PROBABILISTIC REGIME RECOGNITION AND DAMAGE ESTIMATION FOR ROTARY AND FIXED WING AIRCRAFT

A Dissertation
Presented to
The Academic Faculty

By

Dakota Musso

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Engineering
School of Mechanical Engineering

Georgia Institute of Technology

December 2021

PROBABILISTIC REGIME RECOGNITION AND DAMAGE ESTIMATION FOR ROTARY AND FIXED WING AIRCRAFT

Thesis committee:

Dr. Jonathan Rogers, Chair
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Aldo Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Anirban Mazumdar
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Alper Erturk
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Jonnalagadda V. R. Prasad
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: November 19, 2021

I am, and ever will be, a white-socks, pocket-protector, nerdy engineer, born under the second law of thermodynamics, steeped in steam tables, in love with free-body diagrams, transformed by Laplace and propelled by compressible flow.

Neil Armstrong

For my wife, family, and friends.

ACKNOWLEDGMENTS

I would first like to thank my advisor Dr. Jonathan Rogers for his support and mentorship throughout my time in graduate school. I graduated from a small undergraduate university that is most certainly not known for its research, so I know Dr. Rogers took a chance when he invited me to join his lab. I am both grateful and glad that he did. Dr. Rogers instills a desire to achieve and explore in research, and he has pushed me to become a far better engineer than I thought could be. His technical expertise combined with his care and dedication to his grad students has made my time at Georgia Tech incredibly rewarding.

I want to thank my fellow labmates (both former and current): Brian Eberle, Umberto Saetti, Kevin Webb, Jared Elinger, Joey Myers, Geordan Gutow, Adam Garlow, and Sam Kemp, for their invaluable advise both in terms of research and with regards to navigating grad school. Together we had a healthy lab culture that was both fun and productive. Specifically, I want to extend a special thanks to Brian Eberle, who took me under his wing, and gave me just enough pep-talks to keep me going whenever classes and research were getting exceptionally tough.

I need to thank my wife and best friend Meghan, without her love, support, and (daily) encouragement, I doubt I would have made it to this point at all. I also want to thank the rest of my family for their support throughout my time during both undergraduate and graduate school.

I would like to thank the members of my committee: Dr. Aldo Ferri, Dr. Alper Erturk, Dr. J.V.R. Prasad, and Dr. Anirban Mazumdar for their guidance and constructive critique of my research. Finally, I want to thank all of my professors, classmates, friends and family not mentioned by name for your input, advise, and listening ears.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AOB Angle of Bank

AUC Area Under the [ROC] Cuve

CBM Condition Based Maintenance

CDF Cumulative Density Function

DCM Damage Computation Module

DOF Degree of Freedom

HIVE Helicopter Integrated Vehicle Simulation Environment

HMM Hidden Markov Model

HUMS Health and Usage Monitoring System

IAT Individual Aircraft Tracking

IMM Interacting Multiple Model

INS Inertial navigation system

ITS Inverse Transform Sampling

OEM Original Equipment Manufacturer

PDF Probability Density Function

PMF Probability Mass Function

RB Rule-Based

ROC Receiver Operating Characteristic

RR Regime Recognition

SISO Single Input Single Output

NOMENCLATURE

α	Angle of attack
$\bar{c}$	Mean aerodynamic chord
$\bar{q}$	Dynamic pressure
β	Angle of sideslip
δ_a	Aileron control surface deflection
δ_e	Elevator control surface deflection
δ_r	Rudder control surface deflection
δ_{th}	Throttle control surface deflection
b	Wing span
$c_1 ... c_9$	Inertia constants
C_l, C_m, C_n	Non-dimensional body-axis moment coefficient about X,Y,Z
C_X, C_Y, C_Z	Non-dimensional body-axis force coefficient about X,Y,Z
$E_t(r)$	Expected time of regime r
$F_t(r)$	Expected time fraction of regime r
g	Earth gravitational constant
h	Aircraft altitude above mean sea level
h_{eng}	Engine angular momentum about X body-axis
m	Aircraft mass
p, q, r	Body-axis angular rates about X,Y,Z
P_a	Actual engine power level
S	Wing reference area
T	Engine thrust

u, v, w	Body-axis translational velocity about X,Y,Z
V_t	Airspeed of the aircraft
x_E, y_E	Planar earth-axis positions X,Y
$.9_6$	Percentile corresponding to 99.9999%
Δt	Duration of timestep
$\gamma_{\eta,i}$	Damage estimated as the η percentile of damage distribution for flight sequence index i
$\mathcal{N}(a, b)$	Gaussian (normal) distribution with mean a and variance b
$\mathcal{U}[a, b]$	Uniform distribution from a to b
μ	Mean
σ	Standard deviation
σ^2	Variance
θ, ϕ, ψ	Euler pitch, roll, and yaw angles
$\vec{m}$	Vector of measured vehicle states
$\vec{r_i}$	Measurement vector of filter i
$\vec{s_i}$	State vector of filter i
$d_{r,k}$	Damage rate from regime r and component k
$F(X)$	Cumulative density function of random variable X
N_f	Number of mode-matched filters
p_b^k	Probabilities of bank b and filter k
$P_j(X_j)$	Probability mass function of random variable X at timestep j
$p_r^{t_j}$	Probability of regime r at timestep j
Q	Model error covariance matrix
R	Measurement error covariance matrix
V_h	Maximum level forward flight speed of the vehicle
x, y, z	Position coordinates of the vehicle mass center in the inertial frame
$X_{a:b}$	Sum of random variables X from timesteps a to b
$X_{j,k}$	Damage random variable at timestep j for component k

SUMMARY

Regime recognition is a critical tool used for condition-based maintenance, fatigue life prediction, and creation of usage spectra for military and commercial rotorcraft. Regime recognition is the process by which aircraft state data is analyzed and vehicle maneuvers are identified with respect to time. While a variety of regime recognition algorithms are currently in use, many current algorithms suffer from an over-reliance on training data or poor classification performance.

This work seeks two main goals, the first is to introduce a new type of regime recognition algorithm based on a multiple model adaptive estimation scheme, known as an interacting multiple model (IMM) estimator. IMM estimators use a bank of dynamic models to evaluate the likelihood of the system existing in one of various possible dynamic modes. In the regime recognition context, each mode represents the system operating in a given maneuver regime. Compared with other approaches, IMM estimators offer the benefits of probabilistic regime classification and the incorporation of knowledge of the aircraft flight dynamics, which reduces reliance on training data. This dissertation presents a novel formulation of an IMM estimator for regime recognition wherein mode probabilities from a bank of IMM filters are combined in Bayesian framework to yield maneuver regime probabilities.

Second, this work introduces a novel approach to condition based maintenance (CBM) that is designed to be used in conjunction with a probabilistic regime recognition scheme. The proposed methodology results in a probability distribution of incurred damage for specified life-limited components on an individual aircraft basis, with applications to fleet-wide probabilistic damage spectrum. The primary advantage of the proposed methodology is the rigorous treatment of regime uncertainty that is quantified in probabilistic regime recognition. A Gaussian probability density approximation is used to improve computational efficiency of the damage estimation method, and effects of the flight condition on the uncertainty in the damage estimation are investigated.

Following a detailed explanation of the proposed methodologies, results are provided by applying the algorithms to both simulated and real helicopter data, and compared against “standard” rule-based approaches. Lastly, some of the methodologies presented in this dissertation are appropriately modified and applied to fixed wing aircraft and results are shown using simulated data for the F-16.

CHAPTER 1
INTRODUCTION

1.1 Problem Motivation and Background

Helicopters contain numerous components that experience repeated dynamic loading that results in limited part life due to material fatigue. A key goal of modern Condition Based Maintenance (CBM) programs for both military and civilian helicopters is to estimate the useful life remaining for life-limited components, and to replace components on an as-needed basis dictated by the flight loads incurred by a specific vehicle. CBM paradigms for helicopters in the past have embraced a usage spectrum approach in which component replacement timelines are specified by the aircraft manufacturer based on a mutually agreed upon assumed usage spectrum [1, 2]. This assumed usage spectrum is designed to be conservative and may be refined as the aircraft fleet accumulates operational flight hours [3]. However, inaccuracies in the assumed usage spectrum (as documented by Moon *et al.* [4]) can lead to either unnecessary part replacement or unexpected failures [5]. As an alternative to usage spectrum-based approaches, future CBM programs are expected to estimate component life remaining based on actual aircraft usage, a paradigm known as Individual Aircraft Tracking (IAT). This approach to CBM requires collection and analysis of flight data for each aircraft along with continual updating of component life predictions using measured or estimated loads (or damage rates). Several US military specifications [6–8] highlight the need for, and benefits of, individual aircraft tracking schemes for current and future military aircraft.

One avenue to track component usage on a per-aircraft basis is through direct loads measurement or estimation. Because many life-limited components on helicopters are part of the rotating system, direct measurement of loads requires costly modifications to the aircraft with the installation of strain gauges and slip rings. The difficulty involved in maintaining such components may lead to poor reliability [9, 10]. An alternative is to estimate loads from flight dynamic parameters recorded on-board Health and Usage Monitoring System (HUMS) [11–13]. HUMS are playing an increasingly important role in CBM programs for all modern aircraft, but especially helicopters. These systems typically involve sensor packages on-board the aircraft to record rigid-body states, vibrational measurements, and other data; telemetry or data storage packages; and real-time and/or post-flight processing algorithms [14, 15]. These algorithms may serve a variety of purposes to include anomaly detection, creation of usage or loads spectra [16, 17], fatigue prediction [18, 19], or some combination. Both the U.S. military and commercial helicopter operators are increasingly interested in using HUMS data to predict and estimate how flight loads translate into fatigue damage on life-limited components. Significant research over the past several decades has focused on developing various possible algorithms and techniques for this purpose.

Some developed approaches utilize various numerical methods. One such approach to loads estimation using HUMS data employs high-fidelity structural models of the aircraft to predict fatigue loads and estimate component useful life remaining [20–22]. While this so-called “digital twin” approach is promising and potentially transformative, modeling the complex structural dynamics of helicopters to sufficiently high fidelity to accurately predict fatigue loads has proven challenging [23]. Instead of a physics-based structural model approach, numerous authors have proposed estimating loads using black box machine learning methods such as neural networks or regression models, a class of algorithms that are known as *flight loads synthesis*. Such methods have been shown to estimate component strains and load spectra with reasonable accuracy [10, 13, 24–26]. The neural nets employed in these algorithms are trained on a database of flight parameters, aircraft states, and the corresponding component loads – taken from heavily instrumented vehicles [27–30]. As an alternative to neural networks, several authors have employed other types of regression models in which model coefficients are derived from empirical training data [31–33]. For example, Ory and Lindert [34] used a reconstruction method to develop a model of the air loads on helicopter blades for the purpose of damage estimation.

While flight loads synthesis techniques have been successfully employed for fixed-wing aircraft, the complexity of helicopter structural and aerodynamic loads means that such approaches require fairly complex regression or machine learning models, with resulting requirements for large training data sets. Such large training data sets, which must include both direct loads measurements and corresponding rigid body flight data, are not always available or may be expensive to obtain [35, 36]. As a result, the rotorcraft community has relied on approaches that bypass the need for loads estimation. Instead of estimating loads continuously, the helicopter Original Equipment Manufacturer (OEM) measures loads for a discrete set of flight regimes using highly-instrumented aircraft, and converts this loads data to fatigue damage rates for each life-limited component on a per-regime basis [37–39]. Component replacement timelines can then be developed from an assumed usage spectrum, or an updated usage spectrum derived from employing Regime Recognition (RR) with large quantities of flight data [40, 41].

Initial design usage spectrum are often supplied by the manufacturer based on the expected use of the aircraft. This assumed usage spectrum must revised as needed to match the actual usage as closely as possible. The manufacturer must be notified if the actual usage differs significantly from the supplied usage spectrum so that it can be adjusted. In the past, surveys were conducted using paper questionnaires that were supplied to the pilots. The pilots filled out information such as mission type, gross weight, altitude, flight maneuver and associated time, velocity, vertical acceleration, wind speed, etc. This information is then analyzed by an engineer to update the fleet usage [42, 43]. It was found by the Royal Dutch Navy, that an extension of 10% in the airframe life of their Lynx fleet could be achieved by frequent updating of the aircraft usage spectrum [44]. However, manually recording and analyzing flight regimes has the potential to introduce a large amount of human error, resulting in the usage spectrum either being over or under conservative with its estimates. With the increase in availability of HUMS data and computational power, the U.S. Military, among others agencies, have been relying more on regime recognition to produce more accurate usage spectra.

Regime recognition involves the post-flight classification of recorded flight data into a set of discrete categories, called maneuver regimes, in which fatigue damage rates are known for life-limited components. Many condition-based maintenance processes for rotorcraft currently employed by the U.S. military and civilian operators rely on regime recognition for fatigue damage prediction [45, 46]. For military operations, ADS-79D provides guidance on the development and certification of RR algorithms [8]. This document (ADS-79D) specifies fairly stringent accuracy requirements – for instance, that regime recognition codes should be able to demonstrate 97% classification accuracy. This is a difficult goal to meet given that there is no single, universally accepted definition for maneuvers in the vehicle state space, and that execution of the same maneuver can vary greatly between pilots. Work done by Warner and Rogers [47] examines various methodologies for assessing accuracy of regime recognition codes. Their paper describes in detail how the lack of universally accepted regime definitions imposes limits on the accuracy of regime recognition. As shown through the examples in [47], the regime recognition problem is arguably ill-posed due to the requirement that continuous state data be deterministically classified into discrete, ill-defined regime categories. Nevertheless, the simplicity of RR and its ability to predict fatigue damage without the need for complex models means that, despite its limitations, regime recognition still forms the basis for many condition based maintenance programs.

Generally, three types of regime recognition codes have been developed to date: rule-based [39, 48–52], motion primitive based [53], and Hidden Markov Model (HMM) based [54, 55]. Rule-based algorithms use a set of logical rules based on state- or parameter-dependent thresholds to perform regime classification at each measurement timestep. For some maneuvers that appear similar, tiered logic using data from previous timesteps may be employed to differentiate one maneuver from another. A fully detailed formulation of an example rule-based regime recognition algorithm is provided in [52]. In recent work, Saetti and Rogers [53] developed an RR algorithm that leverages motion primitives to distinguish between trim and maneuver flight segments in a non-causal approach. Trim segments are classified into particular trim regimes based on conditional logic, while maneuver segments are classified into particular maneuvers. Dynamic Time Warping is used to compensate for maneuver initial condition, duration, and aggressiveness. (Note that in this dissertation, trim and maneuver segments are treated as synonymous.) Alternatively, the Hidden Markov Model approach to regime recognition proposed in [54, 55] performs classification using a Gaussian mixture model, where the mixture coefficients are computed recursively for each measurement time based on the previous measurement. HMM algorithms rely on labeled training data and do not contain any internal model of the vehicle dynamics.

There are unique benefits and drawbacks of rule-based, motion primitive based, and HMM based approaches to regime recognition. Rule-based algorithms are relatively easy to construct, but lack a generalized recursion technique that conditions the current classification on the previous one. They also perform classification in a binary way — either the aircraft is flying a regime with 100% certainty, or it is not with 100% certainty. This deterministic classification does not properly acknowledge the inherent uncertainty in the regime recognition process. In the motion primitive based approach, the conditional logic is applied on a per-segment basis, rather than on a per-time-step basis. This avoids the typical high-frequency switching behavior seen in many rule-based methods. However, similar to a rule-based method, the motion primitive approach classifies the flight sequence deterministically (i.e. the aircraft *is* flying a particular regime, or it is *not* flying a particular regime). In contrast with these two methods, HMM algorithms leverage a Markov Model during classification, providing a recursion mechanism that enforces filter memory. Furthermore, they classify regimes probabilistically, providing a likelihood associated with each regime at each timestep. However, HMM algorithms do not incorporate any flight dynamic model. Instead, the HMM algorithm is a data-driven method that is trained solely on whatever scripted flight test data is available. As a result, flight dynamic models cannot be leveraged to any extent by these algorithms and they are only as accurate as the scripted flight test data on which they have been trained. This lack of internal model can also cause issues with certification of the algorithm among some agencies that may wish to employ it.

From the results of either of the two deterministic algorithms discussed above (rule-based and motion primitive based), the damage incurred on each component is straightforward to calculate, as an assumed damage rate for each life-limited component per each maneuver is specified. However, regime recognition algorithms that result in probabilistic estimates of the flown regime at each time step (i.e. they produce a probability distribution over the regimes rather than a deterministic classification) are not straightforward to translate into damage estimates. Currently no technique is available to translate probabilistic regime classifications to fatigue damage estimates for a specified component.

1.2 Work Overview

The work of this dissertation seeks to address two main topics. First, by documenting the design and testing of a new type of regime recognition algorithm for helicopters using Interacting Multiple Model (IMM) estimators. An IMM estimator is a recursive Bayesian filtering algorithm for systems that exhibit multiple modes of dynamic behavior [56]. In the regime recognition context, each regime may be considered a different dynamic mode of the system. At each timestep of recorded data, the IMM computes innovations between the predicted state associated with each mode and the measured observation. Then the probability of each regime is computed based on these innovations, a pre-defined transition probability matrix, and the probabilities from the prior timestep. Through this probabilistic mode determination, the IMM incorporates uncertainty quantification into the regime recognition process. It also inherently leverages a flight dynamic model of the aircraft and thus does not rely strictly on training data from scripted flights.

Secondly, this dissertation documents a novel technique to compute fatigue damage estimates from probabilistic regime classifications. The proposed method for fatigue damage estimation is designed to be used in conjunction with probabilistic RR algorithms such as the Hidden Markov Model method from [54, 55, 57] or the IMM method to be discussed in Chapter 2, and also documented in References [58, 59]. In the proposed approach, the fatigue damage incurred by each component at a given time step of flight data is considered as a random variable with a Probability Mass Function (PMF) dictated by the identified regime and the damage rates in each regime. Incurred damage over a flight is then estimated by summing the damage random variables from each timestep. This results in a probability distribution for the total damage incurred by each component over a flight, which can be used for IAT purposes to evaluate remaining useful life. In order to improve the computational performance of the algorithm, a Gaussian approximation is made that eliminates the need to convolute a large numbers of random variables. This modification results in minimal loss of accuracy but reduces computation time to a nearly-trivial amount. While the focus of this work is primarily on damage estimation for IAT, an extension of the methodology is discussed that allows it to be employed for creation of a damage-per-100 hours spectrum for fleet-wide use.

Results are presented for using both simulated data for an SH-60B and F-16, and real data gathered from a generic utility-scale helicopter. Probabilistic damage predictions from the proposed algorithm are compared against those from a deterministic algorithm. Results show that the probabilistic method produces fatigue damage distributions that mirror the uncertainty in the RR process. When regime estimates have low uncertainty, the damage distribution has low variance and closely matches deterministic predictions. However, when regime estimates have high uncertainty, the corresponding damage distribution has large variance, indicating significant uncertainty in the incurred damage. These results highlight the main advantage of the probabilistic approach (i.e. uncertainty in the RR process is translated in a rigorous way into the resulting fatigue damage estimates). Furthermore, the approach is general enough that additional sources of uncertainty (e.g. uncertainty in the damage rates caused by variability in regime execution as described above) can be easily incorporated to produce a unified probabilistic framework that captures all significant sources of uncertainty in the damage estimation process.

1.2.1 Dissertation Outline

This dissertation proceeds as follows: Chapter 2 begins with an overview of the IMM filter. The regime recognition algorithm is then presented as a set of IMM filters and Kalman filters specific to one or more vehicle states, whose mode probabilities are combined to form the regime probabilities. Results are presented for several simulated flight sequences of the SH-60B. Accuracy of the regime recognition results are assessed through the use of accuracy metrics that are applicable to probabilistic classification. Results are compared with regime recognition outputs from a rule-based classification scheme, and performance tradeoffs are highlighted. Chapter 3 provides a detailed description of the probabilistic damage estimation algorithm. A Gaussian approximation is introduced and justified, and analysis is performed to study the factors that influence variance in the damage distribution. An algorithmic extension is also presented for use in fleet-wide usage spectrum development. Simulation results comparing performance of the proposed technique against a deterministic approach are shown. Chapter 4 takes the proposed methodologies and algorithms and sees them applied to actual HUMS data gathered from a generic single engine transport helicopter. Chapter 5 shows the presented algorithms applied to a fixed-wing aircraft with little modification. Results are shown using simulated data generated from a six-Degree of Freedom (DOF) model of a General Dynamics F-16. Finally, Chapter 6 summarizes the main contributions of work and gives recommendations for areas of future research.

CHAPTER 2

PROBABILISTIC REGIME RECOGNITION

2.1 Regime Recognition

Regime recognition is an important aspect of condition-based maintenance for modern helicopters, and involves the post-flight classification of flight data into regime categories. These classifications are then used to predict fatigue damage and vehicle usage spectra. While several regime recognition algorithms have been proposed to date, many suffer from an over-reliance on training data or poor accuracy when presented with flight data that does not precisely match one of the defined regimes. This dissertation introduces a new type of regime recognition algorithm based on IMM estimators, which use a bank of dynamic models to evaluate the probability of the system existing in one of various possible dynamic modes. In the regime recognition context, each of the dynamic modes corresponds to a particular regime. The proposed recognition algorithm offers advantages over other methods in that it provides a probabilistic classification of flight data, thereby explicitly acknowledging uncertainty in the recognition process. Furthermore, the algorithm is model-based, reducing reliance on training data. Following a detailed description of the methodology, results are provided by applying the algorithm to simulated data towards the end of this chapter, and actual flight test data in Chapter 4. Results show significant performance improvements compared with a typical rule-based recognition scheme.

2.2 IMM Estimator Overview

The Interacting Multiple Model estimator is a special type of multiple model adaptive estimation scheme in which the system is considered to exist in one of several possible modes [56]. IMMs are in widespread use in radar processing [60], missile tracking [61], and a variety of other aerospace and signal processing applications [62]. The IMM estimator contains a bank of (extended) Kalman filters corresponding to each system mode. A Bayesian mixing step combines the contributions of each mode-matched filter to arrive at an overall state estimate, as well as a set of probabilities of the system existing in each of the modes at the specific time. A brief summary of the IMM algorithm is presented here, while a detailed mathematical description is available in [56].

A diagram of a general IMM filter is shown in Figure 2.1. A set of N_f Kalman filters are propagated at each timestep – one for each dynamic mode of the system. Let the mode i filter have state vector $\vec{x_i}$ with a state propagation equation given by,

$$\vec{x_i} = f_i(\vec{x_i}, \vec{u_i}) \tag{2.1}$$

where $\vec{u_i}$ is a vector of control inputs, specific to the i^{th} dynamic mode. The covariance matrix associated with this mode is denoted as P_i The k^{th} filter step is computed as follows, with reference to Figure 2.1. First, the N_f models are propagated to the k^{th} measurement time. The measurement at update step k is denoted as $z(k)$. The residual for filter i is computed as,

$$\vec{r_i}(k) = z(k) - H_i(k)\vec{x_i}(k^*) \tag{2.2}$$

where $\vec{x_i}(k^*)$ is the update step k and $H_i(k)\vec{x_i}(k^*)$ is the predicted measurement. The residual covariance for filter i is given as,

$$S_i(k) = H_i(k)P_i(k^*)H_i(k)^T + R_i(k) \tag{2.3}$$

where $P_i(k^*)$ is the predicted state error covariance at update k, $H_i(k)$ is the measurement matrix for filter i, and $R_i(k)$is the measurement error covariance for filter i. The state and covariance updates for filter i are given by,

$$\vec{x_i} = \vec{x_i}(k^*) + W_i(k)\vec{r_i}(k) \tag{2.4}$$

$$P_i(k) = P_i(k^*) + W_i(k)S_i(k)W_i(k)^T \tag{2.5}$$

where $W_i(k)$ is the Kalman gain matrix. This propagation and measurement update step is repeated for all N_f filters.

Following the individual mode-matched filter updates, the mode probabilities are computed as follows. First, the likelihood function for mode i is computed according to,

$$\Lambda_i(k) = \mathcal{N}(r_i(k); \vec{0}; S_i(k)) \tag{2.6}$$

where $\mathcal{N}(a(k); \vec{0}; M)$ denotes the zero-mean multivariate normal distribution with covariance matrix M evaluated at a $\vec{a}$. Finally, the mode probabilities are computed as,

$$\mu_i(k) = \frac{1}{c}\Lambda_i(k)\sum_{j=1}^{N_f} p_{ij}\mu_j(k-1) \tag{2.7}$$

In Equation 2.7, c is a normalizing coefficient and p_{ji} is the Markov transition probability from mode j to mode i. The Markov transition probabilities p_{ji} are constants that are defined based on anticipated system behavior [56]. Also, the initial mode probabilities $\mu_i(0)$ may be set based on knowledge of the initial system mode, or can be set to uniform values $\mu_i(0) = \frac{1}{N_f}$ if no information about the initial mode of the system is available. The filtering process defined in Equations (2.1) to (2.7) is repeated for each of the N_f mode-matched filters to produce a probability distribution over the set of modes. Note that additional interaction and mixing steps are performed, but are not described here since the mode probabilities in Equation 2.7 are the primary quantity of interest in this work (details of these additional steps, as well as a more complete overview of the IMM are provided in [56]).

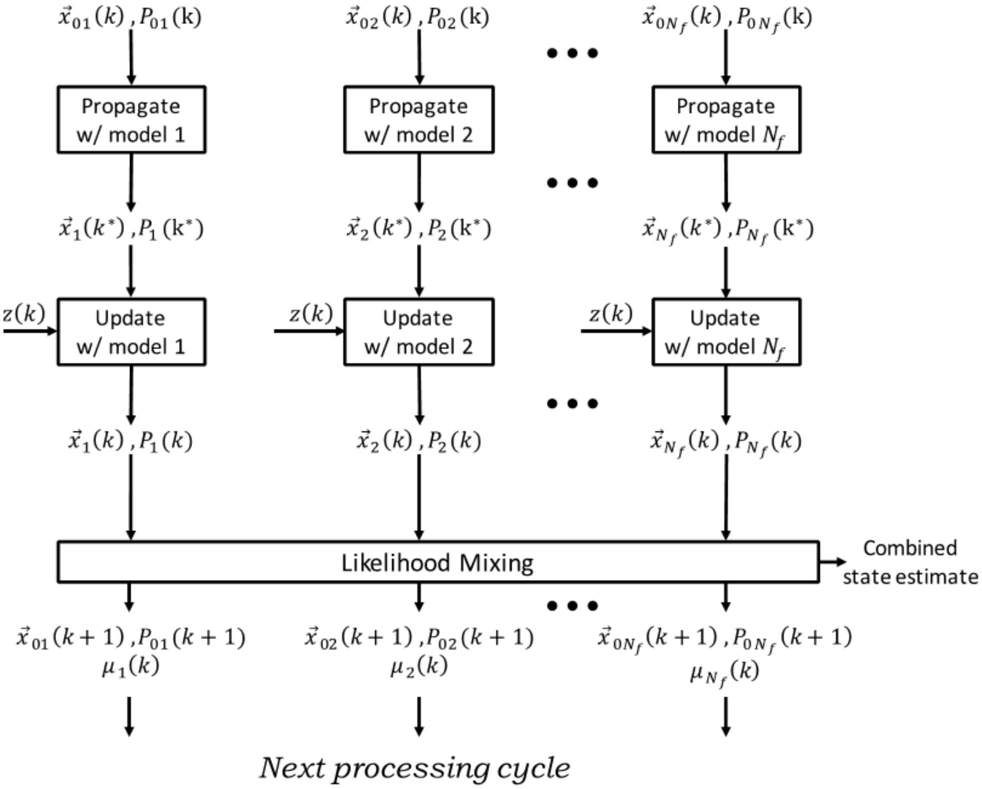

Figure 2.1: Overview of the IMM Filter.

From a qualitative standpoint, the IMM filter propagates a bank of independent dynamic models. When a measurement is received, the residuals of each model are compared against one another. In general, a filter that exhibits a smaller residual will lead to a higher probability that the system is in that mode, compared to a filter with a higher residual. In addition, the mode transition probability matrix defines the probability that the system transitions from one mode to another at each filter timestep. These Markov transition probabilities and the measurement residuals together determine the mode probabilities per Equation 2.7.

There are several key tuning parameters in an IMM filter – the model error covariance (Q_i for filter i), the measurement error covariance (R_i for filter i), and the mode transition probability matrix. The values of these matrices govern both the steady-state behavior and the rate at which the filter switches between modes. Note that Q_i and R_i must be defined for each filter in the IMM, and the transition probabilities matrix is $N_f \times N_f$. Thus, the number of tuning parameters can become quite large as N_f and the size of the state space grows. This can lead to complications and poor performance as discussed in [63]. In the context of this paper, the scaling issues associated with the tuning parameters of the IMM filter lead to specific design choices as described in the next section.

2.3 IMM-Based Regime Recognition for Rotorcraft

2.3.1 Algorithm Overview

The key claim underlying this work is that each maneuver regime defined for helicopter regime recognition purposes can be recognized as a unique dynamic mode of the system, and that the mode probabilities associated with each mode define the probability that the helicopter is executing that maneuver regime at the given timestep. Thus, the regime recognition problem is solved by computing the mode probabilities μ_i for $i = 1, ..., N_f$ at each timestep of the recorded data. Note that since the mode probabilities exist in the continuous interval $[0, 1]$, the regime recognition problem is solved probabilistically. There are usually non-zero probabilities of the system being in multiple modes (i.e., executing multiple maneuvers) at once.

Consider a standard IMM filter constructed to solve the regime recognition problem. There are typically upwards of 100 different maneuver regimes defined for a given vehicle, and the typical state space for each dynamic mode might range from 5-10 state variables. This leads to an enormous number of tuning parameters – the mode transition probability matrix of 100×100 elements, as well as over 100 different Q and R matrices (one for each filter) which can range in size from 5×5 to 10×10. The sheer number of resulting tuning parameters makes this treatment of the problem impractical, as it can lead to poor performance and excessive development effort.

Instead, an alternative approach is proposed here. A set of IMM filters and Kalman filters is defined, each with a low-dimensional state vector. Each IMM filter or Kalman filter is referred to as a *bank*. The mode probabilities for each filter bank are then combined to form a joint probability. Because maneuver regimes are defined as unique regions in the joint probability domain, the joint probabilities form the final maneuver regime probabilities. This overall process is shown in Figure 2.2. Each of the filter banks produces specific estimates for the mode probabilities of its internal modes. These mode probabilities are then combined to form joint probabilities, which are mapped to the overall maneuver regime probabilities.

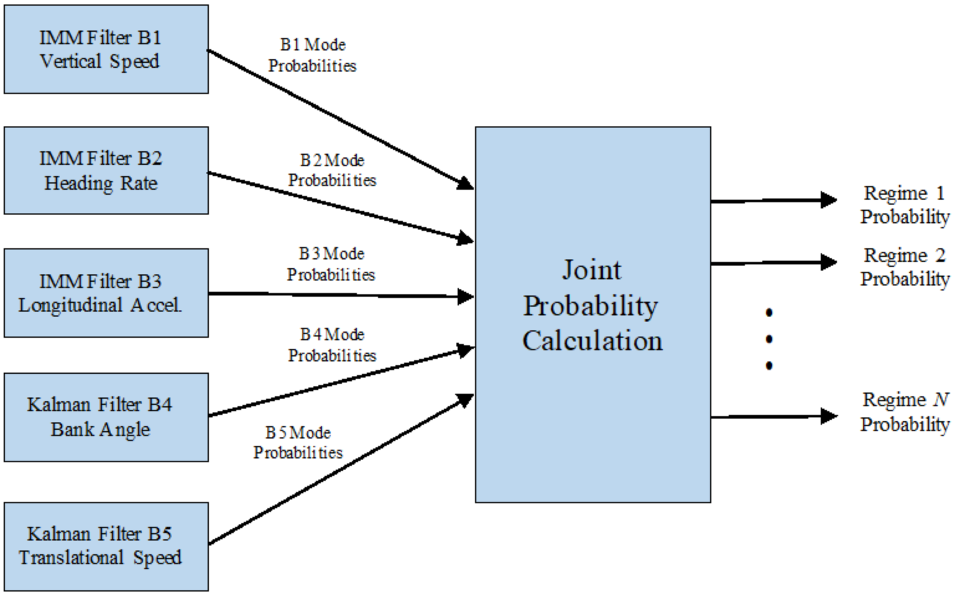

Figure 2.2: IMM Estimator Framework for Maneuver Regime Recognition.

The primary benefit of this scheme is the low dimensionality of the state vectors within each filter bank (as will be described below). Therefore, the filter includes only a handful of tuning parameters. In simulation experiments during development, this type of filter architecture exhibited more robust performance and much less sensitivity to tuning parameters compared to a single IMM filter with a larger state space that includes all regimes.

It should also be noted that the models used in each of the Kalman filters or IMM filters in Figure 2.2 can be of varying fidelity. A wide range of rotorcraft flight dynamic models of varying fidelity have been developed in the literature, ranging from point mass models [64, 65] to 6DOF models driven by aerodynamic loads computed from blade element theory

[66] or computational fluid dynamics [67, 68]. A comprehensive treatment of rotorcraft simulation techniques, as well as discussion of modeling trade offs, is provided in [69]. In this work, a reduced 6-degree-of-freedom state vector is used by the mode filters and the vehicle models embedded in each filter are kinematic only in nature. This design, in which forces and moments are not modeled explicitly in the filter, reduces any dependence on complex aerodynamic models while also providing suitable performance in the studies performed here.

The remainder of this section details each of the filter banks as well as the scheme for computing joint probabilities. Note that the implementation here is simply a prototype and is capable of identifying only 18 common maneuver regimes. However, the filter architecture is easily scalable, and implementation of the necessary banks and filter modes to be able to identify the full range of maneuver regimes for a given helicopter platform would follow similar procedures to those detailed here.

2.3.2 Measured Vehicle States

For the purposes of this work, the regime recognition algorithm is assumed to have access to the following measured state vector at an appropriate update rate (10 Hz were assumed for these studies, but the update rate can vary depending on application and sensing capabilities):

$$\vec{m} = \begin{Bmatrix} z \\ \dot{z} \\ \phi \\ \psi \\ \dot{\psi} \\ u \end{Bmatrix} \tag{2.8}$$

In Equation 2.8, z is the helicopter altitude (with respect to an arbitrary reference, positive up) and ϕ and ψ are the Euler roll and yaw angles, respectively. Define the standard helicopter body frame in which unit vectors $\vec{I_B}$, $\vec{J_B}$, and $\vec{K_B}$ point respectively towards the front of the aircraft, out the right side of the aircraft, and down, with the origin at the mass center (North-East-Down convention). Then u in Equation 2.8 is the $\vec{I_B}$ component of the velocity of the mass center with respect to the wind-fixed frame. Note that these values can usually be obtained from standard Inertial navigation system (INS) packages and air data systems typically included in flight data recorders or Health and Usage Monitoring Systems. Also note that the states in Equation 2.8 are selected only to facilitate computation of this particular prototype filter – a larger, more complete measurement vector may be needed to drive an IMM-based algorithm that estimates all maneuver regimes. Finally, the regime recognition algorithm detailed in this work does not use measurements of pilot inputs. This is because nearly all maneuver regimes are defined based on the helicopter states rather than control inputs, and thus conditioning regime classifications on the inputs introduces unnecessary complications in the recognition process for most regimes compared to a classification method based on vehicle states only. It is possible that identification of certain maneuvers (e.g. control reversals) will require the inclusion of control inputs in the data set considered by the algorithm; however, such developments are not considered in this work.

2.3.3 Vertical Speed, Heading Rate, and Acceleration IMM Filters

This section presents the vertical speed, heading rate, and acceleration IMM filters, all of which share a common design. The vertical speed IMM filter will be presented in detail, followed by brief discussions of the analogous designs of the other filters. The goal of the vertical speed IMM filter is to estimate the probability that the vehicle is in a climb, descent, or level flight condition. An IMM is implemented for this purpose defining these three conditions as separate dynamic modes. The state vector and measurement vector for this IMM filter are given respectively by,

$$\vec{s_1} = \begin{Bmatrix} z \\ \dot{z} \end{Bmatrix} \tag{2.9}$$

$$\vec{r_1} = \begin{Bmatrix} z^* \\ \dot{z}^* \end{Bmatrix} \tag{2.10}$$

Within this IMM filter, five Kalman filters are defined. Each filter uses an identical internal state vector but a different dynamic model. The dynamic model of the k^{th} filter is given by,

$$\vec{s}_1^k = \begin{Bmatrix} \hat{\dot{z}} \\ 0 \end{Bmatrix}, \qquad \hat{\dot{z}} = \begin{cases} \bar{\dot{z}} & \text{if } \bar{\dot{z}} \in [\dot{z}_{th,k}, \dot{z}_{th,k+1}] \\ \dot{z}_{th,k} & \text{if } \bar{\dot{z}} < \dot{z}_{th,k} \\ \dot{z}_{th,k+1} & \text{if } \bar{\dot{z}} > \dot{z}_{th,k+1} \end{cases} \tag{2.11}$$

where $\bar{\dot{z}}$ is the IMM's estimate of the vertical speed. Note that the vertical speed $\hat{\dot{z}}$ used in the k^{th} filter is bounded by two threshold values $\dot{z}_{th,k}$ and $\dot{z}_{th,k+1}$. These thresholds bound a given vertical speed range and are given in Table 2.1. The five filters in this bank are designed to differentiate between climb maneuvers $(k = 1)$, slow climb maneuvers $(k = 2)$, constant altitude maneuvers $(k = 3)$, slow descent maneuvers $(k = 4)$, and descent maneuvers $(k = 5)$. Note that the differentiation between climb and slow climb (and descent and slow descent) maneuvers is made because typical climb and descent rates in hover are smaller in magnitude compared to forward flight. The resulting mode probabilities p_1^1, p_1^2, p_1^3,p_1^4, and p_1^5 (analogous to the μ_i mode probabilities in the previous section) are used in the joint probability calculation to estimate the final regime probabilities, as will be discussed below.

During the estimation process, if the k^{th} filter estimates $\dot{z}$ to be within the range $[\dot{z}_{th,k}, \dot{z}_{th,k+1}]$, then the $\dot{z}$ estimate is used in the dynamic model in Equation 2.11. If $\dot{z}$ is estimated to be less than $\dot{z}_{th,k}$, then $\dot{z}$ in the dynamic model is set equal to $\dot{z}_{th,k}$. Likewise, if $\dot{z}$ is estimated to be greater than $\dot{z}_{th,k+1}$, then $\dot{z}$ used in the dynamic model is set equal to $\dot{z}_{th,k+1}$. As a result of this design, the dynamic model in the filter bank which corresponds to the current vertical speed of the aircraft will propagate the estimated altitude more accurately than the other filters, yielding the lowest residual error at the next measurement timestep (neglecting measurement noise). This filter will therefore yield the highest likelihood value in Equation 2.6, and the mode probability for this filter will be pushed higher per Equation 2.7.

Table 2.1: IMM Filter Thresholds

Threshold	Vertical Speed ($\dot{z}_{th,k}$), ft/min	Heading Rate ($\dot{\psi}_{th,k}$), deg/s	Acceleration ($\dot{u}_{th,k}$), m/s^2	Roll Angle, deg
1	∞	$-\infty$	$-\infty$	-52.5
2	300	-2.3	-0.1g	-37.5
3	90	2.3	0.1g	-15
4	-90	∞	∞	15
5	-300	–	–	37.5
6	-∞	–	–	52.5

g = 32.12 ft/s^2

The goal of the heading rate IMM filter is to estimate the probability that the vehicle is in a turning flight condition using filter modes corresponding to right yaw rate, left yaw rate, and constant yaw rate flight conditions. Likewise, the acceleration IMM filter estimates the probability that the aircraft is in a longitudinal acceleration or deceleration condition. The state vectors for these IMM filters (labeled $\vec{s}_2$ and $\vec{s}_3$, respectively) are given in Equation 2.12, while the measurement vectors (labeled $\vec{r}_2$ and $\vec{r}_3$, respectively) are given in Equation 2.13.

$$\vec{s}_2 = \begin{Bmatrix} \psi \\ \dot{\psi} \end{Bmatrix} \qquad \vec{s}_3 = \begin{Bmatrix} u \\ \dot{u} \end{Bmatrix} \tag{2.12}$$

$$\vec{r}_2 = \begin{Bmatrix} \psi^* \\ \dot{\psi}^* \end{Bmatrix} \qquad \vec{r}_3 = \{u^*\} \tag{2.13}$$

Similar to the vertical speed IMM filter, three mode-matched Kalman filters are defined within each of the heading rate and acceleration IMM filters. The dynamic models for the k^{th} filter of the heading rate and acceleration IMM filters are given respectively by Equation 2.14 and Equation 2.15,

$$\vec{s}_2^{\,k} = \begin{Bmatrix} \hat{\dot{\psi}} \\ 0 \end{Bmatrix}, \qquad \hat{\dot{\psi}} = \begin{cases} \overline{\dot{\psi}} & \text{if } \ \overline{\dot{\psi}} \in [\dot{\psi}_{th,k}, \dot{\psi}_{th,k+1}] \\ \dot{\psi}_{th,k} & \text{if } \ \overline{\dot{\psi}} < \dot{\psi}_{th,k} \\ \dot{\psi}_{th,k+1} & \text{if } \ \overline{\dot{\psi}} > \dot{\psi}_{th,k+1} \end{cases} \tag{2.14}$$

$$\vec{s}_3^{\,k} = \begin{Bmatrix} \hat{\dot{u}} \\ 0 \end{Bmatrix}, \qquad \hat{\dot{u}} = \begin{cases} \overline{\dot{u}} & \text{if } \ \overline{\dot{u}} \in [\dot{u}_{th,k}, \dot{u}_{th,k+1}] \\ \dot{u}_{th,k} & \text{if } \ \overline{\dot{u}} < \dot{u}_{th,k} \\ \dot{u}_{th,k+1} & \text{if } \ \overline{\dot{u}} > \dot{u}_{th,k+1} \end{cases} \tag{2.15}$$

Where $\overline{\dot{\psi}}$ an $\overline{\dot{u}}$ are the IMM filters' estimates of the yaw rate and change in forward speed. As in the vertical speed IMM, the heading rate and acceleration used by the k^{th} filters to drive the dynamic models are saturated between the threshold values for each filter ($\dot{\psi}_{th,k}$ and $\dot{\psi}_{th,k+1}$) for the heading rate IMM filter, and $\dot{u}_{th,k}$) and $\dot{u}_{th,k+1}$). The values used for these thresholds in this study are given in Table 2.1. The mode probabilities for the heading rate IMM filter are p_2^1, p_2^2, and p_2^3, corresponding to a left yaw rate, approximately constant yaw rate, and right yaw rate, respectively. Similarly, the probabilities output from the acceleration filter bank are denoted as p_3^1, p_3^2, and p_3^3 which correspond to deceleration, approximately constant speed, and acceleration, respectively. These mode probabilities are used subsequently in the joint probability calculation to estimate the final regime probabilities.

2.3.4 Bank Angle Filter

One issue that arises when using an IMM to distinguish different maneuver regimes is that some maneuver regimes do not exhibit distinct dynamic modes, but rather they are differentiated by some boundary in the kinematic state space. An example is the two regimes: Right Turn at 30°Angle of Bank (AOB), and Right Turn at 45°AOB. These two maneuvers exhibit the same overall dynamics, with the only difference between them being the estimated value of the roll angle state. While the modes of the vertical velocity filters differ in the range of values of a dynamic state (vertical velocity), the turn angle regimes differ only in the value of a kinematic state (roll angle). Thus, an IMM cannot be used to differentiate between them.

An alternative approach that is consistent with the overall probabilistic framework is to use the state probability density estimated by a standard Kalman filter to estimate the probability associated with different regimes. This process is used in formulation of the bank angle filter. This fourth filter “bank”, which is actually a single Kalman filter, has state and measurement vectors given by,

$$s_4 = \{\phi\} \tag{2.16}$$

$$r_4 = \{\phi^*\} \tag{2.17}$$

This filter uses a white noise process model given by $\dot{s}_4 = \{0\}$. At each measurement update, the filter produces a mean estimated roll angle value $\overline{\phi}$ and a standard deviation of the roll estimate error σ_ϕ, as depicted in Figure 2.3. Define a set of six threshold values $\phi_{th,k}$, $k = \{1, ..., 6\}$ that define the boundaries between turning regimes. Several of these boundaries are illustrated in Figure 2.3, with values provided in Table 2.1. At each measurement update, the resulting roll angle Probability Density Function (PDF) defined by the Gaussian distribution with mean $\overline{\phi}$ and standard deviation σ_ϕ is processed to obtain the probability that the actual roll angle lies within certain bounds. This is done by integrating the PDF within the boundaries of each threshold as shown in Figure 2.3, resulting in the probability of the state existing within each threshold boundary. The thresholds selected for this filter in Table 2.1 are designed to differentiate between the regimes Left Turn 45°AOB, Left Turn 30°AOB, Level Flight, Right Turn 30°AOB, and Right Turn 45°AOB, respectively. Thus, the filter yields five probability values (p_4^1, p_4^2, p_4^3, p_4^4, and p_4^5) corresponding to each of these regimes.

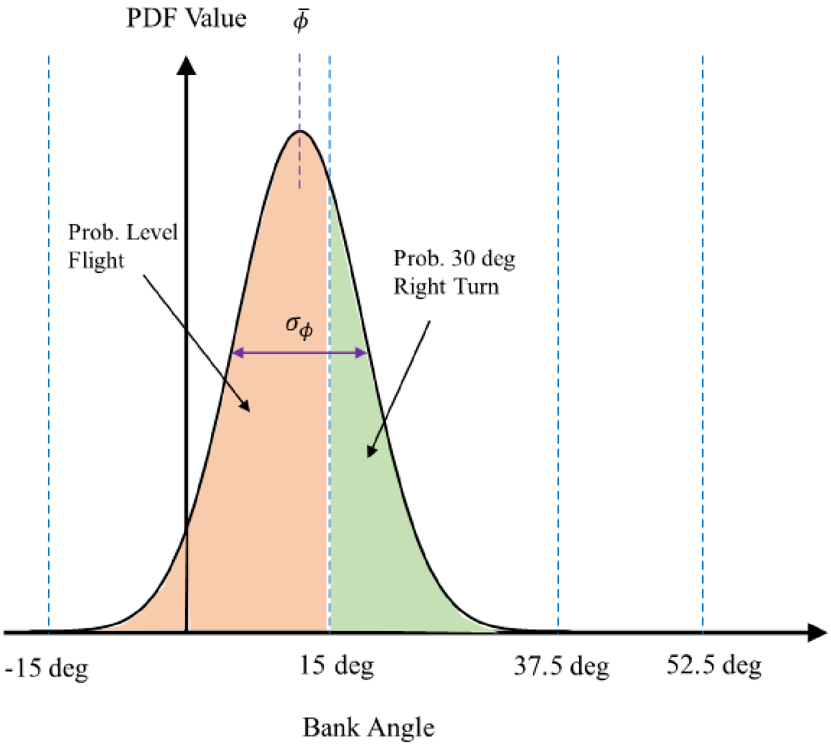

Figure 2.3: Diagram of Probabilistic Roll Angle Estimate.

2.3.5 Translational Velocity Filter

In general, rotorcraft regimes may be placed in two categories: hover-based regimes, and forward flight-based regimes (with a few exceptions such as ground-based maneuvers, sideward and rearward flight, etc.). The goal of the translational velocity filter bank is to estimate the probability that the helicopter is in a hover-based regime, or a forward flight-based regime in one of several speed ranges. Ideally, this filter would be implemented in an analogous fashion to the vertical speed IMM filter in which the state vector is composed of the position and velocity states, and each mode-matched filter propagates a dynamic model that enforces saturated limits on the velocity estimate used in propagation. However, this requires measurements of the kinematic state, i.e. position, in order to implement because the residual error from each mode-matched filter is computed between the propagated position and measured position. This residual error is then used to compute the mode probabilities for each filter in the IMM. In the case of the vertical speed IMM filter, altitude measurements are typically available. However, for operational security reasons, it is oftentimes desired that location information not be used in the regime recognition process as such data may be sensitive in military missions. Thus it is assumed that the x and y position states of the aircraft are not available for use by the regime recognition algorithm.

Table 2.2: Thresholds for Translational Velocity Filter

Threshold	Value
$u_{th,1}$	0
$u_{th,2}$	$0.05V_h$
$u_{th,3}$	$0.15V_h$
$u_{th,4}$	$0.25V_h$
$u_{th,5}$	$0.35V_h$
$u_{th,6}$	$0.45V_h$
$u_{th,7}$	$0.55V_h$
$u_{th,8}$	$0.65V_h$
$u_{th,9}$	$0.75V_h$
$u_{th,10}$	$0.85V_h$
$u_{th,11}$	$0.95V_h$
$u_{th,12}$	$1.05V_h$
$u_{th,13}$	∞

As a result of this restriction, the translational velocity filter is implemented in an analogous manner to the bank angle filter, using a single Kalman filter. The state and measurement vectors for this filter are given by $s_5 = \{u\}$ and $r_5 = \{u^*\}$, respectively, and a white noise process model is used. The resulting state PDF is integrated inside the boundaries listed in Table 2.2 to determine the probability that the aircraft is in a hover-based regime or within a forward speed regime in a given speed range. Table 2.2 lists the boundaries of integration defining each of the 12 speed regions, yielding probabilities $\{p_5^1, p_5^2, ..., p_5^{12}\}$. Note that for the purposes of this work, only positive vehicle airspeeds are considered and thus a rearward flight regime cannot be detected with this implementation of the algorithm. However, simply extending the threshold set found in Table 2.2 into the negative direction, and modifying the appropriate regime joint probability formation (Table 2.3) is all that is required.

2.3.6 Joint Probability Calculation

The final step in the regime recognition process is to estimate the probabilities associated with each regime. To obtain the probabilities of each regime, the probabilities from each of the five filters can be multiplied together to form joint probabilities, which are of the form $p_1^i \times p_2^j \times p_3^k \times p_4^q \times p_5^r$ where $i \in \{1, ..., 5\}, j \in \{1, 2, 3\}, k \in \{1, 2, 3\}, q \in \{1, ..., 5\}$, and $r \in \{1, ..., 12\}$. Thus, there is a total of 2,700 unique joint probabilities (i.e., the number of combinations of individual filter probabilities). These joint probability values may be grouped together as appropriate into maneuver regime categories. A regime is therefore a label applied to certain groups of joint probabilities.

Before defining the specific joint probabilities for each maneuver, some additional notation for the mode probabilities is defined for convenience. First, the mode probabilities for the translational velocity filter bank can be grouped into categories for hover-based and forward flight-based maneuvers. Define the hover-based regime probability as,

$$p_{hov} = p_5^1 \tag{2.18}$$

and the total forward flight-based regime probability as,

$$p_{trans} = \sum_{i=2}^{12} p_5^i \tag{2.19}$$

When assigning regime labels to sets of joint probability values, it is oftentimes the case that a particular regime label will apply regardless of the probabilities associated with a particular bank. For instance, in the "hover" regime, the vehicle roll angle may be irrelevant and thus the hover regime label definition will not depend on the probabilities output from the roll angle filter. For notational convenience in these cases, denote the sum of probabilities across an entire filter m as,

$$p_m^{all} = \sum_i^n p_m^i \tag{2.20}$$

where n is the number of probability values output from filter m. Of course, the sum in Equation 2.20 is equal to 1, but because the regime labels will be assigned to complete joint probabilities from each of the five filters, the notation in Equation 2.20 is maintained in the regime definitions for consistency.

Table 2.3 presents the maneuver regimes implemented, along with the formulas for computing their probabilities in terms of the individual probabilities from each filter. As an example, the hover regime probability is defined as the probability that the vehicle is not climbing or descending (p_1^3) multiplied by the probability that the vehicle is not changing heading (p_2^2) multiplied by the probability that the vehicle forward speed is near zero (p_{hov}). The other quantities in this formula $(p_3^{all}$ and $p_4^{all})$ are equal to one and are included only to ensure that probabilities from all filters are represented in each formula. Note that the joint probabilities listed in Table 2.3 represent only 1,788 of the 2,700 possible combinations of individual filter probabilities. The remaining 912 combinations represent either flight regimes that are not implemented in this study, or non-physical flight regimes (such as Right Turn 30°AOB with a left-ward heading rate in forward flight at $0.5V_h$, which may not be physically possible for the helicopter to fly). Thus, the remaining joint probabilities that are not assigned via the groupings in Table 2.3 are summed together to form the probability of a general *unknown regime*. This is analogous to the "Unknown" label assigned by threshold-based regime recognition codes if a flight regime is processed that conflicts with all of the threshold definitions. The total probability of the regimes defined in Table 2.3, plus the Unknown regime probability, sums to 1.

Table 2.3: Regime Probability Formulas

Regime	Formula
Hover	$p_1^3 \times p_2^2 \times p_3^{all} \times p_4^{all} \times p_{hov}$
Left Hover Turn	$p_1^3 \times p_2^1 \times p_3^{all} \times p_4^{all} \times p_{hov}$
Right Hover Turn	$p_1^3 \times p_2^3 \times p_3^{all} \times p_4^{all} \times p_{hov}$
Axial Climb	$(p_1^1 + p_1^2) \times p_2^2 \times p_3^{all} \times p_4^{all} \times p_{hov}$
Axial Descent	$(p_1^4 + p_1^5) \times p_2^2 \times p_3^{all} \times p_4^{all} \times p_{hov}$
Forward Flight $0.xV_h$	$(p_1^2 + p_1^3 + p_1^4) \times p_2^{all} \times p_3^2 \times p_4^3 \times p_5^{x+1}$
Accelerate	$(p_1^2 + p_1^3 + p_1^4) \times p_2^{all} \times p_3^3 \times p_4^3 \times p_{trans}$
Decelerate	$(p_1^2 + p_1^3 + p_1^4) \times p_2^{all} \times p_3^1 \times p_4^3 \times p_{trans}$
Left Turn 30°AOB	$(p_1^2 + p_1^3 + p_1^4) \times p_2^{all} \times p_3^2 \times p_4^2 \times p_{trans}$
Left Turn 45°AOB	$(p_1^2 + p_1^3 + p_1^4) \times p_2^{all} \times p_3^2 \times p_4^1 \times p_{trans}$
Right Turn 30°AOB	$(p_1^2 + p_1^3 + p_1^4) \times p_2^{all} \times p_3^2 \times p_4^4 \times p_{trans}$
Right Turn 45°AOB	$(p_1^2 + p_1^3 + p_1^4) \times p_2^{all} \times p_3^2 \times p_4^5 \times p_{trans}$
Descent	$p_1^5 \times p_2^{all} \times p_3^{all} \times p_4^3 \times p_{trans}$
Climb	$p_1^1 \times p_2^{all} \times p_3^{all} \times p_4^3 \times p_{trans}$
Left Descending Turn	$p_1^5 \times p_2^{all} \times p_3^{all} \times (p_4^1 + p_4^2) \times p_{trans}$
Left Climbing Turn	$p_1^1 \times p_2^{all} \times p_3^{all} \times (p_4^1 + p_4^2) \times p_{trans}$
Right Descending Turn	$p_1^5 \times p_2^{all} \times p_3^{all} \times (p_4^4 + p_4^5) \times p_{trans}$
Right Climbing Turn	$p_1^1 \times p_2^{all} \times p_3^{all} \times (p_4^4 + p_4^5) \times p_{trans}$

2.4 Evaluation Metrics

When developing regime recognition algorithms, validation must be performed using simulated or experimental scripted flight test data to ensure suitable accuracy. For instance, Aeronautical Design Standard 79, entitled "Condition Based Maintenance System for U.S. Army Aircraft," [8], dictates required levels of accuracy for regime recognition codes. Thus, for a particular code some method to evaluate accuracy is necessary both for potential certification and for performance comparisons to other codes. This section will introduce evaluation metrics for the probabilistic classification strategy employed in this work. Note that, in so-called ordinary (or deterministic) classification, an observation is assigned to a single class deterministically – i.e. the classifier outputs a single class label for each observation. In contrast, the probabilistic classifier derived here outputs a probability distribution over all of the classes for each sample observation.

Performance of an ordinary classifier can be measured through the confusion matrix

[70]. The confusion matrix is a table showing the true class and the hypothesized class, and several common metrics such as the false positive rate, true positive rate, accuracy, precision, and recall can be computed from it [71]. For a probabilistic classifier, application of the confusion matrix is not as straightforward because each observation results in a probability distribution over the classes, rather than a single classification. Following the technique proposed by Wang *et al.* [72], a *probabilistic confusion matrix* is used in this work wherein the confusion matrix elements are computed as a weighted sum of each classification and the weights are the probability values assigned to the hypothesized class. To formalize this, the (i, j) element of the $n \times n$ confusion matrix is given by,

$$C_{ij} = \frac{\sum_{k \in K} p(i, k)}{\sum_{r=1}^{n} C_{rj}} \tag{2.21}$$

where K is the set of all timesteps where maneuver regime j was actually flown, and $p(i, k)$ is the probability value assigned to maneuver i at timestep k. Note that the matrix elements are normalized by the sum of each column, which is equal to the total number of timesteps that the regime is flown. This has the effect of weighting the regimes such that a “perfect classifier” would return all ones down the main diagonal, and zeroes everywhere else. Also note that this general definition of the confusion matrix can be applied to deterministic classifiers as well, since in the deterministic case $p(i, k) = \{0, 1\} \; \forall \; i, k$. Thus, this generalized confusion matrix can be used to compare performance of the proposed probabilistic algorithm with a deterministic threshold-based one.

A second measure that will be used in this work to measure classification performance is the Receiver Operating Characteristic (ROC) graph [71]. ROC graphs are a method to visualize classifier performance, and can be applied effectively in the case of probabilistic classifiers. To introduce ROC, define a true positive t_{ii} as an instance (timestep) in which the classifier hypothesized regime i, and the actual regime flown was i. Denote anytime that the actual maneuver i was flown as t_i. The true positive rate for regime i is then,

$$TP_i = \frac{\sum t_{ii}}{\sum t_i} \tag{2.22}$$

which has an ideal value of 1. Likewise, define a false positive t_{ij} as an instance in which the classifier hypothesized regime i, and the actual regime flown was j where $i \neq j$. Denote anytime that the actual maneuver flown was $j \neq i$ as t_j. The false positive rate for regime i can then be defined as,

$$FP_i = \frac{\sum t_{ij}}{\sum t_i} \tag{2.23}$$

which has an ideal value of zero. For a probabilistic classifier, these definitions cannot be immediately applied since each timestep classification results in a probability distribution rather than a singular hypothesis. To use these metrics for a probabilistic classifier, the probabilities of each regime can be thresholded by a value $r \in [0, 1]$ such that, if the probability of regime i is greater than r, then the classifier is said to have hypothesized regime i (and thus t_i is incremented). Note that if $r < 1$, this may result in multiple regime hypotheses for each observation, since the probabilities of multiple regimes may be above the threshold r.

A receiver operating characteristics graph is formed by applying a given threshold r to the classification results, measuring TP_i and FP_i, and plotting them on the x and y axes, respectively. This result is then repeated for a discretized set of r values between 0 and 1, resulting in a line. An example ROC graph is shown in Figure 2.4 (black curve). Note that when $r = 1$, the probabilistic classifier never selects regime i and thus commits no false positive errors but also gains no true positives (thus, the point $(0, 0)$). Similarly, when $r = 0$, the probabilistic classifier always selects regime i and thus has a perfect true positive score of 1, but also a maximum false positive score of 1.

The integrated area under the ROC curve is a single metric that can be computed to measure classification performance. A perfect classifier will yield $TP_i = 1$ and FP_i = 0 for any r, meaning that the RR code assigned 100% probability to the correct (flown) regime at every timestep. This results in an optimal ROC curve as shown in blue in Figure 2.4. The integrated area under this curve is 1, which is the optimal Area Under the [ROC] Cuve (AUC) value. Likewise, the worst-case ROC curve is shown in red in Figure 2.4. For this classifier, $TP_i = FP_i$ for any r which represents the strategy of randomly guessing a class at each observation. No realistic classifier should perform worse than this, and thus the red curve represents worst-case performance with an associated worst-case AUC value of 0.5. A deterministic classifier has no r value, but its performance can still be plotted as an ROC curve. For a deterministic classifier, there is a single true positive and false positive rate for a classified sequence. This value can be plotted on the ROC graph, and connected to the points $(0, 0)$ and $(1, 1)$ as shown in green in Figure 2.4. The AUC value for the deterministic classifier can then be compared to that from a probabilistic classifier.

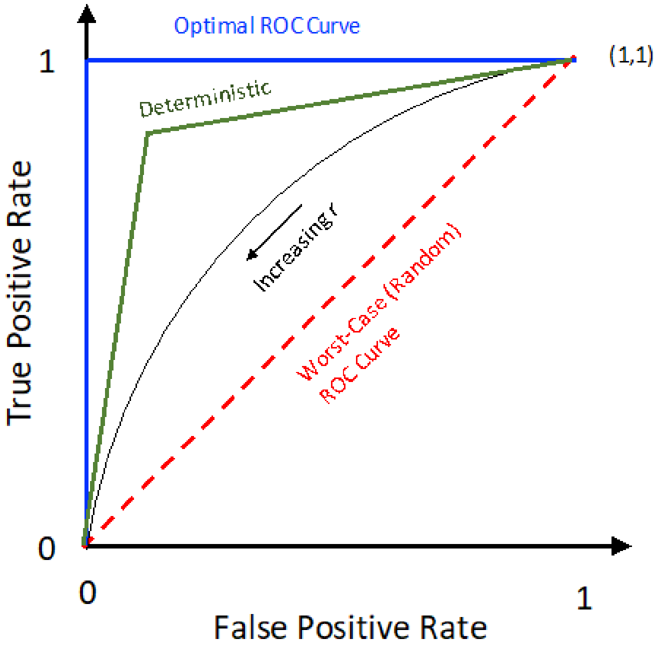

Figure 2.4: Example ROC Graph Showing Various Probabilistic Classifiers.

2.5 Simulation Model

The simulation model used to generate the HUMS flight data for use in the following examples is from the Georgia Institute of Technology's Helicopter Integrated Vehicle Simulation Environment (HIVE). HIVE utilizes a high-fidelity dynamic flight model based on the ARMCOP model developed by Talbot *et al.* and Chen (Refs. [73–75]). HIVE also incorporates several advanced modeling additions for increased accuracy including ground effect, dynamic inflow, blade stall, and quasi-steady second-order flapping dynamics. The aircraft model contains 22 total states consisting of: twelve rigid-body vehicle states, three dynamic inflow states, one for the rotor speed, and six for the main rotor flapping states. The simulation propagates all model states using a 4^{th} order Runga-Kutta numerical integration method, with a timestep of 1 ms.

Rotor Forces and Moments

The forces and moments generated by the main rotor are calculated using a numerical blade-element approach. While dynamic stall and compressibility effects are neglected, complete aerodynamic look-up tables (See Ref. [76]) are used to model any blade elements in static stall. For computation, each blade was divided into 15 radial sections, and integrated at 30 azimuthal locations around the main rotor head. The model includes first-order harmonic flapping, with flapping states $\beta_0,\ \beta_{1s},$ and β_{1c}, which are propagated using a second-order flapping equation [77]. HIVE uses a buildup approach where computed forces and moments are summed to obtain the aerodynamic forces exerted on the rest of the vehicle. Contributions from the fuselage and stabilizer components are included using a linear model of the aerodynamics, and down-wash effects are included using a bluff-body model. Forces and moments exerted by the tail rotor are calculated using Newton-Raphson iteration with the assumption of uniform inflow. The rotation rate of the tail rotor is determined by scaling the main rotor rotation rate by a constant factor. Additional details regarding the rotor model can be found in Refs. [78, 79].

Dynamic Inflow

A variety of dynamic inflow models have been developed in the past several decades. These models include (but are not limited to) linear models [80], free-vortex wake models [81], and state-space wake models derived using indicial theory [82]. These models range in complexity, and the trade off between accuracy and computational burden must be determined based on the purpose of the studies to be performed. Due to the versatile nature of the HIVE simulation platform, HIVE utilizes the Pitt and Peters model (detailed in References [83, 84]) for computing the dynamic inflow for the main rotor. This three-state model is widely used for its balance of accuracy and computational efficiency, and has been extensively studied and experimentally verified [85]. The three states (denoted as $\lambda_0,\ \lambda_s,$ and λ_c) describe the distribution of the induced flow ratio over the disk swept by the main rotor according to the following equation,

$$\lambda_i(r,\psi_{MR}) = \lambda_0 + \lambda_s \frac{r}{R} sin(\psi_{MR}) + \lambda_c \frac{r}{R} cos(\psi_{MR}) \tag{2.24}$$

where r is the radial distance from the rotor hub, R is the main rotor radius, and ψ_{MR} is the main rotor azimuth angle in radians. These states are then evolved according to the dynamic equation given by,

$$[\mathbf{M}] \begin{bmatrix} \dot{\lambda_0} \\ \dot{\lambda_s} \\ \dot{\lambda_c} \end{bmatrix} + [\hat{\mathbf{L}}]^{-1} \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} = \mathbf{C} \tag{2.25}$$

where $\mathbf{M}$ and $\hat{\mathbf{L}}$ are matrices that are calculated based on blade loads, aerodynamics, rotor inertial properties, as well as sideslip and wake angles, and $\mathbf{C}$ is the resulting vector of force and moment coefficients expressed in the body-frame. Complete details of the inflow model can be found in References [83, 84].

Ground Effect

The HIVE also includes a ground effect correction for when the vehicle is within two rotor diameters of the ground. This ground effect correction modifies the dynamic inflow given by Equation 2.24. At steady-state (i.e. $\dot{\lambda} = 0$) eq. (2.24) yields,

$$\mathbf{C} = [\hat{\mathbf{L}}]^{-1}\lambda_{ss} \tag{2.26}$$

where λ_{ss} is the three state vector representing the steady-state inflow. An inflow correction factor of $\Delta w/w_0$ (as described by Heyson [86]) can be used to model ground effect in forward flight. The correction factors obtained by Heyson – using experimental data – is only valid at the center of the rotor, and decreases from root to tip of the blades, and thus a 0.8 multiplier is used to compute the average value of the inflow correction across the entire rotor disk [86]. The steady-state inflow is then modeled by,

$$\lambda_{ss,IGE} = \left(1 - 0.8\frac{\Delta w}{w_0}\right)\lambda_{ss} \tag{2.27}$$

This correction is then applied to Equation 2.24 by adjusting the dynamic inflow model given by $\mathbf{C}$ so that λ tends towards $\lambda_{ss,IGE}$. Therefore, at steady-state, the following equation can be used to model the dynamic inflow when the vehicle is in ground effect (IGE),

$$[\mathbf{M}]\begin{bmatrix}\dot{\lambda}_0\\ \dot{\lambda}_s\\ \dot{\lambda}_c\end{bmatrix} + [\hat{\mathbf{L}}]^{-1}\begin{bmatrix}\lambda_0\\ \lambda_s\\ \lambda_c\end{bmatrix} = \left(1 - 0.8\frac{\Delta w}{w_0}\right)\mathbf{C} \tag{2.28}$$

Note that in Equations (2.27) and (2.28) the correction factor $\frac{\Delta w}{w_0}$ is referred to as the *ground-induced interference velocity*, and is found via a lookup table in [86] using rotor height above ground and wake angle, as inputs.

Synthetic HUMS

A companion program to HIVE is used to generate an output file that is structurally identical to that found in some real HUMS outputs used for helicopter regime recognition. This compatibility of file output allows for comparisons to be made more easily between the results of the proposed IMM regime recognition and enterprise-grade rule-based RR software. The helicopter used in this work is the Sikorsky SH-60B. HIVE includes allowable minimum and maximum actuator deflection limits and deflection rates. The main rotor collective has a range limit of 9.9°to 25.9°, the lateral cyclic limit of ±8°, and the longitudinal cyclic has a range limited to -12.9° to 16.5°. An actuator rate limit of 40°/s is enforced. Table 2.4 lists the SH-60B parameters used in the simulation studies found within this dissertation.

Table 2.4: SH-60B Model Parameters

Parameters	Value
Helicopter gross weight	16,000 lbs
Number of main rotor blades	4
Main rotor blade chord	1.73 ft
Main rotor radius	26.83 ft
Main rotor blade moment of inertia	1491 slug ft^2
Main rotor height above ground (water line)	12 ft
Main rotor normal operating speed	27.0 rad/s
Main rotor blade airfoil (simulation)	SC 1095

Virtual Pilot

In conjunction with the HIVE program, a *virtual pilot* algorithm, developed by Fowler *et al.* [87], was used. Control inputs to the simulated aircraft are driven by a so-called “virtual pilot” that is capable of executing simulated missions in a similar manner to a human pilot. This control law module plug-in has a catalog of known maneuvers, each with a corresponding control law. When a maneuver is commanded, the pre-determined control loops are activated and/or deactivated, and the appropriate setpoints are updated based on the maneuver parameters. The commanded maneuver sequence (also referred to as *flight cards*) are loaded into the program as a script prior to simulation. The HIVE, in conjunction with the *virtual pilot* algorithm, has been found to generate realistic flight data that can be used for regime recognition validation and verification [78, 87].

2.6 Results

A set of example cases is presented in this section to demonstrate performance of the IMM-based regime recognition algorithm using both simulated data for the SH-60B and experimental flight data from a generic single engine transport helicopter. The purpose of these example cases is to illustrate various aspects of algorithm performance and compare regime recognition outputs with a threshold-based algorithm.

Two examples below compare the IMM-based regime recognition algorithm outputs to those generated from a threshold-based code. The specific threshold regime recognition code used for these comparisons was developed for the SH-60B and uses hierarchical rule-based methods to classify regimes based on state thresholds. An extensive description of the algorithm and the development process underlying it is available in Refs. [52] and [87]. Note that this algorithm has been validated extensively using scripted flight test data from SH-60B test flights (where pilots fly prescribed maneuvers for specified times, creating a truth dataset against which regime recognition algorithm outputs can be compared for validation).

The IMM-based algorithm used to generate results throughout this section includes multiple tuning parameters: the measurement covariance matrices for each filter R_i, the model error covariance matrices for each filter, and the mode transition probabilities matrices for each IMM filter. These parameters were tuned using both simulated and actual flight data such that proper performance was observed in terms of reasonable rise times during maneuver transitions, robustness to noise, and accurate classification results for simulated data for which truth regimes were known. The same filter tuning was used for all examples in this section.

2.6.1 Example Flight Sequence 1

The first example uses a fairly short flight segment of simulated data. The maneuver sequence flown (in simulation) for this example is shown in Table 2.5. Beginning from forward flight at approximately 135 ft/s, the helicopter performs a gentle left turn at 16°AOB, then a slow climb at 300 feet per minute (fpm), and then recovers to forward flight. Several relevant state time histories for this flight sequence, namely roll angle, altitude, and altitude rate are shown in Figure 2.5. Note that in these figures, the solid line represents the simulated flight data and the dashed line is the state estimate from either the IMM filter or the Kalman filter associated with this state. In this initial example, no measurement noise is included (although it is in subsequent examples).

Table 2.5: Maneuver Sequence for RR Example 1

Maneuver Start Time, s	Flown Regime	Parameter
0	Forward Flight	$0.5V_h$
5	Left Turn	16°AOB
60	Climb	300 fpm
115	Forward Flight	$0.5V_h$
135	End	–

Regime recognition results for this example are shown in Figure 2.6 for the threshold-based code, and in Figure 2.7 for the IMM-based algorithm. In Figure 2.6 the classified regime is marked with a black square at each timestep. This figure shows that the threshold algorithm correctly identifies the left turn (after an initial transient during the entry to the turn) between 9.8-55.8 sec, but then for the remainder of the flight identifies the flown regime as Forward Flight $0.5V_h$. The threshold algorithm's behavior in this example is a clear illustration of the limits of threshold-based codes – if a sustained flight regime observed in measured data does not fit clearly into one of the canonical definitions for the regime set, the threshold code is forced to make a deterministic classification anyway. In this case, the left turn maneuver exceeded the bank angle threshold set in this code for a

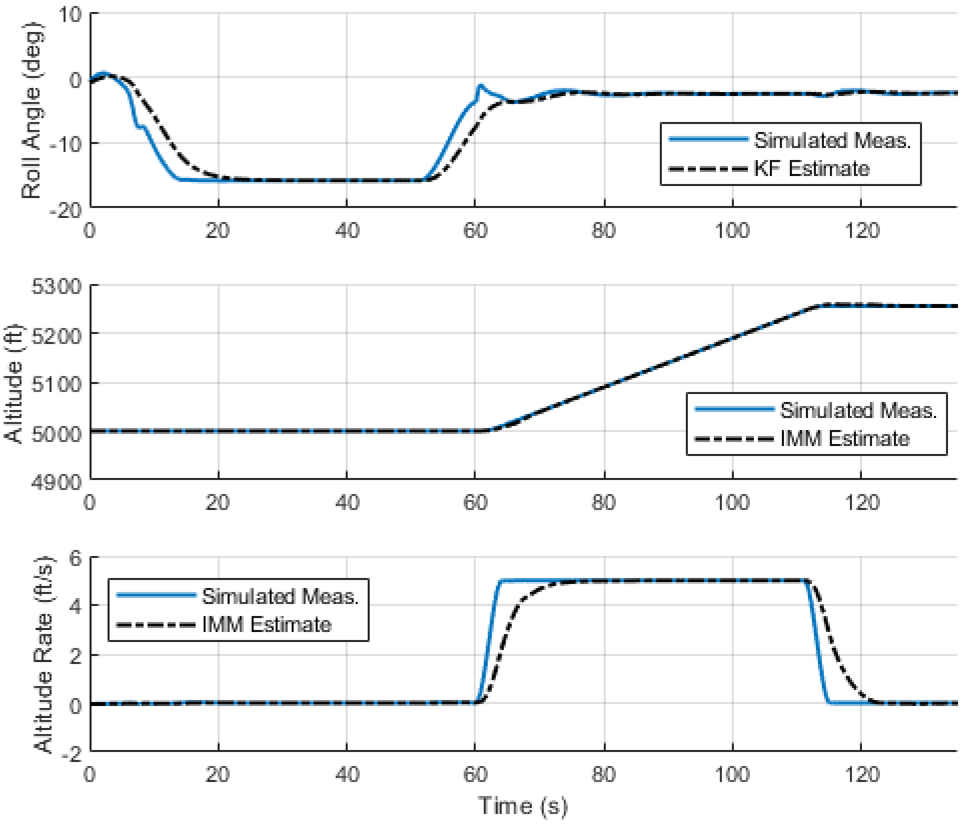

Figure 2.5: Selected State Time Histories for Example 1, Simulated Flight Data (Meas. = measurement).

"Left Turn" regime and thus this portion was classified accordingly. However, the threshold algorithm "misses" the climb regime because the climb rate is slower than that for a typical "Climb" regime as defined by the algorithm thresholds. Thus, although the slow climb is, in reality, a maneuver that is between a typical "Climb" and "Forward Flight" segment, the threshold algorithm classifies it deterministically as "Forward Flight". This deterministic classification may even lead to damage underprediction (if a shallow climb yields more component damage than "Forward Flight"), although such implications are not explored here.

Figure 2.7 shows the performance of the IMM-based algorithm for this example. Unlike the threshold-based algorithm, the IMM-based algorithm provides partial classifications to multiple regimes throughout the flight segment. As the helicopter transitions into the shallow left turn, the "Left Turn 30°AOB" increases from near zero probability to about

0.6 probability, while "Forward Flight $0.5V_h$" decreases from about 1 to about 0.4. Similarly, as the aircraft enters the slow climb segment, the "Climb" probability increases from about zero to roughly 0.5, and "Forward Flight $0.5V_h$" decreases from about 1 to about 0.5. These results make sense since the left turn and slow climb maneuvers as flown are somewhere in between the "Left Turn" and "Forward Flight" (or "Climb" and "Forward Flight") maneuvers as they would be typically flown. The IMM-based estimator provides the regime recognition code the flexibility to perform this partial classification into two regimes, rather than a threshold-based code which must perform a 100% classification into one of the two regimes. This case illustrates the difficulty in applying deterministic classification to the regime recognition problem since there are no clear, universally-recognized boundaries between regimes in the state space. In contrast, the probabilistic classifications more accurately reflect the fact that a flight condition can be a blend between two canonical regimes.

To compare performance quantitatively for this example, the AUC metrics are computed from the results in Figure 2.6 and Figure 2.7. The threshold code yielded a Forward Flight AUC of 0.709, Left Turn AUC of 0.982, and Climb AUC of 0.5. Note that the Climb AUC of 0.5 is the worst possible score for a classifier, which is due to the fact that the threshold code missed the entire climb maneuver. The IMM code AUC values are significantly higher at 0.848, 0.956, and 0.989 for Forward Flight, Left Turn, and Climb, respectively.

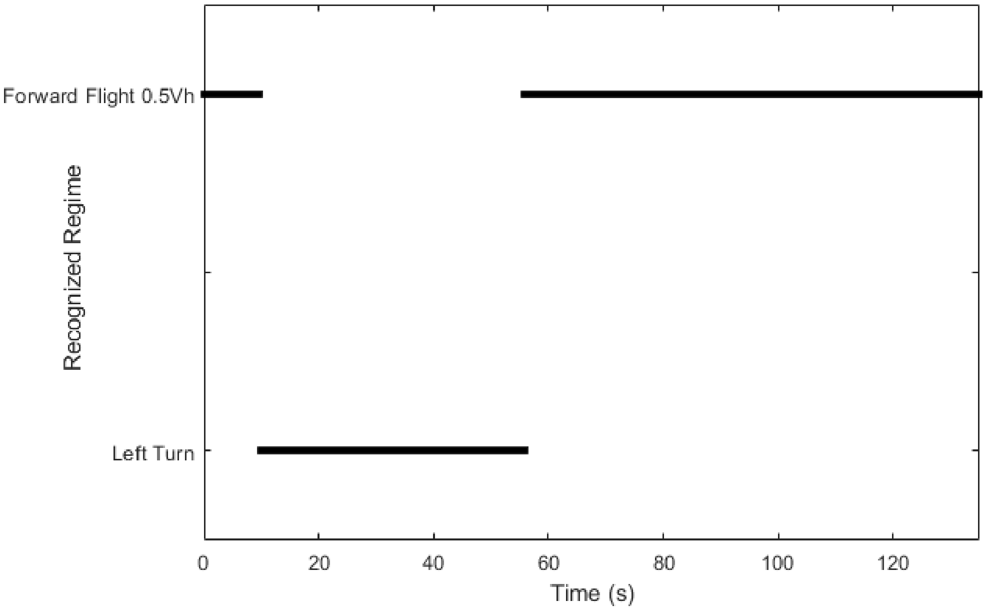

Figure 2.6: Threshold Algorithm Regime Recognition Results, Example 1.

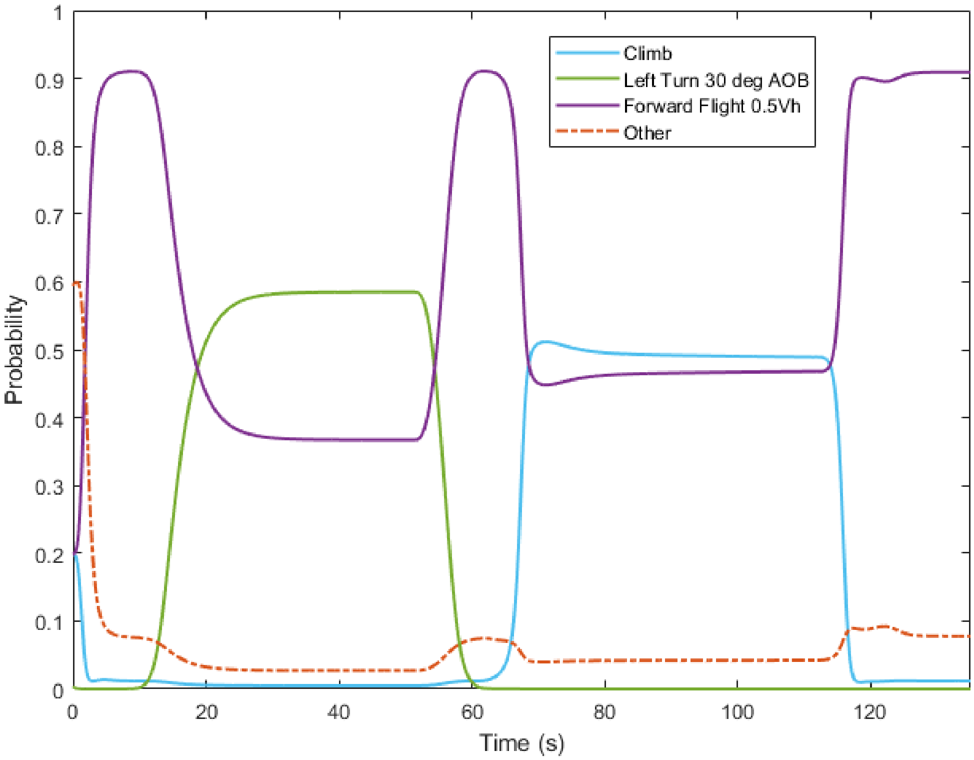

Figure 2.7: IMM-Based Regime Recognition Results, Example 1.

2.6.2 Example Flight Sequence 2

The second example considers a longer flight sequence consisting of more maneuvers to explore the performance of the IMM-based algorithm more in depth. The flight sequence flown in simulation for this example is provided in Table 2.6. Note that the aircraft performs several climbs, turns, and descents, as well as a hover-to-forward flight transition. Prior to using the data for regime recognition, zero-mean Gaussian noise is added to the state time histories, where the noise standard deviations for each state were derived from engineering judgment and experience with real HUMS measurement data. Several relevant state time histories, showing both the noisy (simulated) measurement values and the associated IMM/Kalman filter estimates from the associated filter, are shown in Figures 2.8 and 2.9.

Table 2.6: Maneuver Sequence for RR Example 2

Maneuver Start Time, s	Flown Regime	Parameter
0	Hover	–
35	Acceleration	–
50	Forward Flight	$0.5V_h$
80	Climb	500 fpm
115	Climbing Right Turn	500 fpm, 30°AOB
140	Climbing Right Turn	500 fpm, 40°AOB
165	Right Turn	40°AOB
200	Right Turn	30°AOB
230	Forward Flight	$0.5V_h$
270	Left Turn	30°AOB
305	Descending Left Turn	500 fpm, 30°AOB
330	Descent	500 fpm
360	Forward Flight	$0.5V_h$
395	End	–

Figures 2.10 to 2.12 show the regime recognition results for the two algorithms. Figure 2.10 shows regime recognition results for the threshold algorithm where the noisy measurements are directly input to the RR code. In this figure, it regularly appears that timesteps are classified into multiple regimes simultaneously – in fact, this overlap in the plotting occurs because the threshold algorithm switches classifications repeatedly between

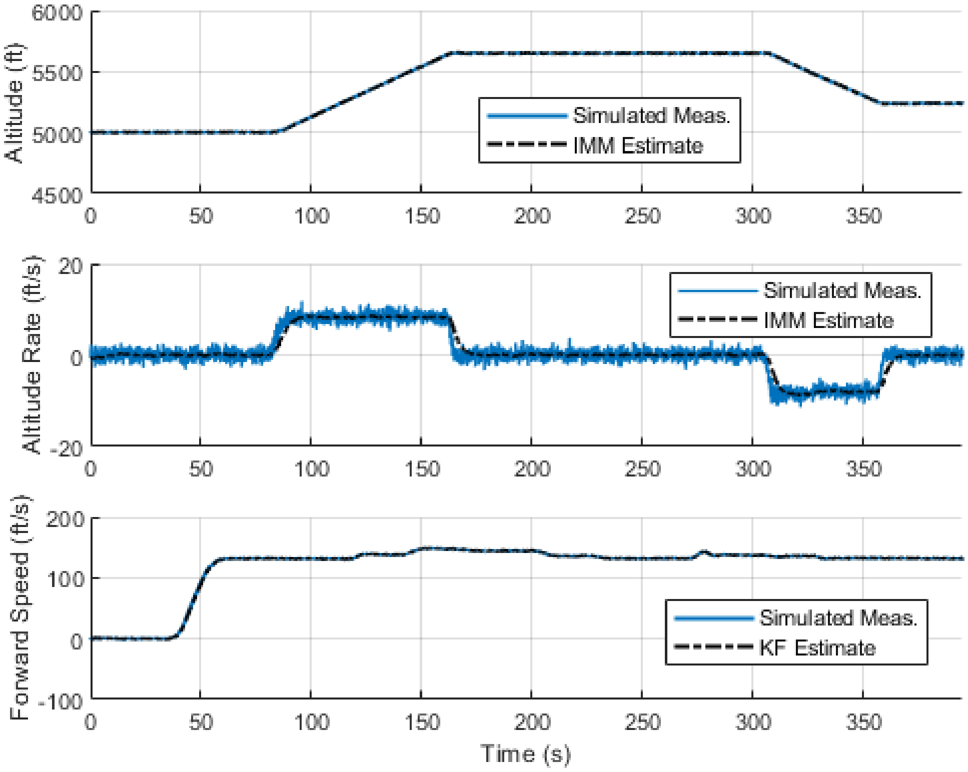

Figure 2.8: Selected State Time Histories for Example 2, Simulated Flight Data.

adjacent timesteps. This high-frequency switching (also present in Figure 2.11) appears as overlapping classifications in the plot. Note that while the regimes are recognized correctly at many timesteps, the algorithm switches rapidly between classifications even during steady-state portions of the maneuver. This is because noise in the measurements causes the states to cross the threshold boundaries between regimes almost on a per-timestep basis. Figure 2.11 shows the threshold RR code results using measurements that have been smoothed using a moving average smoother with a two-second sliding window. These results are much improved over those shown in Figure 2.10, but some high-frequency switching behavior is still notable in climbing and turning segments of the flight. Figure 2.12 shows the IMM RR code results when using the raw, noisy measurements, where the gray-scale value denotes the maneuver probability. Note that, in general, regimes are correctly classified during steady-state and uncertainty about classification, primarily during transients, is addressed by assigning non-zero probabilities across several maneuvers.

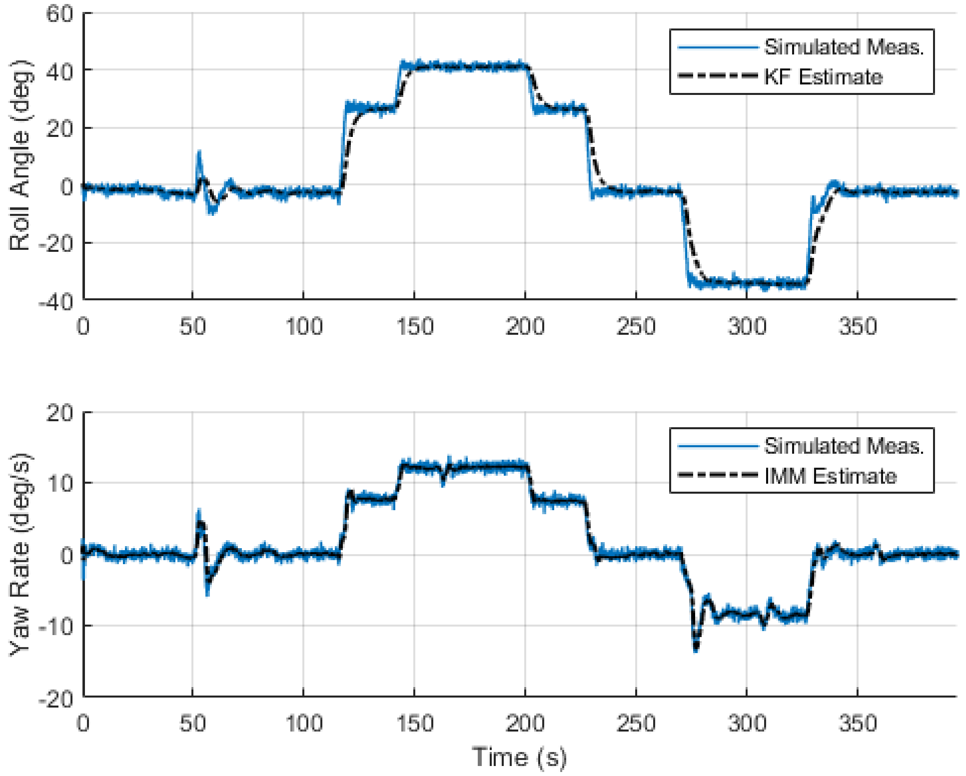

Figure 2.9: Selected State Time Histories for Example 2, Simulated Flight Data.

To quantitatively compare the results, figs. 2.13 and 2.14 show the probabilistic confusion matrices for the smoothed threshold RR results and the IMM RR results, where only rows and columns for actual flown regimes are included. These matrices can be compared qualitatively to show that the IMM RR code yields better classification performance, indicated by the fact that the confusion matrix for the IMM case more closely resembles the identity matrix. Normalized accuracy, computed as the number of true positives divided by the total number of true positives plus false positives in the normalized confusion matrix, is 64.6% for the smoothed threshold RR code versus 84.1% for the IMM RR code. Table 2.7 shows the AUC metrics for each of the regimes identified by each code, and a subset of the ROC curves for the IMM code results is shown in Figure 2.15. In Table 2.7, the average AUC is computed by summing all the AUC values multiplied by their respective fraction of the flight time executing each maneuver. For all maneuvers, the AUCs yielded by the IMM-based code are as high, or significantly higher, than those yielded by the threshold-based code. The average AUC value of 0.976 for the IMM RR code is significantly better than the 0.788 exhibited by the threshold code. In addition to the IMM algorithm's ability to partially classify flight data that occurs in between standard maneuvers, this example illustrates the benefits of the algorithm's probabilistic classifications in handling maneuver transients. Furthermore, the filtering of noisy measurements is handled naturally by the IMM algorithm's internal Kalman filters, which also serves to improve performance.

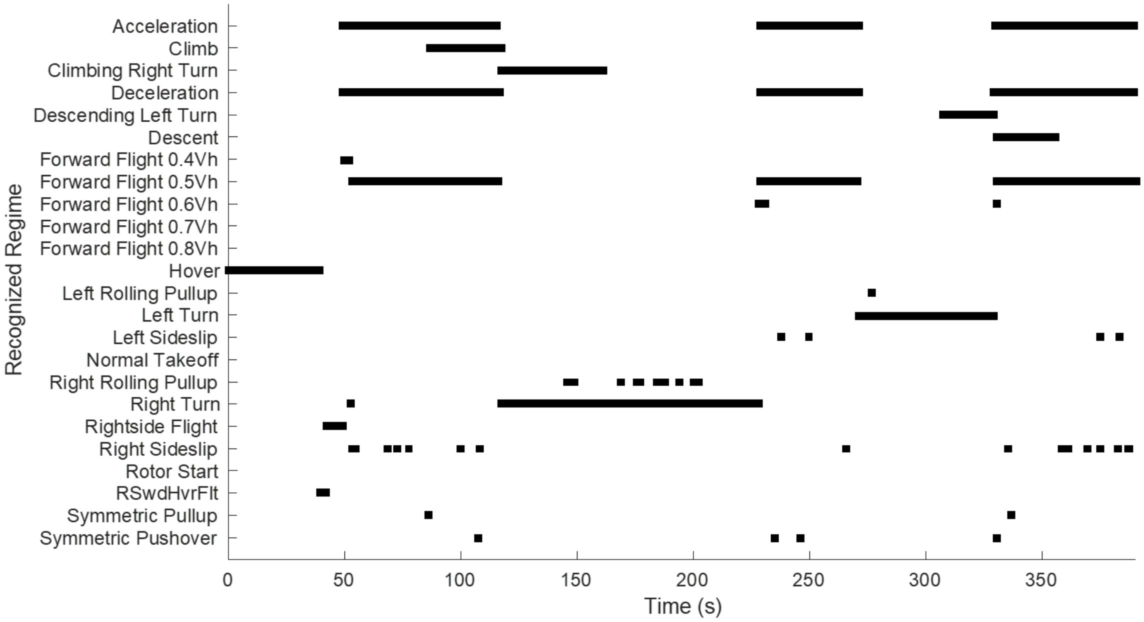

Figure 2.10: Threshold-Based RR Classification Results, Example 2, No Measurement Smoothing.

Table 2.7: AUC Values for Example 2 Regime Recognition Results

Maneuver	Threshold-Based Code	IMM-Based Code
Acceleration	0.649	0.910
Climb	0.726	0.973
Climbing Right Turn	0.694	0.983
Descending Left Turn	0.622	0.992
Descent	0.500	0.999
Forward Flight $0.5V_h$	0.885	0.972
Hover	0.996	0.996
Left Turn	0.959	0.989
Right Turn	0.947	0.999
Average (Weighted)	**0.788**	**0.976**

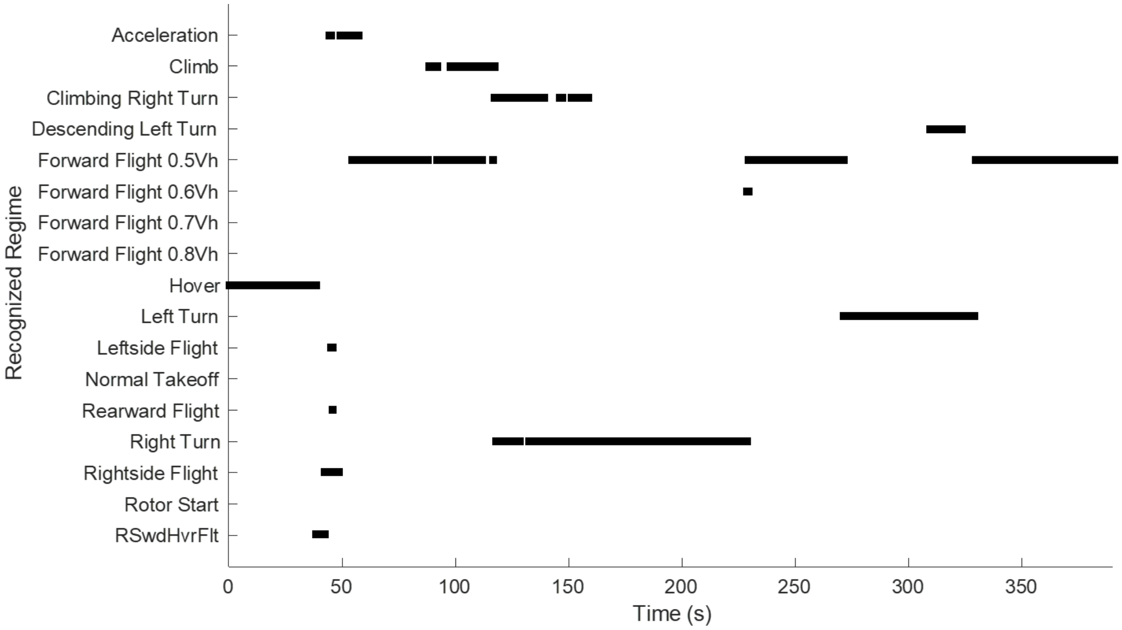

Figure 2.11: Threshold-Based RR Classification Results, Example 2, Smoothed Measurements.

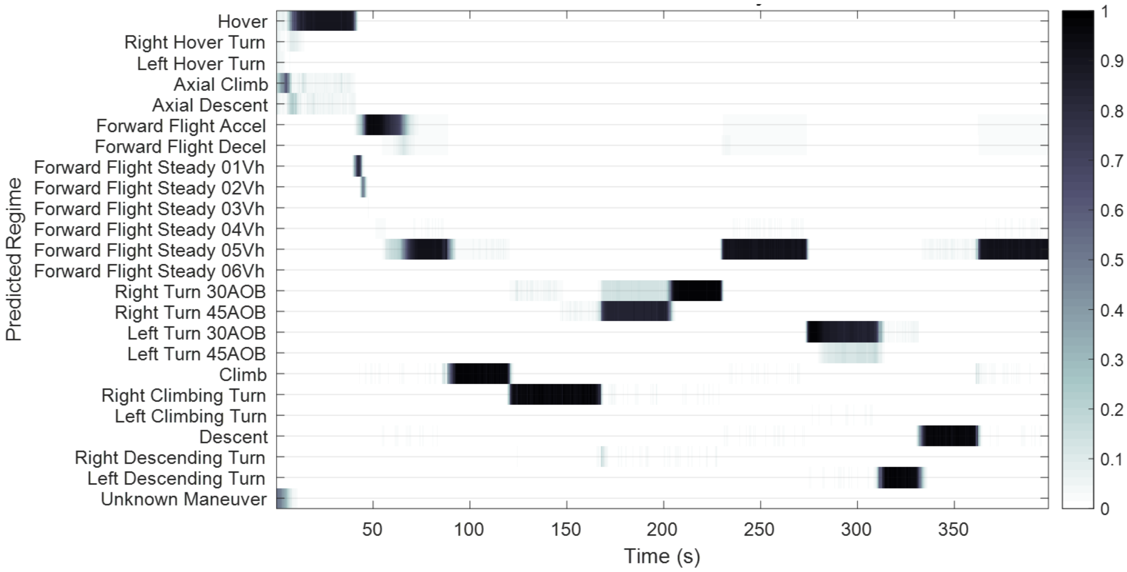

Figure 2.12: IMM-Based RR Classification Results, Example 2, No Measurement Smoothing.

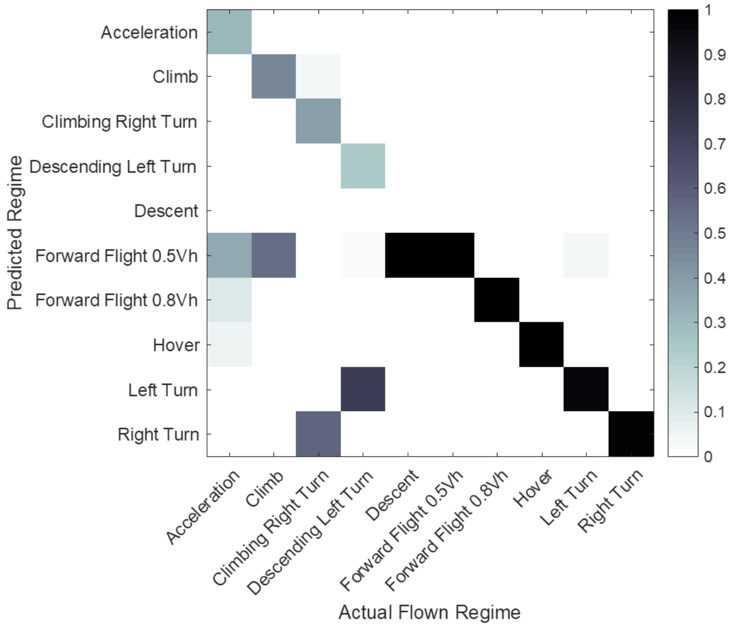

Figure 2.13: Probabilistic Confusion Matrices for Threshold RR Results.

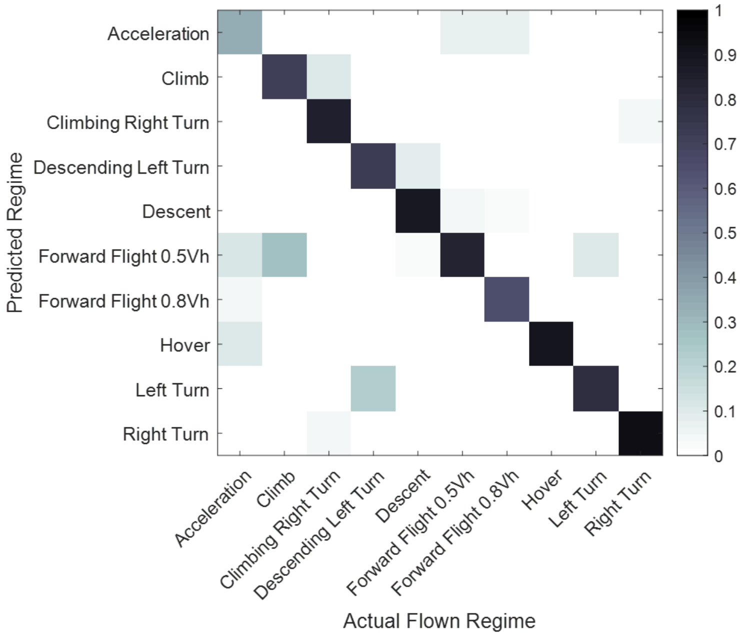

Figure 2.14: Probabilistic Confusion Matrices for IMM RR Results.

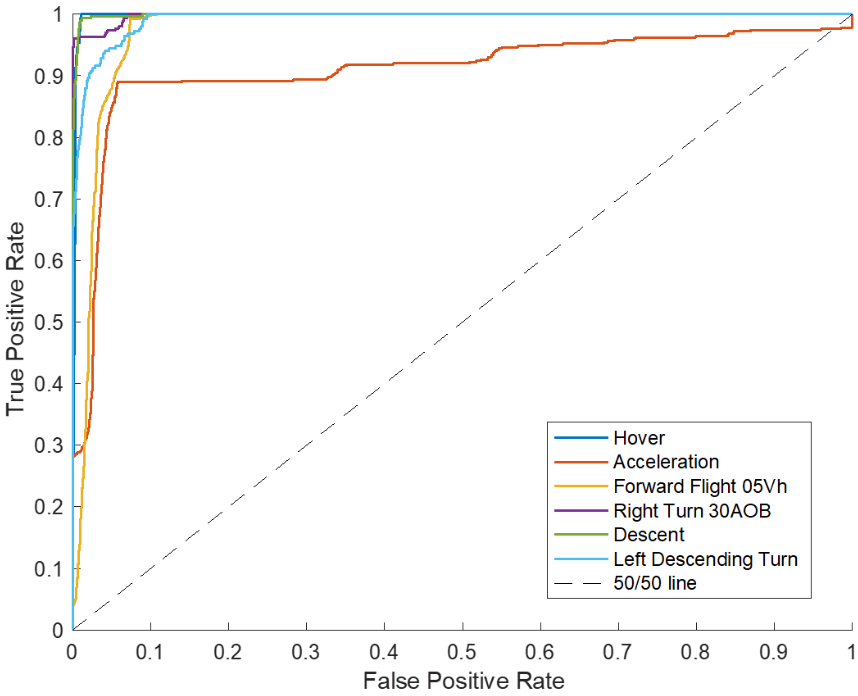

Figure 2.15: ROC Curves for Selected Regimes, IMM Regime Recognition Results.

CHAPTER 3

PROBABILISTIC DAMAGE ESTIMATION

Condition-based maintenance programs for modern helicopters rely on algorithmic techniques to estimate the useful life remaining for life-limited components. Regime recognition-based CBM programs involve a regime recognition step and a damage estimation step in which damage is calculated based on the identified regimes. However, to date there has been no method developed to convert probabilistic regime distributions to damage estimates. This dissertation proposes a technique to compute a probability distribution over the fatigue damage for each life-limited component directly from the regime probability distributions. The method treats the incurred damage at a given time as a random variable, and accumulates the total damage incurred as a sum of random variables. The damage distribution at each time is computed from the regime distribution and the regime damage rates. A primary advantage of the approach is that it captures uncertainty in the regime recognition process by treating damage as a random variable rather than a deterministic value. Simulation results illustrate the benefit of the probabilistic approach over a deterministic method, particularly for flights where there is significant uncertainty in the flown regimes.

3.1 Damage Estimation Algorithm

This section begins with an overview of the proposed Individual Aircraft Tracking CBM scheme using probabilistic RR and damage estimation. This is followed by a detailed description of the damage estimation methodology, which operates on the regime probabilities for each timestep of flight data identified by the RR code. Two additional subsections discuss a computational implementation using a Gaussian approximation for the damage distributions, as well as an analysis of the factors that drive uncertainty in the damage estimates.

3.1.1 Damage Estimation Process Overview

The IAT paradigm envisioned here is depicted in Figure 3.1. For each flight (or set of flights) of a given aircraft, HUMS data is recorded and processed by a probabilistic regime recognition module. Note that flight data may be processed after each flight, or after a batch of flights, as shown in Figure 3.1. The probabilistic RR modules may, for example, be based on the Hidden Markov Model algorithm of Refs. [54, 55, 57], the IMM scheme discussed in chapter 2 (also documented in Refs.[58, 59]), or some other method, but are differentiated from rule-based RR methods in that they produce a probability distribution over the regime set at each timestep of flight data. The output of the probabilistic regime recognition algorithm is a probability distribution over the regime set at each timestep of flight data.

These regime distributions are provided to a *Damage Computation Module (DCM)*, which is responsible for estimating the fatigue damage incurred for all life-limited components over the course of the flights being processed. Typically, a deterministic RR approach to damage calculation would apply the damage rates identified for each regime over the time duration spent in each regime to calculate the total damage on each component. However, in a probabilistic setting, regime classifications are non-deterministic at each timestep, but rather take the form of probability distributions. Thus, there is no single damage rate that can be applied to a given component at a given time, since probabilities may be assigned to several regimes at once. This necessitates the development of a complementary *Damage Computation Module* for use with probabilistic regime recognition, which is the focus of this chapter.

The Damage Computation Module proposed here treats the damage incurred by each component at each time interval of flight data as a random variable. Thus, the total damage incurred by a component over a flight (or set of flights) is a sum of random variables,

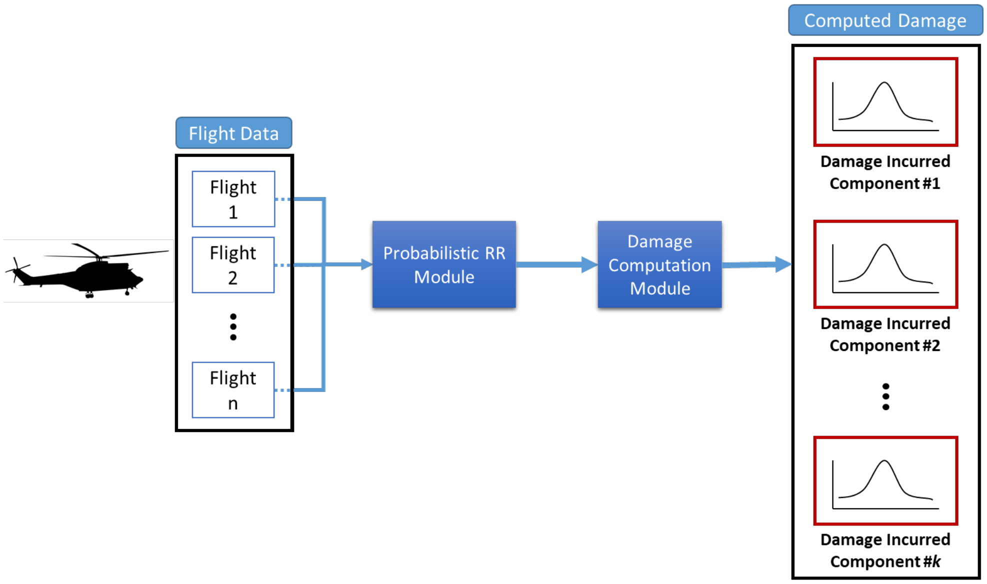

Figure 3.1: Individual Aircraft Tracking Data Flow.

which itself is a random variable. Thus, the proposed DCM converts the regime probability distributions at each timestep of flight data to damage probability distributions for each component at each timestep. Then, as depicted in Figure 3.1, the DCM accumulates the damage estimates over the course of the flights being processed to produce a total damage probability distribution for each life-limited component. A mathematical description of the DCM is provided in detail in following subsection (also documented in Refs. [88, 89]).

3.1.2 Methodology of the Damage Calculation

Useful life consumed, due to material fatigue, in a dynamic component can be calculated using Palmgren-Miner's rule of linear cumulative damage [90], given by Equation 3.1,

$$DF = \sum_{i=1}^{K} \frac{c_i}{N_i} \tag{3.1}$$

In Equation 3.1, c_i is the number of cycles a component has experienced at a given unique stress level and N_i is the number of cycles to failure at that same stress level. Failure occurs when the damage fraction $DF = 1$. Since loads themselves are never measured or estimated in the context of regime-based damage estimation, the goal of the DCM in this work is to estimate the damage fraction incurred by all life-limited components for a specific aircraft over a given set of flights. This is accomplished through the determination of damage rates for each regime and each component, which can be estimated using Palmgren-Miner's rule in conjunction with rainflow counting techniques [91]. It is assumed in this work that damage rates for each regime have already been established and are available for use by the DCM.

Let the damage rate for regime r and component k be denoted as $d_{r,k}$. Furthermore, let the regime probabilities identified at the j^{th} timestep of flight data be given by $\{p_1^{t_j}, p_2^{t_j}, ...p_n^{t_j}\}$, where $\sum_{r=1}^{n} p_r^{t_j}$ and n is the total number of regimes in the regime set (so $r \in \{1, 2, ..., n\}$). These regime probabilities form a PMF over the regime set. Then, the damage incurred by component k at timestep t_j is also a random variable, denoted as $X_{j,k}$. For the remainder of this dissertation, the subscript k denoting the component will be dropped for simplicity of notation – however, it is important to remember that the below damage computations must be performed for each component using their respective damage rates in each regime. The PMF over X_j can be calculated by assigning a probability of $\sum_{r \in R} p_r^{t_j}$ to the damage value $d_r \Delta t$ (where Δt is the length of the timestep) for all R regimes that have the same damage rate d_r for this component. The resulting damage PMF at this timestep is denoted as $P_j(X_j)$. This process models the damage incurred by a particular component at timestep j as a discrete random variable, with a PMF computed from the regime probabilities identified at timestep t_j and the damage rates in each regime.

As a simple example, consider a case with only four regimes and damage rates for the k^{th} component given by $d_r = \{0.001, 0.005, 0.001, 0.007\}$ sec^{-1} for regimes $r \in \{1, 2, 3, 4\}$. Assume that the probabilistic RR code identifies regime probabilities at timestep t_j given as $p_r^{t_j} = \{0.4, 0.1, 0.2, 0.3\}$. Then, assuming $\Delta t = 0.1$ sec, the PMF over X_j is shown in Figure 3.2. Note that the probabilities for regimes 1 and 3 contribute to the same damage value of 0.0001 because they share the same damage rate for component k. It should be mentioned that the damage rates here and elsewhere in this section are orders-of-magnitude higher than actual damage rates, and are inflated only for the purposes of illustrative examples in this section.

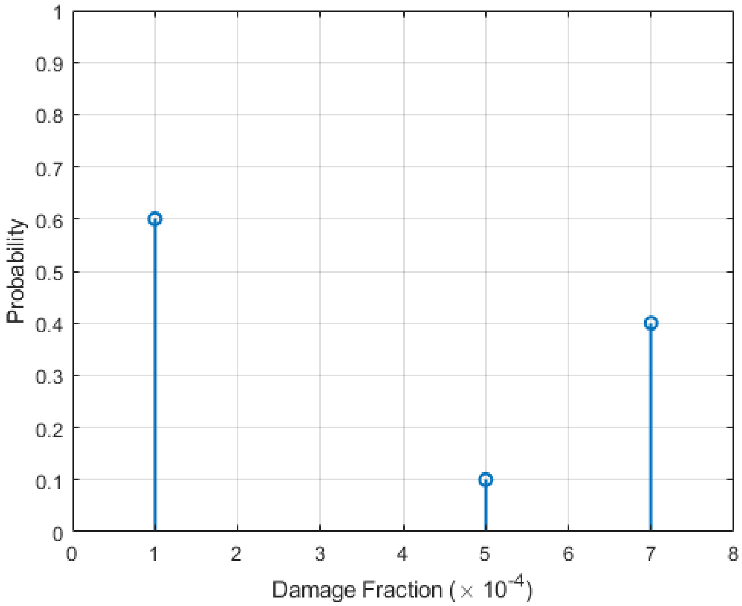

Figure 3.2: Probability Mass Function Example for Incurred Damage at Timestep j, X_j.

Because the damage incurred by the k^{th} component at a single timestep is a random variable, the damage incurred over M timesteps (or, a flight or set of flights) is also a random variable given by,

$$X_{1:M} = X_1 + X_2 + ... + X_M \tag{3.2}$$

This sum of discrete random variables may be found through the process of discrete convolution [92]. Consider the damage at two timesteps, X_1 and X_2, with PMFs given by $P_1(X_1)$ and $P_2(X_2)$, respectively. Then the PMF of $X_{1:2} = X_1 + X_2$ is given by,

$$P_{1:2}(X_{1:2}) = \sum_{X_2 \in R_2} P_1(X_{1:2} - X_2) P_2(X_2) \tag{3.3}$$

where R_2 is the support of X_2. Equation 3.3 is the standard method of computing the

PMF of the sum of two discrete random variables and can be applied recursively to find the damage PMF of $X_{1:M}$, representing the cumulative damage incurred over M timesteps.

To illustrate the process of adding two damage PMFs, consider example probabilistic RR data for two adjacent timesteps t_1 and t_2 in Table 3.1. The notional damage rate (inflated in this example, for illustrative purposes only) is listed as well. Note that the damage rate is listed in the units of damage fraction per timestep for illustrative purposes. Using the data in Table 3.1, the PMFs for X_1 and X_2 can be obtained as shown in Figure 3.3. Finally, using the convolution equation in Equation 3.3, the cumulative damage distribution $P_{1:2}(X_{1:2})$ may be computed using the steps shown in Table 3.2. A plot of $P_{1:2}(X_{1:2})$ is shown in Figure 3.3.

Table 3.1: RR and Damage Rate Information for Two Timestep Example

		Probability	
Regime	Damage/timestep	t_1	t_2
Forward Flight	0.005	0.4688	0.4115
Right Turn 30°AOB	0.006	0.5055	0.5611
Right Turn 45°AOB	0.008	0.0059	0.0065
Other	0.008	0.0198	0.0209

Table 3.2: Convolution Calculations for Damage Summation from Two Timestep Example

$X_{1,2}$ PMF	Intermediate Expression	Value
$P_{1:2}(X_{1:2=0.010})$	$(0.4688)(0.4115)$	0.1929
$P_{1:2}(X_{1:2=0.011})$	$(0.4688)(0.5611) + (0.5055)(0.4115)$	0.4711
$P_{1:2}(X_{1:2=0.012})$	$(0.4688)(0) + (0.5055)(0.5611) + (0)(0.4115)$	0.2836
$P_{1:2}(X_{1:2=0.013})$	$(0.4688)(0.0274) + (0.5055)(0) + (0)(0.5611) + (0.0257)(0.4115)$	0.0234
$P_{1:2}(X_{1:2=0.014})$	$(0.5055)(0.0274) + (0)(0) + (0.0257)(0.5611)$	0.0283
$P_{1:2}(X_{1:2=0.015})$	$(0)(0.0274) + (0.0257)(0)$	0
$P_{1:2}(X_{1:2=0.016})$	$(0.0257)(0.0274)$	0.0007

In this work, the damage fractions at each timestep are assumed to be random variables (with example PMFs shown in Figure 3.3), although the uncertainty in the damage fraction stems only from uncertainty in the regime flown; the damage rates for each regime are assumed to be deterministic and known. It is possible to incorporate uncertainty in the damage rates for each regime into the above probabilistic damage estimation process,

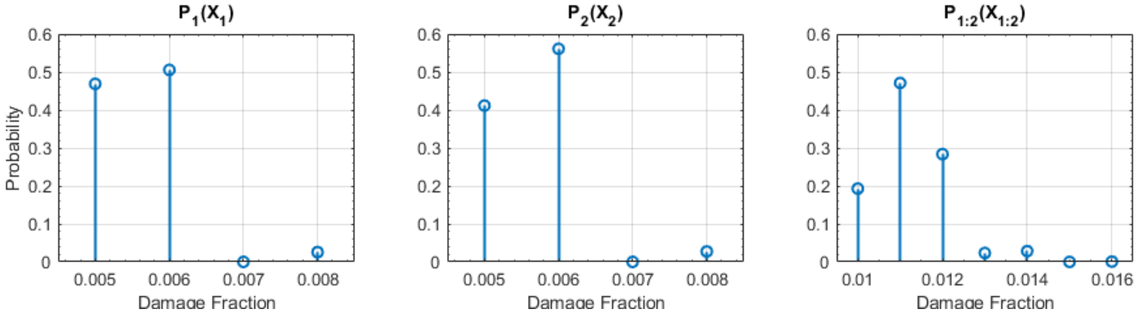

Figure 3.3: Example Sum of Damage From Two Adjacent Timesteps.

which would result in a more complete treatment of the problem in a probabilistic manner. This will be discussed further in the next subsection.

3.1.3 Gaussian Approximation of the Damage

The above convolution methodology produces the exact damage PMF over M timesteps from the individual damage PMFs at each timestep; however, it is computationally intensive and scales poorly to large amounts of flight data. Direct implementation of the discrete convolution of M random variables scales as $\mathcal{O}(M^2)$, and although more efficient approaches exist (such as Fast Fourier Transform-based methods, which scale as $\mathcal{O}(M\ log(M))$ [93]) even these approaches are likely infeasible for the very large datasets commonly encountered in CBM. In this section, a Gaussian approximation approach with computational complexity of $\mathcal{O}(M)$ is proposed that significantly reduces computation time with minimal loss of accuracy.

The key insight underlying the Gaussian approximation is that the damage random variables at each timestep are strongly mixed. That is, while the correlation between X_j and X_{j+1} (i.e., the damage random variables at two adjacent timesteps) is likely to be high because the aircraft flight condition is likely to be similar, the farther apart in time that two timesteps are, the less correlation is expected to be observed in the damage incurred at each timestep. Strong mixing for a sequence of random variables implies that the dependency between any two random variables decreases the further they are apart in the sequence and goes to zero as the distance between them in the sequence goes to infinity [94]. Importantly, it has been shown that the probability distribution of a sum of random variables with finite variance subject to strong mixing will converge to a Gaussian distribution [95–99]. In the context of the current work, let the covariance between the two damage random variables at adjacent timesteps X_j and X_{j+1} be given by $\sigma_{j,j+1}$. The variables X_j and X_{j+1} are strongly correlated since the damage incurred at the next timestep is likely to be similar to the damage incurred at the current timestep. However, it is assumed that the damage random variables exhibit strong mixing meaning that $\sigma_{j,j+N} \to 0$ as $N \to \infty$. This is certainly the case, as the regime flown (and thus the damage incurred) at a time in the far distant future is completely independent of the regime flown, and damage incurred, at the current time.

Given this assumption of strong mixing, the probability density function of the cumulative damage will be a Gaussian for any appreciable length of flight data. The mean and variance which define the cumulative damage PMF can be obtained by summing the means and variances of the individual timestep PMFs, $P_j(X_j)$, which themselves are not necessarily Gaussian. Let the mean and variance of $P_j(X_j)$ be denoted as μ_j and σ_j^2, respectively. The mean and variance of a PMF can be found by computing the first and second central moments respectively, using the following equations for $r \in \{1, ..., n\}$ defined regimes over the j^{th} timestep,

$$\mu_j = \sum_{r=1}^{n} p_r^{t_j} \, d_r \Delta t \tag{3.4}$$

$$\sigma_j^2 = \sum_{r=1}^{n} p_r^{t_j} \, (d_r \Delta t)^2 - \mu_j^2 \tag{3.5}$$

Then the PMF of the cumulative incurred damage, $P_{1:M}(X_{1:M})$, for a value of M representative of typical flight data segments approximates a Gaussian PDF with mean $\mu_{1:M}$ and variance $\sigma^2_{1:M}$ where,

$$\mu_{1:M} = \mu_1 + \mu_2 + ... + \mu_M \tag{3.6}$$

$$\sigma^2_{1:M} = \sigma^2_1 + \sigma^2_2 + ... + \sigma^2_M \tag{3.7}$$

In other words, $X_{1:M} \sim \mathcal{N}(\mu_{1:M}, \sigma^2_{1:M})$. Note that under this approximation, the cumulative damage $X_{1:M}$ is now considered a continuous rather than discrete random variable, which is more realistic from a physical perspective.

The numerical approach to computing the Gaussian approximation of $P_{1:M}(X_{1:M})$ using Equations (3.6) and (3.7) is $\mathcal{O}(M)$ and is thus extremely computationally efficient. Furthermore, the approximation is nearly exact considering the number of data points usually processed for a single flight (typically tens of thousands). Numerical comparisons between $P_{1:M}$ obtained from the convolution in Equation (3.3) and the Gaussian approximation for 100 timesteps of simulated flight data showed essentially a perfect match between the two densities. A further advantage of the Gaussian approximation approach is that the Cumulative Density Function (CDF) of the total damage, denoted as $F_{1:M}(X_{1:M})$, is easily obtained. One reason why this may be important is that, in a notional CBM program using damage PDFs for life-limited components, it may be desired that components be replaced when $F_{1:M}(1) = P_{1:M}(X_{1:M} < 1) < 0.999999$, signifying that there is a one in one million chance that the component has exceeded a damage fraction of 1. Since $P_{1:M}$ is a Gaussian distribution, computing $F_{1:M}$is trivial even when M is very large.

As a final point of comparison against deterministic damage estimation approaches, recall that deterministic (rule-based) regime recognition is a special case of probabilistic RR in which regimes are assigned either 1 or 0 probabilities at each timestep (see discussion in [59]). At the j^{th} timestep, the deterministic regime classification results in a damage PMF given by $P_j(X_j = d_r \Delta t) = 1$, where d_r is the damage rate of the identified regime. Computing $P_{1:M}(X_{1:M})$ using either the convolution or Gaussian approximation approach yields a deterministic result for the damage estimate, i.e., $P_{1:M}(X_{1:M} = D) = 1$ where $D = \sum_{j=1}^{M} d_{rj}\Delta t$ and d_{r_j} is the damage rate of the regime identified at the j^{th} timestep. This is the identical result as would be produced by a deterministic damage estimation approach which simply sums the damage fractions at each timestep. Thus, this type of deterministic damage estimation may be viewed as a special case of the probabilistic DCM proposed here.

Damage Rate Uncertainty

The Gaussian approximation discussed above is useful for computing the damage incurred across all timesteps in a computationally efficient way. However, the methodology assumes the damage rates for each regime take on a discrete value. This assumption is not necessarily realistic. While regime-based approach avoids the need for direct load measurement and/or estimation, Boorla and Rotenberger [40] have established that fatigue damage in a given regime may vary widely due to a range of factors including maneuver severity, aircraft weight, and pilot technique, reinforcing the findings of other authors such as Schaefer [100]. This damage variability within regimes has often been stated as one of the main drawbacks of a regime-based approaches to component damage estimation. While the employment of probabilistic regime recognition can incorporate regime uncertainty into the damage estimate, additional work is needed to properly incorporate *damage rate* uncertainty. Fortunately, the Gaussian approximation discussed above provides a perfect opportunity for the inclusion of the uncertainty present in the damage rates into the total fatigue damage estimate.

First, let the nominal damage rate – of a given component – for regime r (out of n total defined regimes in the RR library) be denoted as d_r, and the associated variance over a single timestep be denoted as $\sigma^2_{d_r\Delta t}$. As before, let the probability of regime r identified by the probabilistic RR at the j^{th} timestep be denoted as $p_r^{t_j}$. In this way, the Gaussian approximation of the equivalent damage over timestep t_j (given by eqs. (3.4) and (3.5)) can be modified to include the damage rate uncertainty. This will be done in accordance with the methodology of variance estimation in combined Gaussian mixture distributions found in Ref. [101], yielding the following equations,

$$\mu_{eq,j} = \sum_{r=1}^{n} p_r^{t_j}\; d_r \Delta t \tag{3.8}$$

$$\sigma_{eq,j}^2 = \sum_{r=1}^{n} p_r^{t_j}\left[\sigma_{d_r\Delta t}^2 + (d_r\Delta t)^2\right] - \mu_{eq,j}^2 \tag{3.9}$$

Now the equivalent mean and variance of the incurred damage over the j^{th} timestep can be found by computing $\mu_{eq,j}$ and $\sigma_{eq,j}^2$ in Equations (3.8) and (3.9) respectively. These equivalent means and variances for each timestep can then be summed using Equations (3.6) and (3.7) to form the new cumulative damage Gaussian PDF that effectively captures the damage uncertainty stemming from both major sources (i.e. uncertainty from the flown regime and uncertainty from the damage rates). It should be noted that Equations (3.8) and (3.9) are generalized in the sense that if a discrete damage rate was to be used (i.e. $\sigma_{d_r\Delta t}^2 = 0$) they would reduce to Equations (3.4) and (3.5).

3.1.4 Damage Estimation Uncertainty Analysis

When considering fatigue damage as a random variable driven by uncertainty in regime recognition (as well as damage rates, potentially), it is interesting and important to characterize the factors that influence the amount of uncertainty in damage estimates. Since the incurred damage at each timestep is modeled in this context as a random variable with the PDF $P_j(X_j)$, the variance σ_j^2 provides a proper metric of the uncertainty in the damage estimate at the j^{th} timestep. The question posed in this section is then, what distribution of regime probabilities leads to maximum uncertainty in the damage estimate?

To answer this question, consider the regime probability estimates at the j^{th} timestep. Let the highest damage rate over all regimes in the regime set be given by $d_{r,max}$, and the minimum damage rate (usually zero) by $d_{r,min}$. Then X_j is bounded above by $a = d_{r,max}\Delta t$

and below by $b = d_{r,min}\Delta t$. With these bounds defined, the distribution $P_j(X_j)$ with mean μ_j can take on infinitely-many forms depending on the probabilities over the regime set at the j^{th} timestep produced by the RR code. It can be shown [102, 103] that if the variance of X_j is maximized over every possible μ_j, then the greatest possible variance is achieved when $P_j(X_j = a) = 0.5$ and $P_j(X_j = b) = 0.5$, with $P_j(b < X_j < a) = 0$. This means that the maximum uncertainty in the damage at a given timestep, as measured by the variance in X_j, occurs when the aircraft flies on the boundary between the most damaging regime(s) and the least damaging regime(s). More precisely, the maximum uncertainty condition is achieved when the regime recognition code identifies a total of 50% probability that the aircraft is flying the most-damaging regimes, and a total of 50% probability that the aircraft is flying the least-damaging regimes. While such conditions may sound unlikely to occur, in practice large-variance estimates of the damage distribution $P_j(X_j)$ may occur routinely because high angle-of-bank turns are often associated with high damage rates for some rotor components. During transitions from level flight to turns, or during shallow turns, probabilistic RR codes can produce mixed probabilities of level flight (which often produce low or zero damage rates) and turns (which often produce higher damage rates), leading to high-variance damage distributions during these timesteps. It is important to note that the above condition corresponds to the largest uncertainty in the damage estimate (maximum σ_j^2), rather than the largest estimate of the mean damage (maximum μ_j). Clearly, the mean estimated damage μ_j is maximized when $P_j(X_j = a) = 1$, i.e., the RR code identifies the most damaging regime with probability 1.

3.2 Application to Fleet-Wide Damage Spectrum

This section documents an approach to use the proposed damage estimation methodology to create an updated damage spectrum for an aircraft fleet based on observed usage. Oftentimes, rotorcraft manufacturers provide component replacement timelines based on number of hours flown and an assumed usage spectrum. Studies have shown that such usage spectra may be inaccurate [4], and thus there may be a desire to create updated usage spectra and component replacement timelines based on observed usage. This approach to CBM, in which the damage fraction per N hours is specified, may be used as an alternative to IAT in cases where IAT is deemed too expensive or infeasible for operational reasons. A damage spectrum may also be used to augment IAT if HUMS data is corrupted or unreliable due to a faulty sensor. This section will detail a method to create updated component replacement timelines through the application of the probabilistic RR approach and damage computation module to large-scale fleet data.

Consider a large set of HUMS data from a fleet of N_a aircraft, and suppose that a component damage spectrum in terms of damage fraction per 100 hours is desired. Then, with reference to Figure 3.4, a random sample of 100 flight hours can be drawn from the fleet data and processed with both the probabilistic RR code (for example, from [59]) and the DCM outlined in the prior section. Assuming the total data consists of M timesteps of flight data, this produces the damage distributions $P_{1:M}(X_{1:M})$ for each component, representing the estimated damage fraction incurred during this 100 hour sample of flight data. These damage distributions can be reduced to a single conservative estimate of the damage per 100 hours by storing the damage value corresponding to the 0.999999 (six-nines) percentile, given by $F_{1:M}(0.999999) = D$. This process of randomly sampling 100 hours of flight data, computing $P_{1:M}$ and $F_{1:M}$, and storing the six-nines damage fraction D_i (where the i subscript is added for the i^{th} random sampling of flight data) is repeated numerous times to create a bootstrapped statistic on the damage per 100 hours. Assuming this process is repeated N_s times, various bootstrapped statistics of D_i, $i \in 1, 2, ...N_s$ (such as the mean or median) can be used to create a single damage per 100 hours estimate for each component. Note that the mean or median of the set of D_i's may be an appropriate statistic in this case because conservativeness is already built into the estimated values through the six-nines selection criteria applied to each PDF from the randomly-sampled flight data. Nevertheless, the desired bootstrapped statistic (mean, median, or otherwise) to be used as the single damage per 100 hours estimate is a regulatory decision that may involve additional considerations and analysis not explored here. Note that the intent of the process outlined in this section is to produce fatigue damage spectra that explicitly account for the uncertainty in the regime recognition process.

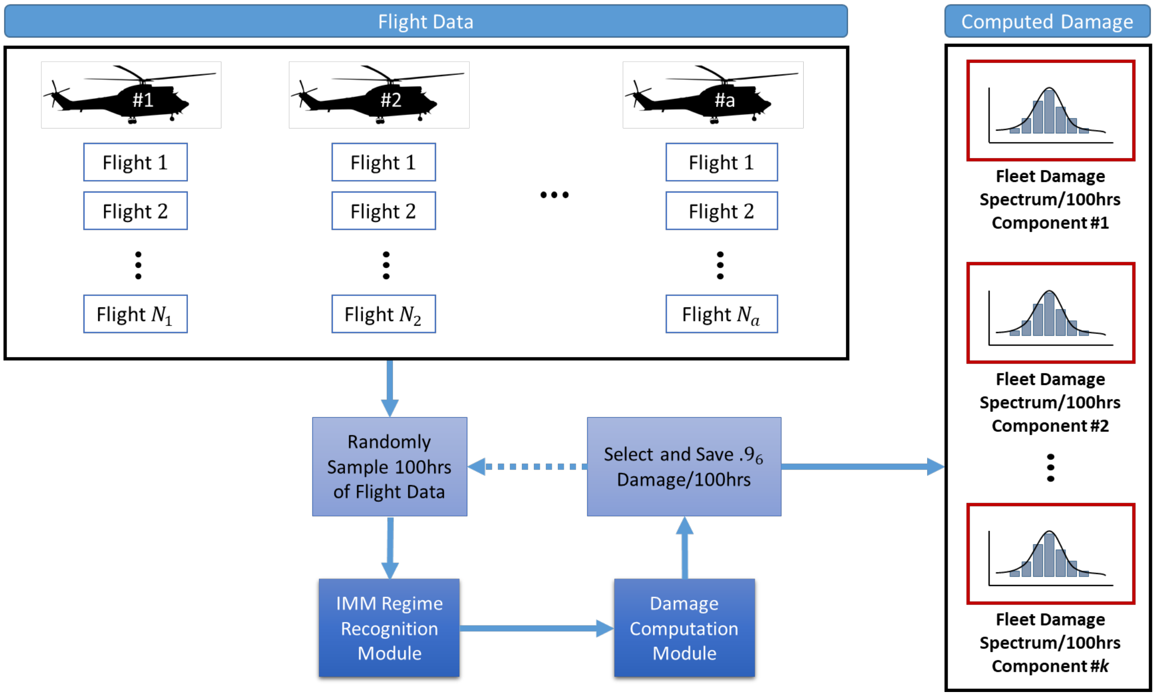

Figure 3.4: Visualization of Process to Create Updated Damage Spectrum Using Probabilistic Damage Computation.

3.3 Results

The proposed damage computation method described in Sections 3.1 and 3.2 is illustrated in this section using simulated flight data in a series of test cases. The simulation model used in this section is the same model and *virtual pilot* discussed in Section 2.5, as well as documented in [78, 87]. To obtain the data used in this study, simulated states of the aircraft were recorded at 10 Hz to model the data rates of typical HUMS devices

One of the goals of this section is to compare performance of the proposed probabilistic damage estimation scheme with a deterministic approach based on rule-based regime recognition. Rule-Based (RB) regime recognition uses thresholds to define sections in the aircraft state-space that map directly to a particular regime in order to make a deterministic classification. The simulation studies and comparisons shown in this section use only a limited set of regimes (listed in Table 3.3), and thus for the purposes of this work a basic RB-RR algorithm was developed using guidelines obtained from [39, 48–52, 104, 105]. The thresholds used to classify regimes in this basic RB-RR code are shown in Table 3.3, where the thresholds are applied to the state listed in the second column. The probabilistic results shown in this section are generated with the IMM-RR algorithm discussed in Chapter 2. The same regime thresholds used in the rule-based code in Table 3.3 are used to define the individual models in the IMM-RR code, which is important so that the RR and damage estimation results can be meaningfully compared. The final column of Table 3.3 lists notional component damage rates (in damage fraction per timestep) for an example component, to be used in the damage estimation studies throughout the remainder of this section. Note that the damage rates in Table 3.3 are for example purposes only and are not meant to represent actual damage rates for a particular component or set of regimes.

3.3.1 Flight Condition Effects on Uncertainty

The first simulated example will illustrate the damage computation process and explore the basic effects of flight condition on damage estimation uncertainty. Two simulated flight segments were flown using the flight card shown in Table 3.4. In the first flight segment, labeled the Well-Defined flight segment, the two Right Turn maneuvers are flown at ±30°AOB which falls squarely within the range of the defined Right Turn (30°AOB) regime listed in Table 3.3. The second flight segment, labeled the Mixed flight segment, performs the same right turns except at slightly higher ±37.5°AOB. This places the aircraft state between two regimes – Right Turn (30°AOB), and Right Turn (45°AOB), which have differing damage rates. Figure 3.5 shows simulated flight data for these two flight cards, showing time histories for the roll angle and the body-frame x velocity component, denoted as u.

Table 3.3: Regime Definition Information

Regime	Threshold State	Lower Threshold	Upper Threshold	Damage/s
Forward Flight	Forward Speed	$0.1V_h$	$0.95V_h$	0
Right Turn 30°AOB	Roll Angle	15°	37.5°	1.0424×10^{-7}
Right Turn 45°AOB	Roll Angle	37.5°	52.5°	1.6280×10^{-6}
Left Turn 30°AOB	Roll Angle	-37.5°	-15°	6.1125×10^{-8}
Left Turn 45°AOB	Roll Angle	-52.5°	-37.5°	8.4980×10^{-7}
Forward Accelerate	Longitudinal Accel.	$0.1g$	∞	–
Descent	Rate of Climb	$300\ ft/min$	∞	1.6288×10^{-6}

Table 3.4: Flight Card for Example 1

Maneuver Start Time, s	Flown Regime	Parameter (Well-Defined)	Parameter (Mixed)
0	Forward Flight $0.5V_h$	–	–
30	Right Turn	30°AOB	37.5°AOB
60	Forward Flight $0.5V_h$	–	–
80	Left Turn	-30°AOB	-37.5°AOB
110	Forward Flight $0.5V_h$	–	–
130	End	–	–

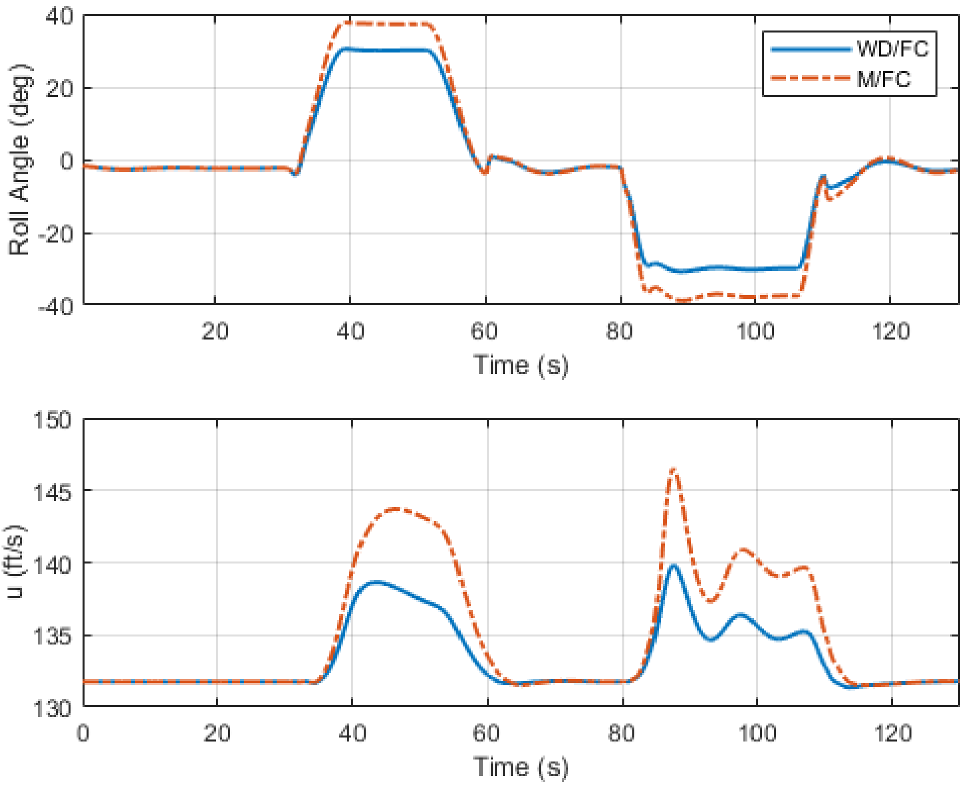

Figure 3.5: Selected State Time Histories for Example 1.

Figures 3.6 and 3.7 show the regime recognition results for the Well-Defined flight segment and Mixed flight segment respectively, where the IMM-RR results are shown on the top plots, and the RB-RR results are shown on the bottom plots. For the Well-Defined flight segment, both the IMM and RB algorithms produce essentially identical results, correctly identifying the regimes throughout the flight. For the Mixed flight segment, during the first turn the IMM algorithm estimates a roughly 50% probability that the regime is Right Turn (30°AOB) and 50% probability that the regime is Right Turn (45°AOB), with analogous results during the second turn. In contrast, the rule-based code must make a deterministic classification and instead switches between 30°AOB and 45°AOB classifications during both turns. Importantly, although the aircraft is flying on the border between two regimes, the rule-based code does not provide any quantification of the uncertainty in the recognized regime outputs, but instead switches between deterministic classifications of the two bounding regimes. This is a primary drawback of rule-based methodologies in general as noted in Refs. [58, 59], but also has implications for damage estimation as well.

The estimated damage was computed from the regime recognition results shown in in Figures 3.6 and 3.7 by applying the DCM using the damage rates in Table 3.3. For the probabilistic RR results, this results in damage PDFs for the Well-Defined and Mixed flight segments, while for the deterministic RR results, a single damage value for each flight segment is computed. The results are shown in Figure 3.8. Note that the deterministic damage predictions are different between the two flight segments as would be expected, but the ambiguity regarding the regimes flown in the Mixed flight segment is not represented in any way in the deterministic damage estimates. In contrast, the damage PDFs shown for the Well-Defined and Mixed flight segments differ not only in their mean predictions, but also in their variance. Figure 3.8 shows that the variance of the estimated damage from the Mixed flight segment is about three times greater than that for the Well-Defined flight segment, reflecting higher uncertainty in the damage stemming from higher uncertainty in the regime classifications. This translation of uncertainty in regime classifications to

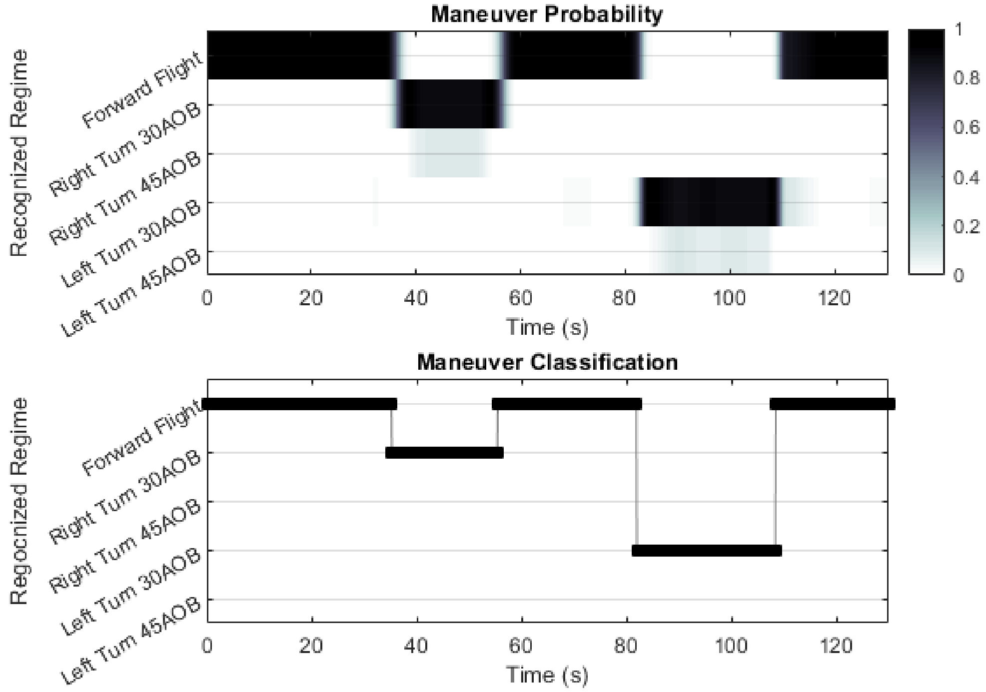

Figure 3.6: Regime Recognition results for IMM (top) and RB (bottom) algorithms (Well-Defined Flight Segment).

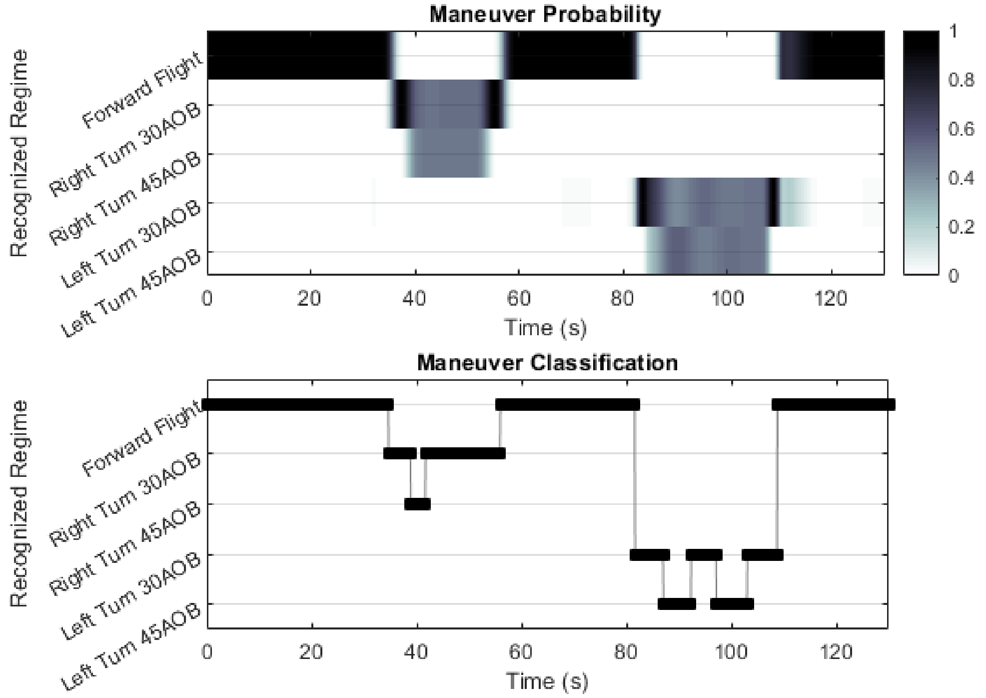

Figure 3.7: Regime Recognition results for IMM (top) and RB (bottom) algorithms (Mixed Flight Segment).

larger variance in the damage estimates is a key benefit of the probabilistic RR and damage estimation approach that is clearly lacking in the comparison deterministic method. It is also important to recognize that although most of the Well-Defined flight segment is flown well within the defined regime boundaries, there is still uncertainty in the regimes flown while the aircraft transitions between different regimes, as illustrated in Figures 3.6 and 3.7. As a result, the probabilistic damage estimate for the Well-Defined flight segment has a low but non-zero variance.

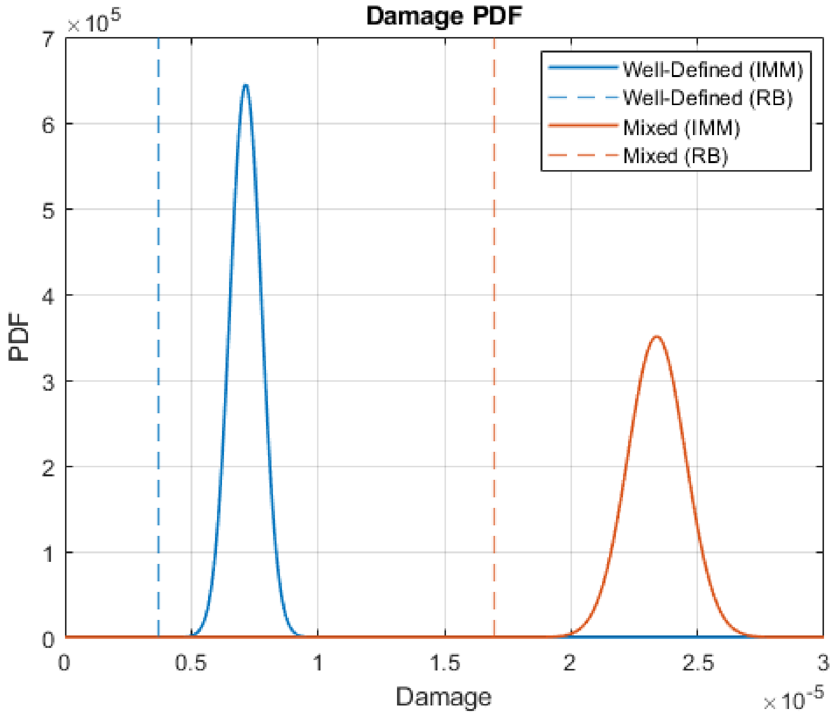

Figure 3.8: Damage Estimates and Distributions for Example 1.

3.3.2 Monte Carlo Analysis of the Effects of Flight Condition Uncertainty

To further reinforce the benefits of the probabilistic damage estimation scheme over a deterministic approach, a Monte Carlo study is performed to analyze how small perturbations to the aircraft state over a nominal flight segment can cause large uncertainty in RR outputs. Specifically, the study shows how uncertainty in the recognized regimes affects the overall damage estimates for both the deterministic and probabilistic damage algorithms.

Five hundred simulated flights of the flight card shown in Table 3.5 were performed. For each Monte Carlo trial, the bank angle parameters in column 3 of Table 3.5 were perturbed by a normally-distributed random value with zero mean and standard deviation of 1°. Thus, for every Monte Carlo trial, a total of six random samples were created and added to each turn setpoint value in the flight card in Table 3.5 executed by the *virtual pilot*. In practice, because the commanded turns are on the boundaries between the Forward Flight, 30°AOB, and 45°AOB turn regime definitions, these perturbations mean that in each trial of the Monte Carlo simulation the aircraft is either just over or just under each of the thresholds for a 30°AOB or 45°AOB turn. This means that the rule-based regime recognition code changes its classifications between Forward Flight, 30°AOB turn, or 45°AOB turn on a flight-to-flight basis, even though the angle-of-bank perturbations are only on the order of 1 deg between the turns maneuvers in each trial. Figure 3.9 shows a time history of selected states for the nominal flight segment (with no random perturbations), while Figure 3.10 shows the RR outputs from the rule-based and probabilistic codes for this nominal flight. In Figure 3.10, the probabilistic RR results in the top plot show approximately 50% probabilities between the Forward Flight and 30°AOB turn regimes for the shallow turns, and 50% probabilities between the 30°AOB or 45°AOB turns during the steeper turns, as expected. In contrast, the rule-based code switches between them periodically and is forced to make a deterministic classification even though these turns are conducted on the boundaries between regimes.

For each Monte Carlo trial, the damage was computed for the RR results using the DCM described in Section 3.1. For the IMM probabilistic RR results, this produced a damage PDF for each trial. The mean and 99.9999th percentile ($.9_6$) values were computed for each trial. A histogram of these values is shown in the top two plots of Figure 3.11. The damage was also computed for the deterministic rule-based RR results, with a histogram of these values shown in the third plot in Figure 3.11. For each of the top three plots in Figure 3.11, the damage value computed from the nominal flight segment (with no random

Table 3.5: Flight Card for Example 2

Maneuver Start Time, s	Flown Regime	Parameter	Perturbation
0	Forward Flight 0.5V_h	–	–
10	Right Turn	15°AOB	$\mathcal{N}(0, 1\ deg^2)$
55	Right Turn	37.5°AOB	$\mathcal{N}(0, 1\ deg^2)$
115	Right Turn	15°AOB	$\mathcal{N}(0, 1\ deg^2)$
160	Forward Flight 0.5V_h	–	–
190	Left Turn	-15°AOB	$\mathcal{N}(0, 1\ deg^2)$
235	Left Turn	-37.5°AOB	$\mathcal{N}(0, 1\ deg^2)$
295	Left Turn	-15°AOB	$\mathcal{N}(0, 1\ deg^2)$
340	Forward Accelerate	–	–
375	Max Forward Speed	–	–
420	End	–	–

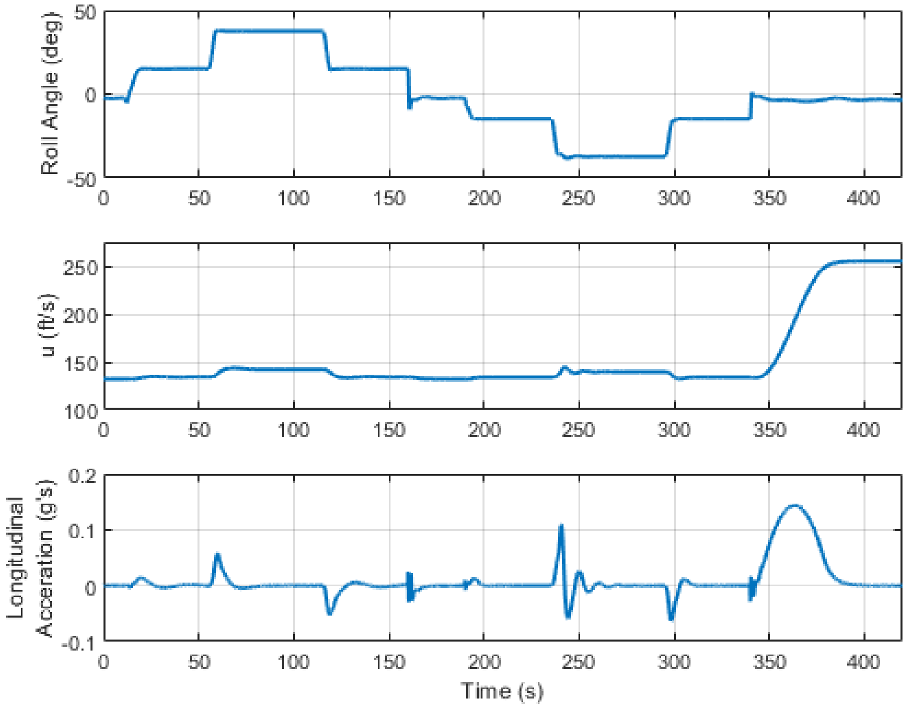

Figure 3.9: Nominal State Time History for Flight Card in Table 3.5.

perturbations) is shown with a red line, and the mean of the Monte Carlo trials is shown with the blue dashed line. Note that the histograms of the mean and $.9_6$ damage values for the IMM results are fairly smooth, with the small perturbations in each trial resulting in small perturbations in the damage estimates. The damage histograms exhibit a close match to normal distributions, which correspond to the normal distributions used to gen-

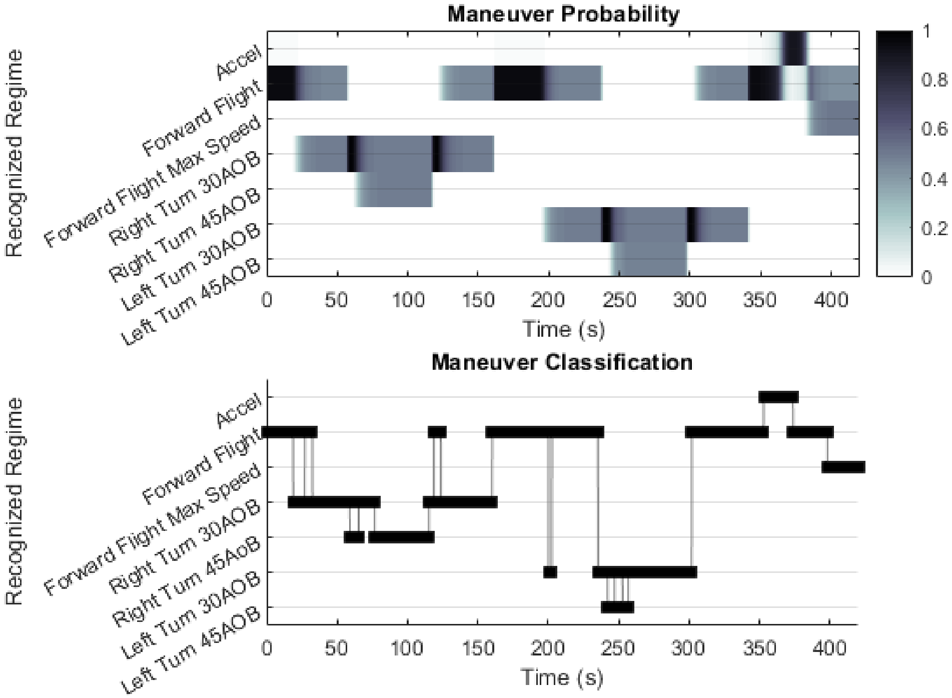

Figure 3.10: Regime Recognition Results for Nominal Flight for Flight Card in Table 3.5 (Top: IMM, Bottom: Rule-Based).

erate the roll angle perturbations. Furthermore, the mean damage from the Monte Carlo trials matches the nominal damage almost exactly. In contrast, the deterministic damage histogram exhibits a multi-modal distribution caused by the discrete "switch" between the maneuver options (Forward Flight, Left Turn 30/45°, Right Turn 30/45°) depending on the perturbation in a given simulation, and the mean damage differs from the nominal damage by more than 10%. The four modes are caused by the large damage incurred by the 45°deg turns – if the 3rd and 7th flown regimes happen to cross the boundaries for this regime (or not), the damage value is shifted significantly. The deterministic damage histogram shows that the damage estimate can vary widely, and fairly discontinuously, given small perturbations in the aircraft state as it flies near the boundaries between defined regimes. This wide variation in the deterministic damage estimate is a result of using a rule-based regime recognition approach that is forced to produce deterministic (100%) classifications, even if there is significant ambiguity in the actual flown regime. Finally, the deterministic results were re-processed with simulated flight data which was perturbed by additive white noise on each of the states, simulating measurement noise (where noise standard deviations were derived from engineering judgment based on experience with actual flight data). The damage estimated from these noise-perturbed trials is shown in the bottom plot in Figure 3.11. In this case, the distinct modes of the distribution disappear as the RR results show more variation in maneuver switching times. While the noise mitigates the discontinuity in the damage estimates due to small perturbations in the state, it does not change the fact that the damage estimates vary widely in response to small changes in the roll angle on the order of a couple of degrees or less. Overall, these results highlight the improved performance of the probabilistic DCM compared to its deterministic counterpart when the aircraft flies for sustained periods on the boundary between defined regimes with different damage rates.

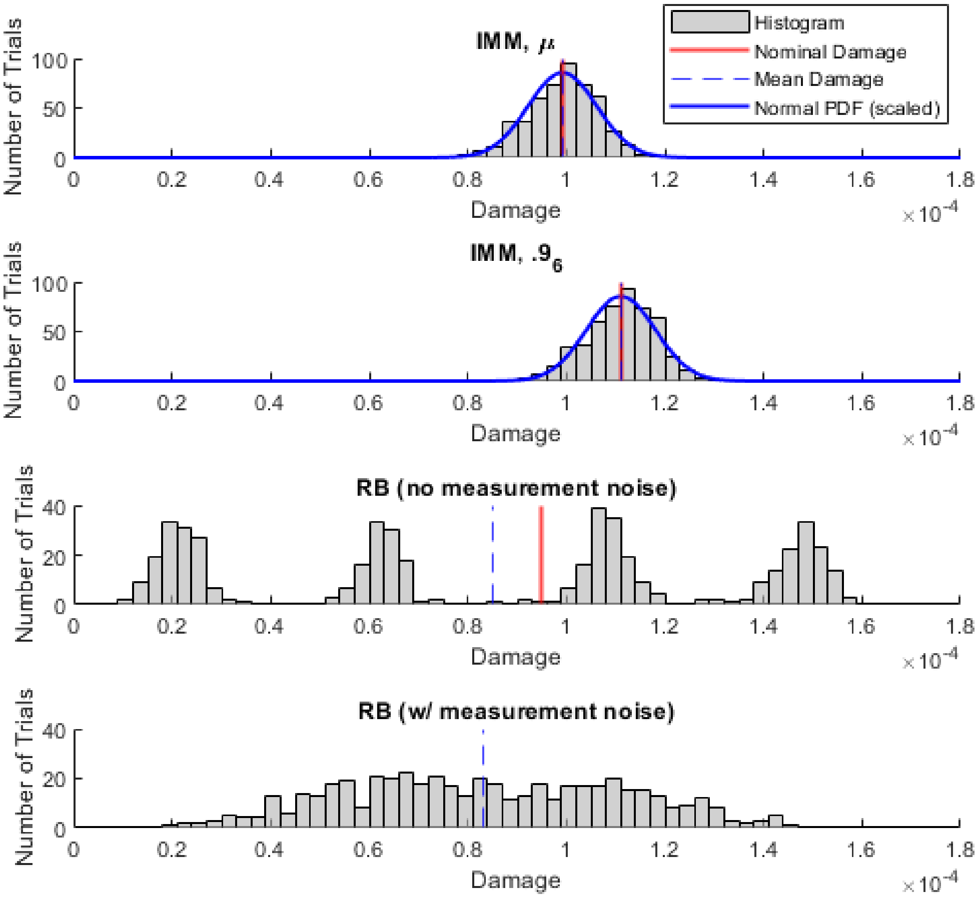

Figure 3.11: Monte Carlo Damage Estimation Results.

To complement this Monte Carlo study, an additional Monte Carlo simulation was performed using only well-defined flight conditions – i.e., flight segments which do not involve the aircraft flying on the border between defined maneuvers. The flight card for this study was the same as in Table 3.5, except that the ±15°roll angle setpoints were changed to ±30°, and the ±37.5°setpoints were changed to ±45°. Even though the same small roll angle perturbations as used in the above example were still included, the roll maneuvers in each Monte Carlo trial fell well within the Left Turn 30/45°and Right Turn 30/45°regime categories. Figure 3.12 shows the damage results for this Monte Carlo simulation using the same RR and damage processing methods employed in Figure 3.11. Note that, because the small perturbations do not cause flight-to-flight changes in the regime estimates for either the deterministic or probabilistic RR outputs, the damage estimates are nearly the same for all trials. For all cases, the mean and nominal damage estimates differ by less than 0.01%, and the spread of each distribution is less than 1% of their respective nominal values. While the nominal damage computed by the IMM (red line in top plot of Figure 3.12) is close to the nominal damage computed by the rule-based methodology in the bottom plot, they are not identical due to the effects of filter tuning in the IMM and the transient response during maneuver switching. Note that sensor noise was not added in this example as the perturbations caused by noise would not affect the recognized regimes, given that the aircraft flew well within the regime boundaries during the entire flight segment. The results in Figure 3.12 clearly show that the probabilistic damage estimation approach produces essentially identical results to the deterministic damage estimator when the level of uncertainty in regime estimates is low. When viewed in conjunction with the first set of Monte Carlo results, it is evident that the probabilistic DCM is a generalization of the deterministic damage estimator that responds more smoothly to uncertainty in regime estimates but provides equivalent results when the aircraft flies well within the defined regime categories.

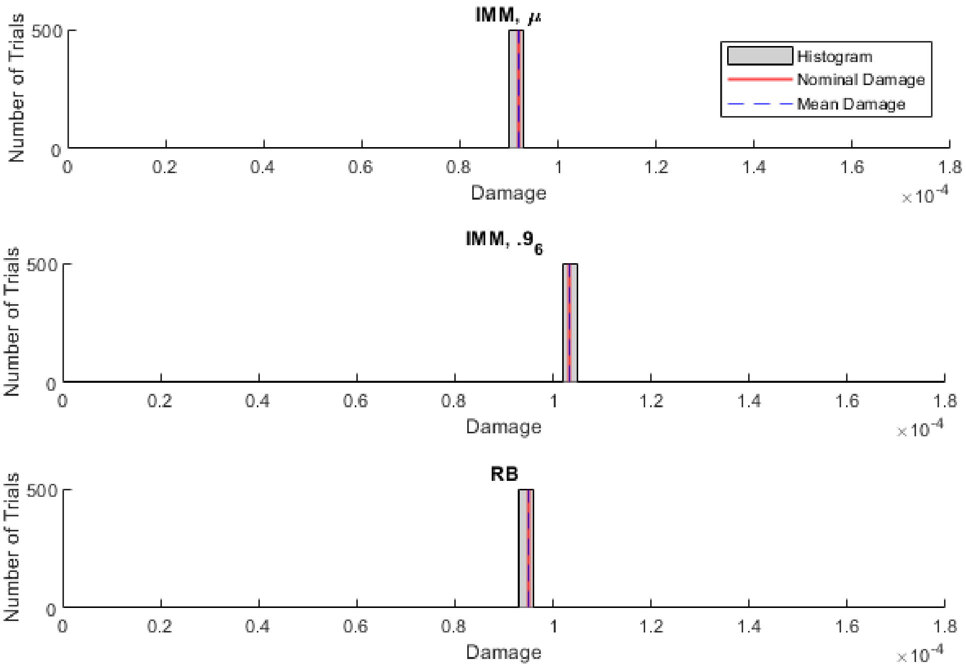

Figure 3.12: Monte Carlo Damage Estimation Results (Well-Defined Nominal Flight Condition).

3.3.3 Creation of Probabilistic Damage Spectrum

As described in Section 3.2, the damage computation methodology can be applied to fleet-wide usage data to create a probabilistic damage spectrum. For the purposes of this work, simulated fleet data was used to demonstrate the overall methodology. An automated flight simulation process was created to generate a large database of randomized flight segments consisting of simple maneuvers. The ordering of these maneuvers, the total duration and maneuver durations, and the setpoints used for each maneuver were randomized.

For each flight in the simulated database, the number of maneuvers is first randomized using the distribution in Table 3.6. Then, the duration of each maneuver is randomly sampled from a uniform distribution with the limits shown in Table 3.6. The maneuver type is then randomly selected as Forward Flight, Turn, Climb/Descent, and Climbing/Descending Turn (except for the first maneuver in the flight card, which is always selected as Forward Flight). Finally, the parameters for each maneuver are randomly sampled using Inverse

Transform Sampling (ITS) [106]. For the Forward Flight segments, the speed setpoint is sampled from a uniform distribution using the limits shown in Table 3.6. For the turn, Climb/Descent, and Climbing/Descending Turn segments, the setpoints are chosen from bimodal distributions shown in Table 3.6, which are built from two truncated normal distributions. For instance, if a Climb/Descent maneuver is commanded, the rate is sampled from the bimodal distribution defined by the union of the two normal distributions in Table 3.6, resulting in either a climb or descent depending on the sampled parameter.

Table 3.6: Flight Parameter Distributions for SH-60B Simulated Database Creation

Parameter	Lower Limit	Upper Limit	Distribution(s)
Number of Maneuvers	15	25	Uniform
Maneuver Duration, s	120	240	Uniform
Angle of Bank, deg	-40	40	$\mathcal{N}(30,4), \mathcal{N}(-30,4)$
Rate of Climb, ft/s	-12.5	12.5	$\mathcal{N}(8.33,2), \mathcal{N}(-8.33,2))$
Forward Speed (V_h)	0.5	0.7	Uniform

Using this automated flight card generation method, a total of approximately 1,000 hours of flight data was generated using the generic helicopter simulation and the virtual pilot. A probabilistic damage spectrum for a given component, providing an estimate of the damage per 100 hours, was created from this data as follows. First, random flights were selected from the database until the total cumulative number of flight hours exceeded 100. These 100 hours of data were then processed by the IMM-RR algorithm and the probabilistic DCM described in Section 3.1, producing a PDF of the damage incurred over those ~100 hours. Let this PDF be denoted as $\mathcal{N}(\mu_T, \sigma_T^2)$. Because this damage estimate corresponds to a series of flights that do not total 100 hours exactly, this PDF must be scaled to represent a probability distribution for damage per 100 hours, denoted here as X_{100}. The PDF for X_{100} can be computed from $\mathcal{N}(\mu_T, \sigma_T^2)$ according to,

$$X_{100} \sim \mathcal{N}\left(\mu_{100}, \sigma_{100}^2\right) = \mathcal{N}\left(\mu_T \frac{100}{T}, \sigma_T^2 \left(\frac{100}{T}\right)^2\right) \tag{3.10}$$

where T is the total cumulative flight time of the randomly-selected flight segments. A percentile of this distribution, for instance, the 99.9999th percentile, is then recorded, representing a conservative estimate of the damage incurred over this random $\sim$100 hour flight segment. Let this percentile value be denoted as $\gamma_(\eta, i)$, where η is the percentile and i is the index corresponding to the randomized flight data set.

This process is repeated for N_f total sets of approximately 100 hours of flight data, creating a set of $\gamma_{\eta,i}$'s where $i \in 1, ..., N_f$. Figure 3.13 shows a histogram of the $\gamma_{\eta,i}$ values for $N_f = 10,000$ randomized sets of flight data created from the simulated flight database and $\eta = .9_6$ (top plot), along with the empirical CDF (bottom plot). Note that the histogram of $\gamma_{\eta,i}$ representing a bootstrapped distribution of the damage per 100 hours for this example component to a desired level of conservativeness represented by $\eta = .9_6$, closely matches a normal distribution. In the current example, the damage per 100 hours is observed to have a mean of 0.175, standard deviation of 0.0044, and range of 0.0336.

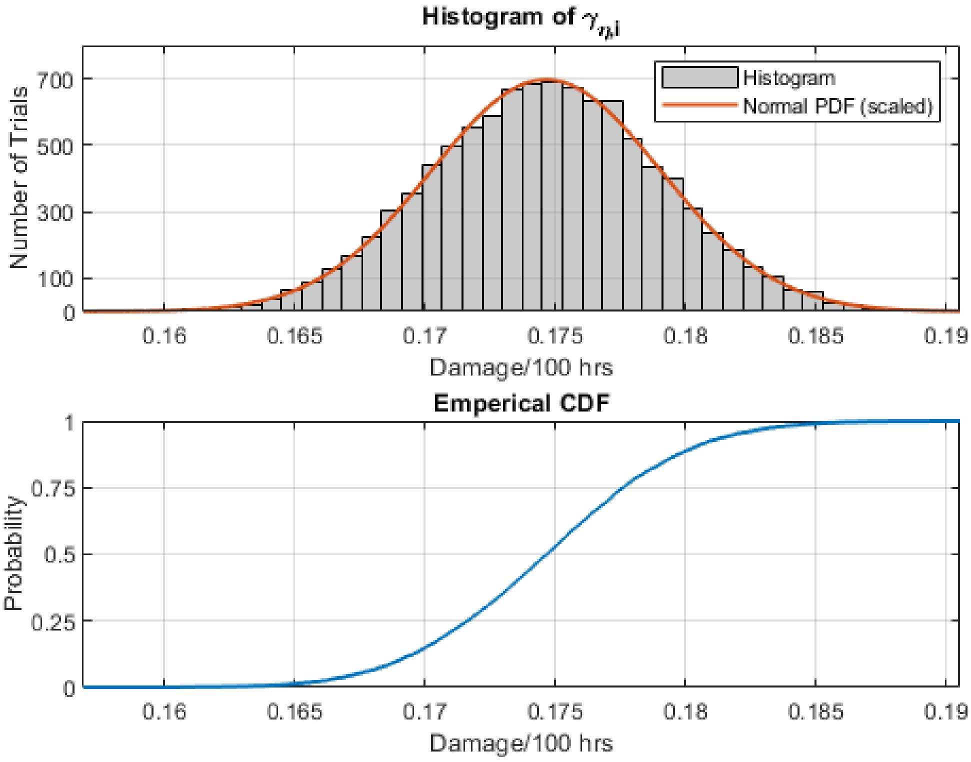

Figure 3.13: Histogram (top) and Empirical CDF (bottom) of Damage Per 100 Hours $\gamma_{\eta,i}$.

In practice, CBM requires a single damage per 100 hour estimate for each life-limited component, not a PDF such as that shown in Figure 3.13. To obtain this, the empirical CDF in Figure 3.13 can be interpolated to (once again) achieve a desired level of conservativeness. Since the $\gamma_{\eta,i}$ values would likely already be chosen using a conservative selection for η (for instance, six-nines), it may be appropriate to choose the mean of the $\gamma_{\eta,i}$ values in Figure 3.13 as the single estimate of damage per 100 hours for this component. Alternatively, additional conservativeness can be enforced by selecting the six-nines value from the empirical $\gamma_{\eta,i}$ values. Appropriate choice of the percentile values used in computing $\gamma_{\eta,i}$, and in the final damage per 100 hours, warrants further study and consideration given current and future regulatory requirements, although such investigation is beyond the scope of this work.

Finally, it is important to point out the difference between computing the damage spectrum using a rule-based RR method and the probabilistic approach used in this section. In a deterministic approach, one can potentially create a histogram of damage values using random $\sim$100 hour flight sequences as performed here. This approach could be used to generate the equivalent plot to Figure 3.13, and to select a damage per 100 hour value based on a desired level of conservativeness from the empirical CDF. However, the deterministic samples used to create the empirical PDF and CDF (equivalent to Figure 3.13) only provide the deterministic RR algorithm's best guess as to the damage incurred (similar, but not the same, as an expected value), providing no information about the uncertainty in the regimes flown. In contrast, for each 100 hour segment of data, the probabilistic RR algorithm and DCM condition the damage estimates on the uncertainty in the regimes flown, and a conservative estimate can then be derived from the resulting distribution. The probabilistic approach therefore provides a mechanism to enforce a level of conservativeness during the regime recognition process, and during the "outer" sampling process in which flight data sequences are chosen to be representative of fleet-wide usage.

CHAPTER 4

APPLICATION TO REAL HUMS DATA

In the prior chapters, methodologies were developed that use the output from an aircraft HUMS and perform probabilistic regime recognition, then translate those probabilistic RR results into fatigue damage estimates for use in IAT and usage spectrum-based programs. Currently there has been no work done to verify the validity of such probabilistic RR algorithms on real-world aircraft data. This chapter seeks to demonstrate that the proposed algorithm is viable on real data. To demonstrate this, a large batch of HUMS from a generic single engine transport helicopter was processed using the algorithms and methodologies proposed in Chapters 2 and 3.

4.1 Helicopter Regime Recognition

In this section, several flight sequences are explored using actual HUMS data. For these datasets, no "truth" data is available, and thus the goal of these examples is not to evaluate the accuracy of the IMM-RR algorithm, but rather to demonstrate its operation using real-world data.

4.1.1 Real Flight Sequence 1

The flight segment examined in this example consists of a series of climbs, descents, climbing turns, and descending turns, although the determination of these flown maneuvers is obtained through subjective analysis of the flight data rather than any recorded truth data from the pilots. Figures 4.1 and 4.2 show selected state time histories from the HUMS data as well as the IMM or Kalman filter state estimates for each state. Note that, due to the proprietary nature of the data, dimensions are removed and only trends are shown. Throughout the approximately 8 min flight segment, the vehicle maintains a fairly constant forward flight speed, performing turns, climbs, and descents. Figure 4.3 shows the recognized regimes from the IMM-based algorithm, where the gray-scale denotes regime probability. Examining the results in depth, the algorithm recognizes a descent regime for about the first 50 seconds with probability near 1, due to the clear descent rate and zero angle of bank forward flight condition. The helicopter then performs a climb in a shallow right turn, leading the IMM algorithm to recognize Climb and Right Climbing Turn regimes with approximately equal probability. After about 120 sec, the angle of bank becomes zero and the algorithm recognizes a Climb regime until about 220 sec. The aircraft then performs a level descent from about 220-350 seconds, with intermittent shallow left turns again correctly classified by the IMM algorithm. The flight segment ends with a Left Descending Turn maneuver, Right Descending Turn maneuver, and finally a level Climb maneuver. Overall, this example shows that the IMM-based regime recognition algorithm can be practically applied to actual data from a HUMS flight data recorder and provides evidence of the algorithm's promise as an alternative approach to regime recognition with unique advantages over state-of-the-practice methods.

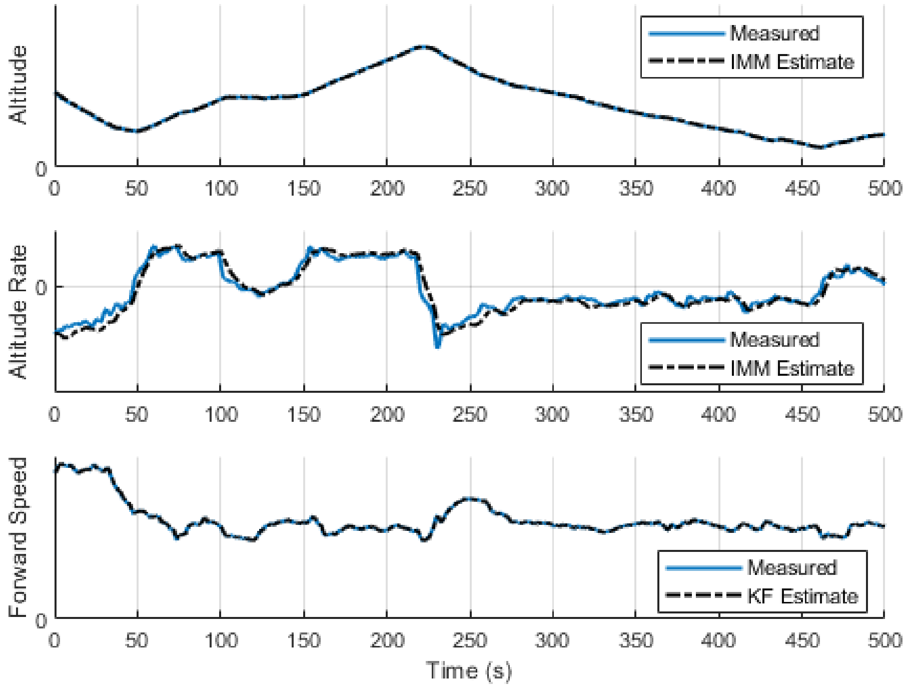

Figure 4.1: Selected State Time Histories for Real Flight Sequence 1, Actual Flight Data.

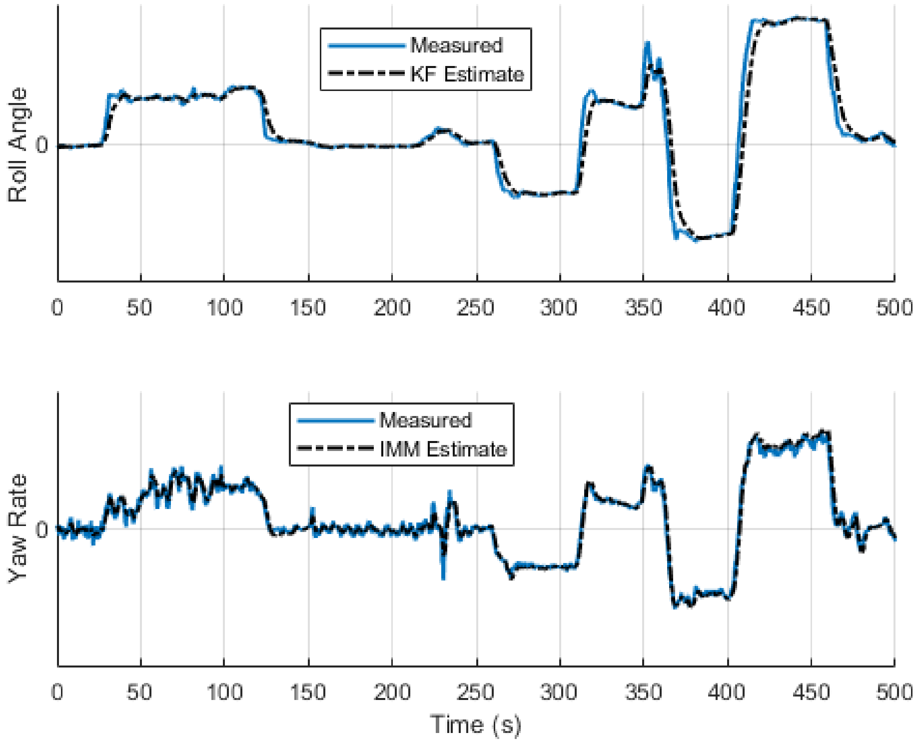

Figure 4.2: Selected State Time Histories for Real Flight Sequence 1, Actual Flight Data.

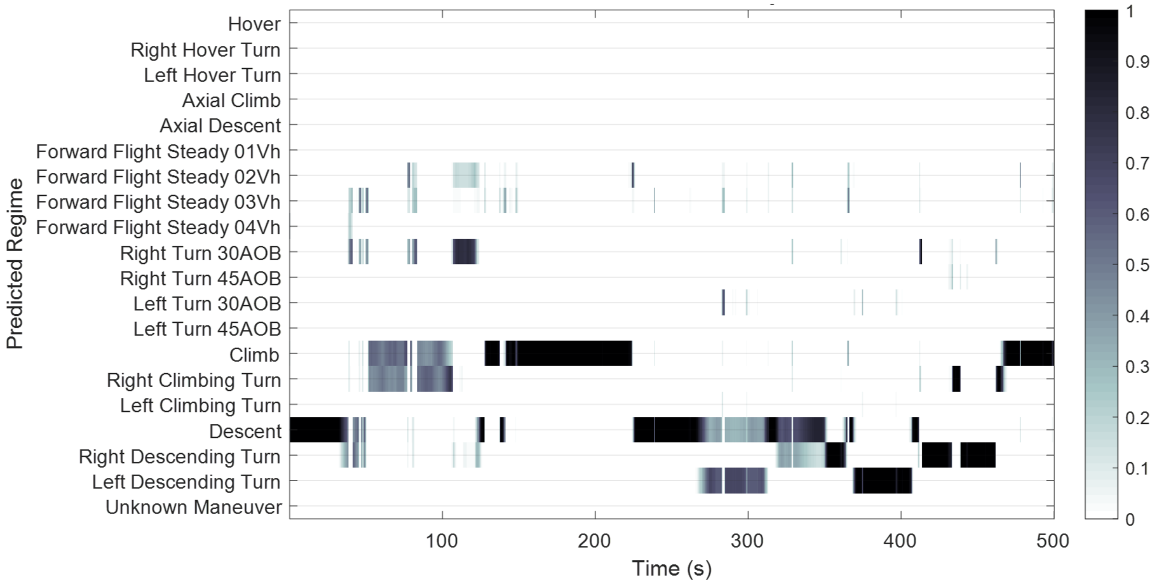

Figure 4.3: IMM-Based RR Classification Results for Real Flight Sequence 1.

4.1.2 Real Flight Sequence 2

In the next two examples, taken from a generic single-engine transport helicopter, no state time histories are available for the sequences due to the potentially sensitive nature of the underlying data. So once again no "truth data" is available, similar to the previous example. However, these examples still allow for the further validation of the proposed IMM-based probabilistic regime recognition. In this example a flight sequence is analyzed, and the results of the IMM-RR can be seen in Figure 4.4. Based on the RR results, it is presumed to be a hover-based sequence, but note that no true regime labels are available. Because only several regimes are active throughout the entire flight, the chart plotted using a line graph. Examining the RR results, the aircraft is shown to be in steady hover with probably greater then 97% for the first 8 minutes of analyzed flight time. This is immediately followed by an abrupt transition to a regime that does not appear in the IMM-RR maneuver library. The IMM code is able to maintain its probabilistic classifications and mark the regime as "unknown." The algorithm then initially begins a mixed classification between "Hover" and "Axial Climb" before classifying the segment as almost exclusively the "Hover" regime. At around the 9 minute mark, the algorithm begins classifying an alternating set of "Hover", "Axial Climb", and "Axial Descent" with high steady-state probability (>95%), before ending the sequence with "Hover."

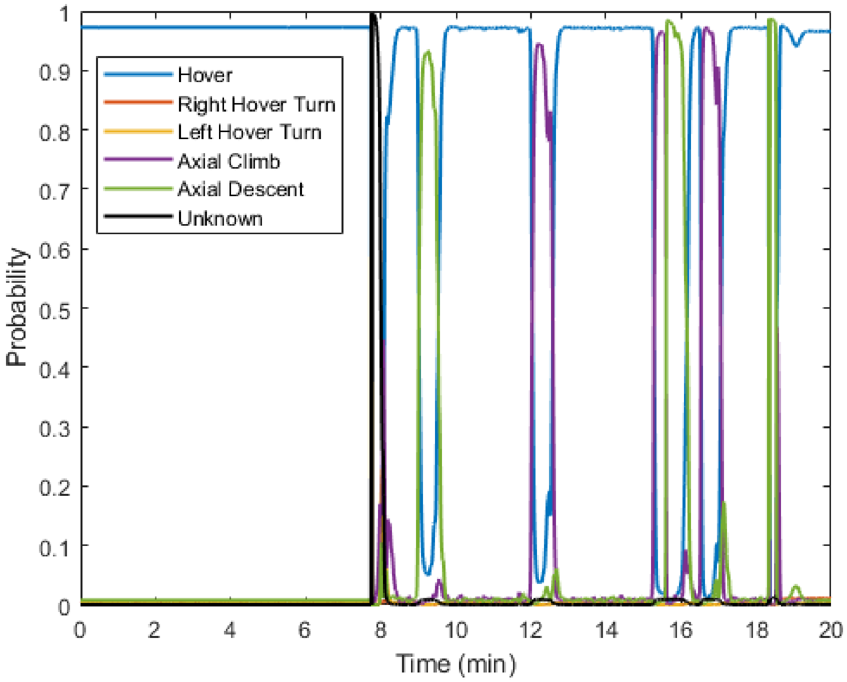

Figure 4.4: IMM-Based RR Classification Results for Real Flight Sequence 2.

4.1.3 Real Flight Sequence 3

In this next example, a sequence is analyzed that is believed to be primarily forward flight based, as evident by the regime recognition results. The results of the IMM-RR can be seen in Figure 4.5. At the start of this analyzed segment, the aircraft can be seen exiting a Climb. Over a five minute period it engages in a sequence of climb and descent based maneuvers before entering steady forward flight. The algorithm continues to classify the aircraft regime as steady forward flight with a probability of greater than 90% from approximately 110-130 minutes. The aircraft then enters a series of climbs, descents, and level flight segments for the remainder of the flight. Notice that in Figure 4.5 the maneuvers transitions are classified smoothly as the probability varies from one to the other. Also note that the steady state probabilities are not strictly 100% or 0% – as would be the case with a rule-based RR methodology – and the IMM-based algorithm is able to mix the probabilities of multiple regimes to best estimate the regime state at any particular timestep.

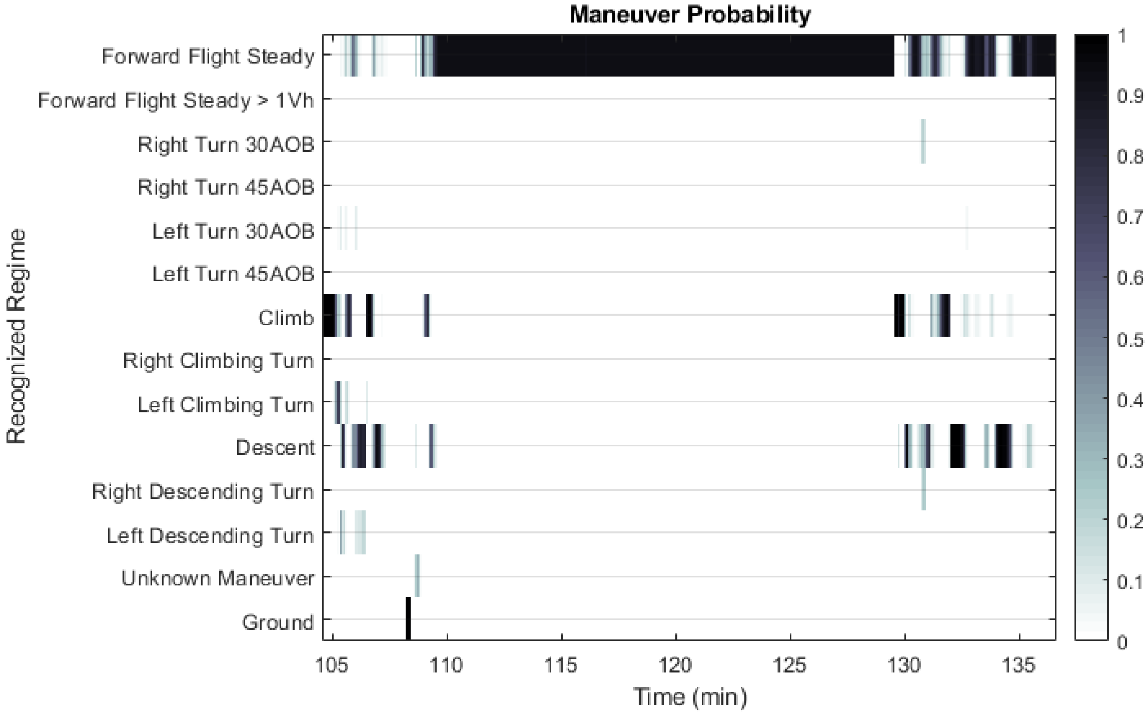

Figure 4.5: IMM-Based RR Classification Results for Real Flight Sequence 3.

4.2 Helicopter Usage Analysis

In traditional (rule-based) regime recognition, the amount of analysis that can be performed on aircraft HUMS data is limited. This limit is due to the deterministic nature of the results. Whereas a probabilistic RR scheme will lend itself more easily to more in depth analyses. This allows for information and relationships to be derived that are simply not possible for a rule-based RR methodology.

Before any usage analysis could be done, regime recognition first had to be performed on all available data. A program was written to open each helicopter flight HUMS file, and perform RR using the IMM methodology discussed in Chapter 2. Because the IMM-RR is based on a Kalman filter, it suffers from some of the same drawbacks. In this case, it was found that missing data in the file stream (i.e corrupted values due to sensor failure or issues with data recording/retrieval) could sometimes cause the IMM to go unstable. In these cases, it was necessary to stop execution of the IMM-RR and save off only data prior to filter instability. If the methods proposed in this dissertation are to ever be deployed within production-grade code, it is recommended that additional pre-processing steps be implemented to remove sections of corrupted data prior to regime recognition. This will insure that the amount of data available for analysis is maximized, and that the results of analysis are not skewed by the vehicle's maneuvering behavior in the earlier portions of the flown missions. After running regime recognition on all available data, just over 500 hours of usable flight data was available for analysis.

4.2.1 Expected Time Analysis

As mentioned above, the use of probabilistic regime recognition allows for greater flexibility when analyzing aircraft usage. It will be useful to convert the regime probabilities at each timesetep into the ratios of "expected time." For this purpose, the *expected time* of each regime r can be computed as follows,

$$E_t(r) = \sum_{j=1}^{n} \Delta t \, p_r^{t_j} \tag{4.1}$$

Where n is the total number of timesteps. Once $E_t(r)$ is computed for all regimes in the RR library, it can then be converted into a fraction of the total time as,

$$F_t(r) = \frac{E_t(r)}{n\Delta t} \tag{4.2}$$

Once the *expected time fraction* is computed, the results can be tabulated and sorted. Table 4.1 shows the results for the processed 500 hr batch of flight data. It was found that Hover and Forward Flight dominate the usage for this particular data set. Forward Climb and Forward Descent were found to be the next most commonly flown regimes. Axial

Climb and Axial descent were flown the next most often. All regimes after that (including the Unknown Regime), are found to have a frequency of less then 5% of the overall flight time.

This type of analysis is analogous to the types of usage case studies that can be easily performed on regime recognition results generated by a rule-based classifier. In a rule-based classifier, each timestep is identified as a particular regime, so the process of computing the *expected time fraction* is straight forward, and could be done as follows,

$$F_{t,RB}(r) = \frac{n_r}{n} \tag{4.3}$$

Where n_r is the number of timesteps identified to be associated with regime r, and n is the total number of timesteps processed.

Table 4.1: Expected Values of Usage Spectrum

Regime Name	Percent
Hover	29.3
Forward Flight Steady	24.0
Climb	11.3
Descent	8.1
Axial Climb	6.5
Axial Descent	6.4
Unknown Maneuvers	4.7
Left Hover Turn	2.2
Right Hover Turn	1.3
Forward Flight Accelerate	1.1
Left Climbing Turn	1.1
Forward Flight Decelerate	0.8
Right Turn 30°AOB	0.8
Left Turn 30°AOB	0.7
Left Descending Turn	0.7
Right Climbing Turn	0.5
Right Descending Turn	0.4
Forward Flight Steady $> 1V_h$	0
Right Turn 45°AOB	0
Left Turn 45°AOB	0

4.2.2 Simultaneous Regimes Analysis

The real benefit of using a probabilistic RR for usage spectrum analysis comes from the ability to extract more useful information then just the expected times and percents. Because a probabilistic regime recognition can give partial classifications of the flown regimes at each timestep (as opposed to a single classification in the case of a rule-based method), an analysis can be performed to see what regimes were active simultaneously, and how often. For this analysis the regime recognition results for the batch of flight data was analyzed by looking only at the timesteps when two regimes each had a probability of greater then 30% at the same timestep. This can be formalized by saying that only the set of timesteps $S = \{t : p_{r1}^{t_j} \geq 0.3 \ , \ p_{r2}^{t_j} \geq 0.3 \ \ \forall \ \ j, r\}$ were considered. From this, it was found that two regimes were simultaneously active (i.e. both had probability ≥ 0.3) for a total of 12.44% of flight time. During analysis, it was recorded which regime pairs appeared together, along with the percent of total flight time that the particulars pairs occurred. The results of that analysis can be seen in Table 4.2, where the first 50 most common regime pairs are shown.

Table 4.2: Regime Pair Usage

Pair #	Regime #1	Regime #2	Percent
1	Forward Flight Steady $0.5V_h$	Forward Flight Steady $0.6V_h$	1.12
2	Hover	Axial Descent	1.04
3	Hover	Axial Climb	0.84
4	Forward Flight Steady $0.4V_h$	Forward Flight Steady $0.5V_h$	0.82
5	Forward Flight Steady $0.6V_h$	Forward Flight Steady $0.7V_h$	0.73
6	Forward Flight Steady $0.3V_h$	Forward Flight Steady $0.4V_h$	0.37
7	Forward Flight Steady $0.7V_h$	Forward Flight Steady $0.8V_h$	0.37
8	Climb	Left Climbing Turn	0.36
9	Axial Descent	Unknown Maneuver	0.36
10	Axial Climb	Unknown Maneuver	0.36
11	Left Hover Turn	Unknown Maneuver	0.36
12	Climb	Descent	0.34
13	Descent	Left Descending Turn	0.31
14	Hover	Right Hover Turn	0.31
15	Hover	Left Hover Turn	0.27
16	Forward Flight Steady $0.2V_h$	Forward Flight Steady $0.3V_h$	0.23
17	Forward Flight Steady $0.5V_h$	Descent	0.22
18	Forward Flight Steady $0.5V_h$	Climb	0.21

Table 4.2 (continued): Regime Pair Usage

Pair #	Regime #1	Regime #2	Percent
19	Forward Flight Steady $0.6V_h$	Descent	0.20
20	Climb	Right Climbing Turn	0.19
21	Axial Climb	Axial Descent	0.19
22	Forward Flight Steady $0.6V_h$	Climb	0.18
23	Forward Flight Steady $0.8V_h$	Forward Flight Steady $0.9V_h$	0.18
24	Forward Flight Steady $0.9V_h$	Forward Flight Steady $1V_h$	0.14
25	Forward Flight Steady $0.4V_h$	Climb	0.13
26	Forward Flight Steady $0.7V_h$	Climb	0.13
27	Hover	Forward Flight Steady $0.1V_h$	0.13
28	Forward Flight Steady $0.4V_h$	Descent	0.12
29	Forward Flight Steady $0.7V_h$	Descent	0.12
30	Right Hover Turn	Unknown Maneuver	0.11
31	Descent	Right Descending Turn	0.11
32	Forward Flight Steady $0.1V_h$	Forward Flight Steady $0.2V_h$	0.10
33	Axial Descent	Descent	0.09
34	Forward Flight Steady $0.5V_h$	Right Turn 30°AOB	0.09
35	Forward Flight Steady $0.8V_h$	Climb	0.09
36	Forward Flight Steady $0.3V_h$	Climb	0.08
37	Forward Flight Steady $0.1V_h$	Rearward	0.08
38	Forward Flight Steady $0.8V_h$	Descent	0.07
39	Forward Flight Steady $0.1V_h$	Descent	0.07
40	Forward Flight Steady $0.3V_h$	Descent	0.07
41	Forward Flight Steady $0.4V_h$	Right Turn 30°AOB	0.06
42	Forward Flight Steady $0.6V_h$	Right Turn 30°AOB	0.06
43	Forward Flight Steady $0.2V_h$	Climb	0.06
44	Climb	Unknown Maneuver	0.06
45	Forward Flight Steady $0.5V_h$	Left Turn 30°AOB	0.05
46	Forward Flight Steady $0.2V_h$	Descent	0.05
47	Left Turn 30°AOB	Left Climbing Turn	0.05
48	Left Turn 30°AOB	Left Descending Turn	0.05
49	Axial Climb	Climb	0.04
50	Forward Flight Steady $0.6V_h$	Left Turn 30°AOB	0.04

The results shown in Table 4.2 can provide some deep insight into the type of maneuvers the aircraft is flying during missions. The majority of cases are logical, and even expected. Because Table 4.2 shows the frequency that two regimes are both active it makes sense that regimes that have similar dynamic behavior would appear together. One situation where two similar regimes could be active is during the transient period when the aircraft is transitioning from one regime to another. An example of this is if the vehicle is transitioning from the Hover regime to an Axial Climb, such as might be the situation in Pair #3. A second possible situation that could cause two regime to be active is if the vehicle is flying on the defined boundary between two regimes. An example of this is if the aircraft is flying in a slow climb. In this situation, the vehicle would be 50% in "Level Flight", and 50% in "Climb." A correctly tuned probabilistic RR would classify the regimes accordingly, leading to the situation were both regimes were identified as active. An example of this might be seen in Pair #18. Although, without access to the underlying aircraft state data, there is no way of determining which of the two situations described could be the cause of two simultaneous regime pairs being active.

When looking at the results of in Table 4.2, some interesting cases can be seen that appear – at first glance – to give contradicting regime pairs. An example of a seemingly erroneous pair can be seen in Pair #12, where the "Climb" and "Descent" regimes are found to both be active simultaneously. However, this behavior can be caused when the aircraft is rapidly changing between regimes. In this situation, the IMM filter response for "Climb" and "Descent" regime probabilities may not have time to decay before the other regime probability begins to rise, leading to the apparent simultaneous activation of two opposed regimes. While this behaviour has been observed during the development and testing of the IMM-RR, more testing is warranted to determine if a similar behavior is present if a different probabilistic RR were to be used (such as an HMM-based algorithm). In production, if two opposed regimes are often flagged as simultaneously active, it could also indicate to the aircraft designer that additional work is needed on the algorithm. It may be the case that the IMM filters need additional tuning to reduce the frequency of occurrence – if the events are determined to be erroneous. If regime boundary definitions are close together in the state-space, steady-state error in the filters can interfere with the regime recognition process by causing regime probabilities to have a baseline value that is large enough to make a regime appear "always active." This can usually be fixed by additional filter tuning, or by appropriately modifying the regime probability formulas (such as those seen in Table 2.3).

4.2.3 Unknown Regime Analysis

For usage spectrum analysis to be effective, the percent of time that the regime recognition algorithm results in a classification of "Unknown Regime" must be kept as low as possible. While some Unknown Regime will almost certainly be present, the exact "acceptable amount" is subjective. For this reason, it may be useful to quantify how much "Unknown Regime" is present in a batch of RR processed flight data. For the available flight data, a set of decreasing thresholds were swept through the IMM-RR results, and the fraction of time the Unknown Regime that was less then the threshold was recorded. Or, more formally, for a given threshold i, let b be equal to the number of timesteps where $p_u^{t_j} < i$ for all n timesteps processed, and where u corresponds to the index for the Unknown Regime. Then the fraction of time that the Unknown Regime is less than or equal to the threshold can be found as,

$$F_t(r = u) = \frac{b}{n} \tag{4.4}$$

This was done for all available processed flight data, and the results are recorded in Table 4.3. In Table 4.3 it can be seen that a threshold of 1 yields an unknown percent of 100%. This simply means that 100% of the time, the unknown regime has a probability of less then or equal to 1 across all timesteps. This makes sense, although the information is not very useful. The more interesting results will be found as the threshold is lowered. For a threshold of 0.5, Table 4.3 states that more then 95% of the time, the Unknown Regime had a probability of 0.5 or less. The results can easily be reversed by taking one minus the percent. For example, for the same threshold of 0.5, it can be seen that only 4.4% of the total time the unknown regime had a probability greater then 0.5. Looking towards the bottom of the table, at a threshold of 0.02, a value of 90.5% can be seen in the right column. This is interpreted as saying, that greater then 90% of the time, the Unknown Regime had a probability of less then 2%. Together, these results provide a good indication that

Table 4.3: Percent of Time the Unknown Regime is Less Than Threshold i

Threshold, i	Unknown $\leq$ Threshold, %
1	100.0
0.9	97.4
0.8	96.7
0.7	96.3
0.6	96.0
0.5	95.6
0.4	95.2
0.3	94.8
0.2	94.2
0.1	93.4
0.05	92.4
0.04	92.1
0.03	91.6
0.02	90.5
0.01	86.7
0	31.3

the Unknown Regime is not overly active. As mentioned before, analyzing the Unknown Regime behavior can provide insight into potential problems. If the Unknown Regime is found to be more active in some batches of data then is considered normal for a particular aircraft, it could indicate faulty sensors or corrupted data are present. This would, of course, depend on what states the regime recognition code requires for correct operation, and how error handling is performed both in the HUMS unit and the RR algorithm. A high presence of the Unknown Regime might also signal to an operator that the aircraft is flying regimes do no not appear in the regime recognition library.

It should be noted that this style of analysis could be performed with any regime, not just the Unknown Regime, and may be useful if an operator or aircraft designer wanted to know what percent of time an aircraft spends in regimes with different probability thresholds. Again, it should be noted that this of in depth analysis is difficult or even impossible with traditional rule-based regime recognition codes.

4.2.4 Thresholded Regime Analysis

This next analysis is similar to that found in the previous section, in that a threshold probability is swept through the results and regime percent-of-time is calculated. However, in this case, all regime are considered simultaneously. For a given threshold i, let k be the number of timesteps where $max(p_r^j) < i \; \forall \; r \in R$ for each timestep j over the number of all timesteps processed n, and where R corresponds to the set of all regimes in the RR library. Then the fraction of time that the highest probability regime is below the given threshold can be calculated as,

$$F_{th} = \frac{k}{n} * 100 \tag{4.5}$$

For each threshold i the most common regime satisfying the given conditions was recorded. The results of the analysis can be seen in Table 4.4.

Table 4.4: Maximum Regime Probability Against Threshold i

Threshold, i	Max Regime Probability $\leq i$, %	Most Common Regime
1.00	100.0	Hover
0.95	59.0	Hover
0.90	45.7	Hover
0.85	36.6	Hover
0.80	30.3	Forward Flight $0.6V_h$
0.75	25.4	Forward Flight $0.6V_h$
0.70	20.7	Forward Flight $0.6V_h$
0.65	16.7	Forward Flight $0.6V_h$
0.60	12.8	Forward Flight $0.6V_h$
0.55	9.0	Forward Flight $0.6V_h$
0.50	5.0	Climb

Analyzing the results can once again provide insight into the aircraft usage. It can be seen in Table 4.4 that when the threshold is set to 1, 100% of the greatest regime probability was less than or equal to 1. Once again, this makes intuitive sense, but is not all that useful. The rightmost column of Table 4.4 states that the most common regime (at Threshold = 1) is Hover. This is matches the findings for the most common regime stated in Table 4.1. As the threshold is lowered, the percent of total time that the regime with greatest probability has a probability over that threshold is also lowered. Taking an example from Table 4.4, it can be seen that 25.4% of the time, the regime with highest probability had a probability of less then 0.75, and the most common regime fitting that specification was Forward Flight at $0.6V_h$. Overall, Table 4.4 can be used to determine how often the probabilistic regime recognition is uncertain about the flown regimes, and which regimes those are. From Table 4.4 it can be concluded that when the IMM-RR is uncertain about the flown regime, it is most likely to be a Hover or Forward Flight based maneuver. This makes sense, as helicopters often spend the majority of their time in those two regime, and when climbs and descents are made, they are often not executed at maximum aggression. This results in a lot of slow climbs and descents, that the IMM-RR classifies as partial maneuvers. To reiterate, this is one of the primary benefits of a probabilistic regime recognition scheme over its rule-based counterparts as documented in Chapter 2, as well as in References [58] and [59].

4.3 Helicopter Damage Estimation

The end-goal of many regime recognition schemes is for use in Condition Based Maintenance for fatigue damage estimation of life-limited components. In this section, the actual HUMS data used for regime recognition and usage analysis in the last section will be used. The techniques applied will be those proposed in Chapter 3. While the regime boundary definitions remain the same as in previous chapters (see Tables 2.1 and 2.2), the damage rates for this chapter are given in Table 4.5. Note that these damage rates are for demonstration purposes only and are not meant to represent any true component or set of regimes.

Table 4.5: Regime Damage Rates

Regime	Parameter	Damage/s
Hover	–	0
Right Hover Turn	–	2.89×10^{-9}
Left Hover Turn	–	2.41×10^{-9}
Axial Climb	–	4.66×10^{-9}
Axial Descent	–	1.90×10^{-9}
Forward Flight Accelerate	–	6.62×10^{-8}
Forward Flight Decelerate	–	6.62×10^{-8}
Forward Flight Steady	$0.1V_h$ – $0.9V_h$	0
Forward Flight Steady	$1V_h$	3.38×10^{-8}
Forward Flight Steady	$> 1V_h$	6.33×10^{-8}
Right Turn	$30°$AOB	1.23×10^{-7}
Right Turn	$45°$AOB	1.35×10^{-6}
Left Turn	$30°$AOB	5.99×10^{-8}
Left Turn	$45°$AOB	7.99×10^{-7}
Climb	–	2.62×10^{-7}
Right Climbing Turn	–	5.20×10^{-6}
Left Climbing Turn	–	4.00×10^{-7}
Descent	–	1.55×10^{-7}
Right Descending Turn	–	4.97×10^{-6}
Left Descending Turn	–	3.13×10^{-7}
Unknown Maneuver	–	5.20×10^{-6}

4.3.1 Cumulative Damage

The first example will be to compute the cumulative damage of a *theoretical* component using all available HUMS data. The damage was computed using the Gaussian approximation discussed in Section 3.1.3, and the damage rates in Table 4.5, resulting in the cumulative damage Gaussian seen in Figure 4.6. The total damage was computed twice. Once where the standard deviation on the damage rates were set to zero for all regimes (seen as the blue line). The second time, the standard deviation of each damage rate was set to 20% of its nominal value (seen as the orange line). For example, for a given damage rate $d_r = 2.89 \times 10^{-9}$, the standard deviation for that damage rate would be set to $\sigma_{d_r} = 0.2 * (2.89 \times 10^{-9}) = 0.579 \times 10^{-9}$. The mean of both distributions was 0.6367. This implies that an estimated 63.7% of the components useful life has been consumed.

To remain conservative, one would likely select a high percentile value (e.g. $.9_6$) for the purpose of part replacement timeline creation. The standard deviation of the Gaussian corresponding to 20% damage rate standard deviation is 2.78×10^{-4}, while the standard deviation of the Gaussian corresponding to zero damage rate standard deviation is 2.56×10^{-4}, which as expected, is smaller. The difference between these two standard deviations is very small relative to the number of hours processed. However, it should be noted that the regime usage analysis in the Section 4.2 indicated that the vehicle flew mostly low or non-damaging regimes, and only approximately 12% of the IMM-RR was uncertain about what regime was being flown. Together, this means the IMM-RR, although still probabilistic, was behaving *more like* a rule-based RR algorithm. Put another way, because the IMM-RR resulted in high regime certainty $\sim$90% of the time, the corresponding damage estimate had less overall uncertainty and the effects of additional uncertainty from the damage rates were reduced.

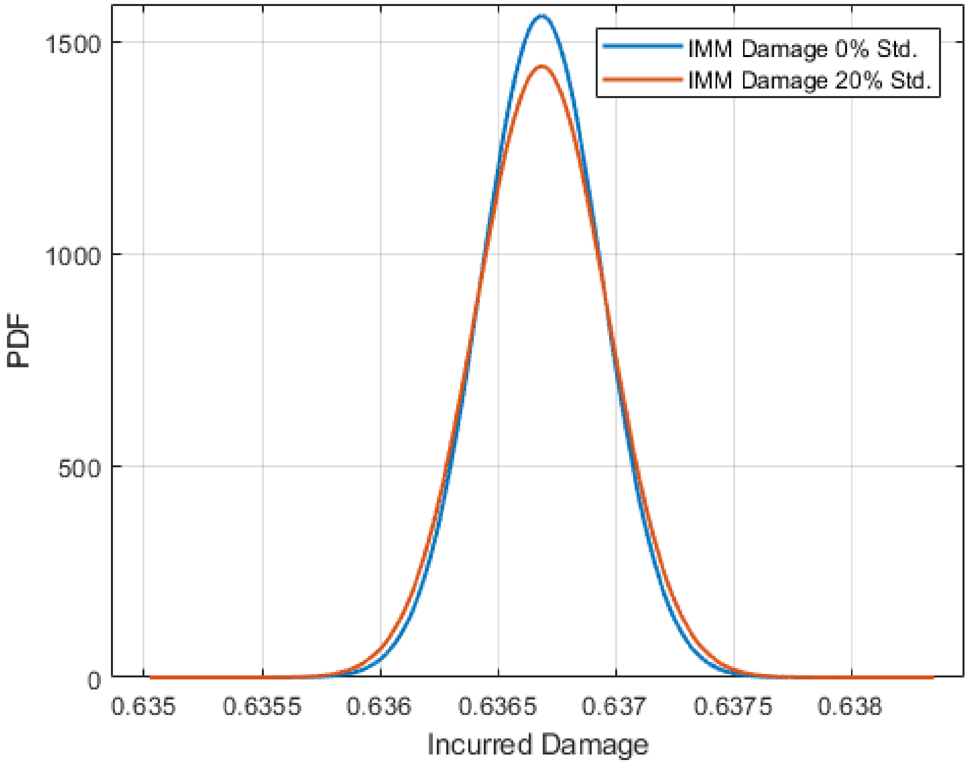

Figure 4.6: Cumulative Damage of Helicopter Component.

4.3.2 Effects of Saturating Probabilistic RR Results

An important factor when evaluating different regime recognition and condition based maintenance scheme combinations is the level of conservativeness in the resulting damage estimate. To effectively evaluate the IMM-RR results, a comparison needed to be made with a different methodology. Because rule-based RR results were unavailable for the batch of helicopter HUMS data processed, a saturation method was employed as a substitute to create a *pseudo* rule-based RR. This is a reasonable substitute because the regime identified with the highest certainty will be promoted to the singularly representative regime. This is not unlike what happens within a rule-based RR code, where a single regime is selected with 100% certainty at each timestep, as discussed in Chapter 1. The saturation of the IMM-RR will be accomplished by taking the regime probability array at each timestep and assigning a value of 1 to the highest probability, and 0 to all others. The algorithm to create this new *saturated IMM-RR* result is,

$$p^j_{r,sat} = \begin{cases} 1 & \text{if} \quad p^j_r = max(p^j) \ \forall \ r \in R \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

Where p^j is the regime probability array at timestep j generated from a probabilistic regime recognition, and R is the set of all regimes in the RR library. As a simple example, consider a case with only four regimes $r \in \{1, 2, 3, 4\}$. Assume the probabilistic RR code identifies regime probabilities at timesteps $j \in \{1, 2, 3\}$ as $p^1 = \{0.2, 0.6, 0.1, 0.1\}$, $p^2 = \{0.3, 0.4, 0.2, 0.1\}$, $p^3 = \{0.4, 0.3, 0.3, 0.0\}$. The resulting saturated values would be $p^1_{sat} = \{0, 1, 0, 0\}$, $p^2_{sat} = \{0, 1, 0, 0\}$, and $p^3_{sat} = \{1, 0, 0, 0\}$.

Using the algorithm in Equation 4.6, all the results generated from using the IMM-RR on the entire batch of HUMS data were saturated, and the incurred damage was computed. The same techniques, set points, damage rates, etc. as those used to generate the results in Figure 4.6 with 20% standard deviation on the damage rates. The results of the damage estimation of the saturated IMM-RR results are seen in Figure 4.7. Immediately it can be seen that both the mean and differ between the two PDFs. Because the saturated RR assumes no uncertainty in the regimes, the fact that there is any variance in the saturated results may not be immediately obvious. However, the variance present is entirely the result of the uncertainty in the damage rates, as is evident from Equation 3.9. The mean of the saturated RR results (seen in orange) is a reduction of 2.03% from the mean resulting from the full probabilistic treatment of the regime uncertainty (seen in blue). The variance, however, is decreased from 7.67×10^{-8} to 1.11×10^{-8}, a reduction by a factor of 6.93. This result clearly points to the effects regime uncertainty has on the uncertainty in the damage estimate. Initially, it may seem that a reduction in the mean estimated damage of only about 2% is not significant. However, it has already been shown that the aircraft flew mostly non-damaging maneuvers, meaning this change in mean damage was caused by a much smaller subset of the overall flight time. If the aircraft were to fly only high-damaging regimes, the difference between the two variances would be much greater. Additionally, even if the aircraft were to continue flying with the same regime usage spectrum, a difference of only 2% can cause a significant difference in the number of hours remaining on a life-limited component. Assuming the aircraft were to continue flying with the same regime usage seen in Table 4.1, the saturated form of the IMM-RR damage estimation would over-predict the available remaining component life by 16.8 hours, using the mean values of the damage PDFs. If higher percentile values were used, the difference in estimated remaining-life would be even greater. A difference of 16.8 hours could be the difference between a vehicle being greenlit for a mission, or being grounded for repairs and maintenance. Or, in a worst case scenario, this discrepancy in estimated remaining life might be the difference between safe operation and catastrophic component failure.

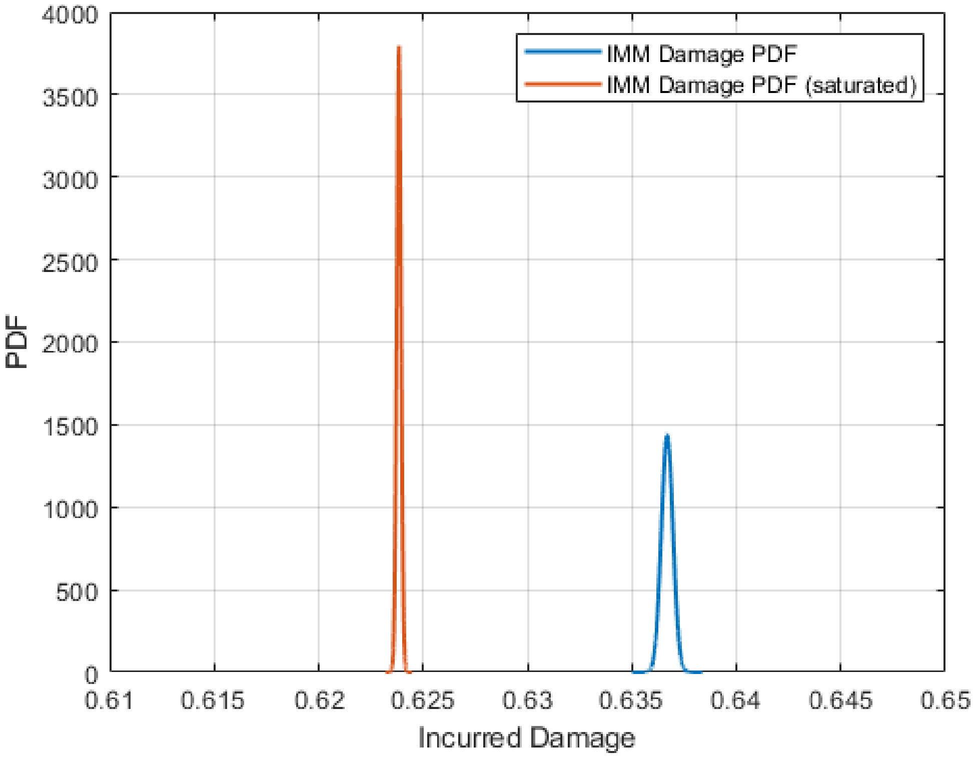

Figure 4.7: Damage PDF with IMM-RR and Saturated IMM-RR Results.

In order to make claims about which methodology is more conservative, it is important to gain a more full understanding of what is happening to in the damage estimate of the saturated case. To this end, a study was performed to analyze the damage rate "promotions" that resulted from the use of the algorithm in Equation 4.6. At each timestep, when Equation 4.6 was applied, a check was performed to see if the damage rate associated with $max(p^j)$ was greater, lesser, or equal to the damage rate associated with the regime that was assigned the second highest probability by the IMM-RR. In this way, it can be determined if the damage rate actually applied to each timestep is over- or under-estimating the damage.Figure 4.8 shows the results of the damage rate promotions.

In Figure 4.8 it can be seen that a *lesser* damage rate was promoted 68% more often then a *greater* damage rate. Or, put another way, this shows that the damage rate with the lower nominal value is being promoted more often then the damage rate with the higher nominal value. This implies that when the probabilistic regime recognition results are saturated

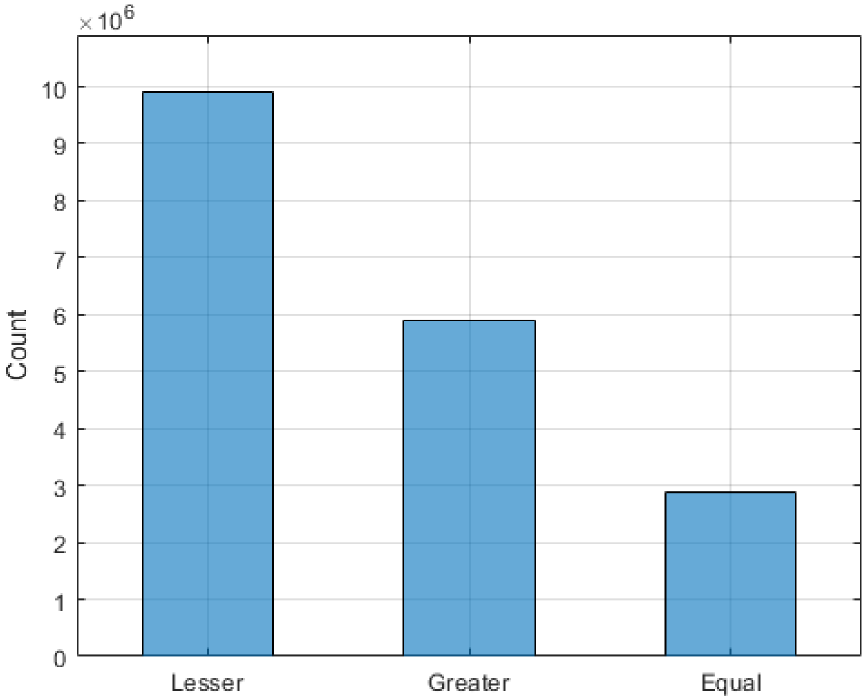

Figure 4.8: Damage Rate Promotions Using Saturation Algorithm.

using the algorithm in Equation 4.6, the cumulative damage estimate is under-predicted for the batch of flight data processed, when compared to using the full probabilistic treatment of the cumulative damage estimate as described in Chapter 3. While the results of Section 4.3.2 are technically only valid for the vehicle usage of the flight data processed, coupled with the damage rates used in Table 4.5, some generalizations can still be made. Because aircraft tend to spend a lot of time flying in fundamental maneuvers (i.e. level forward flight, hover, etc.) the damage rate associated with those fundamental regimes will get promoted more often if a saturation algorithm is used. Also, because a rule-based regime recognition algorithm shares similar characteristics, in terms of resulting regime classifications, to those of saturating a probabilistic RR output, many (if not all) of the comparisons and conclusions made here would be valid for both.

Because condition based maintenance programs are useful, not only for cost savings, but for personnel safety as well, it is important to know how different algorithms and methodologies affect the conservativeness of the damage estimate. The results of this section indicate that full treatment of the regime and damage rate uncertainty provides a more conservative damage estimate over rule-based or saturation-based methods.

4.3.3 Helicopter Damage Spectrum

There are many reasons why individual aircraft tracking may not be feasible. Corrupted data, faulty sensors, and a myriad of other reasons may prevent the retrieval or availability of the data necessary for successful IAT. For this reason, it may be desirable to perform usage spectrum analysis on a batch of flight data in order to gain insight into the estimated damage of the vehicle (or fleet of vehicles) with respect to a fixed time rate.

A simple averaging of the total fatigue damage rate per component is one approach that could be taken. In this way IAT could be performed on an available batch of flight data. Then the mean value of the damage could be selected and divided by the number of hours processed to provide an "average damage per time." However, a simple time-averaging approach will not properly address uncertainty in the vehicle usage.

An alternative approach is that proposed in Section 3.3.3 where a bootstrapped statistical method is used to encapsulate variability in the aircraft usage spectrum. In this study, the methods proposed in Section 3.3.3 (whose process flow is shown in Figure 3.4) are employed on the real-world HUMS data. In accordance with the proposed methodology, random flights are selected from the total batch of available data and the damage is estimated using the IMM-RR and DCM discussed in Chapters 2 and 3. The level of conservativeness chosen was set to both the 99.9999th and 50th percentile so that comparisons could be made between the two. In the same way as before, Equation 3.10 was used to scale the damage random variable to 100 hours at each iteration of the re-sampling. The damage rates for a theoretical component were set equal to those found in Table 4.5. Figure 4.9 shows the resulting histogram of the $\gamma_{\eta,i}$ for $i \in \{1, ..., 10,000\}$ randomized sets of flights taken from all available in the batch for the 99.9999th percentile of damage during each iteration of 100 hours. The histogram in Figure 4.9 is observed to approximate a normal distribution, this is inline with the simulation based findings in Section 3.3.3. For this case study, the damage per 100 hours probability distribution was found to have a mean of 0.1225, a standard deviation of 0.0184, and a range of 0.1426. Note that the damage for the entire batch of flight data (518 hrs) computed using the IAT method in Section 4.3.1 can be scaled to yield an average damage rate of 0.1227 per 100 hrs (shown as the vertical red line in Figure 4.9). This results in a percent difference of 0.14% compared to the mean damage per 100 hrs computed using the bootstrapped method. This further validates the bootstrapped method as a viable solution to estimating the damage of the vehicle when an Individual Aircraft Tracking approach is not possible.

In Figure 4.10 the empirical CDF is shown, created from the underlying data in the histogram for both the 99.9999th percentile ($.9_6$) and the 50th percentile (μ). It can be seen that the empirical CDF corresponding to the 99.9999th and 50th percentile are almost identical. This in part, helps to answer a question that was asked in Section 3.3.3 about whether the mean value, or a highly conservative value, should be selected from the damage spectrum CDF for use in practically applying the damage rate to a vehicle(s) for CBM. Because the vehicle (in the cases presented here) flew mostly non/low-damaging regimes with relatively low regime uncertainty the uncertainty present in the damage estimate at the end of each 100 hr randomly sampled block was also relatively low, resulting in a very small difference between the two distributions (99.9999th and 50th percentile).

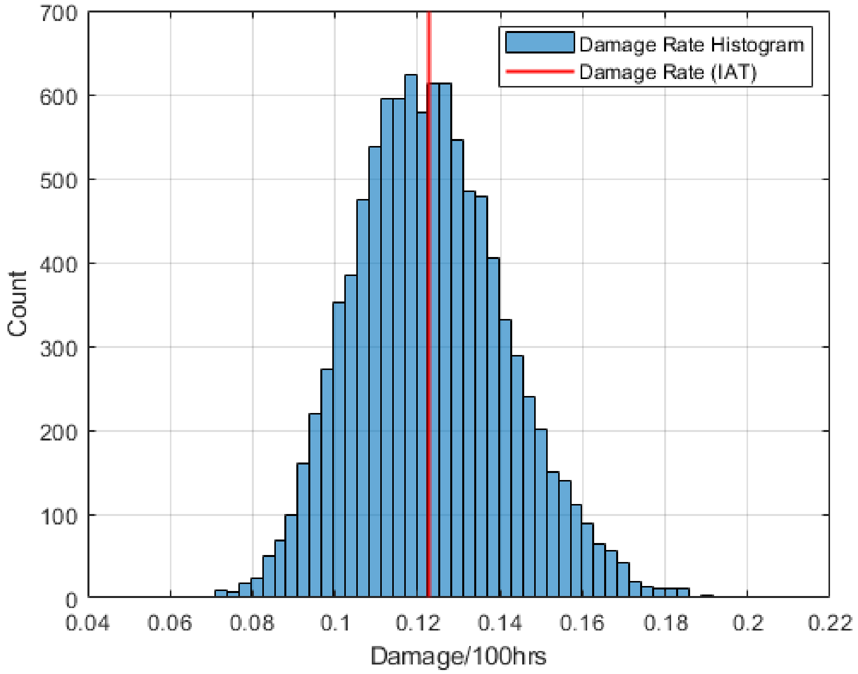

Figure 4.9: Histogram of Damage Per 100 hours at the 99.9999th Percentile.

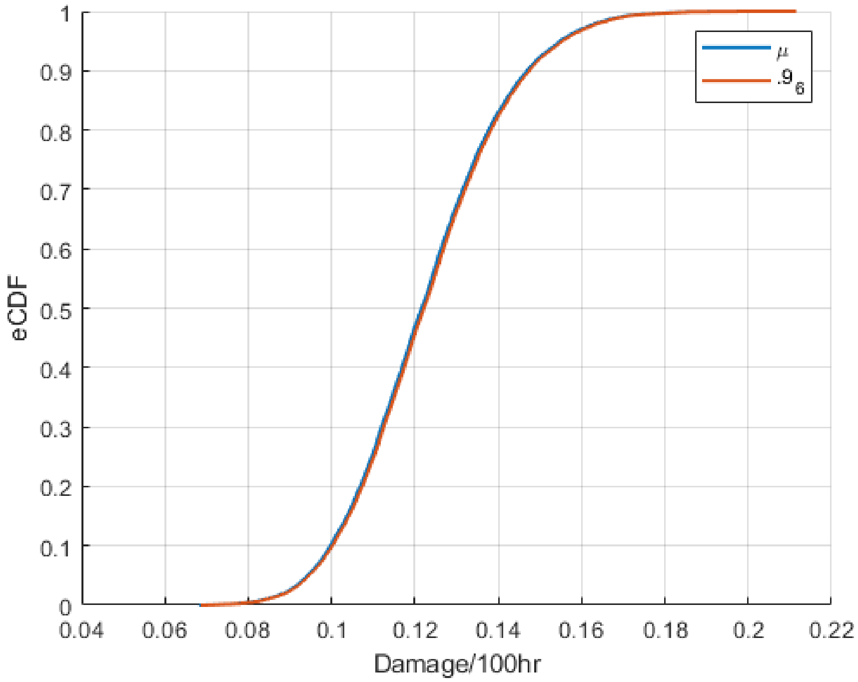

Figure 4.10: Empirical CDF of Damage Per 100 hours.

CHAPTER 5

APPLICATION TO FIXED WING AIRCRAFT

The regime recognition and damage estimation algorithms presented in Chapters 2 and 3 have – so far – only been used on helicopters. However, there is nothing in either algorithm formulation that precludes its use on fixed wing aircraft. As discussed in Chapter 1, fixed wing aircraft often make use of strain gauges and load sensors mounted directly on the vehicle airframe. Load data can then be collected and analysed using the flight load synthesis techniques mentioned in Chapter 1. For this reason, the damage estimation algorithms presented in Chapter 3 may not provide any additional useful information about the damage state of the aircraft components. However, the probabilistic regime recognition algorithms in Chapter 2 and usage analysis techniques in Chapter 4 can provide useful information about the type of missions being executed. This chapter will seek to employ the proposed IMM-RR algorithms on data collected from a simulated General Dynamics F-16.

5.1 Aircraft Dynamic Model

In this section, a nonlinear aircraft will be described. This model, taken from Reference [107], is the basis for a program that was written to provide the necessary aircraft state data for applying the IMM-RR algorithm – in a similar way that HIVE was used to generate HUMS data in Chapters 2 and 3. It should be noted that not all necessary information for defining the aircraft model is present in this section. A general overview is provided, but some additional state equations, aerodynamic coefficients, and lookup tables, that may be required for implementation, are omitted here. A full detailing of the model used can be found in Reference [107]. Some aircraft properties for the F-16 are given in Table 5.1, the remaining properties as well as all aerodynamic lookup tables can be found in Reference [108].

Table 5.1: F-16 Selected Properties

Property	Value	Unit
Weight	20,500	lbs
Moment of Inertia J_{xx}	9,496	Slug-ft^2
Moment of Inertia J_{yy}	55,814	Slug-ft^2
Moment of Inertia J_{zz}	63,100	Slug-ft^2
Moment of Inertia J_{xz}	982	Slug-ft^2
Wing Span	30	ft
Wing Area	300	ft^2
Mean Aerodynamic Chord	11.32	ft

5.1.1 Assumptions

The aircraft is assumed to be rigid, with constant mass, and symmetry about the X-Z plane. The mass density of the aircraft is assumed constant, and internal components (such as the motion and quantity of fuel) are ignored. External forces and moments acting on the aircraft are assumed to be only from from gravity, propulsion, and aerodynamics. A flat earth approximation is used, with a stationary atmosphere. Flights are limited to 50,000 feet in altitude and speeds less than Mach 1. The gravitational pull of the Earth is assumed to be a uniform field, such that the center of mass and center of gravity are coincident on the aircraft. All sign conventions follow the standard right-hand rule [109].

5.1.2 Aircraft States and Controls

For the aircraft model, the following state vector is used,

$$\boldsymbol{x} = [V_t \ \alpha \ \beta \ \phi \ \theta \ \psi \ p \ q \ r \ x_E \ y_E \ h \ P_a]^T \tag{5.1}$$

where, V_t is the airspeed of the aircraft, α is the angle of attack, β is the angle of sideslip, ϕ, θ, ψ are the Euler angles, p, q, r are the body-axis angular rates, x_E, y_E are the X and Y earth-axis positions respectively, h is the aircraft altitude above sea level, and P_a is the engine power level expressed as a percent. In addition to the vehicle states the following control state vector was used,

$$\boldsymbol{u} = [\delta_{th} \;\; \delta_e \;\; \delta_a \;\; \delta_r]^T \tag{5.2}$$

In this equation, the variables correspond to the control surface deflection for the throttle (δ_{th}), elevator (δ_e), aileron (δ_a), and rudder (δ_r).

The set of coupled first-order nonlinear differential equations that govern the dynamics of the aircraft can be represented as,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right) \tag{5.3}$$

with output equations,

$$\boldsymbol{y} = \boldsymbol{h}\left(\boldsymbol{x}, \boldsymbol{u}\right) \tag{5.4}$$

5.1.3 Equations of Motion

There are six nonlinear rigid body equations of motion used to model the translational and rotational dynamics of the aircraft [107]. These are made up of three force equations and three moment equations.

Force Equations

$$\dot{u} = rv - qw - g\sin\theta + \frac{\bar{q}SC_x + T}{m} \tag{5.5}$$

$$\dot{v} = pw - ru - g\cos\theta\sin\phi + \frac{\bar{q}SC_y}{m} \tag{5.6}$$

$$\dot{w} = qu - pv - g\cos\theta\cos\phi + \frac{\bar{q}SC_z}{m} \tag{5.7}$$

Moment Equations

$$\dot{p} = \left(c_1 r + c_2 p + c_4 h_{eng}\right) q + \bar{q} S b \left(c_3 C_l + c_4 C_n\right) \tag{5.8}$$

$$\dot{q} = \left(c_5 p - c_7 h_{eng}\right) r - c_6 \left(p^2 - r^2\right) + \bar{q} S \bar{c} c_7 C_m \tag{5.9}$$

$$\dot{r} = \left(c_8 p - c_2 r + c_9 h_{eng}\right) q + \bar{q} S b \left(c_4 C_l + c_9 C_n\right) \tag{5.10}$$

In addition to the force and moment equations, there are three rotational kinematic equations, and three navigation equations [107]. The rotational kinematic equations describe the relationship between the aircraft Euler angular rates and the body-axis angular rates. The navigation equations describe the relationship between the aircraft translational velocity components expressed in the body axis and the aircraft translational velocity expressed in the earth-axis. These six total equations are given below.

Kinematic Equations

$$\dot{\phi} = p + \tan\theta \left(q \sin\phi + r \cos\phi\right) \tag{5.11}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{5.12}$$

$$\dot{\psi} = \frac{q \sin\phi + r \cos\phi}{\cos\theta} \tag{5.13}$$

Navigation Equations

$$\begin{aligned} \dot{x}_E = {} & u \cos\psi \cos\theta + v \left(\cos\psi \sin\theta \sin\phi - \sin\psi \cos\phi\right) \\ & + w \left(\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi\right) \end{aligned} \tag{5.14}$$

$$\begin{aligned} \dot{y}_E = {} & u \sin\psi \cos\theta + v \left(\sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi\right) \\ & + w \left(\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi\right) \end{aligned} \tag{5.15}$$

$$\dot{h} = u \sin\theta - v \cos\theta \sin\phi - w \cos\theta \cos\phi \tag{5.16}$$

These twelve total equations (given by eqs. (5.5) to (5.16)) are sufficient to describe the nonlinear equations of motion for the aircraft, and can be used in a numerical integration scheme to propagate the aircraft states forward in time. Complete descriptions of each equation can be found in Ref. [107].

5.1.4 Engine Model

The engine power is modeled using a first-order lag filter to relate the commanded power level to the actual power response. The equation governing this relationship is given by,

$$\dot{P}_a = \frac{1}{\tau_{eng}} \left(P_c - P_a\right) \tag{5.17}$$

where P_c is the commanded power level computed as a function of the throttle state δ_{th}, and τ_{eng} is the engine power level time constant. Additional details regarding the engine model used can be found in Ref. [107].

5.1.5 Atmospheric Model

The air density and speed of sound is calculated in accordance with the U.S. Standard Atmosphere model [107, 108]. Air density ρ, dynamic pressure $\bar{q}$, and Mach number M are governed by,

$$\rho = 0.002377 \left(T^*\right)^{4.14} \tag{5.18}$$

$$M = \begin{cases} \frac{V_t}{\sqrt{1.4(1716.3)(390)}} & \text{if} \quad h \geq 35{,}000 \text{ ft} \\ \frac{V_t}{\sqrt{1.4(1716.3)(519T^*)}} & \text{if} \quad h < 35{,}000 \text{ ft} \end{cases} \tag{5.19}$$

$$\bar{q} = \frac{1}{2}\rho V_t^2 \tag{5.20}$$

where $T^* = 1 - 0.703h \times 10^{-5}$ is the computed temperature factor [107].

5.2 Aircraft Control Laws

In order to simulate flights of the aircraft, a control scheme is needed. To this end, a set of controllers was developed using some of the same techniques found in the *virtual pilot* algorithm discussed in prior chapters and found fully documented in References [87] and [78]. Because the set of maneuvers needed to demonstrate the feasibility of IMM-RR on fixed wing aircraft is relatively small (compared to the 100+ maneuvers that may be implemented for deployment on actual vehicles), a simplified control scheme could be used. The control design is based primarily on the autopilot PID controllers for altitude and roll-attitude hold found in *Aircraft Control and Simulation, 2nd Ed.* by Stevens and Lewis [108]. Additional controllers were added using a similar architecture for regulating sideslip, and forward speed. A control update rate of 20 Hz was used for these studies. All four controllers (manipulating control variables δ_{th}, δ_a, δ_r, and δ_e) were tuned and operated independently of each other. More information regarding general PID controller design and tuning can be found in *Modern Control Engineering, 4th Ed.* by Ogata [110].

5.2.1 Controller Architecture

Forward Speed, Roll Angle, and Sideslip Channels

The controllers used for the Forward Speed, Roll Angle, and Sideslip channels were Single Input Single Output (SISO) PID controllers. Each controller used a specified parameter imported from the *flight card* to operate on the relevant aircraft state variable. The engine throttle control variable δ_{th} was manipulated using commanded airspeed V_{tc} as reference, where airspeed was used as an approximation for the forward speed of the aircraft. The aileron control variable δ_a was manipulated using the commanded Euler roll angle ϕ_c as reference. Finally, the rudder control variable δ_r was manipulated using the commanded sideslip angle β_c as reference. Because the sideslip controller was used solely to regulate sideslip to zero for all flights, the sideslip angle setpoint was set to a constant zero in the control law. Control block diagrams for all three channels can be seen in Figure 5.1.

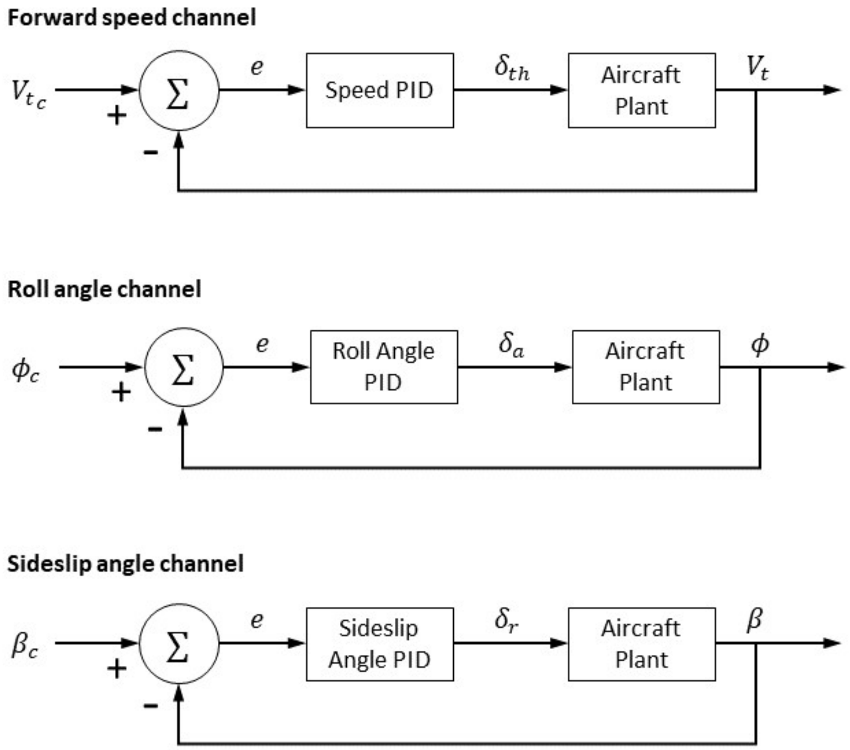

Figure 5.1: Forward Speed, Roll Angle, and Sideslip Control Loop Structures.

Altitude Channel

The controller used for the Altitude channel is a SISO nested-loop PID controller. In this controller, the altitude setpoint h_c is used by the outer-loop PID to generate a pitch angle setpoint θ_c for the inner-loop PID. The inner-loop PID is then used to drive the elevator control variable δ_e. The control block diagram for the altitude hold can be seen in Figure 5.2. If a rate of climb track was desired, it was supplied to the controller as a ramp input to the altitude setpoint, whose slope was set equal to the desired rate of climb. This resulted in satisfactory performance for the type of flights used in these studies – without the additional complexity of a variable structure control architecture, like that used in the *virtual pilot algorithm* developed by Fowler (Refs. [78, 87]).

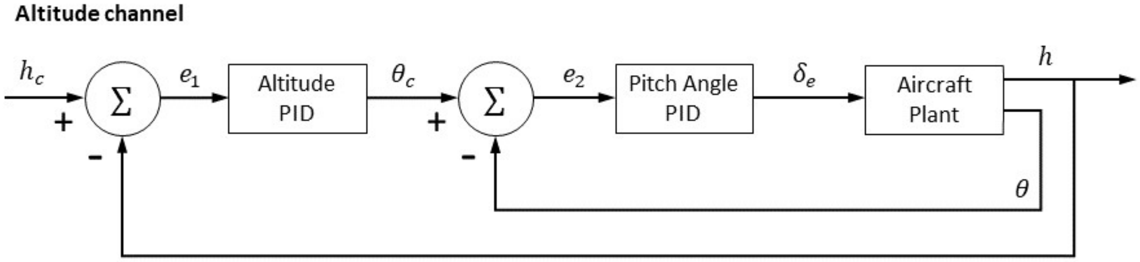

Figure 5.2: Altitude Control Loop Structure.

5.3 Aircraft Simulation

For the purpose of this work, a 4$^{\text{th}}$ order Runge–Kutta scheme for numerical integration was used to propagate the model forward in time with a timestep of 0.01 seconds. A timestep as small as 1 ms was tried, but resulted in a significant increase in computation time with no impactful change in accuracy.

5.3.1 Aircraft Simulation Example with Acrobatic Maneuver

This example shows a fairly short flight segment of simulated data, the purpose of which is to establish the aircraft simulation basic capabilities. The maneuver sequence for this example is shown in Table 5.2. Beginning from forward flight at approximately 500 ft/s, the aircraft performs a right turn at 30°AOB, then transitions directly into a more aggressive right turn at 60°AOB, before recovering to forward flight at 600 ft/s. The aircraft then enters a vertical loop maneuver before exiting the simulation. Several relevant state time histories for this flight sequence, namely forward speed, altitude, and roll angle are shown in Figure 5.3. The maneuvers executed by a high performance and agile aircraft (such as the F-16) can be roughly broken down into two categories. Namely, *standard* maneuvers and *acrobatic* maneuvers. The regime recognition and usage analysis techniques were designed to work on *standard* maneuvers. Therefore, it will be useful to identify the acrobatic maneuver segments in a pre-processing step, before regime recognition takes place. In this dissertation, acrobatic flight segments were defined to be any set of timesteps where the angle of bank exceeded ±60°, or the pilot load factor L_f exceeded a value of ±2 g's. These values were selected for illustrative purposes, and could be changed depending on the needs of the user. The load factor was computed using the following equation [108],

$$L_f = -\frac{a_z + \dot{q}x_s}{g} \tag{5.21}$$

where a_z is the acceleration in the z-direction at the aircraft center of mass, $\dot{q}$ is the derivative of the Euler pitch rate, x_s is the longitudinal distance from the aircraft center of mass to the pilot seat, and g is the earth gravitational constant.

Table 5.2: Maneuver Sequence for F-16 Simulation Example

Maneuver Start Time, s	Flown Regime	Parameter
0	Forward Flight	500 ft/s
5	Right Turn	35°AOB
45	Right Turn	60°AOB
95	Forward Flight	600 ft/s
125	Vertical Loop	–
155	End	–

This pre-processing step was performed on the flight sequence described in Table 5.2. Figure 5.4 shows the plot of the aircraft flight path in 3D space. The solid blue line indicates flight segments where the pre-processor identified *standard* maneuvers, and the dashed black line indicates flight segments identified as *acrobatic* maneuvers.

Figure 5.5 shows the state time history of the pilot load factor calculated using Equation 5.21. During the right turn maneuvers it can be seen how the calculated load factor is greater for the more aggressive turn ($L_f \approx 1.2$ for 35°AOB turn, and $L_f \approx 2.0$ for 60°AOB turn). It can then be seen that during the level flight portion of the flight sequence (95 – 125 seconds) the load factor was approximately 1g, this is expected as the pilot is not experiencing any acceleration other then that due to gravity. During the vertical loop maneuver (125 – 155 seconds), the pilot load factor exceeds 2 g's for the majority of the maneuver, then is rapidly reduced to near zero as the aircraft exits the loop.

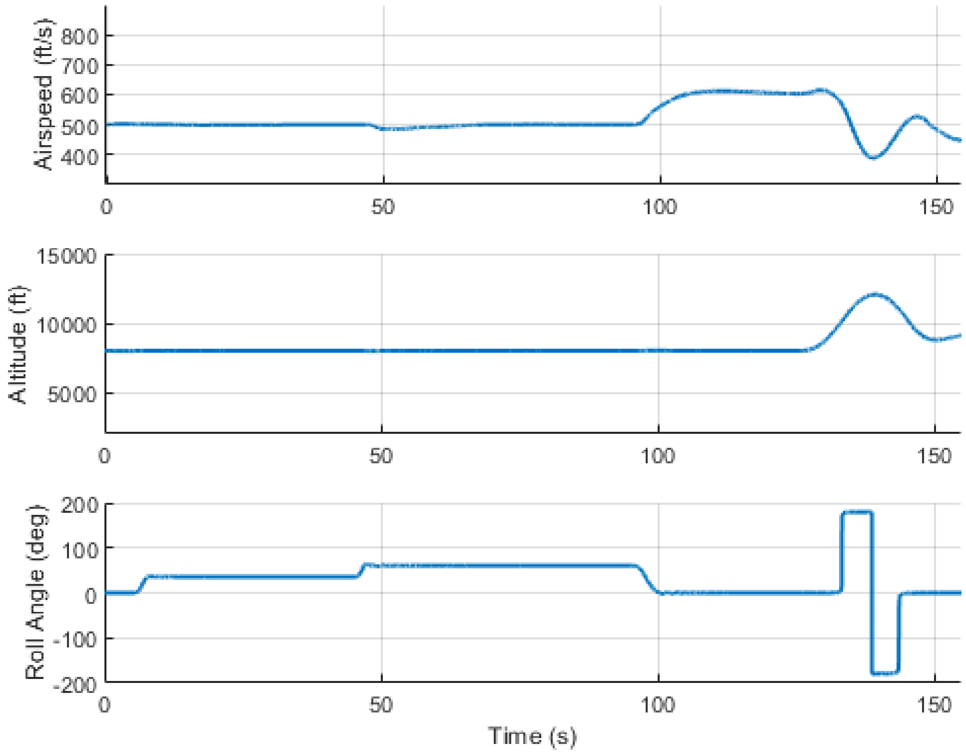

Figure 5.3: Selected State Time History for F-16 Simulation Example.

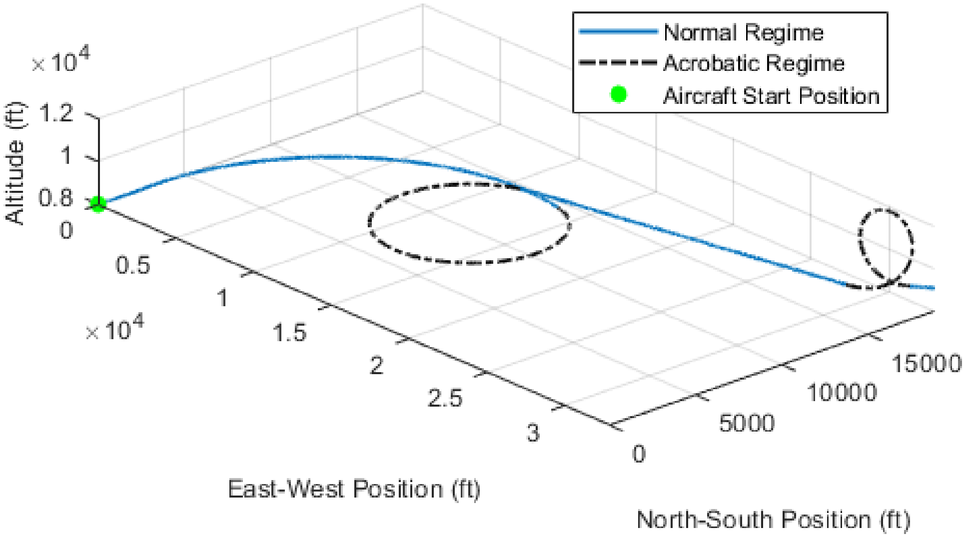

Figure 5.4: Aircraft Flight Path with Identified Acrobatic Segments for F-16 Simulation Example 1.

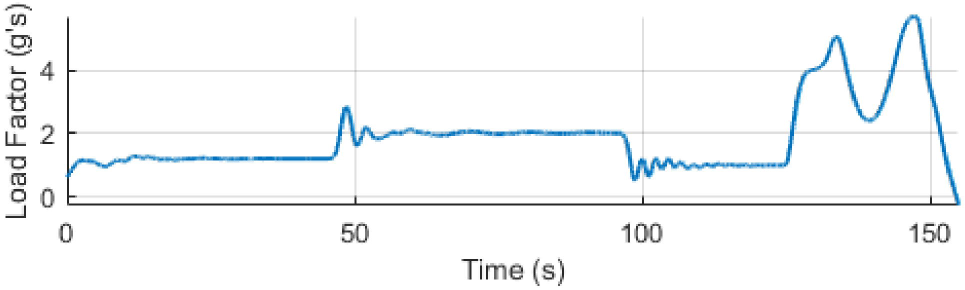

Figure 5.5: Pilot Load Factor Time History for F-16 Simulation Example.

5.4 Fixed Wing Regime Recognition

Before usage spectrum analysis can take place, regime recognition must be performed. The IMM-RR discussed in detail in Chapter 2 can be designed to work on fixed wing aircraft data (such as the F-16) with little modification. Mainly, the modifications involve removing any maneuvers from the RR library (Table 2.3) that correspond to hover-based maneuvers (i.e. Hover, Left/Right Hover Turn, Axial Climb/Descent). Setpoints for regime boundary definitions can also be modified if deemed necessary by the user, although for consistency in this dissertation, any maneuvers with a common name are given the same boundary definitions. Because aircraft handling characteristics vary substantially between the F-16 and SH-60, additional tuning was required for some filters. It should be noted that this additional tuning was quite minimal and was accomplished using only a handful of test sequences generated using the simulation methodology outlined above.

5.4.1 RR Example Sequence

This example uses a fairly short segment of simulated data. The maneuver sequence flown for this example is shown in Table 5.3. The aircraft begins in forward flight at approximately 600 ft/s for the first twenty seconds, the aircraft then enters a climbing right turn at 30°AOB and a 20 ft/s rate of climb. The aircraft then levels off to constant altitude while remaining in a 30°AOB turn and increasing forward speed to 800 ft/s, before entering a left descending turn at -15°AOB and 20 ft/s rate of descent. Finally the aircraft levels out to steady forward flight and reduces forward speed to 700 ft/s before ending the simulation.

Table 5.3: F-16 Maneuver Sequence for RR Example Sequence

Maneuver Start Time, s	Flown Regime	Forward Speed, ft/s	Rate of Climb, ft/s,	Angle of Bank, deg
0	Forward Flight	600	0	0
20	Climbing Right Turn	600	20	30
50	Right Turn	800	0	30
75	Descending Left Turn	800	-20	-15
110	Forward Flight	700	0	0
150	End	–	–	–

Several relevant state time histories for this sequence, namely forward speed, altitude, altitude rate, and roll angle, are shown in Figure 5.6. Note that in this figure, the solid blue line represents the simulated flight data, and the dashed black line represents the IMM Filter (or Kalman Filter) estimate for each respective state. Regime recognition results are shown in Figure 5.7 for the IMM-RR adapted to fixed wing usage. In Figure 5.7 the gray scale denotes the probability of the recognized regimes. The maneuver sequence identified by the IMM-RR closely matches the flight card in Table 5.3. In Figure 5.7, between seconds 90 and 110, the IMM-RR identified a mix of the "Descent" and "Left Descending Turn" maneuvers. The flight card for that portion of the flight specifies a rate of descent of 20 ft/s and an angle of bank of -15°. These parameters put the aircraft well into the definition of a descent, but on the regime boundaries for "level wing", and "left turn." Therefore, the correct regime designation would be somewhere between a level wing descent, and a descending left turn, which is exactly the classification given by the IMM-RR.

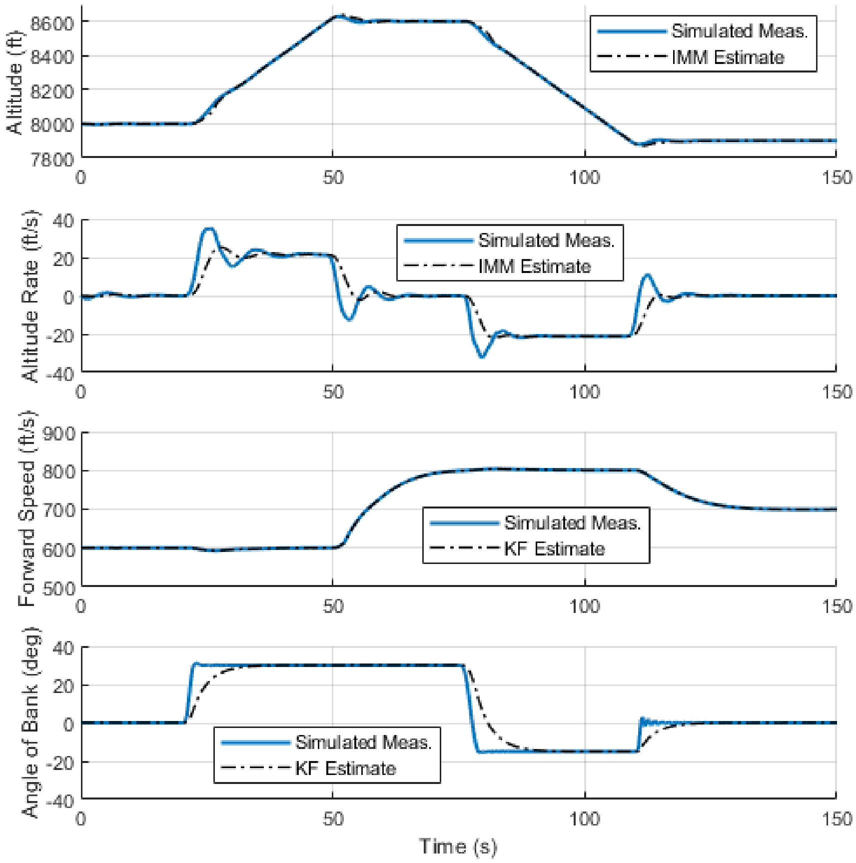

Figure 5.6: Selected State Time Histories for F-16 RR Example, Simulated Flight Data (Meas. = measurement).

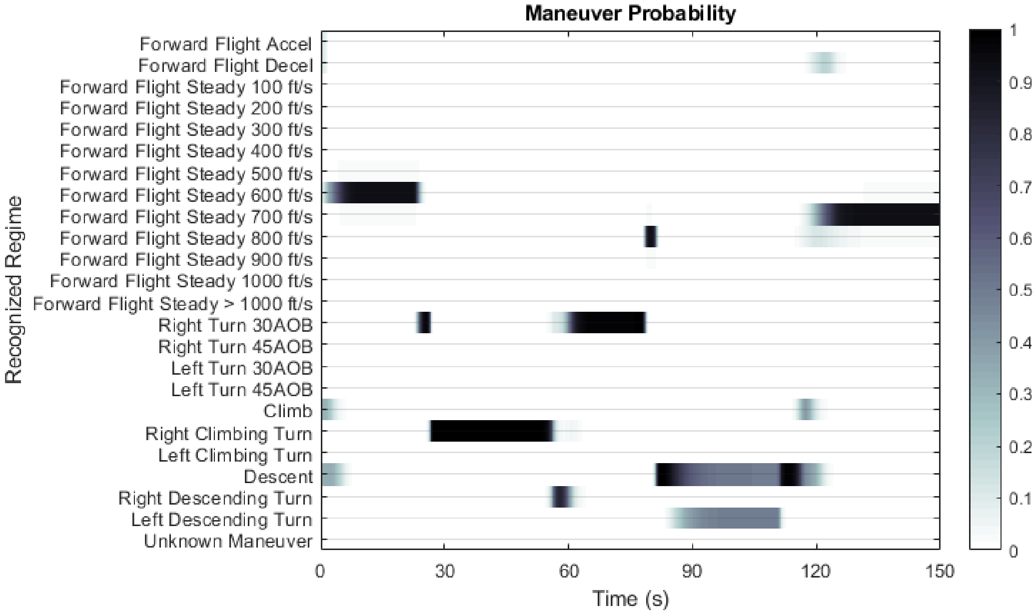

Figure 5.7: IMM-Based Regime Recognition Results for F-16 RR Example.

5.5 Fixed Wing Usage Spectrum

Before the usage analysis of a simulated F-16 can be analyzed, a large batch of flight data must be created. 1,000 hours of flight data were generated using the parameter distributions found in Table 5.4, and the same techniques described in detail in Section 3.3.3 for the SH-60 simulated database. In the same way as before, each flight in the simulated database was generated by first randomizing the number and duration of maneuvers using the uniform distributions found in Table 5.4. Then the maneuver type is randomly selected as Forward Flight, Climb/Descent, and Climbing/Descending turn. Similar to the creation of the SH-60 database, the first maneuver is always selected as Forward Flight. Finally, the the parameters for each maneuver are sampled using ITS [106] from the parameter distributions created using Table 5.4.

Table 5.4: Flight Parameter Distributions for F-16 Simulated Database Creation

Parameter	Lower Limit	Upper Limit	Distribution(s)
Number of Maneuvers	15	25	Uniform
Maneuver Duration, s	120	240	Uniform
Angle of Bank, deg	-50	50	$\mathcal{N}(30,4), \mathcal{N}(-30,4),$ $\mathcal{N}(45,4), \mathcal{N}(-45,4)$
Rate of Climb, ft/s	-30	30	$\mathcal{N}(10,4), \mathcal{N}(-10,4))$
Forward Speed, V_h	0.5	0.8	Uniform

All the same techniques for usage analysis discussed in Section 4.2 can be applied to the probabilistic regime recognition results for fixed wing aircraft. However, there is an additional methodology that can be applied to probabilistic RR that may be more useful in cases were the usage spectrum is the end-goal of the flight data analysis. This technique involves the use of a bootstrapped statistical method.

First, consider a large set of HUMS data from of a fleet of N_m aircraft, and suppose that a usage spectrum in terms of per 100 hours is desired. Then with reference to Figure 5.8, a random sampling of 100 flight hours can be selected from the batch of available feet data and processed with a probabilistic RR code. Then the expected time for each regime

r can be computed using Equation 4.1 for each set of 100 hrs, and stored in $E_{r,i}$, where i is the i^{th} random sampling of data. This process can then be repeated a large number of times (e.g. 10,000) to extract various statistics about the regime usage (such as mean fraction of overall flight time and standard deviation). This process was performed on the batch of 1,000 hours of simulated F-16 flight data with 10,000 re-samples, the results of which can be seen in Table 5.5. The two key statistics extracted from the data set are *computed usage* and *computed standard deviation*. In Table 5.5 the computed usage refers to the *expected time* that the aircraft spent in each regime (expressed as a percent), and the computed standard deviation is the measure of how dispersed the computed usage is from the mean (with units of percentage points).

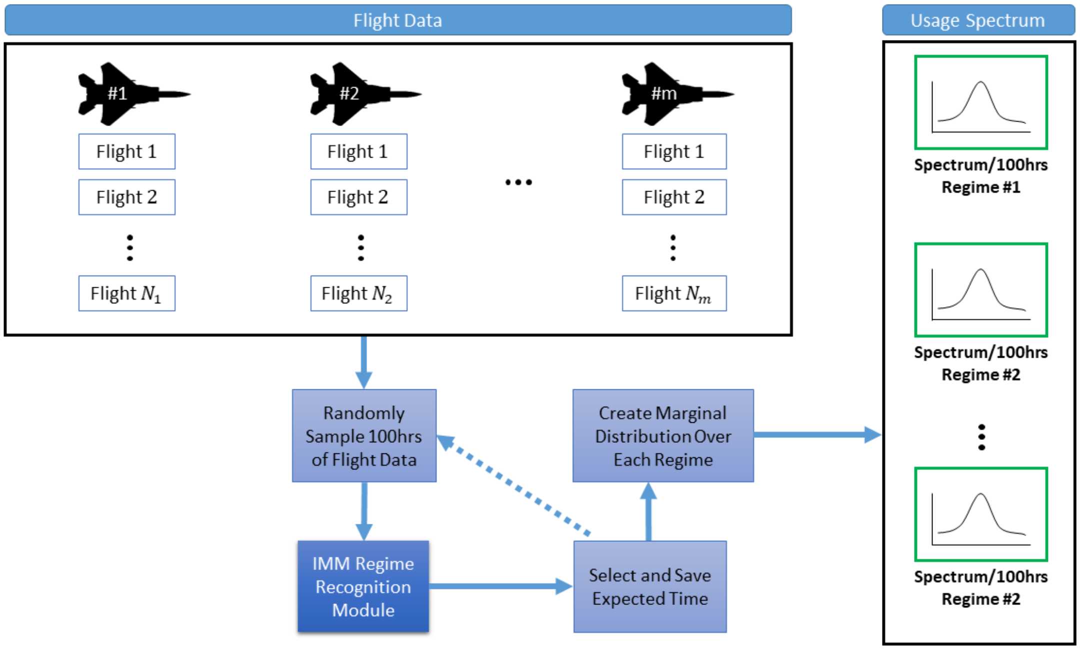

Figure 5.8: Visualization of Process to Create Regime Usage Spectrum.

In order to check the accuracy of the computed results, a comparison to "truth data" needs to be made. This "truth data" was found by taking the *flight cards* generated for each random flight in the 1,000 hours of simulated data, and tracking how much time the aircraft was commanded to fly in each regime across all flights. In this way, the actual expected usage could be computed for each regime across the entire data set by dividing the total commanded time for each regime by the total flight time simulated. Formally this can be expressed as,

$$U_t(r) = \frac{T_{c,r}}{T_{total}} \tag{5.22}$$

Were $T_{c,r}$ is the total command time flown for regime r, and T_{total} is the total simulated flight time across the entire data set.

This provides a benchmark that the proposed methodology can be compared to, but it should be noted that the "actual usage" and the "computed usage" will not match exactly. This is because the "actual usage" is based on the discrete commanded regime, and cannot account for the variation in regime setpoints stemming from the sampling distributions. For example, if the aircraft was commanded to fly a "Right Climbing Turn," the rate of climb and angle of bank setpoints are specified from the sampling distributions found in Table 5.4. If the setpoints are not sampled at exactly the middle of the regime boundary definitions, then the IMM-RR will return primarily a classification of "Right Climbing Turn," but may also return partial classifications of "Climb," "Level Flight," "Left Turn 30°," etc. These partial classifications will result in a discrepancy between the "computed usage" and the "actual usage" of the aircraft.

Another source of discrepancy is in the transient response of the aircraft. For example, the flight card might command that the aircraft fly for some set amount of time in a particular regime (e.g. Right Turn 30° for 60 seconds), but if the aircraft was flying a left turn in the prior maneuver, it will take some time for the aircraft to change its flown regime to match the the new command. This transient time cannot be accounted for, leading to another source of discrepancy between the "expected usage" and the "actual usage" of the aircraft.

Finally, it should be noted that some maneuvers are not explicitly commanded, but are the result of a change in setpoint from one maneuver to the next. An example of this would be the *Accelerate* and *Decelerate* maneuvers, which are the results of the aircraft first being commanded one particular forward speed, then subsequently being commanded a different forward speed in the next maneuver.

Table 5.5 shows the actual usage as specified by the generated flight cards and the "expected usage" calculated using the bootstrap method in conjunction with the IMM-RR. It can be seen that even with the sources of discrepancies discussed above, the computed and actual aircraft usage have close agreement across all regimes, with a largest difference of less then 2 percentage points.

This shows that the bootstrapped method was able to effectively extract regime usage statistics from the available data set. This methodology, along with the other usage analysis techniques discussed in Section 4.2, allows for an aircraft operator to make full use of probabilistic regime recognition schemes to analyze aircraft usage, either individually or as a fleet.

Table 5.5: Regime Usage Spectrum for Simulated F-16 Database

Regime	Actual Usage (A), %	Expected Usage (C), %	Difference |A - C|, %	Computed Std., %	Computed C.V., %
Forward Flight 600 ft/s	14.59	15.80	1.21	0.566	3.58
Forward Flight 700 ft/s	9.93	10.31	0.38	0.582	5.65
Forward Flight 500 ft/s	9.89	9.49	0.43	0.534	5.63
Forward Flight 800 ft/s	9.65	9.10	0.55	0.532	5.85
Descent	7.82	6.14	1.68	0.372	6.06
Climb	7.50	5.66	1.84	0.370	6.54
Left Turn 30°AOB	6.86	8.05	1.19	0.499	6.20
Left Turn 45°AOB	6.84	7.21	0.37	0.534	7.41
Right Turn 45°AOB	6.67	6.68	0.01	0.492	7.37
Right Turn 30°AOB	6.59	8.21	1.62	0.531	6.47
Right Descending Turn	3.45	2.48	0.97	0.290	11.69
Right Climbing Turn	3.44	2.56	0.88	0.283	11.05
Left Climbing Turn	3.39	2.54	0.85	0.309	12.17
Left Descending Turn	3.38	2.63	0.75	0.273	10.38
Forward Flight 400 ft/s	0.00	0.20	0.20	0.012	6.00
Forward Flight 900 ft/s	0.00	0.20	0.20	0.012	6.00
Forward Flight Accel.	–	1.41	–	0.040	2.84
Forward Flight Decel.	–	0.74	–	0.027	3.65
Unknown Maneuver	–	0.57	–	0.052	9.12

Std. = Standard Deviation
C.V. = Coefficient of Variation

CHAPTER 6
CONCLUSION

6.1 Contributions

1. A novel algorithm has been developed for rotorcraft regime recognition that leverages interacting multiple model estimation. The proposed approach casts the regime recognition problem in a Bayesian framework, in which the vehicle is a hybrid dynamical system and flight regimes are the discrete dynamic modes. A key advantage of the proposed algorithm is that it yields probabilistic estimates of the vehicle existing in a regime, rather than discrete classifications provided by alternative approaches. Given that there are no clear universally-recognized boundaries between maneuver regimes in the state space, the proposed algorithm acknowledges uncertainty in the classification process by assigning a probability distribution across the regimes at each measurement time. Results illustrate the advantages of the proposed approach compared to threshold-based algorithms.

2. The proposed RR algorithm's ability to assign partial regime classifications allows for a more accurate assessment of the regime(s) that are actually flown at a particular time. Evaluation metrics such as the probabilistic confusion matrix, normalized accuracy, and AUC values show significant improvements when compared to a traditional rule-based approach. For some of the example flights shown, normalized accuracy was improved by 20 percentage points and the AUC score was increased from 0.79 to 0.98.

3. The proposed algorithm yields smooth physically-meaningful results during the transitions between maneuvers, eliminating the high-frequency switching that is present in rule-based RR. The IMM-RR methodology was also found to operate very fast.

The algorithms were found to process 100 hours of flight data in approximately 10 hours using a single core of an i7-4710MQ processor with 16GB of memory.

4. A framework for computing fatigue damage using probabilistic regime recognition has been developed, with specific methods developed for both individual aircraft tracking and creation of fleet-wide damage spectra. The underlying damage estimation algorithm converts probabilistic regime classifications into probability densities over the incurred damage for each life-limited component on the aircraft.

5. A Gaussian approximation was employed and justified based on the strong-mixing property, providing a means to rapidly compute the damage distributions for large sets of flight data. A generalized version of the Gaussian approximation was then introduced to capture not only the uncertainty present in the flown regimes, but also the uncertainty present in the damage rates. Results showed an increase in the variance of the damage estimate when all sources of uncertainty were accounted for. The Gaussian approximation of the incurred damage was found to process 100 hours of probabilistic regime recognition results in under 10 seconds using a single core of an i7-4710MQ processor with 16GB of memory.

6. Demonstrated primary advantage of the probabilistic framework over deterministic counterparts is that the probabilistic methodology produces damage distributions that depend on the underlying uncertainty in regime classifications, which can be significant when aircraft fly near the boundaries of defined regimes. Simulation results show that the damage estimates produced by the probabilistic framework respond more smoothly to small changes in the vehicle states, and that the methodology can be viewed as a generalization of deterministic methods for computing damage. Monte Carlo studies showed how when the aircraft is flown on the boundaries of the regime definitions, the smoothly varying nature of the IMM resulted in a unimodal distribution of the estimated damage, where the mean and nominal damage values

differed by only 0.1%. This is contrasted with the rule-based approach were the mean and nominal damage differed by more than 10%, and had a range of greater then 3 times that of the IMM methodology.

7. The presented algorithms were applied to a large batch of actual helicopter HUMS data, and the results were consistent with the simulation based findings. The methodologies were also adapted to work on HUMS data from a simulated F-16. Results demonstrated application of regime usage analysis in both rotary and fixed wing aircraft, where the bootstrapped methodology was shown to be able to accurately generate the aircraft usage spectrum, differing by less than 2 percentage points for the worst case regime, in the study performed.

6.1.1 Concluding Remarks

These algorithms have been found to be very computationally efficient, and due to the modular nature of the *joint probability module's* use of IMMs, the proposed regime recognition methodology is a prime candidate for computational parallelization. Because the interacting multiple model estimators leverage banks of dynamic models, minimal training data is needed to effectively tune the filters for proper function. In addition, data filtering is handled internally through use of the IMMs, reducing the number of pre-processing steps required for data analysis.

Overall, the probabilistic schemes for both regime recognition and damage estimation have been shown to be promising alternatives to standard deterministic methods for future condition-based maintenance programs.

6.2 Recommended Future Work

As with any research, there are many interesting questions raised along the way. While some of these questions must be answered for completeness, there are also those that are beyond the scope of the current work being done. This section presents two areas of potential research that could be explored now that the work presented in this dissertation is concluded.

6.2.1 Regime Recognition and Damage Estimation of Other Events

The regime recognition and damage estimation framework proposed in this dissertation focused on flown aircraft regimes. While flown maneuvers are often the cause of a large portion of the damage incurred by life-limited components, there are other events that may contribute to component wear. Some examples include control reversals, and ground-air-ground cycles. While these events are generally viewed as discrete, it may be useful for their inclusion into the regime recognition and damage estimation to be cast into a probabilistic framework. In this way, a single unified probabilistic approach could be used to model all sources of uncertainty in the damage estimate.

6.2.2 Real-Time Component Damage Reduction

Because the regime recognition and damage estimation scheme is based on the use of Interacting Multiple Model estimators and Extended Kalman Filters – both of which are causal filters – there is nothing preventing the use of these algorithms in real-time. The primary obstacles would include access to real-time state data of the aircraft, and enough computing power to perform the regime recognition without significant delay.

The proposed algorithms could then be used to provide a real-time damage estimate of the aircraft based on the current maneuver(s) being flown. It could then be possible to include the real-time damage estimate as a feedback element in a control loop to enforce a ”low damage mode.” In this mode, the aircraft pilot control inputs could be modified to reduce the damage incurred on various components in real-time, resulting in increased component life during missions where high aircraft agility and responsiveness is deemed less important.

6.2.3 Real-Time Regime Control

In an autonomous vehicle that is commanded to fly a regime (rather than commanded to track some set points in the vehicle state space) the IMM-RR output could be used in real-time to estimate the aircraft's current regime. This real-time RR output could then be used as either a reference in a controller feedback loop, or be used as a gain-scheduling input in the regime controllers themselves. Implementation of this method would be highly dependent on the controller design, but the IMM-RR could provide the necessary feedback to form a full closed-loop regime control scheme.

6.2.4 Multivariate Damage Rates

The damage estimation framework discussed in this dissertation assumes the damage rates to be fixed in nominal value for each regime, with the option to describe each damage rate as a Gaussian distribution. It is likely that in the future, more complete damage rate functions will be developed by the OEMs. One such possible multivariate damage rate function is one that is not only dependent on the flown regime, but also on certain concurrent aircraft states. In this case, it would be beneficial to modify the discussed framework to include multivariate damage rate functions. Because the IMM can provide a combined state estimate, it would be possible to extend the Damage Computation Module to use the combined aircraft state estimates as inputs to the regime multivariate damage rate functions.

It might also be possible to further analyze the IMM-RR output to identify the transient regime behavior as the aircraft switches from one regime to another. If the OEM is able to provide damage rates that change based on maneuver transitions, with further development the IMM approach to regime recognition and damage estimation could be made capable of identifying these transient portions of the flight segments and use the appropriate damage rates accordingly.

REFERENCES

[1] Frank G Polanco. “Estimation of Structural Component Loads in Helicopters: a Review of Current Methodologies”. In: *Defence Science and Technology Organization Melbourne (Australia)* (1999).

[2] John P Ryan et al. *Helicopter Fatigue Load and Life Determination Methods*. Tech. rep. Dayton Univ OH Research Inst., 1975.

[3] Suresh Moon and Clyde Simmerman. “The Art of Helicopter Usage Spectrum Development”. In: *Journal of the American Helicopter Society* 53.1 (2008), pp. 68–86.

[4] S. Moon, D. Menon, and G. Barndt. “Fatigue Life Reliability Based on Measured Usage, Flight Loads and Fatigue Strength Variations”. In: *52nd Annual Forum of the Vertical Flight Society, Washington D.C., June 4-6*. 1996.

[5] Mark Voskuijl, Michel JL van Tooren, and Daniel J Walker. “Condition-based flight control for helicopters: An extension to condition-based maintenance”. In: *Aerospace Science and Technology* 42 (2015), pp. 322–333.

[6] Jack Bell. “Condition based maintenance plus DoD guidebook”. In: *Department of Defense: Washington, DC, USA* (2008).

[7] *Department of Defense Handbook: Aircraft Structural Integrity Program, General Guidelines*. MIL-HDBK-1530A. Jan. 2002.

[8] *Aeronautical design standard 79-A handbook for conditioned based maintenance system on US army aircraft*. ADS-79D-HDBK. Mar. 2013.

[9] C. Cheung et al. “Real-Time Nonlinear Model Predictive Control of a Helicopter in Autorotation”. In: *69th Annual Forum of the Vertical Flight Society, Phoenix, Arizona, May 21-23*. 2013.

[10] C. Cheung et al. “Expanded fatigue damage and load time signal estimation for dynamic helicopter components using computational intelligence techniques”. In: *70th Annual Forum of the Vertical Flight Society, Montreal, Canada, May 20-22*. 2014.

[11] Mauricio Alexandre et al. “Perspectives on Health and Usage Monitoring Systems (HUMS) of helicopters”. In: *MATEC Web of Conferences*. Vol. 314. EDP Sciences. 2020.

[12] JD Roe and DG Astridge. “The health and usage monitoring of helicopter systems-the next generation”. In: *Proc. AHS 41st Forum (Fort Worth, TX, 15-17 May, 1985)*. 1986, pp. 175–87.

[13] KF Fraser. “An overview of health and usage monitoring systems (HUMS) for military helicopters”. In: (1994).

[14] Stanley E Woodard et al. “Development and flight testing of an adaptable vehicle health-monitoring architecture”. In: *Journal of aircraft* 41.3 (2004), pp. 531–539.

[15] Li-Farn Yang and Irina Ioachim. “Adaptive estimation of aircraft flight parameters for engine health monitoring system”. In: *Journal of aircraft* 39.3 (2002), pp. 404–408.

[16] Linda K Kliment et al. “Usage and flight loads analysis of King Airs in aerial firefighting missions”. In: *Journal of Aircraft* 52.3 (2015), pp. 910–916.

[17] Kamran Rokhsaz et al. “Flight Loads Spectra of a Fleet of Heavy Air Tankers”. In: *Journal of Aircraft* 56.5 (2019), pp. 1858–1868.

[18] SM Spottswood and HF Wolfe. “Comparing fatigue life estimates using experimental and spectral density based probability distributions”. In: *Journal of aircraft* 39.3 (2002), pp. 493–498.

[19] RMV Pidaparti and MJ Palakal. “Neural network approach to fatigue-crack-growth predictions under aircraft spectrum loadings”. In: *Journal of Aircraft* 32.4 (1995), pp. 825–831.

[20] Edward Glaessgen and David Stargel. “The digital twin paradigm for future NASA and US Air Force vehicles”. In: *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*. 2012, p. 1818.

[21] Chenzhao Li et al. “Dynamic Bayesian network for aircraft wing health monitoring digital twin”. In: *Aiaa Journal* 55.3 (2017), pp. 930–941.

[22] D Guivarch et al. “Creation of helicopter dynamic systems digital twin using multibody simulations”. In: *CIRP Annals* 68.1 (2019), pp. 133–136.

[23] Andrew P Vechart et al. “Leveraging digital clones for prognostics health management”. In: *AIAC18: 18th Australian International Aerospace Congress (2019): HUMS-11th Defence Science and Technology (DST) International Conference on Health and Usage Monitoring (HUMS 2019): ISSFD-27th International Symposium on Space Flight Dynamics (ISSFD)*. Engineers Australia, Royal Aeronautical Society. 2019, p. 910.

[24] Catherine Cheung, Julio J Valdés, and Matthew Li. “Exploration of flight state and control system parameters for prediction of helicopter loads via gamma test and machine learning techniques”. In: *Real World Data Mining Applications*. Springer, 2015, pp. 359–385.

[25] Michael T Manry, Cheng-Hsiung Hsieh, and Hema Chandrasekaran. “Near-optimal flight load synthesis using neural nets”. In: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No. 98TH8468)*. IEEE. 1999, pp. 535–544.

[26] LA Flitter, K McCool, and DJ Haas. “Rotor Load System Monitoring Using a Neural Network Based Approach”. In: *AIAA Modeling and Simulation Technologies Conference, New Orleans, LA*.

[27] AB Cook et al. “Artificial neural networks for predicting nonlinear dynamic helicopter loads”. In: *AIAA journal* 32.5 (1994), pp. 1072–1077.

[28] Hesham Azzam. “A practical approach for the indirect prediction of structural fatigue from measured flight parameters”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 211.1 (1997), pp. 29–38.

[29] Margerye Hoffman. “Improving predictions for helicopter usage monitoring”. In: *Naval Air Warfare Center, Proceedings of the 1994 USAF Structural Integrity Program Conference p 743-765(SEE N 96-34936 12-05)*. 1996.

[30] RH Cabell, CR Fuller, and WF O’Brien. “Neural network modelling of oscillatory loads and fatigue damage estimation of helicopter components”. In: *Journal of Sound and Vibration* 209.2 (1998), pp. 329–342.

[31] David Haas. “Determination of helicopter flight loads from fixed system measurements”. In: *32nd Structures, Structural Dynamics, and Materials Conference*. 1991, p. 1012.

[32] David J Haas and Robin Imber. “Identification of helicopter component loads using multiple regression”. In: *Journal of Aircraft* 31.4 (1994), pp. 929–935.

[33] S Tang and John Moffatt. “Flight load models for the on-board fatigue usage processing of mechanical components on helicopters”. In: *Dynamics Specialists Conference*. 1994, p. 1718.

[34] H Oery and HW Lindert. “Reconstruction of spanwise air load distribution on rotor blades from”. In: (1992).

[35] Robert A Jacobs et al. “Adaptive mixtures of local experts”. In: *Neural computation* 3.1 (1991), pp. 79–87.

[36] Kelly M McCool, Lance A Flitter, and David J Haas. “Development and flight test evaluation of a rotor system load monitoring technology”. In: *Journal of the American Helicopter Society* 45.1 (2000), pp. 19–27.

[37] P Shanthakumaran et al. “Usage Based Fatigue Damage Calculation for AH-64 Apache Dynamic Components”. In: *The American Helicopter Society 66th Annual Forum, Phoenix*. 2010.

[38] Tony M Page, Westar A Senior Aerospace Engineer, and Ms Deta Adams. “CBM Fleet Implementation and Maintenance Decision Making Through Fatigue Life Management: Utilization and Parts Remediation Processes”. In: (2009).

[39] Yi Lu and R Christ. “AH-64 D Apache Longbow structural usage monitoring system”. In: *AHS International, 58 th Annual Forum Proceedings-*. Vol. 1. 2002, pp. 152–162.

[40] Raghupati Boorla and Kevin Rotenberger. “Load Variability of a Two-Bladed Helicopter”. In: *Journal of the American Helicopter Society* 42.1 (1997), pp. 15–26.

[41] Suresh Moon and Nam Phan. “Component fatigue life reliability with usage monitor”. In: *Annual Forum Proceedings-American Helicopter Society*. Vol. 63. 2. American Helicopter Society, Inc. 2007, p. 1077.

[42] Lt Cdr Jon Milsom, N. Toulas, and P. Irving. “A study of usage variability and failure probability in a military helicopter”. In: *Third International Conference on Health and Usage Monitoring-HUMS2003*. Citeseer. 2002, p. 81.

[43] Dinesh Menon, Gene Barndt, et al. “Fatigue Life Reliability Based on Measured Usage, Flight Loads, and Fatigue Strength Variations”. In: *Journal of the American Helicopter Society* 56.3 (2011), pp. 1–18.

[44] JG Van Der Horst and L Melissant. “Lynx life extension activities in the Royal Netherlands navy”. In: *25th European Rotorcraft Forum, Rome, Italy, Sep 14-16*. 1999.

[45] Scott Maley, John Plets, and Nam D Phan. “US Navy Roadmap to Structural Health and Usage Monitoring-The Present and Future”. In: *Annual Forum Proceedings-American Helicopter Society*. Vol. 63. 2. American Helicopter Society, Inc. 2007, p. 1456.

[46] Catherine Cheung and Nicholas C Bellinger. "Roadmap for Helicopter Life Usage Monitoring". In: *Thirteenth Australian International Aerospace Congress*. 2009, pp. 1–13.

[47] Jonathan Warner and Jonathan Rogers. "Novel Diagnostic Metrics for Regime Recognition Verification and Validation". In: *75th Annual Forum of the Vertical Flight Society, Philadelphia, PA, May 13-16*. 2019.

[48] GL Barndt and Suresh Moon. "Development of a fatigue tracking program for Navy rotary wing aircraft". In: *Annual Forum Proceedings-American Helicopter Society*. Vol. 50. American Helicopter Society. 1994, pp. 1359–1359.

[49] S Moon and N Phan. "Rotary Wing Aircraft Regime Recognition Algorithm Development & Validation". In: *American Helicopter Society 64th Annual Forum, Montreal, Canada*. 2008.

[50] David Haas et al. "Using flight data to improve operational readiness in naval aviation". In: *Annual Forum Proceedings-American Helicopter Society*. Vol. 64. 2. Citeseer. 2008, p. 1559.

[51] R Teal et al. "Regime recognition for MH-47E structural usage monitoring". In: *Annual Forum Proceedings-American Helicopter Society*. Vol. 53. American Helicopter Society. 1997, pp. 1267–1284.

[52] Gene Barndt, Subhasis Sarkar, and Charles Miller. "Maneuver regime recognition development and verification for H-60 structural monitoring". In: *Annual Forum Proceedings-American Helicopter Society*. Vol. 63. 1. American Helicopter Society, Inc. 2007, p. 317.

[53] Umberto Saetti and Jonathan Rogers. "Motion Primitive Approach to Rotorcraft Regime Recognition". In: *Journal of the American Helicopter Society* (2021).

[54] S Wu, E Bechhoefer, and D He. "A Practical Regime Recognition Approach for HUMS Applications". In: *63rd Annual National Forum of the American Helicopter Society, Virginia Beach, VA*. 2007.

[55] David He, Shenliang Wu, and Eric Bechhoefer. "A regime recognition algorithm for helicopter usage monitoring". In: *Aerospace Technologies Advancements* (2010), pp. 391–404.

[56] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[57] David He, Shenliang Wu, and Eric Bechhoefer. “Development of Regime Recognition Tools for Usage Monitoring”. In: *2007 IEEE Aerospace Conference*. IEEE. 2007, pp. 1–11.

[58] D. Musso and J. Rogers. “Interacting Multiple Model Algorithms for Rotorcraft Regime Recognition”. In: *75th Annual Forum of the Vertical Flight Society, Philadelphia, PA, May 13–16*. 2019.

[59] D. Musso and J. Rogers. “Interacting Multiple Model Estimation for Helicopter Regime Recognition”. In: *Journal of Aircraft* 57.6 (2020), pp. 1134–1147.

[60] Shau-Shiun Jan and Yu-Chun Kao. “Radar tracking with an interacting multiple model and probabilistic data association filter for civil aviation applications”. In: *Sensors* 13.5 (2013), pp. 6636–6650.

[61] William J Farrell. “Interacting multiple model filter for tactical ballistic missile tracking”. In: *IEEE Transactions on Aerospace and Electronic Systems* 44.2 (2008), pp. 418–426.

[62] Efim Mazor et al. “Interacting multiple model methods in target tracking: a survey”. In: *IEEE Transactions on aerospace and electronic systems* 34.1 (1998), pp. 103–123.

[63] Xiao-Rong Li and Yaakov Bar-Shalom. “Multiple-model estimation with variable structure”. In: *IEEE Transactions on Automatic control* 41.4 (1996), pp. 478–493.

[64] TL Wood. “High energy rotor system”. In: *American Helicopter Society 32nd Annual National V/STOL Forum, Washington, DC*. 1976.

[65] B. Eberle and J. Rogers. “Real-Time Nonlinear Model Predictive Control of a Helicopter in Autorotation”. In: *76th Annual Forum of the Vertical Flight Society, Virginia Beach, VA, Oct 6-8*. 2020.

[66] T Sakthivel and C Venkatesan. “Flight Dynamics of Helicopter Under Steady Maneuver: Interesting Observations”. In: *Journal of Aircraft* 54.4 (2017), pp. 1595–1604.

[67] Eliot W Quon et al. “Unsteady Reynolds-averaged Navier-Stokes-based hybrid methodologies for rotor-fuselage interaction”. In: *Journal of aircraft* 49.3 (2012), pp. 961–965.

[68] Sung Nam Jung et al. “Loose fluid-structure coupled approach for a rotor in descent incorporating fuselage effects”. In: *Journal of Aircraft* 50.4 (2013), pp. 1016–1026.

[69] Mark E Dreier. *Introduction to helicopter and tiltrotor flight simulation*. American Institute of Aeronautics and Astronautics, 2007.

[70] Ethem Alpaydin. “Introduction to machine learning. 2004”. In: *Cover, Copyright Page, Table of Contents for* (), pp. 1–327.

[71] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.

[72] Xiao-Ning Wang et al. “Probabilistic confusion entropy for evaluating classifiers”. In: *Entropy* 15.11 (2013), pp. 4969–4992.

[73] Peter D Talbot et al. “A mathematical model of a single main rotor helicopter for piloted simulation”. In: (1982).

[74] Robert TN Chen. *A simplified rotor system mathematical model for piloted flight dynamics simulation*. Ames Research Center, 1979.

[75] Robert TN Chen. *Effects of primary rotor parameters on flapping dynamics*. Vol. 1431. Nasa, 1980.

[76] Robert E Sheldahl and Paul C Klimas. *Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines*. Tech. rep. Sandia National Labs., Albuquerque, NM (USA), 1981.

[77] Z. Sunberg, N. Miller, and J. Rogers. “A Real-time expert control system for helicopter autorotation”. In: *Journal of the American Helicopter Society* 60.2 (2015), pp. 1–15.

[78] Lee Fowler. “A Virtual Pilot Algorithm for Synthetic HUMS Data Generation”. MA thesis. Atlanta, GA: Georgia Institute of Technology, Dec. 2015.

[79] Zachary Sunberg. “A Real Time Expert Control System For Helicopter Autorotation”. MA thesis. College Station, TX: Texas A&M University, 2013.

[80] David A Peters, David Doug Boyd, and Cheng Jian He. “Finite-state induced-flow model for rotors in hover and forward flight”. In: *Journal of the American Helicopter Society* 34.4 (1989), pp. 5–17.

[81] Roberto Celi. “State-space representation of vortex wakes using the method of lines”. In: *Journal of the American Helicopter Society* 50.2 (2005), pp. 195–205.

[82] W Johnson. “Time-domain unsteady aerodynamics of rotors with complex wake configurations.” In: *VERTICA*. 12.1 (1988), pp. 83–100.

[83] Dale M Pitt and David A Peters. “Theoretical prediction of dynamic-inflow derivatives”. In: (1980).

[84] David A Peters and Ninh HaQuang. “Dynamic inflow for practical applications”. In: (1988).

[85] G. Gaonkar and D. Peters. “Effectiveness of current dynamic-inflow models in hover and forward flight”. In: *Journal of the American Helicopter Society* 31.2 (1986), pp. 47–57.

[86] Harry H Heyson. *Ground effect for lifting rotors in forward flight*. Vol. 234. National Aeronautics and Space Administration, 1960.

[87] Lee Fowler et al. “A Virtual Pilot Algorithm for Synthetic HUMS Data Generation”. In: *Journal of the American Helicopter Society* 62.4 (2017), pp. 1–14.

[88] D. Musso and J. Rogers. “Probabilistic Damage Estimation for Rotorcraft Condition-Based Maintenance”. In: *77th Annual Forum of the Vertical Flight Society, Virtual, May 10–14*. 2021.

[89] D. Musso, U. Saetti, and J. Rogers. “Probabilistic Fatigue Damage Estimation for Rotorcraft Life-Limited Components”. In: *Journal of Aircraft* (2021), pp. 1–13.

[90] Milton A Miner. “Cumulative damage in fatigue”. In: (1945).

[91] Yu Chee Tong, Ross A Antoniou, and Chun H Wang. “Probabilistic fatigue life assessment for helicopter dynamic components”. In: (2004).

[92] Michael J Evans and Jeffrey S Rosenthal. *Probability and statistics: The science of uncertainty*. Macmillan, 2004.

[93] Caleb Ju and Edgar Solomonik. “Derivation and analysis of fast bilinear algorithms for convolution”. In: *SIAM Review* 62.4 (2020), pp. 743–777.

[94] Richard C Bradley. “Basic properties of strong mixing conditions. A survey and some open questions”. In: *Probability surveys* 2 (2005), pp. 107–144.

[95] Norbert Herrndorf. “A functional central limit theorem for strongly mixing sequences of random variables”. In: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 69.4 (1985), pp. 541–550.

[96] Robert J Serfling. “Contributions to central limit theory for dependent variables”. In: *The Annals of Mathematical Statistics* 39.4 (1968), pp. 1158–1175.

[97] Walter Philipp. “The central limit problem for mixing sequences of random variables”. In: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 12.2 (1969), pp. 155–171.

[98] Shin S Ikeda. “A note on mixingale limit theorems and stable convergence in law”. In: *Communications in Statistics-Theory and Methods* 46.19 (2017), pp. 9377–9387.

[99] Aryeh Dvoretzky. “Asymptotic normality for sums of dependent random variables”. In: *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*. University of California Press. 1972, pp. 513–535.

[100] Carlg Schaefer. “The effects of aerial combat on helicopter structural integrity”. In: *AHS, Annual Forum, 45 th, Boston, MA*. 1989.

[101] Lidija Trailovic and Lucy Y Pao. “Variance estimation and ranking of Gaussian mixture distributions in target tracking applications”. In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* Vol. 2. IEEE. 2002, pp. 2195–2201.

[102] Martin Egozcue and Luis Fuentes Garcia. “The variance upper bound for a mixed random variable”. In: *Communications in Statistics-Theory and Methods* 47.22 (2018), pp. 5391–9395.

[103] Tongseok Lim and Robert J McCann. “Geometrical Bounds for Variance and Recentered Moments”. In: *Mathematics of Operations Research* (2021).

[104] Suresh Moon and Nam Phan. “Rotorcraft Gross Weight and Center of Gravity Prediction Using Regime Recognition”. In: *Journal of the American Helicopter Society* 60.4 (2015), pp. 1–8.

[105] Jason Thomas et al. *Implementation of Structural Health Monitoring for the USMC CH-53E*. Tech. rep. Naval Air Systems Command, Patuxent River MD, Structures Div., 2010.

[106] Macaulay Okwuokenye and Karl E Peace. “A comparison of inverse transform and composition methods of data simulation from the Lindley distribution”. In: *Communications for Statistical Applications and Methods* 23.6 (2016), pp. 517–529.

[107] Frederico R Garza and Eugene A Morelli. “A Collection of Nonlinear Aircraft Simulations in MATLAB”. In: (2003). NASA/TM-2003-212145.

[108] Brian L Stevens and Frank L Lewis. *Aircraft Control and Simulation, 2nd Ed.* John Wiley & Sons, 2003.

[109] Joshua F Whitney and Heather M Whitney. “The right-hand rule”. In: *A Handbook of Mathematical Methods and Problem-Solving Tools for Introductory Physics*. 2053-2571. Morgan & Claypool Publishers, 2016, 7-1 to 7–3.

[110] Ogata Katsuhiko. *Modern Control Engineering, 4th Ed.* Prentice Hall, 2002.

VITA

Dakota Musso was born and raised in San Jose, California. After graduating high school in 2014, Dakota began attending George Fox University in Newberg, OR where he received a B.S. in Mechanical Engineering in the spring of 2018. During his time as an undergraduate, Dakota interned with Tesla, Inc. as part of the automation prototyping team working onsite at the solar cell manufacturing division in Fremont, CA. Following the completion of his bachelor's degree, Dakota continued his studies by moving across country in July 2018 to attend the Georgia Institute of Technology, where he pursued an M.S. and Ph.D., both in Mechanical Engineering. His technical interests include applied dynamics, system simulation and design, and nonlinear control and estimation.