parameterization of the covariance function makes it difficult to deal with multiple outputs. This problem was also first tackled in the field of geostatistics under the name co-kriging (see section 3.2.3 of [16] and [22]).

The method generally involves using an independent model for each output. Nevertheless, by treating each output separately some information is lost and the learning process is suboptimal. In [75] a semiparametric model for nonlinear regression involving multiple outputs is proposed. Here GPs are used as linear combinations in a number of latent channels to capture dependencies that may exist in the outputs. The different latent channels are described by different covariance functions and one has to estimate the hyperparameters for all the covariance functions. A multi-task GP was introduced in [6] where a model learns a shared covariance function over a set of input features.

An alternative approach, Dependent Gaussian Processes (DGPs) was proposed in [7, 8] where Gaussian processes are treated as white noise sources convolved with smoothing kernels. This allows the Gaussian process to handle multiple and coupled outputs. The sole dependency between different tasks come from sharing parameters of the underlying covariance function. The advantage of this approach over [6] is the ability to handle a different number of features for each output within a single covariance function. This helps to construct a GP model even when a particular output is only partially observable.

3.3 System Modeling with Dependent Gaussian Processes

The objective of dependent Gaussian processes is to infer multiple outputs jointly while capturing any dependencies. This extends the basic Gaussian process model (see Section 2.5.2) to account for multiple outputs from a system. It involves defining a valid positive-semi definite kernel to capture the relationship between system outputs. The sole dependence between different tasks comes from sharing the parameters of the underlying covariance function. An overview of the DGP theory is provided in this section and for more details refer to Chapter 3 in [8].

In dependent Gaussian processes [7, 8], the GP model can capture dependency between outputs $\mathbf{y} \in \mathbb{R}^N$ of N dimension. Consider a collection of variables $\boldsymbol{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M]$ with $\mathbf{y} = [y_1, \ldots, y_N]$ where $\mathbf{y} \in \mathbb{R}^N$ which have a joint distribution $p(\boldsymbol{Y}|\mathbf{C}, \boldsymbol{X})$. The *design matrix* $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ consists of $\mathbf{x} = [x_1, \ldots, x_D]$ where $\mathbf{x} \in \mathbb{R}^D$ is a D-dimensional input vector and M is the total number of input processes. $\mathbf{C} = C(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\Theta})$ is a parameterized covariance function with hyperparameters $\boldsymbol{\Theta}$. The output of the GP model is a normal distribution, expressed with a mean and variance. The mean value represents the most likely outcome and variance can be interpreted as the confidence level of the outcome. The key advantages of this approach is its ability to provide uncertainty estimates, model flexibility, and to learn noise and smoothness estimates from test data [66].

3.3.1 Assumptions

In order to satisfy the conditions for a valid kernel function certain assumptions are made. First, the system inputs are independent and stationary processes. This means the joint probability distribution does not change within the input space. Thus, the covariance function $C(\mathbf{x}_i, \mathbf{x}_j)$ between inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ is only dependent on their distance. These are valid for the chosen inputs (see equation (3.4.2)) given that the aircraft is flying under its stall angle of attack in a narrow region of the flight envelop when the conditions are steady for a wings-level flight with no sideslip [35]. If there is a need to model outside of this bound then a non-stationary multi-output covariance function such as the one given by equation (9) in [84] is required. Next, the prior underlying process has zero mean. If constructing a model for a larger region where the prior underlying process changes throughout the flight envelop it is possible to use an explicit basis function (see Section 2.7 in [66]). Finally, the observed data is noisy and it was assumed this distribution is Gaussian white noise. Hence, Gaussian white noise was added to each output.

3.3.2 Modeling with Dependent Gaussian Processes

The objective of dependent Gaussian processes is to infer multiple outputs jointly while capturing any dependencies. This is useful for systems where its outputs are derived from a common set of input sources. By learning them in parallel the performance of the GP model can be improved in comparison to learning them independently. Consider N-output processes $\mathbf{y} = [y_1, \dots, y_N]$ where $\mathbf{y} \in \mathbb{R}^N$ and $\boldsymbol{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_M]$ where M is the total number of observations and $\boldsymbol{X} = [\mathbf{x}_1, \dots, \mathbf{x}_M]$ consists of $\mathbf{x} = [x_1, \dots, x_D]$ where $\mathbf{x} \in \mathbb{R}^D$ is a D-dimensional input vector. The training set is assumed to be drawn from the noisy process represented by

$$y_n(\mathbf{x}) = u_n(\mathbf{x}) + \varepsilon_n(\mathbf{x}) \tag{3.3.1}$$

where $\varepsilon_n(\mathbf{x})$ is stationary Gaussian white noise drawn from $\mathcal{N}(0, \sigma_n^2)$ with variance σ_n^2. $u_n(\mathbf{x})$ is defined by

$$\begin{aligned} u_n(\mathbf{x}) &= \sum_{m=1}^{M} h_{mn}(\mathbf{x}) * x_m(\mathbf{x}) && (3.3.2) \\ &= \sum_{m=1}^{M} \int_{\mathbb{R}^D} h_{mn}(\boldsymbol{\alpha}) x_m(\mathbf{x} - \boldsymbol{\alpha}) d^D \boldsymbol{\alpha} && (3.3.3) \end{aligned}$$

The above equation calculates the sum of convolutions of the h_{mn} kernel connecting the input m to output n. The marginal likelihood of $\boldsymbol{y}$ given inputs $\boldsymbol{X}$ is of the form

$$p(\boldsymbol{y}|\boldsymbol{X}) = \mathcal{N}(0, K(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\sigma}^2\mathbf{I}) \tag{3.3.4}$$

where $\mathbf{K}$ is the kernel matrix, which defines the connection from input m to output n and the noise variance $\boldsymbol{\sigma} \in \mathbb{R}^N$. $\mathbf{K}$ can be described as a function $\mathrm{cov}_{ij}^{y}(\mathbf{x}_a, \mathbf{x}_b)$ which defines the auto $(i = j)$ and cross covariance $(i \neq j)$ between $y_i(\mathbf{x}_a)$ and $y_j(\mathbf{x}_b)$.

$$\mathrm{cov}_{ij}^{y}(\mathbf{x}_a, \mathbf{x}_b) = \mathrm{cov}_{ij}^{u}(\mathbf{x}_a, \mathbf{x}_b) + \sigma_i^2 \tag{3.3.5}$$

Given the assumption that the inputs are stationary, the kernels are likewise stationary. Thus, a separation vector can be defined as $\mathbf{d} = \mathbf{x}_a - \mathbf{x}_b$. Next, $\text{cov}_{ij}^u(\mathbf{x}_a, \mathbf{x}_b)$ is maximized for $E\{u_i(\mathbf{x}_a)u_j(\mathbf{x}_b)\}$. Solving this integral results in

$$\text{cov}_{ij}^y(\mathbf{d}) = \sum_{m=1}^{M} \int_{\mathbb{R}^D} h_{mj}(\boldsymbol{\beta}) h_{mi}(\boldsymbol{\beta} + \mathbf{d}) d^D \boldsymbol{\beta} \tag{3.3.6}$$

The aforementioned equation 3.3.6 defines the auto and cross covariance for the outputs $y_n(\mathbf{x})$. The kernels here were set to be parameterized Gaussians and the integral was solved. Let the kernel h_{mn} be

$$h_{mn}(\mathbf{x}) = \upsilon_{mn} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{mn})^T \mathbf{A}_{mn}(\mathbf{x} - \boldsymbol{\mu}_{mn})\right) \tag{3.3.7}$$

where the hyperparameters are defined by $\upsilon_{mn} \in \mathbb{R}$, $\mathbf{A}_{mn}$ and $\boldsymbol{\mu}_{mn} \in \mathbb{R}^D$. Substituting kernel function (3.3.7) into (3.3.6) and solving the integral results in

$$\text{cov}_{ij}^y(\mathbf{d}) = \sum_{m=1}^{M} \frac{(2\pi)^{\frac{D}{2}} \upsilon_{mi}\upsilon_{mj}}{\sqrt{|\mathbf{A}_{mj} + \mathbf{A}_{mi}|}} \exp\left[-\tfrac{1}{2}(\mathbf{d} - [\boldsymbol{\mu}_{mi} - \boldsymbol{\mu}_{mj}])^T \boldsymbol{\Sigma}(\mathbf{d} - [\boldsymbol{\mu}_{mi} - \boldsymbol{\mu}_{mj}])\right] \tag{3.3.8}$$

where,

$$\boldsymbol{\Sigma} = \mathbf{A}_{mi}(\mathbf{A}_{mi} + \mathbf{A}_{mj})^{-1}\mathbf{A}_{mj}.$$

For a detailed derivation of this covariance function see Appendix A in [7]. The equation (3.3.8) defines the positive definite covariance matrix $\mathbf{K}$ for the combined output processes N. This can be written as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \cdots & \mathbf{K}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{N1} & \cdots & \mathbf{K}_{NN} \end{bmatrix} \tag{3.3.9}$$

A visual representation of this covariance function (3.3.9) is shown in Figure 3.1. In this example there are two dependent output processes with equal number of sample points. The warm colours (red) represent high covariance and the towards the cool colours (blue) represent low covariance. Along the diagonal elements of the matrix are

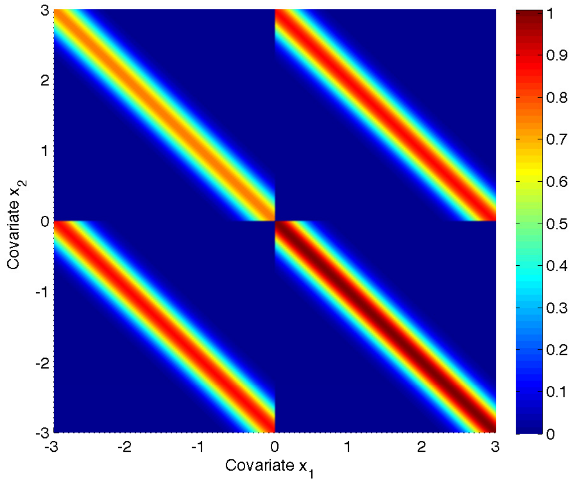


Figure 3.1: Covariance matrix for DGP covariance function with two output processes. The warm colours (red) represent high covariance and cool colours (blue) represent low covariance.

the terms $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$ which show high correlation within each output. The cross terms are the matrices $\mathbf{K}_{12}$ and $\mathbf{K}_{21}$, and these represent the relative dependency between the two outputs.

Now, given a set of test inputs $\boldsymbol{X}_*$, the predictive output $\mathbf{f}_*$ can be obtained by

$$p(\mathbf{f}_*, \boldsymbol{Y}|\boldsymbol{X}_*, \boldsymbol{X}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\boldsymbol{X}_*, \boldsymbol{X}_*) & K(\boldsymbol{X}_*, \boldsymbol{X}) \\ K(\boldsymbol{X}, \boldsymbol{X}_*) & K(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\sigma}^2\mathbf{I} \end{bmatrix}\right) \tag{3.3.10}$$

Since the training output $\boldsymbol{Y}$ is known, the Gaussian distribution can be conditioned on $\boldsymbol{Y}$ to obtain the predictive mean and variance for $\boldsymbol{X}_*$.

$$p(\mathbf{f}_*|\boldsymbol{X}_*, \boldsymbol{Y}, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma}) \tag{3.3.11}$$

where,

$$\begin{aligned} \boldsymbol{m} &= K(\boldsymbol{X}_*, \boldsymbol{X})[K(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\sigma}^2\mathbf{I}]^{-1}\boldsymbol{Y} \\ \boldsymbol{\Sigma} &= K(\boldsymbol{X}_*, \boldsymbol{X}_*) - K(\boldsymbol{X}_*, \boldsymbol{X})[K(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\sigma}^2\mathbf{I}]^{-1}K(\boldsymbol{X}, \boldsymbol{X}_*). \end{aligned}$$

The formulation above assumes the underlying processes have zero mean. This is not a significant limitation, since the mean of the posterior process is not confined to be zero. The outputs were centered to have zero mean on the training set. The subtracted values were then added to the predicted outputs $\mathbf{f}_*$.

3.3.3 Hyperparameter Optimization

The appropriate hyperparameters and noise variance are not known *a priori*. Hence, these have to be inferred from the data itself. From a Bayesian perspective the likelihood of hyperparameters $\boldsymbol{\Theta}$ and noise variance $\boldsymbol{\sigma}^2$ can be thought of as the marginal likelihood. Therefore, the values of $\boldsymbol{\Theta}$ and $\boldsymbol{\sigma}^2$ can be found by maximizing this likelihood. Once these were found predictions on test points can be made by substituting the optimal values of $\boldsymbol{\Theta}$ and $\boldsymbol{\sigma}^2$ into equation (3.3.11).

For a GP model the marginal likelihood is equal to the integral over the product of the likelihood function and the prior density. Both these are in Gaussian forms and product of the two results is another Gaussian. Therefore, the analytical form is given by

$$\begin{align}
p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\Theta},\boldsymbol{\sigma}^2) &= \int p(\boldsymbol{Y}|\mathbf{f},\boldsymbol{X},\boldsymbol{\Theta},\boldsymbol{\sigma}^2)p(\mathbf{f}|\boldsymbol{X},\boldsymbol{\Theta})d\mathbf{f} \tag{3.3.12}\\
&= \int \mathcal{N}(\mathbf{f},\boldsymbol{\sigma}^2\mathbf{I})\mathcal{N}(\mathbf{0},\mathbf{K})d\mathbf{f} \tag{3.3.13}\\
&= \frac{1}{(2\pi)^{\frac{\sum_{i=1}^{N} M_i}{2}}|K(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\sigma}^2\mathbf{I}|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\boldsymbol{Y}^T(K(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\sigma}^2\mathbf{I})^{-1}\boldsymbol{Y}\right) \tag{3.3.14}
\end{align}$$

This involves learning the appropriate hyperparameters and noise variance given the observations. Hyperparameters are free parameters of the covariance function. In this case the parameters of the kernel function are $\boldsymbol{\Theta} = [\upsilon_{mn}, \mathbf{A}_{mn}, \boldsymbol{\mu}_{mn}]$. These parameters are learned by maximizing the log marginal likelihood of the training outputs given the inputs. For numerical reasons the log marginal likelihood is used for calculations. This was first shown in Section 2.5.2. The log marginal likelihood equation was given by (2.5.16). The normalization constant was modified to account for multiple outputs. The log marginal likelihood for the multi-output GP is given by

$$[\boldsymbol{\Theta}, \boldsymbol{\sigma}^2] = \underset{\boldsymbol{\Theta},\boldsymbol{\sigma}^2}{\arg\max}\{\log(p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\sigma}^2))\} \tag{3.3.15}$$

where,

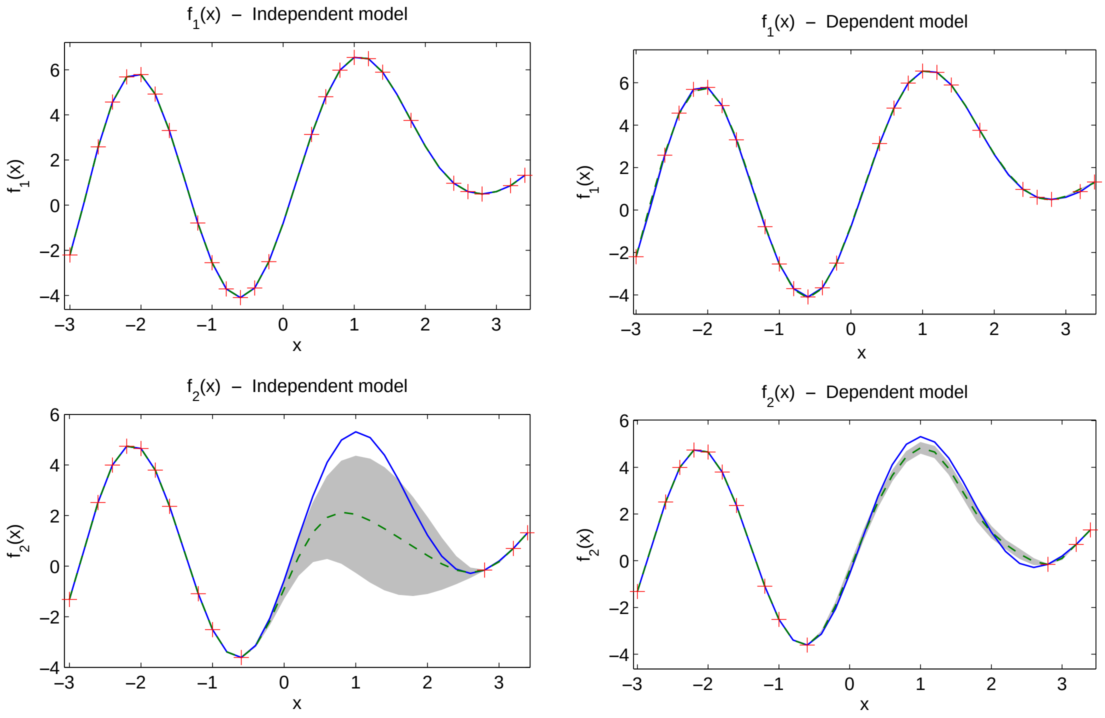
$$\log(p(\boldsymbol{Y}|\boldsymbol{X})) = -\frac{1}{2}\boldsymbol{Y}^T(K(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\sigma}^2\mathbf{I})^{-1}\boldsymbol{Y} - \frac{1}{2}\log|K(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\sigma}^2\mathbf{I}| - \frac{\sum_{i=1}^{N} M_i}{2}\log 2\pi \tag{3.3.16}$$

In GP model training the hyperparameters $\boldsymbol{\Theta}$ and noise variance $\boldsymbol{\sigma}^2$ are optimized. In the model hyperparameters υ_{mn} and $\mathbf{A}_{mn}$ express the relative significance of the associated regressors, $\boldsymbol{\mu}_{mn}$ expresses the relative dependence between the outputs and hyperparameter $\boldsymbol{\sigma}^2$ accounts for the influence of noise. Once these are found, predictions can be made on the posterior outputs $\boldsymbol{f}_*$ by substituting $\boldsymbol{\Theta}$ into (3.3.8) and $\boldsymbol{\sigma}^2$ into (3.3.5).

3.3.4 Two Dependent Outputs

An example application of DGP is given in Figure 3.2 where there are two strongly dependent outputs $f_1(\boldsymbol{X})$ and $f_2(\boldsymbol{X})$ over a $1D$ input space. The training data or the observations are generated by sampling points from the underlying function. From the first function $f_1(\boldsymbol{X})$ 25 points were sampled and from the second function $f_2(\boldsymbol{X})$ 13 points were sampled. For the output $f_2(\boldsymbol{X})$ no samples were taken between -0.5 to 2.5. The model was then built by maximizing the log marginal likelihood in equation (3.3.15) to find the optimal hyperparameters $\boldsymbol{\Theta}$.

The resultant dependent model is shown in Figure 3.2(b) along with the independent model with no output coupling in Figure 3.2(a). Note that the dependent model with DGP has learned the coupling between the two output functions and has predicted output 2 reasonably well compared to the independent model even when there are no observations present. The prediction uncertainty is also considerably low. Therefore, having a one dependent model compared to two independent models has proved to be useful for such problems.



(a) (b)

Figure 3.2: Two dependent output functions $f_1(\boldsymbol{X})$ and $f_2(\boldsymbol{X})$. (a) represent the independent model of the two outputs and (b) shows the dependent model. The solid blue line represent the underlying function, dashed green line is the estimated response from the model and the shaded regions represent 1σ uncertainty for the model prediction.

3.3.5 Computational Complexity

For N output processes and Q data points the covariance matrix scales to NQ leading to $\mathcal{O}((NQ)^3)$ complexity and $\mathcal{O}((NQ)^2)$ storage. The main computational cost is inverting the covariance matrix and this scales poorly with the number of data points. Hence it is computationally expensive compared to the traditional least-squares method. Since the training is done offline this cost is acceptable. For inference it has a complexity of $\mathcal{O}(NQ)$ for the predictive mean and $\mathcal{O}((NQ)^2)$ for the predictive variance.

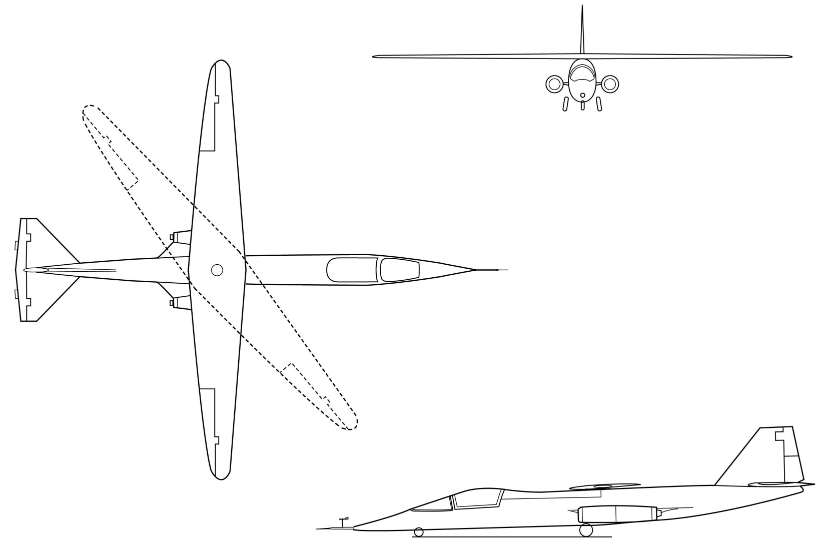


Figure 3.3: Ames Dryden 1 (AD-1) aircraft [54].

3.4 Simulation Results

This section details the implementation process, and test a parametric model and the proposed DGP model to compare the performance. The tests are done on a simulated aircraft, where an accurate model structure is known. The platform chosen was Ames-Dryden-1 (AD-1) aircraft (see Figure 3.3). It has an oblique wing structure which makes it a challenge for system identification. The platform has a high level of cross coupling between its longitudinal and lateral dynamics. The ability to capture dependencies between outputs using DGP was demonstrated.

3.4.1 AD-1 Oblique-Wing Aircraft

The AD-1 was designed to investigate the concept of an oblique-wing configuration [70]. The oblique wing pivots about the fuselage, remaining perpendicular during slow flight like a conventional wing aircraft to provide maximum lift and handling qualities during takeoff and landing, and rotating to angles of up to 60° as the aircraft speed increases to take advantage of the swept-back wing configuration during high speeds. The airplane has a high-fineness ratio fuselage, twin turbojet engines each producing

980 N of thrust at sea level, and a high-aspect-ratio wing. It is 11.8 m in length and has a wingspan of 9.8 m unswept. It has a gross weight of 973 kg, and an empty weight of 658 kg. The primary flight controls are conventional aileron, elevator, and rudder.

The required thrust for a given flight condition is reduced by increasing wing sweep due to decreased aerodynamic (lift induced) drag, permitting increased speed and longer range with the same fuel consumption. Variable-sweep wings are common on many high performance aircraft, including the F-14, F-111 and B-1. Analytical and wind tunnel studies of the concept had shown significant improvements in transonic aerodynamic performance for Mach numbers up to 1.4 with elimination of sonic booms in flight for Mach numbers up to 1.2 [19]. The implication for subsonic transport aircraft would be substantial increase in fuel economy, leading to improved range or reduced takeoff gross weight. Although there are these aerodynamic performance benefits, the platform suffers from lack of directional stability and roll-pitch coupling at high sweep angles leading to poor handling qualities. At 0° wing sweep the aircraft primarily experiences a pitch rate response as expected, whereas at 45° wing sweep there is a significant amount of secondary roll and yaw response [70]. Hence although the concept was first evaluated in 1976 [19] it only remained as a research platform.

3.4.2 Simulator

The simulator for this AD-1 aircraft uses data available in [18, 69] to construct a six Degrees of Freedom (6DOF) non-linear flight model. The simulations were conducted at 3800 m altitude where the research vehicle conducted flight testing [70]. The trim air speed was maintained at 110 kts, less than the thrust limit for level flight at 3800 m. Two configurations of the aircraft were used for training and testing, one at 0° and the other at 45° sweep angle. At 0° sweep the aircraft acts similar to most conventional platforms, with minimal cross coupling in roll and pitch axes. At 45° wing sweep the aircraft experiences high levels of cross coupling, there is low directional stability, unusual trim requirements and the pilot rating indicated degrading handling qualities

[70].

3.4.3 Flight Model Training

The purpose of model training is to design maneuvers to maximize the information content in logged data. To achieve this, system modes must be excited such that the sensitivities of the model outputs to the parameters are high. Standard input flight test maneuvers were used to excite the dynamics while maintaining a trimmed flight condition [35]. This is such that the identified parameters can be constant throughout the maneuver.

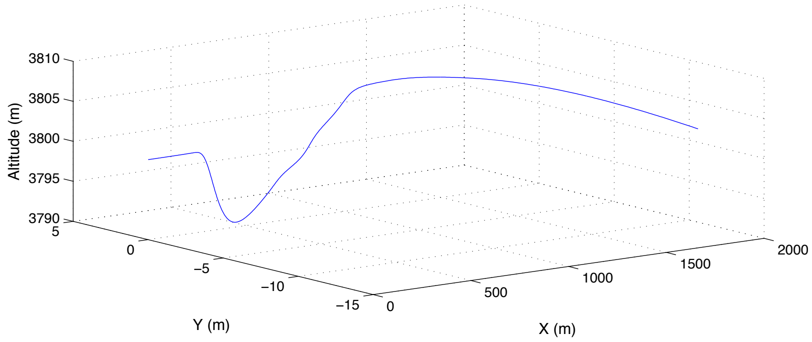
The primary flight controls for AD-1 aircraft are the conventional aileron, elevator and rudder. The excitations were done using these control inputs $\boldsymbol{\delta} = [\delta_e, \delta_a, \delta_r]$, starting from the trimmed surface position. The control deflections are elevator δ_e, aileron δ_a and rudder δ_r. The throttle input was not included as it was maintained constant throughout the maneuver. The inputs for the other controls were optimized using *a priori* knowledge about the dynamic system response. This consisted of a series of doublet control inputs (see Figure 3.4 and 3.5) near the natural frequencies of the dynamic modes. The resultant non-dimensional aerodynamic coefficients for the maneuvers are given in Figure 3.6. The single-surface inputs are used so that individual control surface motions would have low correlation with other control surface motions. This is particularly useful when constructing parametric models with Least Squares (LS) as it helps to segregate the effects from other inputs. Each doublet consisted of a 1s deflection in the positive direction followed by a 1s deflection in the negative direction. These were then separated by a 5s delay of no control input. The chosen amplitudes were 4° for elevator and 5° deflections for rudder and aileron inputs. Longitudinal maneuvers consisted of an elevator doublet; lateral maneuvers consisted of a rudder doublet followed by an aileron doublet. Aileron doublet was performed last to minimize changes in the flight condition [69]. The training flight data from the maneuvers were analysed at 20 Hz which is adequate to capture the highest frequency mode [69].

The cross coupling between flight modes is apparent at 45° sweep angle with the elevator doublet. It can be observed in Figure 3.5 between 3 - 10s that there is high pitch-roll coupling where the primary response of the aircraft is roll. Hence, if the aircraft is required to maintain a pullup maneuver the pilot would require cross controlling of pitch and roll.

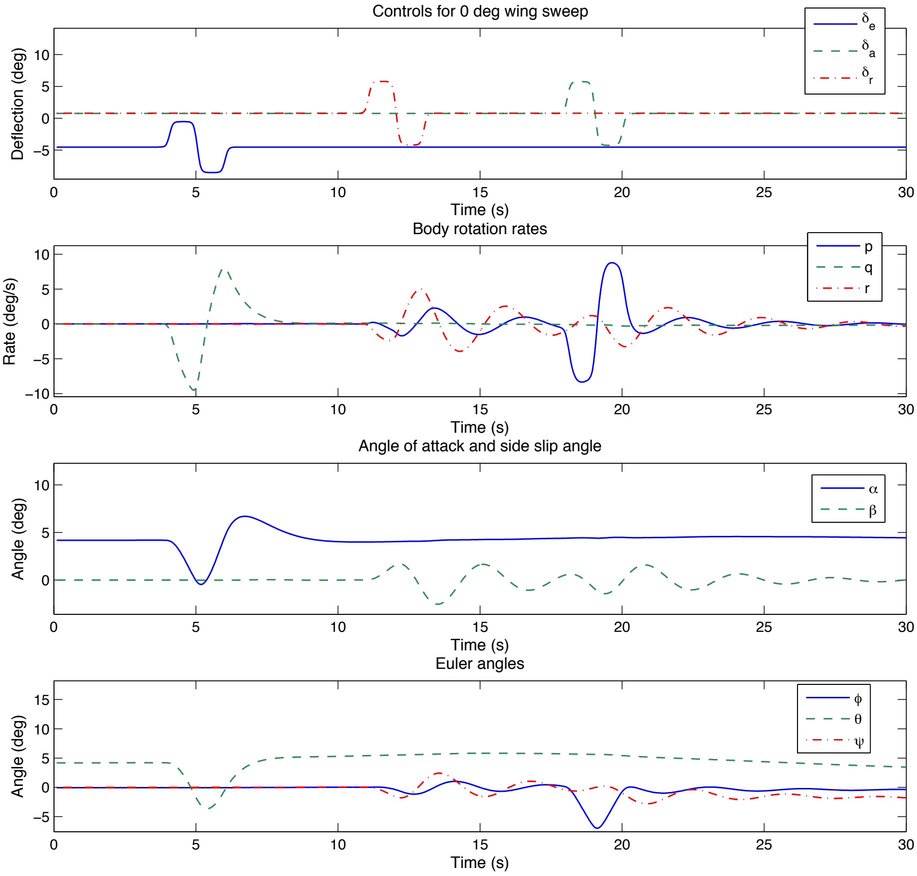
At 0° wing sweep two different flight models were trained using least squares and DGP. At 45° wing sweep three different models were trained, the first using least squares with a new model structure to account for the cross coupling aerodynamic terms, second using least squares on the same model structure as for 0° wing sweep, and the last using DGP. An extra flight model was trained here to demonstrate the performance of having coupled and uncoupled terms in the least squares parametric model structure. More about this is presented in the following sub-section.

Parametric model

The parametric system identification was performed using the least squares estimator that was presented in Section 2.4.1. The structure for the regressor matrix $\mathbf{X}$ in equation 2.4.15 was found by iterative testing for each individual observation (see equations 2.4.9 and 2.4.10). It should be noted that the observation error $\boldsymbol{\varepsilon}$ in equation 2.4.15 was assumed to be zero. The measure of accuracy for each derivative was obtained through Cramér-Rao bound analysis. With respect to (2.4.6), this is a measure of the best possible precision attainable for $\boldsymbol{\vartheta}$ from the information available in measured data $\boldsymbol{z}$. For testing, two different flight models were constructed at the different wing sweep angles to model the corresponding flight dynamics. At 0° wing sweep the model is decoupled into longitudinal and lateral directional terms. At 45° wing sweep the effects of the aerodynamic cross-coupling terms were added to the model. These steps were performed iteratively as the exact structure was previously not known. It was more difficult to model the 45° case as the cross-coupling terms are not similar to a conventional aircraft model. The model terms in $\boldsymbol{\vartheta}$ are the non-dimensional stability and control derivatives $C_{(\cdot)}$ which are a function of the aircraft's

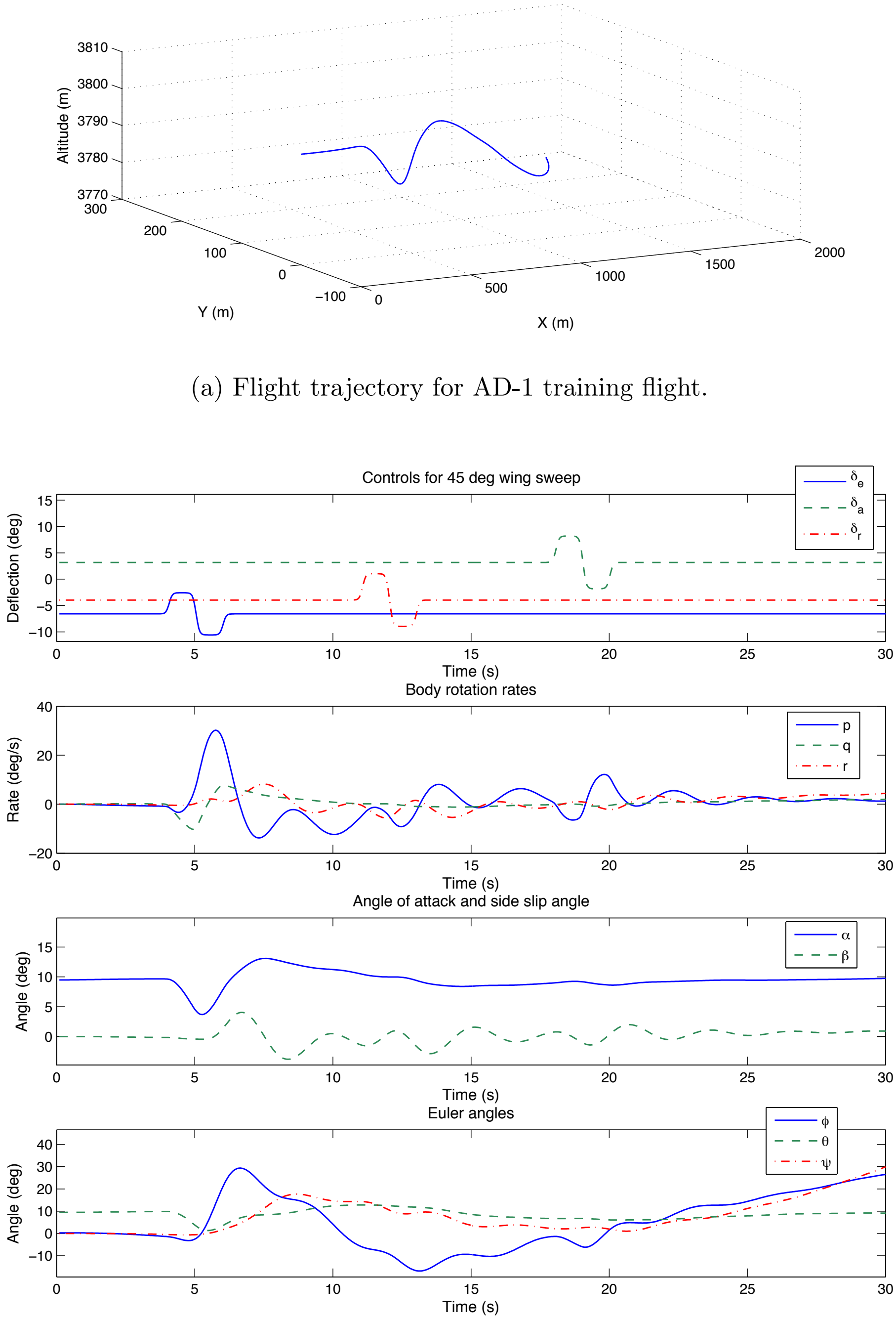


(a) Flight trajectory for AD-1 training flight.



(b) Applied control inputs and the response plots.

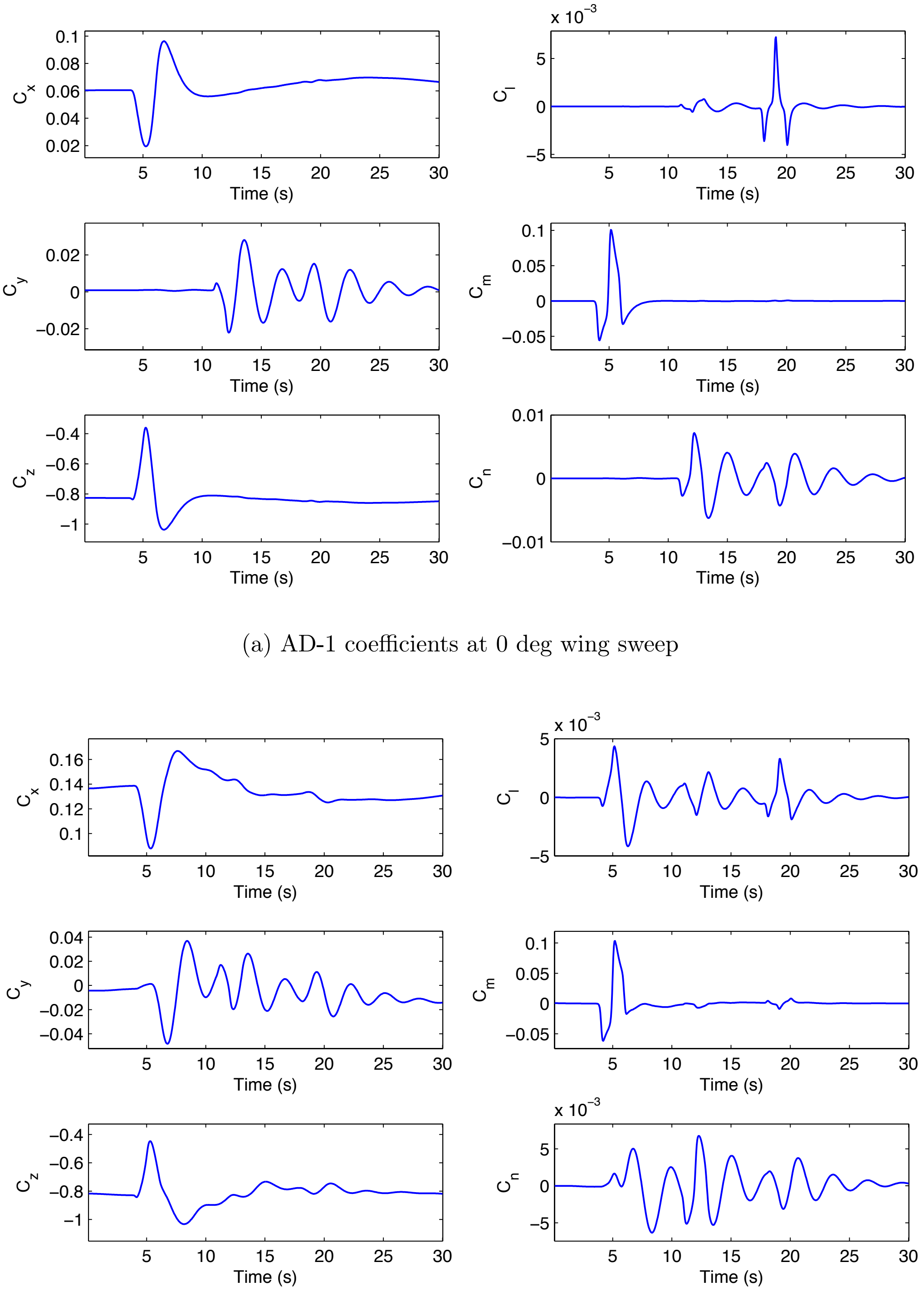
Figure 3.4: AD-1 model training at 0° wing sweep angle. (a) shows the flight path of the simulation. The aircraft starts at (0,0) and at an altitude of 3800 m with an initial speed of 110 kts. (b) shows the doublet control inputs applied on the system and the resultant body axis components.



(a) Flight trajectory for AD-1 training flight.

(b) Applied control inputs and the response plots.

Figure 3.5: AD-1 model training at 45° wing sweep angle. (a) shows the flight path of the simulation. The aircraft starts at (0,0) and at an altitude of 3800 m with an initial speed of 110 kts. (b) shows the doublet control inputs applied on the system and the resultant body axis components.



(a) AD-1 coefficients at 0 deg wing sweep

(b) AD-1 coefficients at 45 deg wing sweep

Figure 3.6: The non-dimensional aerodynamic coefficients for AD-1 model training at (a) 0° and (b) 45° wing sweep.

state and control inputs. To compare the difference, an additional model was trained using the same structure as for the 0° wing sweep for the 45° case.

The equations for 0° wing sweep are

$$\begin{aligned}
C_X &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_{\dot\alpha}}\frac{\dot\alpha\bar c}{2V} + C_{X_{\dot q}}\frac{\dot q\bar c}{2V} + C_{X_{\delta_e}}\delta_e \\
C_Y &= C_{Y_0} + C_{Y_\beta}\beta + C_{Y_{\dot\beta}}\frac{\dot\beta b}{2V} + C_{Y_{\delta_r}}\delta_r \\
C_Z &= C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_q}\frac{q\bar c}{2V} + C_{Z_{\delta_e}}\delta_e \\
C_l &= C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\dot\alpha}}\frac{\dot\alpha\bar c}{2V} + C_{m_q}\frac{q\bar c}{2V} + C_{m_{\delta_e}}\delta_e \\
C_n &= C_{n_0} + C_{n_\beta}\beta + C_{n_r}\frac{rb}{2V} + C_{n_{\delta_r}}\delta_r
\end{aligned}$$

and at 45° wing sweep are

$$\begin{aligned}
C_X &= C_{X_0} + C_{X_\alpha}\alpha + C_{X_\beta}\beta + C_{X_{\dot\alpha}}\frac{\dot\alpha\bar c}{2V} + C_{X_{\dot q}}\frac{\dot q\bar c}{2V} + C_{X_{\delta_e}}\delta_e \\
C_Y &= C_{Y_0} + C_{Y_\beta}\beta + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\
C_Z &= C_{Z_0} + C_{Z_\alpha}\alpha + C_{Z_\beta}\beta + C_{Z_q}\frac{q\bar c}{2V} + C_{Z_{\delta_e}}\delta_e \\
C_l &= C_{l_0} + C_{l_\alpha}\alpha + C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V} + C_{l_r}\frac{rb}{2V} + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_\beta}\beta + C_{m_p}\frac{p\bar c}{2V} + C_{m_q}\frac{q\bar c}{2V} + C_{m_{\delta_e}}\delta_e \\
C_n &= C_{n_0} + C_{n_\alpha}\alpha + C_{n_\beta}\beta + C_{n_r}\frac{rb}{2V} + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r
\end{aligned}$$

where the control deflections are elevator δ_e, aileron δ_a and rudder δ_r. The angle of attack and side slip angle are given by α and β respectively. Angular rates for angle of attack $\dot\alpha$ and side slip $\dot\beta$ were non-dimensionalized with respect to airspeed V, wing mean geometric chord $\bar c$ and wing span b. The model developed for 0° wing sweep is labeled as LS0 and the model developed for 45° wing sweep is labeled as LS45.

DGP model

DGP regression requires measurements of the aerodynamic forces and moments that the aircraft is experiencing, as well as measurements of the regressors. These forces and moments derived in (2.4.9 - 2.4.10) are modeled as the output states to be learned

$$\boldsymbol{Y} = \begin{bmatrix} C_X & C_Y & C_Z & C_l & C_m & C_n \end{bmatrix} \tag{3.4.1}$$

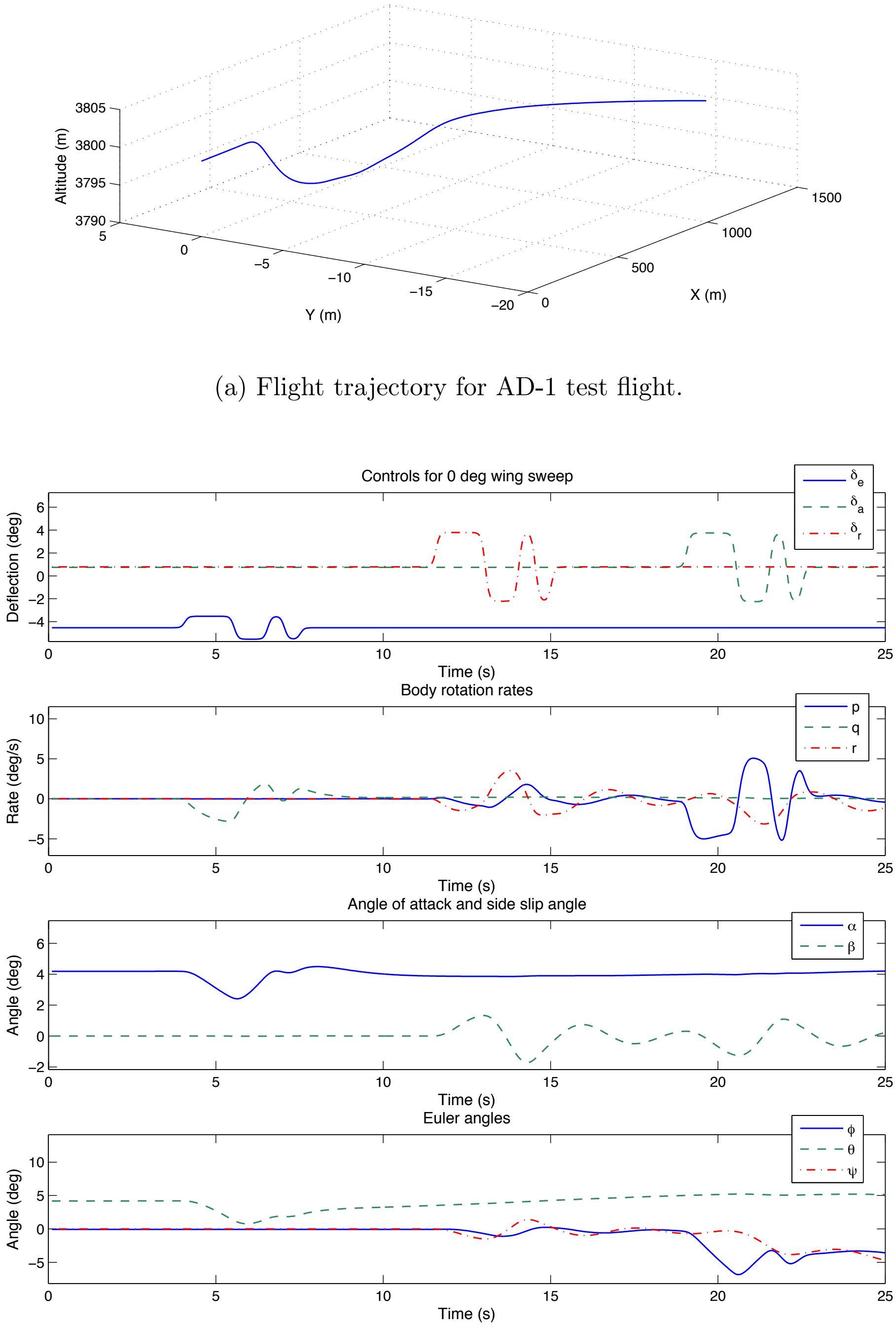
These output states are assumed to be zero mean processes. Having determined the coefficients to perform parameter estimation, the regressors can be measured. These are

$$\boldsymbol{X} = \begin{bmatrix} \mathbf{a}_b & \boldsymbol{\omega}_b & \boldsymbol{\Psi}_b & \alpha & \beta & \boldsymbol{\delta} \end{bmatrix} \tag{3.4.2}$$

The state is composed of the body axis accelerations $\mathbf{a}_b = [a_x, a_y, a_z]$, rotation rates $\boldsymbol{\omega}_b = [p, q, r]$, Euler angles $\boldsymbol{\Psi}_b = [\phi, \theta]$, angle of attack α, side slip angle β, and control inputs $\boldsymbol{\delta} = [\delta_e, \delta_a, \delta_r]$. Rotational rates were non-dimensionalized with respect to true velocity, wing mean geometric chord and wing span. It is assumed that these inputs are independent and stationary to satisfy the condition for the kernel function. The models developed using DGP for 0° wing sweep is labelled as DGP0 and 45° wing sweep is labeled as DGP45.

3.4.4 Flight Model Testing

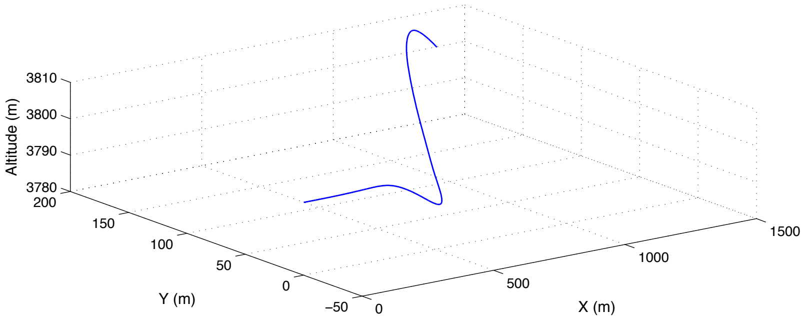
The simulated test flight was designed to validate the learned flight models. The testing was done using another common input maneuver called the 3-2-1-1 [39, 60]. This consists of sequential pulse widths in the ratio of 3-2-1-1. See Figure 3.7 and 3.8 for the control inputs and the resultant angular rates for 0° wing sweep and 45° wing sweep respectively. The '1' in the 3-2-1-1 was chosen to be a 1s pulse width period. The chosen amplitudes were 1° for elevator and 3° deflections for rudder and aileron inputs. These were selected such that the aircraft maintains a stable flight closer to the trimmed condition.



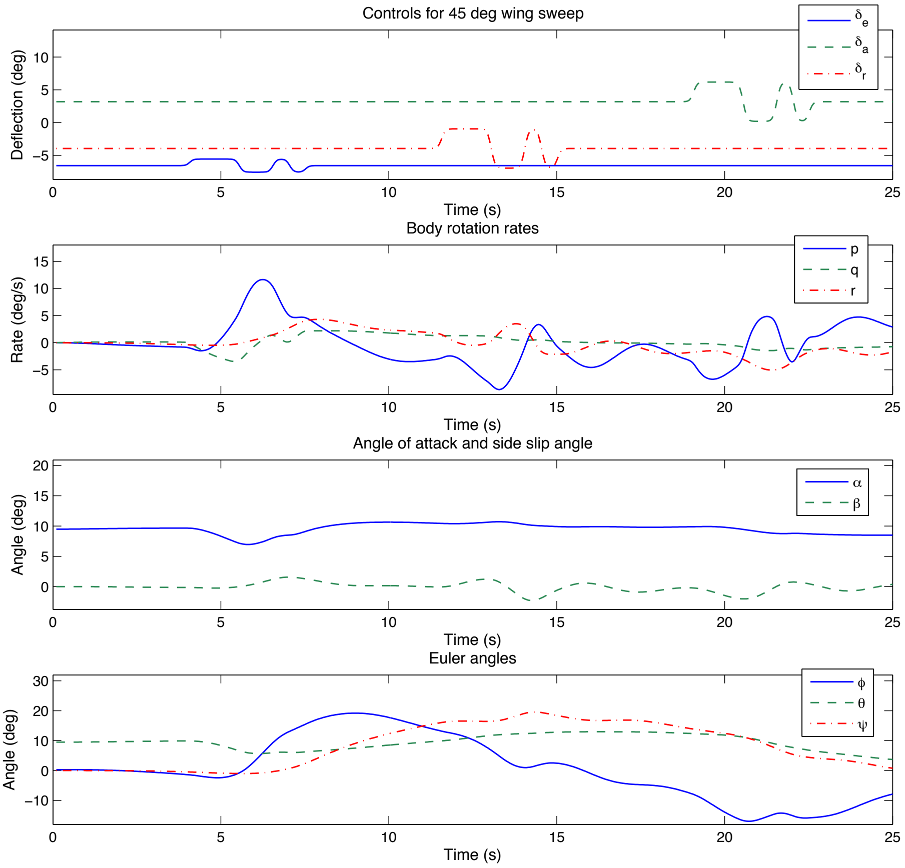
(a) Flight trajectory for AD-1 test flight.

(b) Applied control inputs and the response plots.

Figure 3.7: AD-1 model testing at 0° wing sweep angle. (a) shows the flight path of the simulation. The aircraft starts at (0,0) and at an altitude of 3800 m with an initial speed of 110 kts. (b) shows the 3-2-1-1 control inputs applied on the system and the resultant body axis components.



(a) Flight trajectory for AD-1 test flight.



(b) Applied control inputs and the response plots.

Figure 3.8: AD-1 model testing at 45° wing sweep angle. (a) shows the flight path of the simulation. The aircraft starts at (0,0) and at an altitude of 3800 m with an initial speed of 110 kts. (b) shows the 3-2-1-1 control inputs applied on the system and the resultant body axis components.

3.4.5 Model Robustness Testing

Wind gust were introduced in order to test the model's robustness to unmodeled disturbances. The resultant outputs were then observed to see if they remain bounded of the model. The gust components are firstly modelled in frequency domain using the Dryden Spectra gust model and then transformed into time domain body velocities and rotational rate disturbances. Since these gust components are instantaneous changes in airflow, their effect can be approximated by adding the components directly on to the body velocity and rotational rate vectors of the aircraft. A wind gust of 3 kts was applied to the 0° case. The test maneuver is the same as aforementioned with 1° amplitudes for all the control inputs to maintain the flight closer to the trimmed condition. The 0° model which was trained without any gust was used for testing.

3.4.6 Estimation Results

This section presents the coefficient estimates obtained for the simulated test flight segment. The estimates are from the least squares and DGP models. These were then compared against the coefficients obtained from the non-linear simulator (ground truth). The estimated coefficients and the prediction uncertainties of the simulated test flight are shown in Figure 3.9 and 3.10. In Figure 3.10 the estimates from the uncoupled 0° model is also presented. By observing these results, both the DGP0 and LS0 estimates for 0° , and DGP45 and LS45 estimates for 45° cases are nearly identical to the simulated response. Even in regions of relatively high uncertainty DGP is able to infer a solution closer to the non linear response. This is only possible due to the $\boldsymbol{\mu}_{mn}$ hyperparameter in equation (3.3.8), which describes the outputs that are coupled and translated relative to each other.

The relative errors on the estimated parameters were calculated by comparing against the coefficients obtained from the simulator for the entire simulated test flight segment. The summary of the results are listed in Table 3.1 and 3.2. The median error obtained for each coefficient using the least squares approach and DGP are presented. Additionally, Table 3.2 lists the 0° parametric model (LS0) results. In general, for all

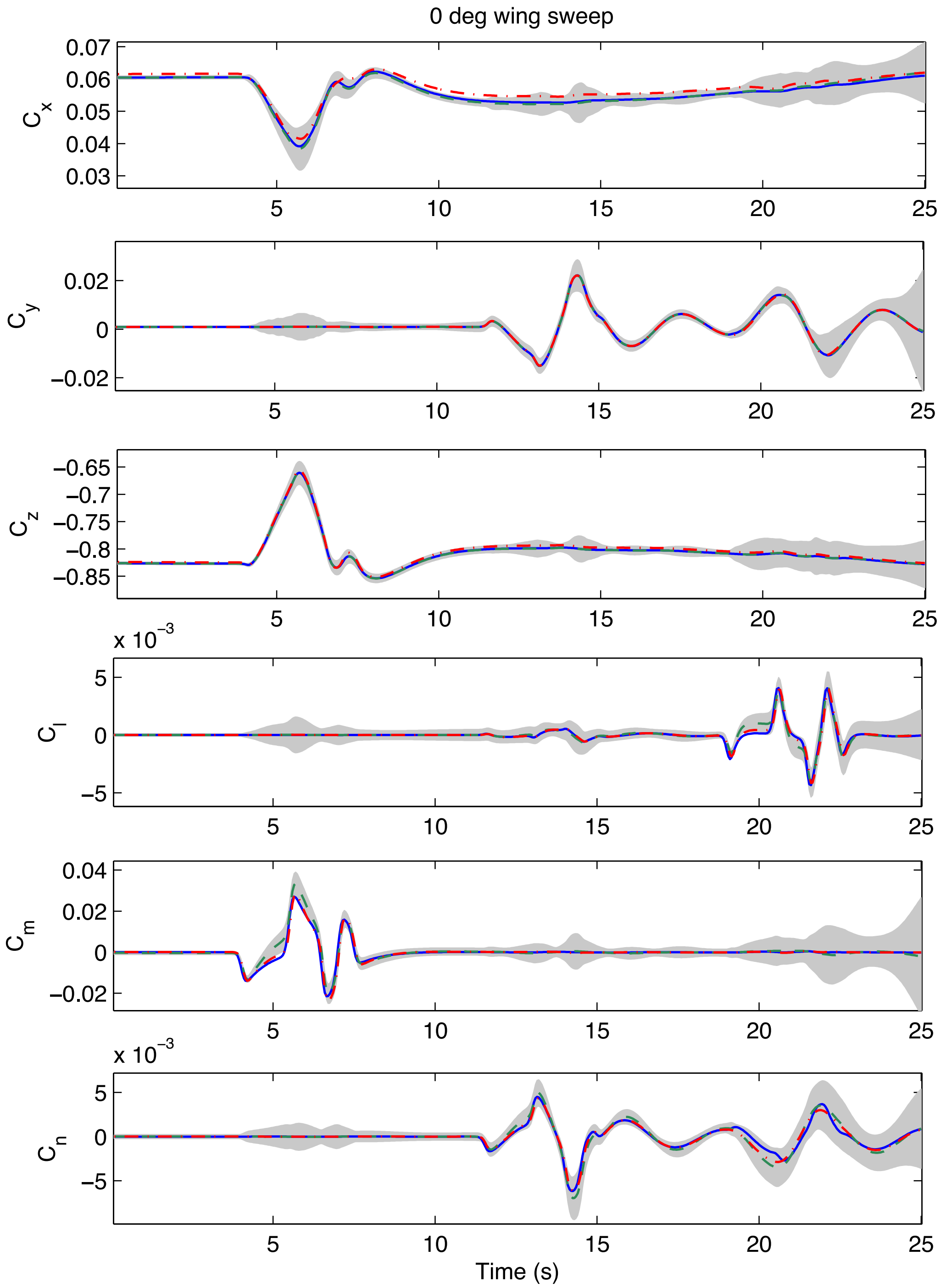


Figure 3.9: AD-1 ground truth (solid blue), least squares (dashed-dotted red) and DGP estimated (dashed green) non-dimensional aerodynamic coefficients with prediction uncertainties (gray) at 0° sweep angle.

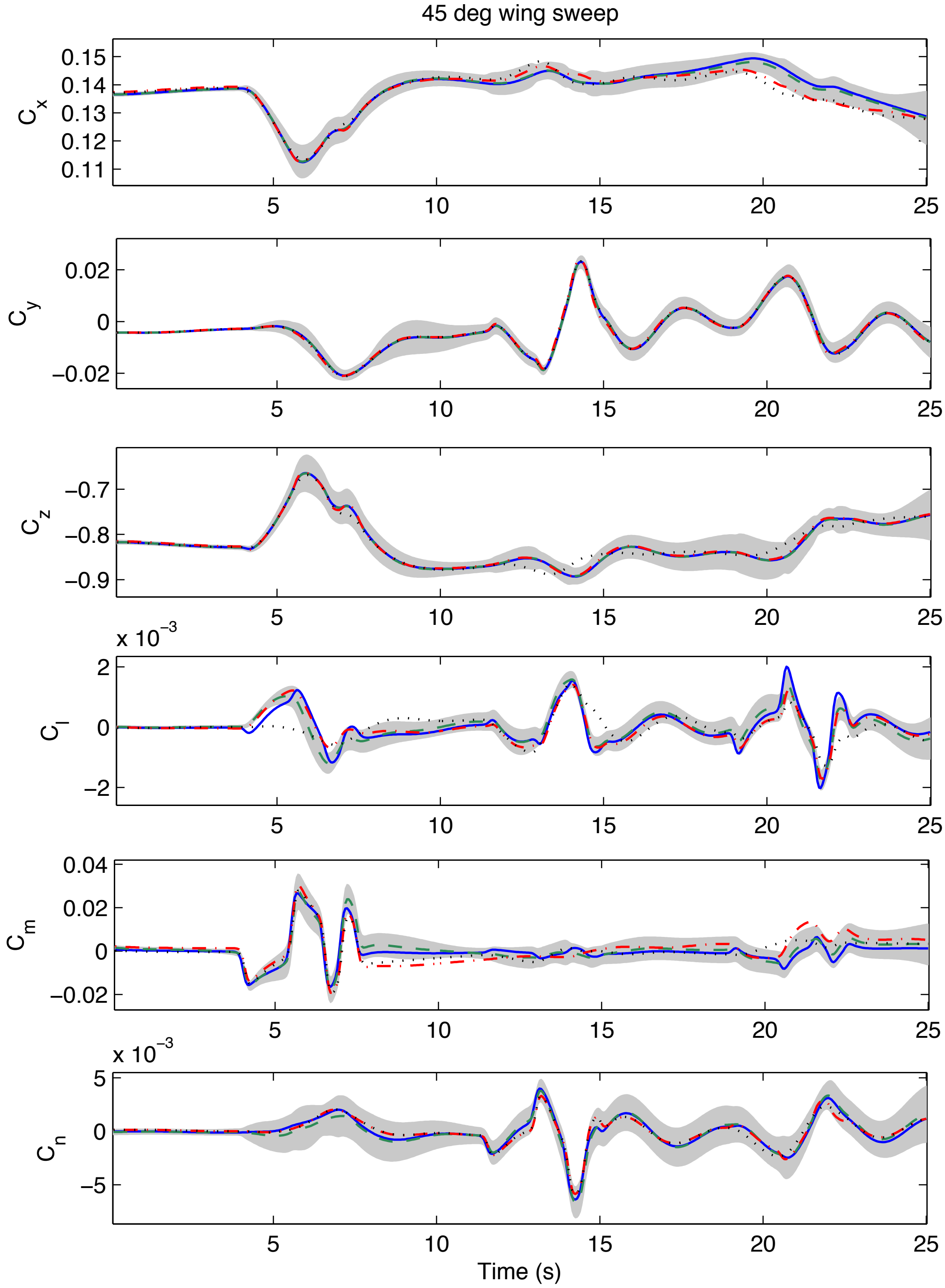


Figure 3.10: AD-1 ground truth (solid blue), least squares (dashed-dotted red) and DGP estimated (dashed green) non-dimensional aerodynamic coefficients with prediction uncertainties (gray) at 45° sweep angle. In here the estimates from the uncoupled model (dotted black) are also shown.

Table 3.1: AD-1 coefficient estimation errors for 0° wing sweep.

% Error	C_X	C_Y	C_Z	C_l	C_m	C_n
LS0	2.645	0.761	0.445	4.675	5.536	3.084
DGP0	0.205	0.048	0.007	12.907	21.244	19.347

Table 3.2: AD-1 coefficient estimation errors for 45° wing sweep.

% Error	C_X	C_Y	C_Z	C_l	C_m	C_n
LS0	0.051	0.041	0.050	57.144	47.878	26.323
LS45	0.004	2.451	0.193	4.640	12.594	23.675
DGP45	0.005	0.036	0.047	25.951	9.520	3.922

the methods force estimates outperform moment estimates by a significant amount. At 0° wing sweep DGP0 model outperforms LS0 model for force coefficient estimates. However, it doesn't perform the same for moment coefficients. It should also be noted that since the average moment coefficient is zero, the expected % errors would be higher than that for the Lift/Drag % errors. This result can be further improved by designing simulated test flight maneuvers to excite the moment coefficients. Also, it should be noted that tests must be conducted to verify the trained hyperparameters of the DGP models as there is no guarantee that the marginal likelihood will not suffer from multiple local maxima. In practice, a technique such as simulated annealing [81] can be used to obtain a good approximation to the global optimum.

In the case for 45° wing sweep the median error on the LS45 response is 7.26%, on LS0 it is 21.92% and on the DGP45 it is 6.58%. Hence, the DGP45 response is as good as the LS45 response without having to know the model structure. This is due to the cross coupling terms in the covariance matrix which learned the dependencies between system parameters. It is also important to note that for a parametric model the goodness of regression can be dependent on how much time and effort was invested in the model selection. Since an accurate model structure is not known *a priori* and because it can vary across the flight envelope, it is a challenge to use parametric techniques to identify aircraft with complex flight dynamics. The proposed method could overcome this.

Table 3.3: AD-1 coefficient estimation errors for 0° wing sweep under wind gust.

% Error	C_X	C_Y	C_Z	C_l	C_m	C_n
LS0	3.138	1.242	0.772	8.615	7.254	5.915
DGP0	0.264	0.078	0.012	17.785	25.736	23.165

In addition, results are obtained for the 0° case under wind gust for both LS0 and DGP0. A summary of these are listed in Table 3.3. The performance of both models have degraded slightly with wind gust. Specially the moment coefficients which are a resultant of the rotational rate vectors have a much greater effect from the added gust components. Similar to the case without gust, the DGP0 model outperforms in force coefficient estimates and not in the moment coefficients. This is because the predictions were made using the same model that was used in the 0° case without gust (see Table 3.1).

These uncertainty measures in predictions are used as the verification measure to validate the model usefulness. The lack of confidence in model prediction is served as the grounds to reject the model as not useful. Thus, if the uncertainty is too high to make an accurate prediction, then the particular flight test data could be learned or manoeuvres could be designed to excite those modes of operation to be added to the GP model. The prediction variance could also be used in falseness validation, whether via specific performance measures such as log-predictive density error or through observation of confidence limits around the predicted output [37].

3.5 Summary and Conclusion

A system identification approach for aicraft based on dependent Gaussian processes is presented in this chapter. The proposed model is non-parametric and alleviates the labor intensive mathematical modeling process. It has the inherent capability to handle system noise and to deal with biases data. It was proved in simulation that the performance of the model is comparable to the classical least squares system identification method. In addition, the approach captures cross coupling between inputs and identifies dependencies between outputs. Hence, the model need not be decoupled with respect to lateral and longitudinal modes. Moreover, DGP provides uncertainty on the predicted estimates, which gives a notion of confidence on the learned parameters. It quantifies validity of the predictions and when they are outside the bounds of the trained model.

The focus of the following chapter involves addressing the increase in computational complexity to handle large data sets and sampling the training data efficiently by selecting points that maximize the prior information. This will provide the capability of modeling the entire flight envelope.

Chapter 4

Local and Global Gaussian Process Approximations

4.1 Introduction

Extending the model to learn the entire flight envelope involves capturing vehicle parameters at different speeds as well as at different dynamic pressures [35]. However, the GP model presented scales poorly with the increase in computational complexity to handle large data sets [27]. In addition, the model does not capture any non-stationary properties in aerodynamics. This chapter endeavours to provide solutions to these problems. Included in this chapter is an approximation on the DGP model to make it scalable and to capture any local properties while maintaining all its positive attributes.

The main problem with GP based design is the computational cost required to invert its $M \times M$ covariance matrix where M is the number of training input points. This matrix inversion has a complexity of $\mathcal{O}(M^3)$ and the cost per prediction is $\mathcal{O}(M^2)$. With DGP this is also scaled by the number of outputs (N), therefore the complexity of inversion becomes $\mathcal{O}((MN)^3)$ and prediction becomes $\mathcal{O}((MN)^2)$. This makes it unfavourable for scaling and becomes prohibitive for large M [66]. In order to address this problem an approximation technique can be used.

The proposed algorithm uses an additive model that combines global and local Gaussian processes to learn a multi-output system. It combines both short and long length-scale phenomena for a multi-output GP. Having a combined approximation makes the model suitable for all regions of the flight envelope. To capture the global properties a new sampling method is introduced to gather information about the output correlations, requiring less number of points to describe the data set. Local properties were captured using a non-stationary covariance function with KD-trees for neighbourhood selection. This makes the model scalable to learn from high dimensional large-scale data sets.

4.2 Related Work

To overcome the computational limitations of GP regression, numerous methods have been proposed to summaries the original input space. The two main approaches for this are based on global and local approximations. The most widely used type is the global approximations. They tend to summarize all the training data via a small set of support points. These include making a reduced rank covariance matrix [7, 61], using induced inputs [17, 62, 72] or by selecting a subset from the training data [91]. A unifying view of these approximation techniques is presented in [63].

For learning a reduced rank GP two main conditions have to be satisfied; one condition is selecting a support set and the other is learning the hyperparameters. In [71] a method is presented for learning the support set for given hyperparameters of the covariance function based on approximating the posterior. This fails to guarantee a good generalized performance. In [61], a greedy algorithm for support set selection is proposed that is based on maximizing the marginal likelihood. A method of learning the hyperparameters of the covariance function given a support set is proposed. In [7] a full GP is approximated with A basis functions reducing the inversion complexity to $\mathcal{O}(A^2M)$ for DGP. It was shown that when the basis function set is equal to the training input set, the resulting reduced rank process is in fact full rank. However, a framework for choosing these basis functions were not presented.

To use induced inputs a small set of latent variables ($\ll M$) is introduced to correspond to input locations [72]. This is referred to as pseudo-inputs and can be placed in arbitrary locations. The input locations do not have to be from the original data input points. These variables can then be optimized as an additional set of hyperparameters. All the data points contribute to each prediction made via the induced inputs. Therefore, it is still a global approximation. This technique scales poorly with the increase in dimensionality. In [91] an information gain strategy was developed based on maximizing entropy. This suffers a significant flaw of not considering the prediction quality of the chosen training points. Hence, some of the sample locations may have “wasted” information.

The other approach is to use local regression, where only the local points are accounted to infer a test point in the chosen region [53, 73, 85]. Here, the correlations are induced using only few data points around their space. This works well in regions where data changes rapidly and there is the requirement to model short length-scale properties. It also means that a function can behave differently for different inputs than a global stationary response. The result is a significantly fast GP. Nevertheless, the performance of this model largely depends on the number of points used from the data set. The widely used approach here is to cluster the input space into smaller subspaces and then make a weighted average over the GP experts [57, 58, 65]. Each expert is a GP in itself and leans from data points within its region. They may also learn different characteristics of the function such length scales, noise variance, etc. It was demonstrated in [58] that local GPs can be used for real-time learning of inverse dynamics for a robot arm. Here, the computational cost is kept fixed by deleting old inputs and updating the matrix with new readings. During this step, informativeness of the data points was not taken into consideration. For expert selection, another approach is to use KD-trees [68, 83]. These can be queried during the inference process to obtain a predefined number of local neighbours near the test data. By utilizing only a small set of points within a set region to make the predictions it reduces computational cost significantly. One disadvantage of local GPs is that the resultant predictions can be unsmooth. The independence between the data blocks lead to a discontinuous nature in the predictions. The smoothness across these boundaries and between the local models can be attained with longer length scales.

One of the more interesting concepts that has recently been addressed is how to combine both the global and local properties together to model short and long length-scale phenomena [73, 82]. Having the best of both worlds can be an added advantage and will make the model suitable for all regions of the data set. However, the approaches in [73, 82] are only applicable for a single output GP. In addition, using induced inputs for the global GP in [73] makes it unscalable for high dimensional problems.

4.3 System Modeling with Gaussian Process Approximations

To solve the computational problem an approximation is required to be made on the original solution. In addition, the DGP covariance function is a stationary and is not scalable with the increase in the number of training input points. Given the stationary assumption on the input data, values of the covariance function $C(\mathbf{x}_i, \mathbf{x}_j)$ between inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ are only dependent on their distance and do not change within the input space. This might only be valid for a narrow region of the flight envelope when the UAV is flying under its stall angle of attack and when the conditions are steady for a wings-level flight with no sideslip [35]. For this model to be valid for a larger region it is required to learn the non-stationary effects. This would also relax the stationarity assumption made in Section 3.3.1. Consequently, to address both the computational and stationary issues, and to still capture the output dependencies, the input dimensions are to be modeled with more than one length-scale. In this case, a more reasonable approach is to use an additive model. Consider N-output processes $\mathbf{y} = [y_1, \ldots, y_N]$ where $\mathbf{y} \in \mathbb{R}^N$ and $\boldsymbol{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$ where M is the total number of training cases where $\mathbf{x} = [x_1, \ldots, x_D]$ where $\mathbf{x} \in \mathbb{R}^D$ is a D-dimensional input vector.

The training set is assumed to be drawn from the noisy process represented by

$$y_n(\mathbf{x}) = u_n(\mathbf{x}) + w_n(\mathbf{x}) \tag{4.3.1}$$

$$= f_n(\mathbf{x}) + j_n(\mathbf{x}) + w_n(\mathbf{x}) \tag{4.3.2}$$

Here the latent function $u_n(\mathbf{x})$ in equation (4.3.1) is replaced by a sum of two functions. One function to capture the output correlations and the other to capture local non-stationary behaviour. $w_n(\mathbf{x})$ is a stationary Gaussian white noise drawn from $\mathcal{N}(0, \sigma_n^2)$ with variance σ_n^2. $f_n(\mathbf{x})$ is the global function defined by

$$f_n(\mathbf{s}) = \sum_{m=1}^{M} h_{mn}(\mathbf{x}) * x_m(\mathbf{x}) \tag{4.3.3}$$

$$= \sum_{m=1}^{M} \int_{\mathbb{R}^D} h_{mn}(\boldsymbol{\alpha}) x_m(\mathbf{x} - \boldsymbol{\alpha}) d^D \boldsymbol{\alpha} \tag{4.3.4}$$

The above equation calculates the sum of convolutions of the h_{mn} kernel connecting the input m to output n. An approximation is made on this function to calculate the global resultant estimate for the new covariance function. The local function $j_n(\mathbf{x})$ is a generalized linear function approximation using linear parametrization with nonlinear basis functions. It is defined by

$$j_n(\mathbf{x}) = \boldsymbol{\kappa} \hat{h}(\mathbf{x}) \tag{4.3.5}$$

where $\hat{h}(\mathbf{x})$ is a vector of nonlinear basis functions and the parameters $\boldsymbol{\kappa} \in \mathbb{R}^D$ is a set of weights to be estimated from the data.

4.3.1 Assumptions

In order to satisfy the conditions for a valid kernel function, certain assumptions are made. First, the system inputs are independent processes. Next, the prior underlying process has zero mean. If constructing a model for a larger region where the prior underlying process changes significantly throughout the flight envelop it is possible to use an explicit basis function (see Section 2.7 in [66]). Finally, the observed data is noisy and it was assumed this distribution is Gaussian white noise. Hence, Gaussian white noise was added to each output.

By not having the restriction of a stationary covariance, the joint probability distribution can change within the input space. Thus, the covariance function $C(\mathbf{x}_i, \mathbf{x}_j)$ between inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ can change within the input space.

4.4 Local and Global Approximations

As discussed in the previous section, making a combined approximation will make the model more suitable for all regions of the flight envelope. In this section, the regimes in which these different approaches work well are investigated. For the global approximate, a mutual information criterion was developed to capture the output correlations. For the local approximate, the covariance function used to capture the non-stationary properties is provided. Finally, the combined approximate and details of the hyperparameter optimisation is presented.

4.4.1 Global Approximation

Here, a subset of the input features is chosen to represent the underlying function. An optimization criterion is proposed to find a feature set that is most informative about locations that are yet to sample while maximizing differential entropy across other output dimensions. It measures the effect of selecting a particular point on the posterior uncertainty of the GP. This technique was first proposed in [40] for optimal sensor placements. We extend this to account for dependence in multiple output dimensions. The proposed algorithm would first sample features that are most informative about other output dimensions and then will go onto sample features that reduce uncertainty in its own dimension. The result is a subset of inputs which captures maximum dependency across outputs, maximum relevance and minimum redundancy.

The purpose of feature selection is to find a feature set $\mathcal{S}$ with k features, which jointly has the largest dependency about the outputs. First, define an N dimensional space with a discrete set of locations Q. This forms a space $|\mathcal{V}| = NQ$ which is the set of all possible points from where it can be sampled from. If considered a subset of the random variables $\mathcal{A} \subseteq \mathcal{V}$, then their joint distribution is also a Gaussian. Starting with an empty set of locations $\mathcal{A} = \emptyset$, points $y_n(\mathbf{x}) \in \mathcal{V} \setminus \mathcal{A}$ are added until $|\mathcal{A}| = k$ where k is the number of chosen features out of NQ possible locations. The goal is to place k features that will give a good prediction throughout $\mathcal{V}$.

This can be achieved by maximizing the mutual information between $\mathcal{A}$ and rest of the space $\mathcal{V} \setminus \mathcal{A}$. Now, in order to account for dependence between multiple dimensions a new variable $\mathcal{B} = [\{y_N(\mathbf{x})\} \setminus y_n(\mathbf{x})]$ is introduced. This captures reduction in uncertainty in the rest of the space for those dimensions. It also forces the choice of the points that are the most correlated within $\mathcal{V}$. Mutual information $I(\mathcal{A}; \mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B}))$ is then calculated between the already chosen points $\mathcal{A}$ and rest of the space not including the correlated outputs $\mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B})$. Searching for the next sample point $\mathcal{A}^*$ is then equal to maximizing this measure

$$\begin{aligned} \mathcal{A}^* &= \quad \max \ \{I(\mathcal{A}; \mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B}))\} \\ &= \operatorname*{arg\,max}_{\mathcal{A} \subseteq \mathcal{S}: |\mathcal{A}| = k} \{H(\mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B})) - H(\mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B}) | \mathcal{A})\} \end{aligned} \tag{4.4.1}$$

The sampling can then be performed incrementally choosing the next point which gives the maximum increase in Mutual Information (MI). This can be derived as

$$\begin{aligned} \Delta I &= I(\mathcal{A} \cup \mathcal{Y}; \bar{\mathcal{A}}) - I(\mathcal{A}; \bar{\mathcal{B}}) \\ &= H(\mathcal{A} \cup \mathcal{Y}) - H(\mathcal{A} \cup \mathcal{Y} | \bar{\mathcal{A}}) - [H(\mathcal{A}) - H(\mathcal{A} | \bar{\mathcal{B}})] \\ &= H(\mathcal{A} \cup \mathcal{Y}) - H(\mathcal{V} \setminus \mathcal{B}) + H(\bar{\mathcal{A}}) - [H(\mathcal{A}) - H(\mathcal{V} \setminus \mathcal{B}) + H(\bar{\mathcal{B}})] \\ &= H(\mathcal{A} \cup \mathcal{Y}) - H(\mathcal{V}) + H(\bar{\mathcal{A}}) - [H(\mathcal{A}) - H(\mathcal{V}) + H(\bar{\mathcal{A}} \cup \mathcal{Y})] \end{aligned} \tag{4.4.2}$$

and the resultant formulation is

$$\Delta I = H(\mathcal{Y} | \mathcal{A}) - H(\mathcal{Y} | \bar{\mathcal{A}}) \tag{4.4.3}$$

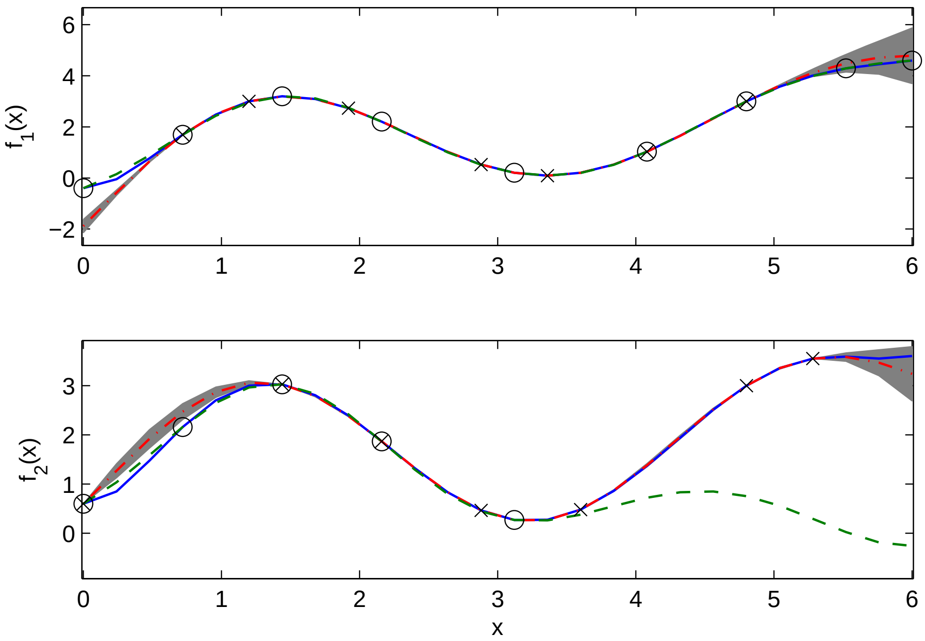


Figure 4.1: The features sampled after 14 iterations using entropy (circles) and MI (crosses) for a multi-output system (solid blue). The DGP approximation on the underlying function using entropy (dash green) and MI (dash-dot red) with prediction uncertainties (gray) are also shown.

where, $\mathcal{Y} = y_n(\mathbf{x})$, $\bar{\mathcal{A}} = \mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B} \cup \mathcal{Y})$ and $\bar{\mathcal{B}} = \mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B})$. Note that equation (4.4.3) forces $\mathcal{Y}$ to be chosen centrally with respect to the unselected locations. Also, the entropy for a Gaussian variable $\mathcal{Y}$ conditioned on a set of variables $\mathcal{A}$ is given by

$$\begin{aligned} H(\mathcal{Y}|\mathcal{A}) &= \frac{1}{2}\log(2\pi e \sigma^2_{\mathcal{Y}|\mathcal{A}}) \\ &= \frac{1}{2}\log \sigma^2_{\mathcal{Y}|\mathcal{A}} + \frac{1}{2}(\log(2\pi)+1) \end{aligned} \tag{4.4.4}$$

This essentially is the entropy at the point of interest and is a monotonic function of the GP variance.

An example of sampling features using this method is shown in Figure 4.1. Here, two underlying functions $f_1(\boldsymbol{X})$ and $f_2(\boldsymbol{X})$ are approximated. It is then compared to the existing entropy based sampling technique [91]. The results are shown after 14 iterations. The MI measure can perform efficient learning of both the functions where entropy based measure is sub-optimal. It is also less prone to sample on the function limits where there would be less information to be learned by placing a feature.

Even though this produces a concise set of informative features, it does so with an initial computational cost of $\mathcal{O}((N(Q-1))^3)$ to compute $H(\mathcal{Y}|\bar{\mathcal{A}})$ when $\mathcal{A} = \emptyset$. In addition, since all the possible positions are evaluated at each iteration, the complexity for selecting k features will be $\mathcal{O}(kh^4)$ where $h = N(Q-1)$. Hence, this is not practical for large n. To solve this problem we adopt a local approximation proved in [40] where $H(\mathcal{Y}|\tilde{\mathcal{J}}) \approx H(\mathcal{Y}|\mathcal{J})$ given $\tilde{\mathcal{J}}$ is a result of removing all elements $\mathcal{X}$ from $\mathcal{J}$ that is $|K(\mathcal{X}, \mathcal{Y})| \leq \varepsilon$ for a small value of ε. It is identified that GP correlations decrease exponentially with the distance between points. Often the variables that are far apart are independent. This gives a bound on the decreasing entropy $H(\mathcal{Y}|\mathcal{J} \setminus \mathcal{X}) - H(\mathcal{Y}|\mathcal{J})$

$$\sigma^2_{\mathcal{Y}|\mathcal{J}\setminus\mathcal{X}} - \sigma^2_{\mathcal{Y}|\mathcal{J}} \leq \frac{K(\mathcal{Y}, \mathcal{X})^2}{\sigma^2_{\mathcal{X}}} \leq \frac{\varepsilon^2}{\sigma^2_{\mathcal{X}}} \tag{4.4.5}$$

by $\varepsilon^2/(\sigma^2\sigma^2_X)$ assuming that each independent Gaussian measurement has an error of at least σ^2. The result is a more efficient way to calculate $H(\mathcal{Y}|\bar{\mathcal{A}})$ at the expense of a small absolute error. The complexity for the operation then reduces to $\mathcal{O}(d^3)$ from $\mathcal{O}(h^3)$ where $d \ll h$. The local approximation was made by selecting d data points using a KD-tree. These local inputs where then used to train the multi-output covariance function given by equation (3.3.8) to approximate $H(\mathcal{Y}|\bar{\mathcal{A}})$. The estimate is substituted into (4.4.3) to evaluate the MI. A feature $\mathcal{A}^*$ is then placed in the location of maximum increase in MI. The result is a compact set of features that maximize prior information to account for output dependency while minimizing redundancy. These can now be used for global approximation of the full DGP at a much lower computational cost.

4.4.2 Local Approximation

Using the global approximations makes it difficult to model very short length-scale phenomena as it would require a dense set of inputs to capture the fast variations. Also the DGP approximation proposed is stationary therefore $|\mathbf{x}_a - \mathbf{x}_b|$ is invariant to translation. Hence, we resort to local experts to account for their own part of space.

This is modeled using a non-stationary local neural network covariance function. The Bayesian interpretation of neural networks is in [55]. Consider a neural network that takes $\mathbf{x}$ as inputs, has one hidden layer with N_H units and then linearly combines the outputs of the hidden units to result the output $j_n(\mathbf{x})$. The function can be written as

$$j_n(\mathbf{x}) = b + \sum_{m=1}^{N_H} \boldsymbol{\kappa}_m \hat{h}(\mathbf{x}, \boldsymbol{\omega}_m) \tag{4.4.6}$$

where $\boldsymbol{\kappa}_m \in \mathbb{R}^D$ are the weights to the hidden unit transfer function $\hat{h}(\mathbf{x}, \boldsymbol{\omega})$ and b is bias of the output. Input-to-hidden weights are $\boldsymbol{\omega}$. The covariance function for a neural network can then be obtained by evaluating $\mathbb{E}_{\boldsymbol{\omega}}[\hat{h}(\mathbf{x}; \boldsymbol{\omega})\hat{h}(\mathbf{x}'; \boldsymbol{\omega})]$ [87]. The resultant covariance is given by

$$\mathrm{cov}^y_{ii}(\mathbf{x}_a, \mathbf{x}_b) = \sigma_n^2 \sin^{-1}\left(\frac{\bar{\beta}_n + 2{\mathbf{x}_a}^T \boldsymbol{\Sigma}_{mn} \mathbf{x}_b}{\sqrt{(1 + \bar{\beta}_n + 2{\mathbf{x}_a}^T \boldsymbol{\Sigma}_{mn} \mathbf{x}_a)(1 + \bar{\beta} + 2{\mathbf{x}_b}^T \boldsymbol{\Sigma}_{mn} \mathbf{x}_b)}}\right) \tag{4.4.7}$$

and hyperparameters of the function are $\boldsymbol{\Sigma}_{mn}$, σ_n and $\bar{\beta}_n$. The value for $\bar{\beta}_n$ is a bias factor. This covariance is much effective in handling discontinuous or rapid changes in data. However, it is a single output covariance and is only used to learn non-stationary behaviour within each output.

Figure 4.2 shows a plot of the neural network covariance function. This example is a single output process with equal number of sample points. It can be observed that the covariance increases with the increase in $|\mathbf{x}_1 - \mathbf{x}_2|$ and reduces when $\mathbf{x}_1$ is approximately equal to $\mathbf{x}_2$. Therefore, a function modeled with this covariance can vary more quickly in some parts of the input space than in the others. Also, notice that for large covariate values of $+\mathbf{x}$ or $-\mathbf{x}$ the covariance approaches a constant value. These observations prove that it is a non-stationary covariance function.

The local approximation uses a similar strategy to [83] where training is performed with the set of all training data, but is applied locally by making a local approximation. The data set comprises of two elements h_{train} training points and h_{eval} evaluation points where $h_{\text{eval}} \gg h_{\text{train}}$. The training points are used to learn the hyperparameters for the GP model and evaluation points together with the training data are used

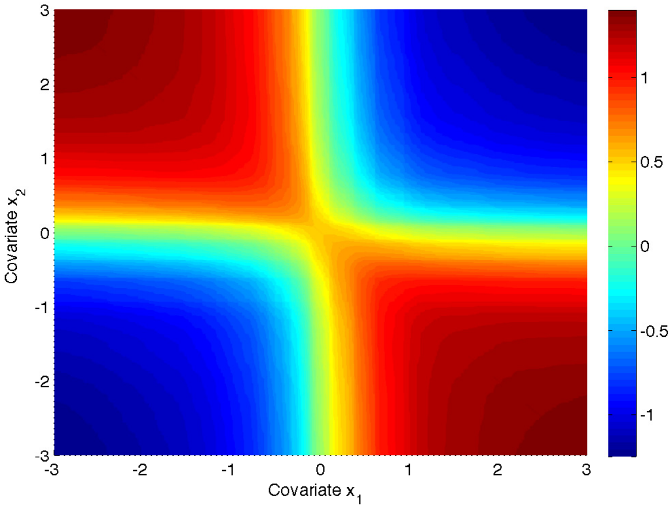


Figure 4.2: Covariance matrix for neural network covariance function. The warm colours (red) represent high covariance and cool colours (blue) represent low covariance.

for inference. The data that consists of $h_{\text{train}} + h_{\text{eval}}$ is stored in a KD-tree structure. When required to perform inference a predefined set of local neighbours d together with the originally learned hyperparameters are used to make the predictions. The outcome is a more locally adaptive GP that is scalable to large scale test flight data sets.

4.4.3 Combined Approximation

The properties of both the local and global GP can be combined. Consider $u_n(\mathbf{x}) = f_n(\mathbf{x}) + j_n(\mathbf{x})$ where the global $f_n(\mathbf{x})$ and local $j_n(\mathbf{x})$ approximation functions are independent. A Gaussian prior can then be placed on both functions $f_n(\mathbf{x})$ and $j_n(\mathbf{x})$ to give them a different covariance function to reflect the belief. The sum of the kernels for these two functions is a kernel as well. The newly formed additive kernel is given by

$$K(\boldsymbol{X}, \boldsymbol{X}) = K_f(\boldsymbol{X}, \boldsymbol{X}) + K_j(\boldsymbol{X}, \boldsymbol{X}) \tag{4.4.8}$$

This construction can be used to add together kernels with different length scales. In addition, note that the sum of two positive definite covariances will always result in a positive definite covariance. Now, given that the sum of two Gaussian variables is a Gaussian the prior for the additive model can be written as

$$p(\boldsymbol{y}|\boldsymbol{X}) = \mathcal{N}(\mathbf{0}, K_f(\boldsymbol{X}, \boldsymbol{X}) + K_j(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{\sigma}^2\mathbf{I}) \tag{4.4.9}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^N$ The marginal distribution and posterior predictive distribution are the same as described in Section 3.3.

The global covariance function $K_f(\boldsymbol{X}, \boldsymbol{X})$ is composed of the multi-output DGP covariance function. On the other hand, the local covariance function $K_j(\boldsymbol{X}, \boldsymbol{X})$ is composed of only independent outputs. Therefore, the components from $K_j(\boldsymbol{X}, \boldsymbol{X})$ are added to the diagonal terms $\mathrm{cov}^y_{ii}(\mathbf{d})$ of the multi-output covariance function in (4.4.8). During implementation, the model $\mathcal{GP}_g$ is first learned from the training data for $\mathbf{K}_f$. Then predictions are made using this model on the n_{train} local inputs to learn an “error model" that captures the difference between target values and the predictions of $\mathcal{GP}_g$. The local approximation model $\mathcal{GP}_l$ is trained from these. The resultant is having one smooth underlying component that primarily captures dependency between outputs and a component that learns a locally concentrated non-stationary structure.

4.4.4 Hyperparameter Optimization

Hyperparameter optimization involves learning the appropriate hyperparameters and noise variance given the observations. Hyperparameters are free parameters of the covariance function. In this case, the parameters of the global kernel function are $\boldsymbol{\Theta}_g = [v_{mn}, \mathbf{A}_{mn}, \boldsymbol{\mu}_{mn}]$ and local kernel function are $\boldsymbol{\Theta}_l = [\boldsymbol{\Sigma}_{mn}, \boldsymbol{\sigma}, \beta_n]$. These parameters are learned by maximizing the log marginal likelihood of the training outputs given the inputs

$$[\boldsymbol{\Theta}, \boldsymbol{\sigma}^2] = \underset{\boldsymbol{\Theta}, \boldsymbol{\sigma}^2}{\arg\max}\{\log(p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\sigma}^2))\} \tag{4.4.10}$$

where,

$$\log(p(\boldsymbol{Y}|\boldsymbol{X})) = -\frac{1}{2}\boldsymbol{Y}^T(K(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\sigma}^2\mathbf{I})^{-1}\boldsymbol{Y} - \frac{1}{2}\log|K(\boldsymbol{X},\boldsymbol{X})+\boldsymbol{\sigma}^2\mathbf{I}| - \frac{\sum_{i=1}^{N} M_i}{2}\log 2\pi. \tag{4.4.11}$$

In GP model training the hyperparameters $\boldsymbol{\Theta}$ and noise variance $\boldsymbol{\sigma}^2$ are optimized. Once these are found, predictions can be made on the posterior outputs $\boldsymbol{u}_{n_*}$ by substituting $\boldsymbol{\Theta}_g$ into (3.3.8), $\boldsymbol{\Theta}_l$ into (4.4.7) and by adding the contribution of these two Gaussians.

4.4.5 Computational Complexity

For N output processes and Q data points let $h = N(Q-1)$. Then training the global GP has an initial complexity of $\mathcal{O}(hd^3)$ to calculate (4.4.3) where d is the chosen number of local neighbours. Next, for each consequent iteration k this calculation requires $\mathcal{O}(kd^4)$ and to evaluate (4.4.1) it requires h comparisons therefore a cost of $\mathcal{O}(kh)$. On the other hand the local GP only has a complexity of $\mathcal{O}(h_{\text{train}}^3 Q)$. Thus, the training process is highly computationally involved than before. Nevertheless, the problem is tractable as it does not scale with a complexity of $\mathcal{O}(NQ^3)$. Plus this does not affect the performance as the model training is performed offline.

For inference, the global GP has a complexity of $\mathcal{O}(z)$ for the predictive mean and $\mathcal{O}(z^2)$ for the predictive variance where z is the final number of points chosen ($z \ll (NQ)$). The local GP has a cost of $\mathcal{O}(d)$ and $\mathcal{O}(d^2)$ for the mean and variance where $d \ll (h_{\text{train}} + h_{\text{eval}})$. In comparison the full GP required a complexity of $\mathcal{O}(NQ)$ for the predictive mean and $\mathcal{O}((NQ)^2)$ for the predictive variance. Thus, it is significantly fast compared to a full GP in making predictions.

4.5 Simulation Testing

In this section the global and local Gaussian process model was implemented and tested to demonstrate its ability to learn aircraft dynamics. This was tested using the AD-1 aircraft simulator (see Section 3.4). Two simulated flights were used here. One to train the model and the other to verify the trained model. For the global approximation, the advantage of using the proposed mutual information criterion for point sampling was demonstrated by comparing against the existing entropy based sampling approach. The combined approximation shows the advantage of incorporating a non-stationary covariance function to learn the local properties. The results were analysed and compared against the simulated response as well as the response from the full DGP.

4.5.1 Flight Model Training and Testing

The simulator experiments in this section were performed with the AD-1 oblique wing flight simulator. A description about the AD-1 aircraft was presented in Section 3.4.1 and the flight simulator details were given in Section 3.4.2. The wing sweep angle was chosen to be 45° for all the experiments because in that setting the flight model contains high level cross coupling within the aerodynamic terms.

For training and testing this section used the same conditions and maneuvers as given in Sections 3.4.4 and 3.4.4 respectively. This gave the opportunity to compare the performance of the approximate GP to the full DGP model that was proposed in Chapter 3. Two main experiments were done here. The first was to test the sampling algorithm for global approximation. The reduction in error on the predicted response was compared with the number of features chosen to model the function. This was then compared to the entropy based sampling approach. The second experiment was to test the learned approximate model on its prediction capability by estimating aerodynamic coefficients for a different simulated flight to the trained one.

The control inputs and the state response for the training maneuver were given in Figure 3.5. It contains a series of doublet control inputs to excite both the lateral and longitudinal dynamics (see Figure 3.5). The coefficient responses result from this maneuver were shown in Figure 3.6 (b). The approximation algorithm uses the whole 30s of this training maneuver and to make the global approximation it sample features (points) that maximize learning across all output dimensions. The test maneuver used 3-2-1-1 control inputs with elevator, rudder and aileron (see Figure 3.8).

To train this GP model, it requires measurements of the aerodynamic forces and moments that the aircraft is experiencing, as well as measurements of the regressors. The force and moment coefficients identified as output states were given by the equation (3.4.1). The regressors are defined by equation (3.4.2). The assumption made on inputs being stationary in Section 3.4.3 was relaxed as the proposed model can learn any non-stationary properties. The global model only used a sub-set of the inputs to learn the function and local model was trained using the complete data set.

4.5.2 Results and Discussion

This section presents the results of the two experiments performed in simulation. The first experiment showed the learning of the global approximate model using a new MI criterion. The second experiment presented with the coefficient estimates obtained for the simulated test flight segment.

Sampling with Mutual Information

This experiment consisted of learning an approximate global GP model. The features for the GP was sampled using MI. It was then compared with the features extracted from the existing entropy sampling technique. The complete duration of the 30s training maneuver was used to sample the feature set. It sampled 250 features across all the dimensions which were determined to sufficient to learn the global approximate function.

Figure 4.3 shows the summary of normalised Root Mean Square (RMS) error over all the dimensions for the training flight segment. The RMS error was calculated for each output and was normalised across all the outputs. The average error across all the six outputs is whats displayed. It showed the error decrement with the increase in number of features was faster with the MI approach. Hence, MI exhibited superior performance in comparison to entropy sampling for increasing set size.

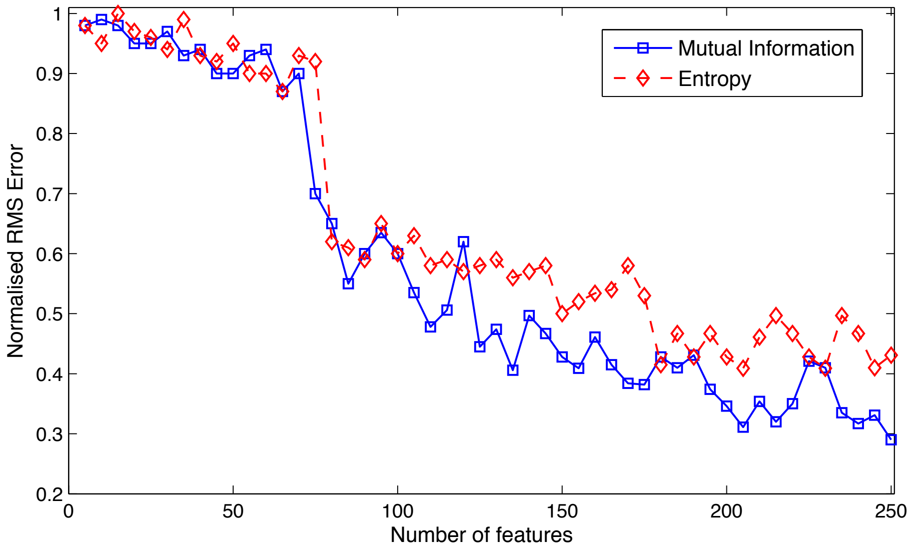


Figure 4.3: Prediction error on the training data during the learning process of the global DGP approximate.

Estimated Aerodynamic Coefficients

For the complete duration of the maneuver both lateral and longitudinal parameters were identified. The estimates were from the global approximate and the combined approximate models. These were then compared to the coefficients obtained from the non-linear simulator (ground truth).

The estimated coefficients and the prediction uncertainties from the combined approximate GP model for the simulated test flight were shown in Figure 4.4. The figure also shows the estimates by only using the global approximate. The model trained for the global Gaussian process approximation at 45° wing sweep was labeled as GA-GP45. The model for the local and global Gaussian process approximation at 45° wing sweep was labelled as LGA-GP45. It should be noted that the global approximate model GA-GP45 contains features that primarily captured information about output correlations. Hence, in some instance it might not capture the underlying response. In particular, this can be seen in the moment coefficients C_l, C_m and C_n. The force response functions C_x and C_z were particularly smooth hence GA-GP45 model was also able capture them with high accuracy. During instances when there was more independent motion GA-GP45 model slightly underperformed.

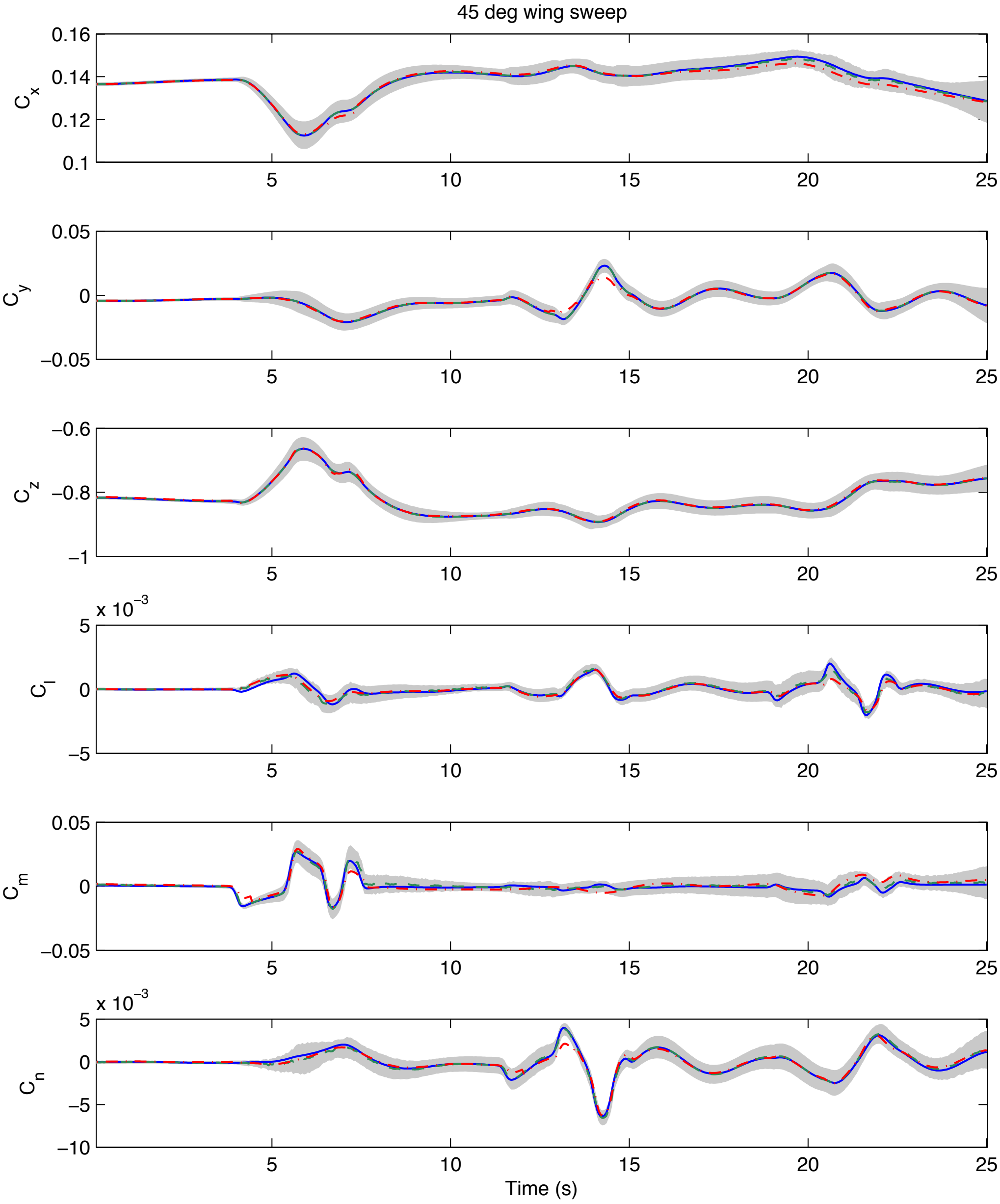


Figure 4.4: AD-1 ground truth (solid blue), global GA-GP45 estimated (dashed-dotted red) and Local and global LGA-GP45 estimated (dashed green) non-dimensional aerodynamic coefficients with prediction uncertainties for LGA-GP45 (gray) at 45° sweep angle.

Table 4.1: AD-1 coefficient estimation errors with GP approximations for 45° wing sweep.

% Error	C_X	C_Y	C_Z	C_l	C_m	C_n
DGP45	0.005	0.036	0.047	25.951	9.520	3.922
GA-GP45	0.028	0.221	0.035	34.187	17.582	7.164
LGA-GP45	0.005	0.040	0.037	28.360	8.172	3.157

For the LGA-GP45 model, the locations where there are less cross coupling between outputs, the predictions are primarily made with the local GP. This is possible due to the local approximate that was used to infer non-linear local properties.

The relative errors on the estimated parameters were calculated by comparing against the coefficients obtained from the simulator for the entire simulated test flight segment. The summary of the results is presented in Table 4.1 together with the results of DGP45, which are the estimated errors using the full DGP that was calculated in Section 3.4.6. The median error on the DGP45 response is 6.58%, on GA-GP45 it is 9.87% and on the LGA-GP45 it is 6.63%. Hence, the LGA-GP45 response is equally a positive response as the DGP45 with a much less computational cost. Also the global approximate model GA-GP45 performed nearly as well as the LGA-GP45 and slightly underperformed when the coefficient response is more independent.

In summary, the model learned by the combined approximation from the training flight was proved to be valid for the entire test flight. It was also shown that by making an approximation the estimates could be made with a much less computational cost without compromising the prediction accuracy.

4.6 Summary and Conclusion

This chapter presented a computationally efficient GP approximation that combines the best of local and global GPs to model a multi-output system. It introduced a mutual information based sampling approach to extract primarily the features that contain information about the output correlations. The global GP captures dependencies between outputs, identifies any coupling between parameters and the local GP captures any correlations within those parameters. Finally, predictions are made with their associated uncertainty which offer a notion of confidence on the learned parameters. Hence, the proposed method can bring forth a more informative flight model that can capture a wide range of the dynamics.

The simulation results indicated the mutual information criteria is better than entropy to sample features from a multi-output GP, both qualitatively and in prediction accuracy. The local GP showed that it can capture fast variations in the data and it was demonstrated that the combined approximation is comparable to the full DGP. Hence, the work can be used to increase the efficiency of DGP models for aircraft system identification.

Chapter 5

Experiments

5.1 Introduction

The final step of system identification process is to test if the identified models have good estimation capability. Experiments were conducted with a real UAV to test the algorithms developed in this thesis. The UAV platform was used for flight data collection and aerodynamic models were learned from these. The measured input data from the sensors were used as inputs to the model to generate the predicted response which was then compared against the measured response. The performance of the learned models were then tested by comparing against the results from different test flights to the trained one.

There were two main experiments conducted to test various aspects of the two algorithms developed in Chapters 3 and 4. The objective of the first experiment was to demonstrate DGP model for learning flight dynamics of a real UAV platform using logged flight data. Two separate flight tests were performed. The data from the first flight test was used to train a DGP model and the data from the second was used to validate the trained model. It also verified the model performance under wind gust in a real flight scenario. The objective of the second experiment was to demonstrate that even after making an approximation on the DGP, it is still possible to estimate the aerodynamic coefficients while maintaining the prediction accuracy. It was also demonstrated that to learn the model for the entire flight envelope learning non-stationary properties is important. Four separate flight tests were performed. The data from the first flight test was used to train an approximate GP model. Other three flights were used to verify the trained model. The extensive tests provided empirical evidence of model robustness.

This chapter provide the details of experimental setup and results. Firstly, the flight test environment and the data collection setup will be described, which were used to test both the algorithms. Secondly, the experiment setup for the full DGP model based estimation will be presented. It includes results from the learned model of the aerodynamic coefficient. Next, how these could be translated into estimating the states is shown. Thirdly, the results from the model identified from local and global Gaussian process approximation algorithm will be presented. Both the training and the test set results are presented and discussed. The goal was to validate the models identified by estimating the aerodynamic response to other test flights. The results were compared to the measured response and checked for the overall performance and robustness.

5.2 Implementation

This section provides an overview of the implementation procedure to learn aerodynamic coefficients from flight testing. The framework can be broken down into three main sections shown in Figure 5.1: flight testing; model training; and predictions through inference.

Flight testing for system identification usually involves designing maneuvers to maximize the information content and collecting flight data. In the next phase the system inputs and outputs are used to learn the hyperparameters for the flight model. Lastly, the chosen DGP models are passed on to estimate the aerodynamic coefficients given the system inputs. The focus of this work is on model training and inference. Hence, it should be noted that the maneuvers implemented may not be optimal in maximiz-

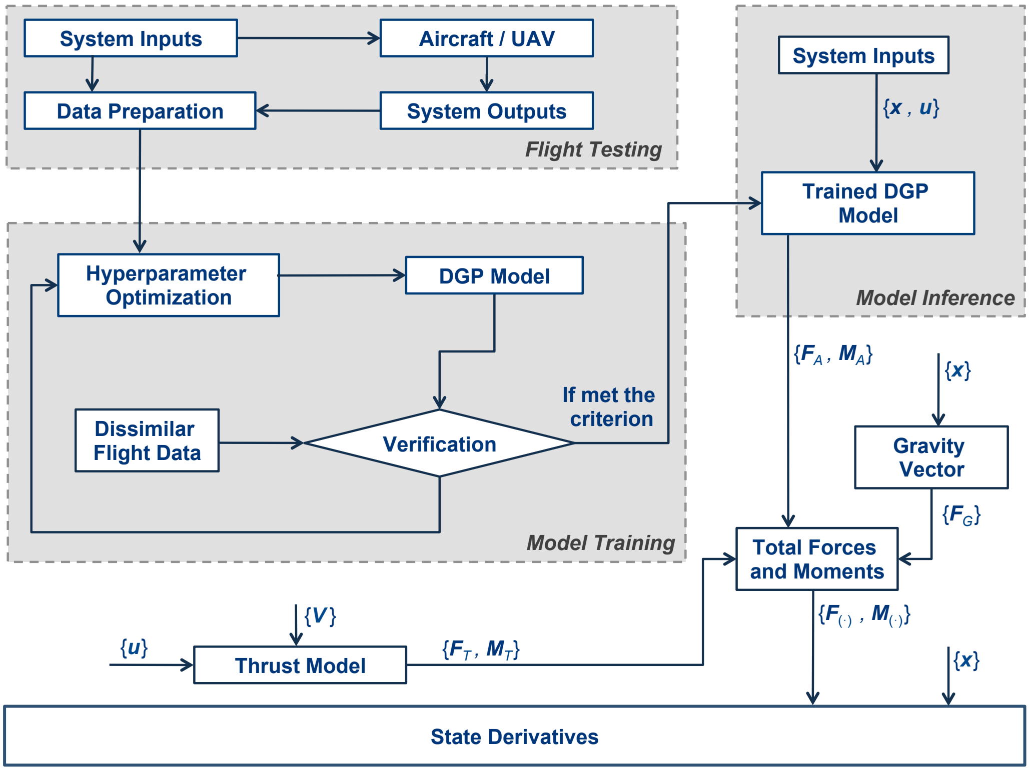


Figure 5.1: Block diagram of the dependent Gaussian Processes aircraft system identification procedure from flight testing, training to model inference.

ing the information content in the collected flight data, although this does not affect the results in using DGP for modeling.

The resultant aerodynamic forces and moments can be calculated using (2.3.21). This is then used along with the gravity vector and the thrust model to calculate the total applied forces and moments on the platform. With the new $\{\boldsymbol{F}_{(\cdot)}, \boldsymbol{M}_{(\cdot)}\}$ estimates and the current state vector of the UAV, state derivatives can be determined. The output can now be used for flight controller design, simulator development and to understand the UAV handling qualities.

5.2.1 Flight Testing

In flight testing for system identification manoeuvres are design to maximize information content in the data. To achieve this, the system modes must be excited such that the sensitivities of the model outputs to the parameters are high and the correlation among the parameters is low [51]. Limiting factors include:

- The measured time series is limited due to desire to maintain equilibrium conditions as well as repeatability. But, also due to finite resources of the UAV such as fuel etc.
- Limitations on excitation of the aircraft due to maximum deflections and rates of controls, pilot response time and aircraft stability.
- Sampling rate of sensors (care must be taken to avoid aliasing or specific resonant frequencies).

There are two approaches for designing inputs for dynamic systems. The first assumes no *a priori* knowledge and so the system must be excited over a broad range of frequencies with nearly constant power for all frequencies. Inputs in this category include frequency sweeps and impulse inputs. The second approach uses *a priori* knowledge to design the inputs to excite a dynamic system response mode. This category includes optimal input designs, along with square wave inputs at or around the estimates of the natural frequencies of the dynamic modes [35].

Before the logged flight data was used for model training and testing it was checked for data compatibility (see Figure 2.2). The data was verified to not have any effects of time delay or any other potential instrumentation issues. This was done by reconstructing the aircraft state response with known rigid-body kinematic equations and comparing the reconstructed response with the measured responses. The summary of rigid-body kinematic equations given in Section 2.3.2 were used to reconstruct the state response of the UAV. Only the data that meet those conditions were used in the experiments conducted in this chapter.

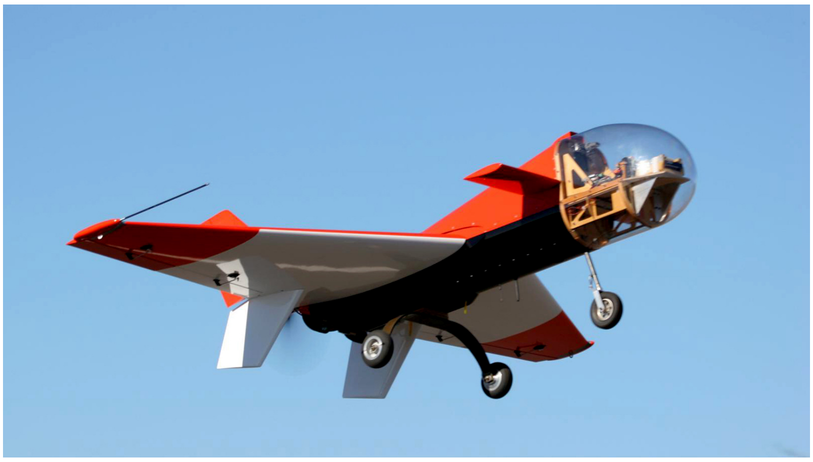


Figure 5.2: The Brumby MkIII UAV.

5.3 Flight Test Procedure

This section describes all the hardware aspects of the system identification procedure, which includes the UAV and its components as well as other aspects of the system which are required to perform a successful mission.

5.3.1 Platform

The platform used for testing is the Brumby MkIII UAV. This UAV is primarily used as a research testbed to demonstrate real-time algorithms for decentralized data fusion [56] and cooperative control strategies involving multiple vehicles [14, 15] at the University of Sydney. It is a delta wing, pusher UAV (see Figure 5.2). Thus, cross-coupled terms are expected to be in the flight model. The platform has a maximum take-off weight of approximately 45 kg, a wing span of 2.8 m and a payload capacity of 13.5 kg. It is capable of flying at $55 - 100$ kts and bank angles of 60 degrees. Its powered by a rear mounted 4-blade propeller engine which has 16 Hp and controlled through rudders and elevons. Maximum flight duration for the UAV is about 40 minutes.

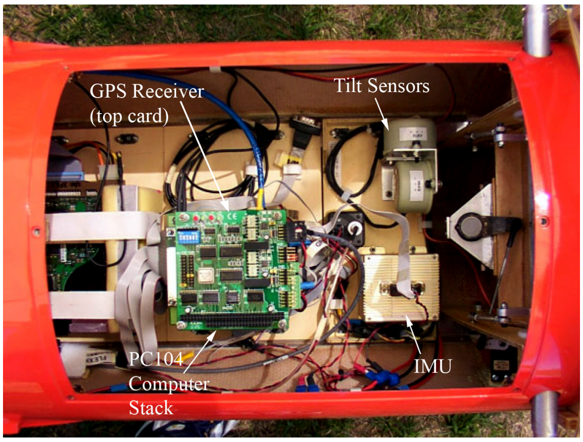


Figure 5.3: Sensors onboard the Brumby MkIII UAV for flight data collection.

5.3.2 Avionics and Sensors

Flight sensors for localisation include an IMU and GPS, and the UAV also has tilt sensors for aligning the navigation system (see Figure 5.3). In addition, it is equipped with a pitot-static system which is attached to the wing to estimate the airspeed, temperature sensors and a sensor for engine RPM.

The on-board navigation system uses differentially corrected GPS to aid the onboard IMU [30, 32]. The data from the IMU is logged at 400 Hz and GPS receivers run at 1 Hz to give an update estimate of the vehicle state (for details see [32]). There are two GPS antennas. One is located in the nose of the platform and the other is located in the wing of the platform. In addition, Differential GPS (DGPS) corrections were transmitted to the UAV from the ground station at 1 Hz to improve the location estimate. The resultant navigation solution is filtered and is logged at 20 Hz. A PC104 computer is used to log this flight data onboard.

The aforementioned navigation solution was used by the Brumby MkIII flight controller [31] to perform over 30 autonomous flights [9], including cooperative missions [15]. Hence, this solution was also used as the ground truth for the tests performed in this chapter as it provided sufficient experimental evidence.

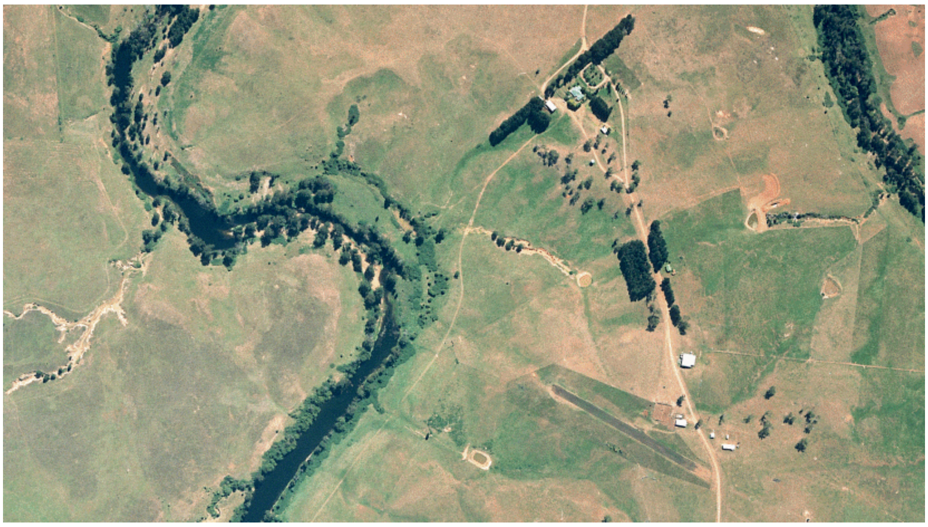


Figure 5.4: Marulan flight test facility. The runway can be seen at the bottom right side of the image.

5.3.3 Test Facility

The flight trials were performed at the University of Sydney flight testing facility in Marulan, New South Wales. Figure 5.4 shows the aerial image around the test area with the runway. It is a 7000 hectare area with a 300 m runway and has ground station facilities for UAV flight tests.

5.3.4 Measurands

The measurands are the measurements of the aerodynamic forces and moments that the aircraft is experiencing, as well as measurements of the regressors. The forces and moments are defined by (3.4.1). The force components were recorded in the body centered frame of reference using the internal IMU. Moments were calculated using the logged rotational rates $\boldsymbol{\omega}_b$, change in rotational rates $\dot{\boldsymbol{\omega}}_b$ and inertia of the platform.

The regressors are defined by (3.4.2). These were logged using the sensors onboard. In addition to the regressors listed in Section 3.4.3, the throttle input δ_T was used to complete the state (3.4.2). This was because during the experiments slight variations to the throttle were made to maintain a stable flight. The inputs from the elevons were separated into aileron and elevator commands based on parity. Low-pass filtered outputs of these regressors were used for training.

The angle of attack (α) and sideslip angle (β) measures were not directly observed using a sensor. Therefore, an approximation was made with

$$\alpha = \tan^{-1}(\frac{w}{V}) \tag{5.3.1}$$

$$\beta = \sin^{-1}(\frac{v}{V}) \tag{5.3.2}$$

where the true velocity is denoted by V, downwards velocity component is w and sideways velocity component is v.

5.4 Experiment 1: Parameter Estimation with DGPs

This experiment was done to test the performance of dependent Gaussian processes to learn UAV stability and control derivatives. It tested the algorithm that was developed in Chapter 3. This experiment include two test flights, one to collect flight data to train the model and second to test the learned model. Details on the measurands for the DGP and flight test procedure is presented. The experiment results show the estimated aerodynamic coefficients as well as the lateral states for the test flight. These were analysed and compared to the measured response. A summary of the test flights for this experiment are presented in Table 5.1.

Table 5.1: Brumby MkIII test flight summary for Experiment 1.

Flight Number	AGL Altitude (m)	Flight Duration (s)	Average Mass (kg)	Average Airspeed (knots)	Purpose
Flight 1	100	250	43.92	78.8	Training
Flight 2	130	1380	44.04	67.2	Testing

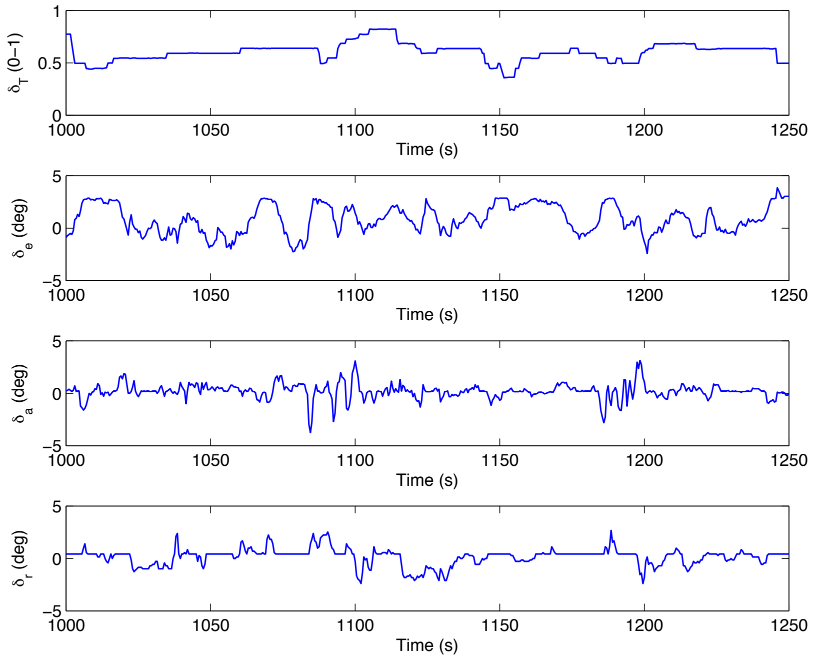
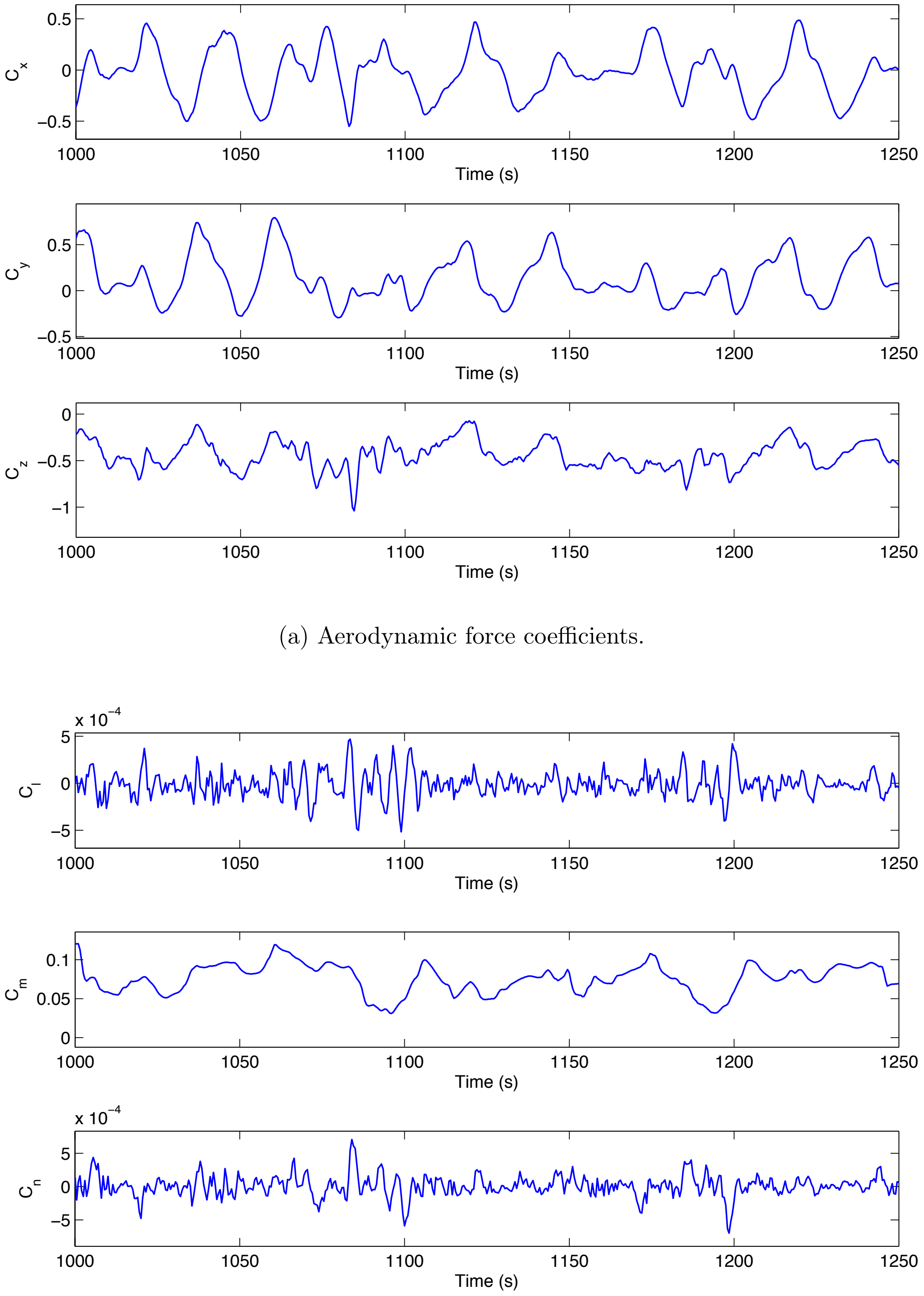


Figure 5.5: Control inputs used on the Brumby MkIII Flight 1.

5.4.1 Flight 1: DGP Model Training

For this initial test, the goal was to learn the lateral dynamics. Also, input design assumed no *a priori* knowledge about the dynamic response of the platform. Hence, control inputs were applied to excite the system over a broad frequency range. This involved the UAV performing lateral maneuvers with aileron and rudder. Elevator inputs were applied to suppress off-axis response and to maintain the UAV flight condition. The variety in shape and frequency of the inputs were to enhance the information content in the data (see Section 9.3 in [35]). A limited control space was explored due to high responsiveness of the platform. The inputs used for training are shown in Figure 5.5 and the resultant aerodynamic coefficients are in Figure 5.6. The training flight was performed at an altitude of 100 m Above Ground Level (AGL). The platform had an average mass of 43.92 kg and maintained an average air speed of 78.8 knots. Training data for the flight was sampled at 10 Hz and the measured responses were tested if they have the sufficient signal-to-noise ratio to capture the dynamics of the platform.



(a) Aerodynamic force coefficients.

(b) Aerodynamic moment coefficients.

Figure 5.6: Flight 1: The measured non-dimensional aerodynamic coefficients for the Brumby MkIII Flight 1.

Of these experimental results, the sample points for training is based on how far away the input points are from the already chosen training samples. Since the posterior variance estimates are a measure of this it can be used to quantify where the next training inputs are to be sampled. This has previously been used for GP point sampling under maximizing differential entropy in [40, 41]. This again is an advantage above the traditional methods in its ability to quantify and inform when outside the bounds of the trained model. Of the training maneuver 7515 points were chosen to learn the hyperparameters and the noise in the output processes by maximizing the log marginal likelihood in equation (3.3.15).

5.4.2 Flight 2: DGP Model Testing

In Flight 2 maneuvers were designed to test the identified lateral dynamics of the system. This involved testing over a complex flight path with lateral orbit type maneuvers as shown in Figure 5.8. The UAV performing the flight test maneuvers are shown in Figure 5.7. This flight had a total flight time of 23 minutes, an average mass of 44.037 kg and maintained an average air speed of 67.2 kts. The tests were performed at an altitude of 130 m AGL. Aerodynamic coefficients were estimated for the whole flight using the trained DGP model. The estimates and the results from this test are presented in the following sub-section.

5.4.3 Experiment 1: Results and Discussion

The estimated coefficients from Flight 2 were compared against those measured coefficients for the entire test flight (Figure 5.8). These measured parameters were calculated using equations (2.3.10) to (2.3.15) and the navigation solution which was used by the Brumby MkIII flight controller. In addition, prior to using the data a compatibility check was performed with kinematic analysis [29]. This ensured the data used for training and testing were consistent and error free.

Figure 5.7: The Brumby MkIII UAV performing flight test maneuvers.

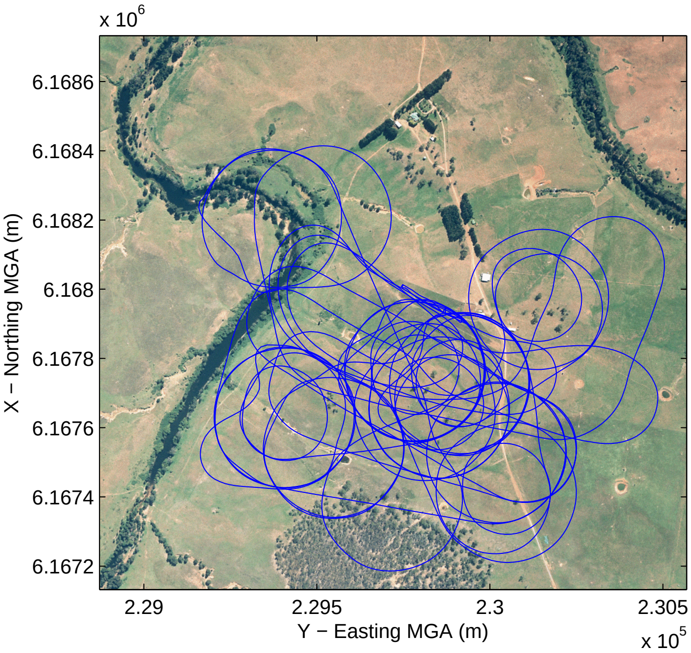


Figure 5.8: Top view of the flight path over the Marulan flight test facility for Brumby MkIII Flight 2.

Estimated Aerodynamic Coefficients

The estimated coefficients and the prediction uncertainties for a segment of the test flight are shown in Figure 5.9. For the complete duration of the flight the lateral

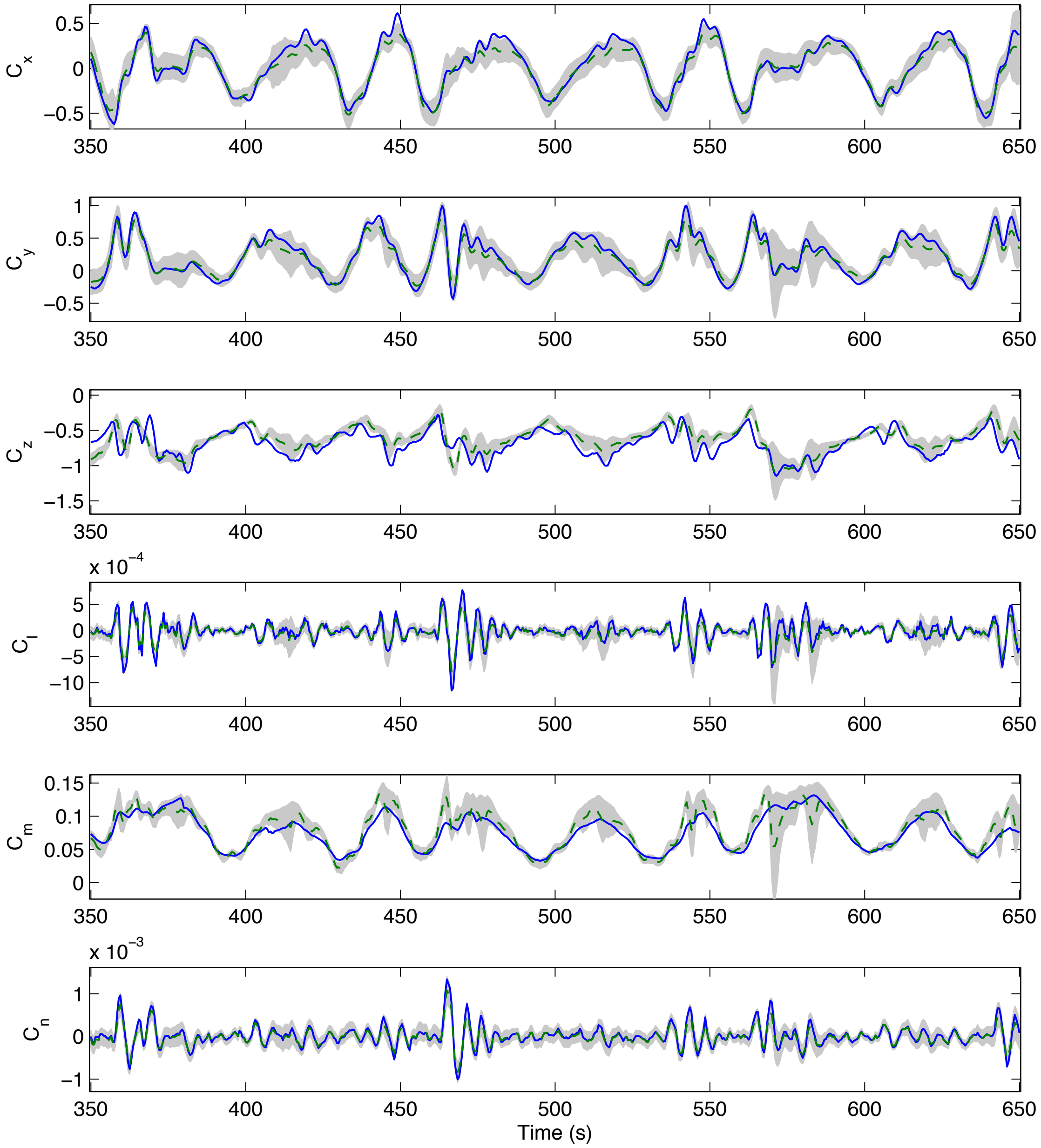


Figure 5.9: Estimated (dashed green) and measured (solid blue) non-dimensional aerodynamic coefficients with prediction uncertainties (gray) for the Brumby MkIII Flight 2.

Table 5.2: Experiment 1 coefficient estimation errors.

% Error	C_X	C_Y	C_Z	C_l	C_m	C_n
DGP	22.758	20.348	13.494	26.573	10.721	25.251

parameters estimated are nearly identical to the measured. Longitudinal estimates are inferred through learning any coupling between the lateral modes. Hence, the error on those parameters are higher which can be noticed specially in the pitching moment coefficient (C_m). This result can be further improved by designing training test flight maneuvers to excite the longitudinal dynamics. Also, using the learned knowledge about the system, optimal inputs can be designed so that the data contains more information.

Figure 5.9 shows that even in the regions of high uncertainty the estimates are fairly close to the measured. This again is due to the dependency learned between parameters, thus making the model valid for a broader flight envelope. The relative errors on the estimated parameters were calculated by comparing against the measured. The results are presented in Table 5.2. The error estimates are relatively high compared to the simulation results obtained for AD-1 in Section 3.4. It was found that the median error on the force coefficients is ±18.87% and on the moment coefficients is ±20.85% of the measured. This is given that the UAV was experiencing wind gusts of up to 11 kts, and there are propagated errors from the sensor noise. Also, the coarsely known mass and inertia properties of the platform and any unmodeled characteristics are captured in the response.

In summary, the model parameters identified from the training flight was proved to be valid for the entire test flight. Next, these coefficient estimates were used to predict the system states. Only the lateral directional dynamics were analyzed since the training maneuvers were designed to extract these.

Estimated Lateral States

The lateral states considered here are dependent on the aerodynamic coefficients. These include the side velocity (v), roll rate (p) and yaw rate (r), which are dependent on C_y, C_l and C_n respectively. The estimated lateral aerodynamic parameters along with the thrust model and the gravity vector were used to find the total applied forces and moments (see Section 2.3.2). These were then used to calculate the current state derivatives using the equations from Newton's second law of motion in translation and rotational forms (see Section 2.3.2). In order to calculate the states, the derivatives were integrated using the 4th-order Runge-Kutta method. The results are shown in Figure 5.10.

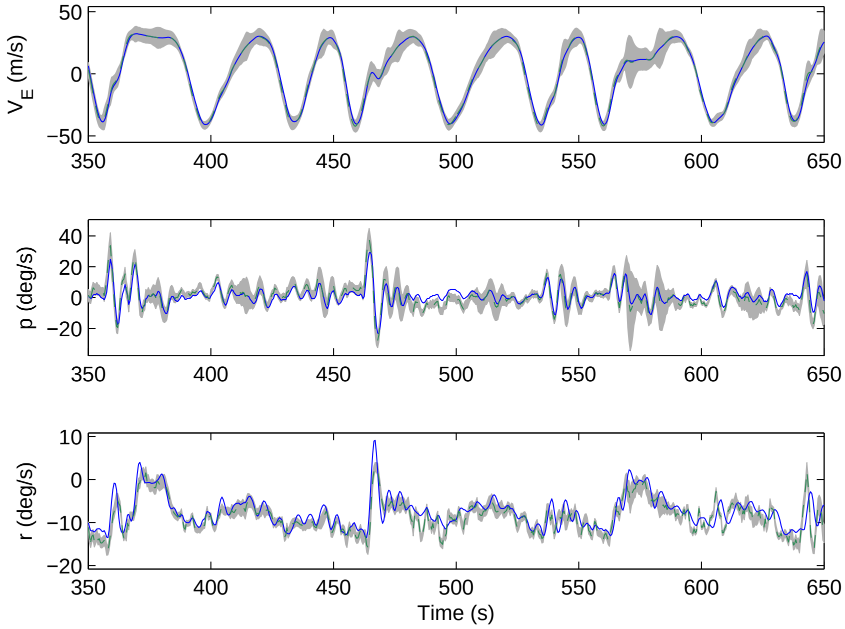


Figure 5.10: Estimated (dashed green) and measured (solid blue) lateral states with prediction uncertainties (gray) for the Brumby MkIII Flight 2.

The estimated outputs were compared against the measured navigation solution. It can be seen in Figure 5.10 that the propagated error from the coefficients to the system states are negligible. The median error for the estimated East velocity was calculated to be 0.076 m/s, for roll rate it was 0.4 deg/s and 0.45 deg/s for yaw rate - a very promising result. These estimates could be improved in the future by designing maneuvers to encapsulate the dynamics in unobserved regions.

5.5 Experiment 2: Parameter Estimation with GP Approximations

This experiment was done to test the performance of multi-output local and global Gaussian process approximations for UAV stability and control derivative estimation. It tests the algorithm that was developed in Chapter 4. Four separate test flights were performed. The data from the first flight test was used to train the approximate GP model and the data from the other flights were used to verify the trained model. A summary of all the test flights for this experiment are provided in Table 5.3. This section details the model training and flight test procedure. It also shows experiment results of the estimated aerodynamic coefficients. These were analysed and compared to the measured response. The advantage of modeling using a combined approximation is demonstrated.

Table 5.3: Brumby MkIII test flight summary for Experiment 2.

Flight Number	AGL Altitude (m)	Flight Duration (s)	Average Mass (kg)	Average Airspeed (knots)	Purpose
Flight 3	100-200	950	43.92	72.50	Training
Flight 4	100-180	1300	44.05	78.01	Testing
Flight 5	120-140	650	44.04	66.07	Testing
Flight 6	100-225	1300	44.04	72.43	Testing

5.5.1 Flight 3: Approximate GP Model Training

The training flight contained maneuvers that excite both lateral and longitudinal dynamics. To learn the lateral dynamics (C_Y, C_l, C_n) the UAV performed orbit type maneuvers primarily with aileron and rudder inputs. To learn the longitudinal dynamics (C_X, C_Z, C_m) maneuvers were preformed with primarily the elevator inputs.

The training flight was performed at two different altitudes of 100 m and 200 m Above Ground Level (AGL). The flight maintained an average air speed of 72.5 knots. Training data for the flight was sampled at 10 Hz, which was determined to be sufficient to capture the dynamics of the platform. The total flight time was 950s of which only 250s of data was used for training the global and local GP. The rest of the data was used as evaluation points (n_{eval}) in the local GP.

5.5.2 Flights 4 - 6: Approximate GP Model Testing

The Flights 4 - 6 were designed to test both the lateral and longitudinal dynamics of the system: Flight 4 was flown for 1300s at 100 - 180 m AGL with primarily lateral maneuvers; Flight 5 was flown for 650s at 120 - 140 m AGL again with primarily lateral maneuvers; Flight 6 was flown for 1300s at 100 - 225 m AGL with both lateral and longitudinal maneuvers. The trials were performed on different days, hence experienced wide range of weather conditions in terms of mean wind and wind gusts. The mass of the platform was also slightly different in each flight due to the varied amounts of fuel carried. Aerodynamic coefficients were estimated for all of these flights.

5.5.3 Experiment 2: Results and Discussion

The goal of this experiment was to learn the lateral and longitudinal dynamics. The estimated coefficients were then compared against those measured coefficients for the entire test flight. These measured parameters were calculated using the data obtained from the navigation solution which was used by the Brumby MkIII flight controller [32]. In addition, prior to using the data a compatibility check was performed with kinematic analysis [35]. This ensured the data used for training and testing are consistent and error free. The following subsection details the results obtained.

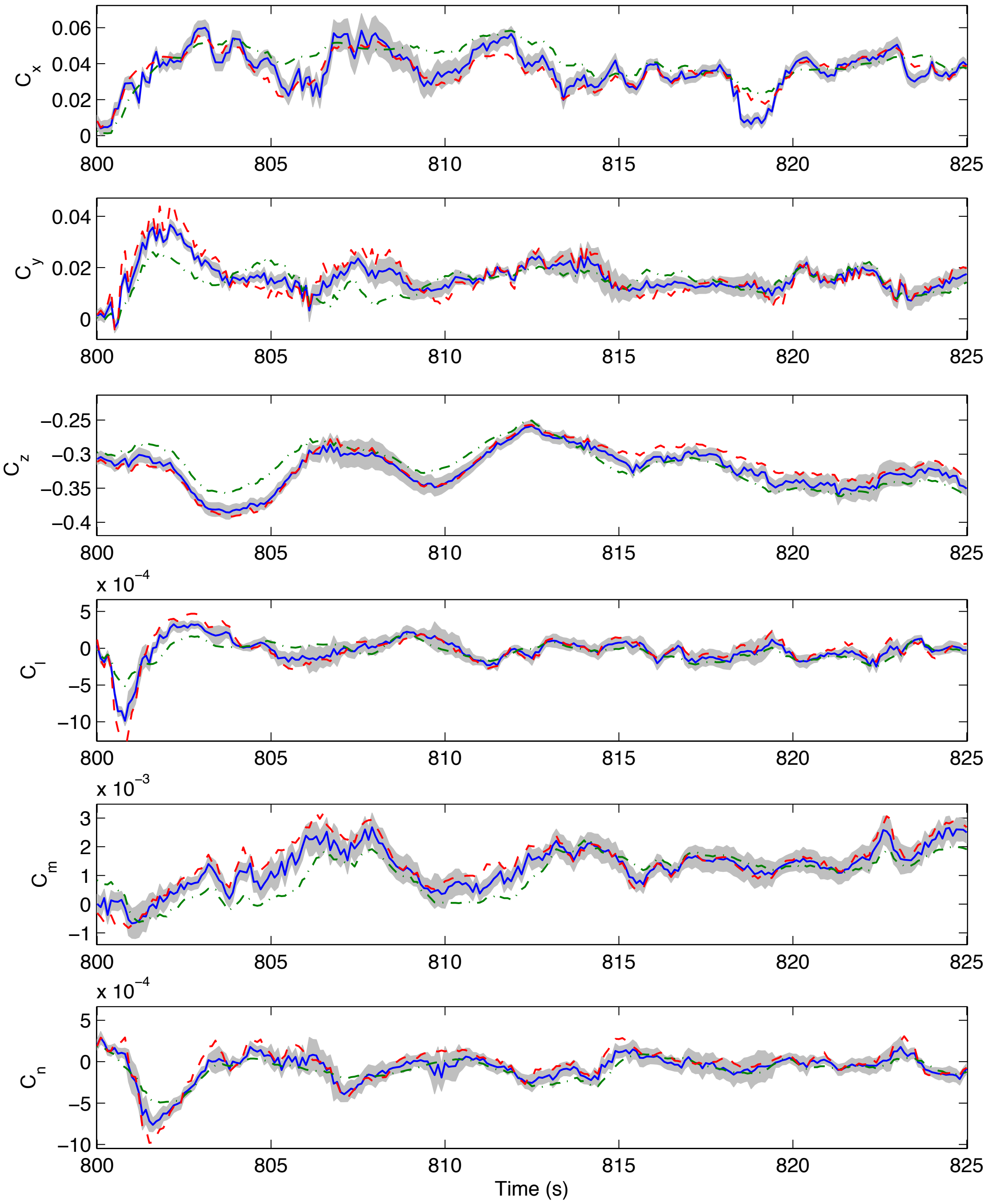


Figure 5.11: Local and global GP estimated (solid blue), global GP estimated (dash-dot green) and measured (dash red) non-dimensional aerodynamic coefficients with prediction uncertainties for the combined approximation (gray) for the Brumby MkIII test Flight 2.

Table 5.4: Experiment 2 coefficient estimation errors.

% Error	C_X	C_Y	C_Z	C_l	C_m	C_n
Flight 4	19.195	23.278	18.365	23.364	22.265	28.173
Flight 5	20.792	24.733	19.925	28.432	19.891	23.714
Flight 6	17.127	19.847	15.174	24.691	17.730	26.245

Estimated Aerodynamic Coefficients

For the complete duration of the flights both lateral and longitudinal parameters were identified. The estimated coefficients and the prediction uncertainties for a segment of test Flight 4 are shown in Figure 5.11. It also shows the estimates by only using the approximated global GP. This is to demonstrate the benefit of utilizing a combined approximation. What is captured by the additional layer of local GP can be seen. The approximated global GP is a much slower moving function and does not capture the fast variations in data. The additional layer of non-stationary covariance function compensated for this gap.

The relative errors on the estimated parameters were calculated by comparing against the measured. The results are presented in Table 5.4 for all the flights. It was found that the median error on the force coefficients is ±19.83% and on the moment coefficients is ±23.83% of the measured. This is comparable to the estimates from the full DGP (see Table 5.2). The error is contributed to by wing gust (8 - 15 kts) and propagated errors from sensor noise. The coarsely known mass and inertia properties of the platform and any unmodeled characteristics are also captured in the response. It is also observed that moment coefficients contained the most error. This could be minimized by designing specific flight test maneuvers to excite those dynamics.

In addition to the predictions made the associated uncertainty can provide a notion of confidence on the estimates. In some instances even in the regions of high uncertainty the estimates are fairly close to the measured. This is due to the dependency learned between parameters from the global approximate. Likewise if there is less cross coupling between outputs inference through global GP would approach zero and the predictions would be primarily made using the local model. This flexibility enabled the model to be valid for a large spectrum of the flight envelope. In summary, the model parameters identified from the training flight was proved to be valid for the entire test flight.

5.6 Summary and Conclusion

This chapter demonstrated the use of Gaussian processes to learn aerodynamic coefficients to model UAV flight dynamics. It was demonstrated through training and extensive testing with real flight data from the Brumby MkIII UAV. The results confirmed that it is possible to use this technique to construct an accurate model for a UAV. These models are more informative and this approach can reduce long-term costs in flight testing. In particular it is useful when identifying parameters for a UAV with high level of cross coupling such as a platform with a delta wing or an oblique wing. In addition, it was verified that DGP models can handle wind gusts experienced in real flight scenarios.

Apart from estimating accurate parameters and model robustness for unmodeled disturbances, the next most useful outcome of the proposed approach is the estimates of uncertainty of the predictions. These can be used as a verification measure to validate the model's usefulness. Lack of confidence in the model predictions can serve as the grounds to learn from new information.

Next, it was demonstrated that flight models can be constructed at a much lower computational cost while retaining the properties that were gained using the full DGP. In the approximation model, the global GP captures dependencies between outputs, identifies any coupling between parameters and the local GP captures any correlations within those parameters. Finally, the predictions are made with an associated uncertainty which gives a notion of confidence on the learned parameters. Hence, the proposed method can bring forth a more informative flight model that can capture a wide range of the dynamics. The consistency in predictions across various maneuvers confirmed that the parameters learned were a accurate representation of the system.

By having the ability to learn with large flight data sets, it can capture wide range of properties in the data.

Chapter 6

Conclusions

This thesis set out to investigate new ways of addressing fundamental challenges in the area of UAV system identification.

System identification for a UAV involves flight testing to collect observation data in order to accurately model the flight dynamics. These models map the platform's control inputs to its dynamic response. They can be used in modern flight controller design, simulator development and to understand the UAV handling qualities. Previous work has been done on developing models using both parametric and non-parametric approaches. The existing parametric approaches require a model structure to regress the observations. This may only be partially known and in some instances may not be available *a priori*. One example of this is developing a model for a new UAV platform without any prior empirical data from an aircraft catalogue such as the DATCOM [21]. In such cases, the modeling process is labor intensive and requires coarse approximations to be made. A limited model structure will restrict its applicability to a narrow band of the flight envelope. Such restrictions can be alleviated by using a non-parametric system identification technique. However, the existing approaches are either limited to operating only within a trained trajectory or do not provide a complete representation of the system.

Another primary challenge in parameter identification is when a model is to be learned that has a high level of cross-coupling between the longitudinal and lateral dynamics.

Examples of this behavior are present in platforms such as a delta wing, an oblique wing or a hammerhead configuration. The goal of this work was to address these challenges in constructing an aerodynamic model from flight testing. Finally, a note on how to use the proposed flight models for flight control is presented.

6.1 Summary of Contributions

The three main chapters of this thesis contain the primary contributions of this work, reviewed below:

- Chapter 3 presented a supervised learning algorithm to learn UAV flight dynamics. It showed how the aerodynamic stability and control derivatives can be modeled using dependent Gaussian processes. We performed an in-depth analysis to identify parameters for the highly coupled AD-1 oblique wing aircraft in simulation. The proposed method does not require *a prior* knowledge of the model structure. We showed how the non-parametric nature of the approach enabled it to capture a wide range of dynamics compared to a parametric method. In addition, we demonstrated the advantage of being able to capture dependencies through cross coupling terms in the covariance function. It enabled the models to be applicable to a broader flight envelope compared to classical system identification methods. By not having to know a prior model structure and its ability to capture output correlations, this method can also be effective for the problem of rotorcraft system identification. Furthermore, the model has the inherent capability to handle noise and biased data. We showed its robustness to unmodeled disturbances. Likewise, the predictions from the model come with uncertainty estimates, which can be used in maneuver design for system identification and in flight controller design. Finally, the method allows the integration of known knowledge about aircraft dynamics. For instance, the model incorporates existing knowledge about applied forces and moments, such as the output from an engine thrust model and the known structural properties of the platform.

- Chapter 4 extended the dependent Gaussian process model for system identification in two ways. First by improving its scalability to learn from large test flight data sets and second by capturing non-stationary properties in aerodynamics. The result is a computationally efficient Gaussian process approximation that combines the best of local and global GPs to model a multi-output system. It uses an additive Gaussian process model that combines both short and long length-scale phenomena for a multi-output GP. By utilizing a combined approximation we were able to learn a broad range of properties in the underlying data. The global GP captures dependencies between outputs and identifies any coupling between parameters. On the other hand, the local GP captures any correlations within those parameters and learns any non-stationary properties. Most of the desired attributes of the full DGP was retained while reducing the computational cost. The work also introduced a new sampling method for DGPs based on a mutual information criterion. It can sample features that primarily contain information about output correlations. Hence, the model requires less data points to describe a dependent multi-output data set. Finally, this was tested with the AD-1 oblique wing flight simulator. Improvement in learning the model with less number of points was shown in comparison to traditional entropy based sampling.

- Chapter 5 presented experimental results from the algorithms developed in Chapters 3 and 4. A Brumby MkIII UAV was used for collecting the flight data for model training and testing. The hardware requirements which includes specifications of the UAV, its avionics and sensors that were required to collect the flight data were presented. The measured input data from the sensors were used as inputs to train the model. The estimation capability of the full DGP model was first tested by comparing the predicted values to the measured response. A comparison was also performed here to show any improvements against the least squares estimator. Next, the results from local and global GP approximation algorithm were presented. Analysis of the results was performed by comparing the measured response to the response generated by using only the global approximation technique. In addition, it was presented how the learned coefficient estimates can be transformed into system states.

6.2 A Note on GP Models for Control

With the proposed approximate GP model, the complete set of state derivatives can be recovered as opposed to [4, 10, 90]. It also is not entirely a black-box prediction model compared to NN. This 6-DOF model can then be linearized at each time step to construct the state and control matrices. Following this, the matrices can be passed on to a control system to calculate the required compensation.

However, it is important to note that the parameters obtained through this method may have less physical meaning as opposed to traditional techniques whereby one can obtain derivative parameters such as C_{m_α}, which dictates longitudinal stability. Hence, the control applicability is limited to modern control approaches such as Model Predictive Control (MPC). Nevertheless, uncertainty estimates obtained can be used to aid the control decisions [43, 44]. They can be integrated into the flight control system or a path planner, which is an added advantage over the traditional techniques. For instance, MPC could change prediction horizon and prediction time step based on the model uncertainty.

This model could be used in the front end of the control design process. It can be particularly useful when developing unconventional UAV designs such as the P.1HH HammerHead UAV [3] or the T-Wing UAV [74] to give an initial estimate of the flight dynamic performance.

Lastly, for real time application the computational cost to obtain predictive mean and variance (see Section 4.4.5) must be faster than the update requirement.

6.3 Future Work

There is a wide scope in future directions to pursue. The models developed with the proposed techniques can be used to construct high fidelity flight simulators, and system outputs can be used to analyse stability, control and handling qualities of the platform. The modern flight control systems rely on feedback of the state vector. This would require the flight model to estimate the state faster than the dominant dynamic mode. For many UAVs, the frequencies of the rigid-body dynamic modes are below 2 Hz (see Section 9.1 [35]). This is faster than the Nyquist frequency and ensures that there is no aliasing of signal frequencies. Hence, the model can effectively capture the important dynamics of the fast modes, such as the short period mode and roll mode. The results can be passed onto a controller to calculate the optimal control inputs required. However, some improvements are still required before using the proposed model to provide state estimates for an on-board flight control systems.

Firstly, the cost to obtain predictive mean and variance must be faster than the update requirement. Hence, the number of points to be selected can be limited. Future work will apply this restriction to the GP approximation algorithm and test its performance. This could be formulated as a cost function and apply when constructing the local and global model presented in Section 4.4.

Currently, to perform local inference the training points are stored in a KD-tree structure and perform inference with a predefined set of local neighbours. This might not necessarily result in a smooth response as local GPs have discontinuities if the training inputs are not sampled in a uniform manner. Instead of using a predefined set of spatially closest training data, the space of the training data can be partitioned using the knowledge of vehicle dynamics. By performing dynamic based partitioning piecewise stationarity can be guaranteed. This would improve the smoothness of the response for the combined approximation.

Future research could also look at extending the models capability to be able to construct a complete flight model for the entire envelope. This involves designing flight maneuvers at different dynamic pressures (altitudes) as well as at different speeds. To construct a complete model, future work should look into ways of using uncertainty on the predicted estimates to design maneuvers to improve the prediction accuracy on the stability and control derivatives.

The inherent capability of the proposed model can capture dependency between outputs and covariance between parameters. This means the problem of system identification does not have to be restricted to solve in the body axis reference frame. It can also be solved in the Earth axes or the inertial reference frame (see Section 2.3.1) where all the body body axis aerodynamic terms are heavily coupled. This would enable the solution to be directly used by a trajectory planner without the need to go through the co-ordinate transformation step.

Bibliography

[1] Parameter estimation techniques and applications in aircraft flight testing. NASA Technical Note D-7647, Washington, D. C., April 1974.

[2] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Proceedings of the 19th Conference on Advances in Neural Information Processing Systems*, 2006.

[3] Piaggio Aero. The P.1HH HammerHead Unmanned Aerial System. URL: http://p1hh.piaggioaero.com/. 2013.

[4] M. J. Allen and R. P. Dibley. Modeling aircraft wing loads from flight data using neural networks. NASA Technical Paper H-2546, Dryden Flight Research Center, 2003.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*, volume 4. Springer, New York, 2006.

[6] E. Bonilla, K. M. Chai, and C. Williams. Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems*, 2007.

[7] P. Boyle. *Gaussian processes for regression and optimisation.* PhD thesis, Victoria University of Wellington, 2006.

[8] P. Boyle and M. Frean. Dependent Gaussian processes. In *Proceedings of the 17th Conference on Advances in Neural Information Processing Systems*, pages 217–224, 2005.

[9] M. Bryson and S. Sukkarieh. Building a robust implementation of bearing-only inertial SLAM for a UAV. *Journal of Field Robotics*, 24(1-2):113–143, 2007.

[10] J. J. Burken, C. E. Hanson, J. A. Lee, and J. T. Kaneshige. Flight test comparison of different adaptive augmentations of fault tolerant control laws for a modified F-15 aircraft. In *AIAA Unmanned...Unlimited Conference*, Seattle, April 2009.

[11] T. G. Butler and D. Michel. *NASTRAN: A summary of the functions and capabilities of the NASA structural analysis computer system.* Scientific and Technical Information Office, NASA, 1971.

[12] A. Coates, P. Abbeel, and A. Y. Ng. Learning for control from multiple demonstrations. In *Proceedings of the 25th International Conference on Machine Learning*, pages 144–151, 2008.

[13] J. D. Colbourne and M. B. Tischler. Flight control design for an unmanned rotorcraft program with a rapid development schedule. In *Proceedings of the American Helicopter Society 57th Annual Forum*, May 2001.

[14] D. Cole. *A Cooperative UAS Architecture for Information-Theoretic Search and Track.* PhD thesis, The University of Sydney, 2009.

[15] D. Cole, A. Goktogan, and S. Sukkarieh. The demonstration of a cooperative control architecture for UAV teams. In *Experimental Robotics*, pages 501–510. Springer, 2008.

[16] N. A. C. Cressie. *Statistics for Spatial Data, revised edition*, volume 928. Wiley, New York, 1993.

[17] L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

[18] R. Curry and A. Sim. In-flight total forces, moments, and static aeroelastic characteristics of an oblique-wing research airplane. NASA Technical Paper 2284, Ames Research Center, 1984.

[19] R. E. Curry and A. G. Sim. Unique flight characteristics of the AD-1 oblique-wing research airplane. *Journal of Aircraft*, 20(6):564–568, 1983.

[20] J. Downs, R. Prentice, S. Alzell, A. Besachio, C. M. Ivler, M. B. Tischler, and M. H. Mansur. Control system development and flight test experience with the MQ-8B fire scout vertical take-off unmanned aerial vehicle (VTUAV). In *Annual Forum Proceedings-American Helicopter Society*, volume 63, 2007.

[21] R. D. Fink et al. USAF stability and control DATCOM. *Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio*, 1975.

[22] P. Goovaerts. *Geostatistics for natural resources evaluation.* Oxford University Press, 1997.

[23] G. Gregorcic and G. Lightbody. Gaussian processes for modelling of dynamic non-linear systems. In *Proceedings of the Irish Signals and Systems Conference*, pages 141–147, 2002.

[24] D. Grimes, R. Chalodhorn, and R. Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Proceedings of Robotics: Science and Systems*, 2006.

[25] N. K. Gupta and W. E. Hall, Jr. Input design for identification of aircraft stability and control derivatives. NASA CR 2493, 1975.

[26] P. Hemakumara and S. Sukkarieh. Non-parametric UAV system identification with dependent Gaussian processes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4435–4441, May 2011.

[27] P. Hemakumara and S. Sukkarieh. Learning UAV stability and control derivatives using Gaussian processes. *IEEE Transactions on Robotics*, 29(4): 813–824, August 2013.

[28] P. Hemakumara and S. Sukkarieh. UAV parameter estimation with multi-output local and global Gaussian process approximations. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5382–5388, May 2013.

[29] R. V. Jategaonkar. *Flight Vehicle System Identification - A Time Domain Methodology*, volume 216. American Institute of Aeronautics and Astronautics, 2006.

[30] J. Kim. *Autonomous Navigation for Airborne Applications*. PhD thesis, The University of Sydney, 2004.

[31] J. H. Kim and S. Sukkarieh. Airborne simultaneous localisation and map building. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 406–411, 2003.

[32] J. H. Kim, S. Sukkarieh, and S. Wishart. Real-time navigation, guidance, and control of a UAV using low-cost sensors. In *Field and Service Robotics*, pages 299–309. Springer, 2006.

[33] V. Klein. Estimation of aircraft aerodynamic parameters from flight data. *Progress in Aerospace Sciences*, 26(1):1–77, 1989.

[34] V. Klein. Aircraft parameter estimation in frequency domain. In *AIAA Atmospheric Flight Mechanics Conference*, Palo Alto, CA, 1998.

[35] V. Klein and E. A. Morelli. *Aircraft System Identification: Theory and Practice*. American Institute of Aeronautics and Astronautics, Education Series, Reston, VA, 2006.

[36] J. Ko, D. J. Klein, D. Fox, and D. Haehnel. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 742–747, 2007.

[37] J. Kocijan. Gaussian process models for systems identification. In *9th International PhD Workshop on Systems and Control: Young Generation Viewpoint*, Izola, Slovenia, October 2008.

[38] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith. Dynamic systems identification with Gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

[39] R. Koehler and K. Wilhelm. Auslegung von eingangssignalen fur die kennwertermittlung ("design of input signals for identification"). Ib 154-77/40, DFVLR Institut fur Flugmechanik, Brunswick, German, 1977 (in German).

[40] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[41] N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. *Advances in Neural Information Processing Systems*, 15:609–616, 2002.

[42] D. J. Linse and R. F. Stengel. Identification of aerodynamic coefficients using computational neural networks. *Journal of Guidance, Control, and Dynamics*, 16(6):1018–1025, 1993.

[43] G. Liu and A. A. Goldenberg. Uncertainty decomposition-based robust control of robot manipulators. *IEEE Transactions on Control Systems Technology*, 4 (4):384–393, 1996.

[44] S. C. Liu, D. L. Tan, and G. J. Liu. Robust leader-follower formation control of mobile robots based on a second order kinematics model. *Acta Automatica Sinica*, 33(9):947–955, 2007.

[45] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms.* Chapter 45. Cambridge University Press, 2003.

[46] R. E. Maine and K. W. Iliff. The theory and practice of estimating the accuracy of dynamic flight-determined coefficients. NASA Reference Publication 1077, Dryden Flight Research Center, 1981.

[47] R. E. Maine and K. W. Iliff. Application of parameter estimation to aircraft stability and control: The output-error approach. NASA Reference Publication 1168, Ames Research Center, 1986.

[48] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

[49] R. K. Mehra, D. E. Stepner, and J. S. Tyler. Maximum likelihood identification of aircraft stability and control derivatives. *Journal of Aircraft*, 11:81–89, 1974.

[50] E. A. Morelli. In-flight system identification. In *AIAA Atmospheric Flight Mechanics Conference*, Boston, MA, 1988.

[51] E. A. Morelli. Flight test validation of optimal input design using pilot implementation. In *Proceedings of the 10th IFAC Symposium on System Identification, Danish Automation Society*, pages 43–48, 1994.

[52] E. A. Morelli. Real-time parameter estimation in the frequency domain. *Journal of Guidance Control and Dynamics*, 23(5):812–818, 2000.

[53] J. Nakanishi, J. A. Farrell, and S. Schaal. Composite adaptive control with locally weighted statistical learning. *Neural Networks*, 18(1):71–90, 2005.

[54] NASA. Ames-Dryden-1 oblique wing aircraft. URL: http://www.dfrc.nasa.gov/Gallery/Graphics/AD-1/index.html. 1998.

[55] R. M. Neal. *Bayesian learning for neural networks.* PhD thesis, University of Toronto, 1995.

[56] E. Nettleton. *Decentralised architectures for tracking and navigation with multiple flight vehicles.* PhD thesis, The University of Sydney, 2003.

[57] D. Nguyen-Tuong and J. Peters. Local Gaussian process regression for real-time model-based robot control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 380–385, 2008.

[58] D. Nguyen-Tuong, M. Seeger, and J. Peters. Model learning with local Gaussian process regression. *Advanced Robotics*, 23(15):2015–2034, 2009.

[59] A. O'Hagan and J. F. C. Kingman. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–42, 1978.

[60] K.-O. Proskawetz. Optimierung stufenförmiger eingangssignale im frequenzbereich für die parameteridentifizierung ("multi-step input signals for parameter estimation, optimized in frequency domain"). In *Zeitschrift für Flugwissenschaften und Weltraumforschung*, volume 9, pages 362–370, 1985.

[61] J. Quinonero-Candela and C. E. Rasmussen. Analysis of some methods for reduced rank Gaussian process regression. *Switching and Learning in Feedback Systems*, pages 98–127, 2005.

[62] J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[63] J. Quinonero-Candela, C. E. Rasmussen, and C. K. I. Williams. Approximation methods for Gaussian process regression. *Large-scale kernel machines*, pages 203–224, 2007.

[64] C. E. Rasmussen. *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression.* PhD thesis, University of Toronto, 1996.

[65] C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems*, 2:881–888, 2002.

[66] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning.* The MIT Press, Cambridge, 2006.

[67] F. C. Schweppe. *Uncertain dynamic systems*, volume 160. Prentice-Hall Englewood Cliffs, NJ, 1973.

[68] Y. Shen, A. Ng, and M. Seeger. Fast Gaussian process regression using KD-trees. *Advances in Neural Information Processing Systems*, 18:1225, 2006.

[69] A. G. Sim and R. E. Curry. Flight-determined aerodynamic derivatives of the AD-1 oblique-wing research airplane. NASA Technical Paper 2222, Ames Research Center, 1984.

[70] A. G. Sim and R. E. Curry. Flight characteristics of the AD-1 oblique-wing research aircraft. NASA Technical Paper 2223, Ames Research Center, 1985.

[71] A. J. Smola and P. Bartlett. Sparse greedy Gaussian process regression. *Advances in Neural Information Processing Systems*, 13, 2001.

[72] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:1257–1264, 2006.

[73] E. Snelson and Z. Ghahramani. Local and global sparse Gaussian process approximations. In *Artificial Intelligence and Statistics*, volume 11, 2007.

[74] R. H. Stone and G. Clarke. The T-wing: a VTOL UAV for defense and civilian applications. *University of Sydney*, 2001.

[75] Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In *Workshop on Artificial Intelligence and Statistics*, volume 10, pages 333–340, 2005.

[76] C. R. Theodore, M. B. Tischler, and J. D. Colbourne. Rapid frequency-domain modeling methods for unmanned aerial vehicle flight control applications. *Journal of Aircraft*, 41(4):735–743, 2004.

[77] M. B. Tischler. System identification methods for aircraft flight control development and validation. *Advances in Aircraft Flight Control*, 1996.

[78] M. B. Tischler and M. G. Cauffman. 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, 1992.

[79] M. B. Tischler and R. K. Remple. *Aircraft and Rotorcraft System Identification: Engineering Methods with Flight Test Examples.* American Institute of Aeronautics and Astronautics, Education Series, Reston, VA, 2006.

[80] M. B. Tischler and C. A. Tumashofski. Flight test identification of SH-2G flapped-rotor helicopter flight mechanics models. *Journal of the American Helicopter Society*, 47(1):18–32, 2002.

[81] P. J. van Laarhoven and E. H. Aarts. *Simulated annealing: theory and applications*, volume 37. Springer, 1987.

[82] J. Vanhatalo and A. Vehtari. Modelling local and global phenomena with sparse Gaussian processes. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 571–578, 2008.

[83] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte. Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10): 812–840, 2009.

[84] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte. Non-stationary dependent Gaussian processes for data fusion in large-scale terrain modeling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1875–1882, 2011.

[85] S. Vijayakumar, A. D'souza, and S. Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634, 2005.

[86] J. F. Wendt. *Computational fluid dynamics: an introduction.* Springer, 2008.

[87] C. K. I. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.

[88] C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. *NATO ASI Series D, Behavioural and Social Sciences*, 89:599–621, 1998.

[89] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8:514–520, 1996.

[90] H. Wu, D. Sun, and Z. Zhou. Model identification of a micro air vehicle in loitering flight based on attitude performance evaluation. *IEEE Transactions on Robotics*, 20(4):702–712, 2004.

[91] J. F. Zubizarreta-Rodriguez and F. Ramos. Multi-task learning of system dynamics with maximum information gain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5709–5715, 2011.