# Automated Data Inspection in Jet Engines

Harjit Hullait


Submitted for the degree of Doctor of

Philosophy at Lancaster University.

September 2019

STOR-i
excellence with impact

# Abstract

Rolls Royce accumulate a large amount of sensor data throughout the testing and deployment of their engines. The availability of this rich source of data offers exciting opportunities to automate the monitoring and testing of the engines. In this thesis we have developed statistical models to make meaningful insights from engine test data.

We have built a classification model to identify different types of engine running in Pass-Off tests. The labels can be used for post-analysis and highlight problematic engine tests. The model has been applied to two different types of engines, in which it gives close to perfect classification accuracy. We have also created an unsupervised approach when there are no defined classes of engine running. These models have been incorporated into Rolls Royce systems.

Early warnings for potential issues can enable relatively cheap maintenance to be performed and reduce the risk of irreparable engine damage. We have therefore developed an outlier detection model to identify abnormal temperature behaviour. The capabilities of the model are shown theoretically and tested on experimental and real data.

Lastly, in a test decisions are made by engineers to ensure the engine complies

with certain standards. To support the engineers we have developed a predictive model to identify segments of the engine test that should be retested. The model is tested against the current decision making of the engineers, and gives good predictive performance. The model highlights the possibility of automating the decision making process within a test.

# Acknowledgements

My PhD has been an emotional journey where I have learnt a lot about myself and feel both proud and humbled by the great work being done by colleagues at STOR-i. I would like to thank David Leslie and Nicos Pavlidis for their support, guidance and patience which I have probably tested far more than I should. A thank you to Steve King for his support on the project. I would like to thank the team at STOR-i including Jon Tawn, Kevin Glazebrook and Idris Eckley for the opportunity and congratulate them on creating a great environment for research and collaboration. I would like to thank the admin staff: Kim Wilson, Jennifer Bull, Wendy Shimmin and Oliver Standford, who have enabled STOR-i to run so smoothly.

I would like to thank my friends who have been a constant source of encouragement and support. Sam Tickle for the limitless enthusiasm and endless patience. To the most unconventional Jake Clarkson, to the most wild David Torres and the most crazy Jack Baker, and to the rest of my STOR-i family, thank you!

My guru has been my guide throughout my life. There is not enough words to express my love and appreciation for all that you have done and continue to do. I like to thank my family without their love and support I would never have started

the PhD let alone finish it.

# Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

Harjit Hullait

# Contents

**Abstract** **I**

**Acknowledgements** **III**

**Declaration** **V**

**Contents** **X**

**List of Figures** **XVI**

**List of Tables** **1**

**1 Introduction and Objectives** **1**

1.1 Introduction . . . . . 1

1.1.1 Jet Engines . . . . . 2

1.1.2 Engine tests . . . . . 3

1.1.3 Contributions and Thesis Outline . . . . . 11

**2 Methodology developed for Jet engine data** **18**

2.1 Introduction . . . . . 18

2.2 Data Visualisation . . . . . 20

2.3 Novelty Detection Approaches . . . . . 23

2.3.1 K-means Model . . . . . 23

2.3.2 Support Vector Machines . . . . . 23

2.3.3 Probabilistic Model . . . . . 24

2.4 Our Approach . . . . . 25

**3 Functional Data Analysis** **27**

3.1 Introduction . . . . . 27

3.2 Functional Principal Component Analysis . . . . . 29

3.2.1 Principal component analysis . . . . . 30

3.2.2 Functional Principal Component Analysis . . . . . 34

3.2.3 Extensions to FPCA . . . . . 44

3.3 Functional Linear Regression . . . . . 49

3.3.1 Historical FLR . . . . . 51

3.3.2 Model Selection for FLR . . . . . 53

3.4 Functional Depth . . . . . 54

3.4.1 Fraiman Muniz depth . . . . . 58

3.4.2 Random Projection depth . . . . . 58

3.4.3 h-modal depth . . . . . 59

3.4.4 Band Depth . . . . . 59

3.4.5 Multivariate Functional Depth . . . . . 60

3.4.6 Other Depth functions . . . . . 61

3.5 Outlier Detection for Functional Data . . . . . 61
3.5.1 Direct approach . . . . . 62
3.5.2 Functional Boxplot . . . . . 63
3.5.3 Outliergram . . . . . 64
3.5.4 Functional Outlier Map . . . . . 65

**4 Classification of manoeuvres in a Pass-Off test 68**

4.1 Introduction . . . . . 68
4.2 Manoeuvre Extraction . . . . . 72
4.2.1 Changepoints . . . . . 72
4.2.2 Extracting Fixed Speed segments . . . . . 78
4.3 Needleman Wunsch . . . . . 78
4.3.1 Thresholding Needleman Wunsch . . . . . 83
4.3.2 Probabilistic Needleman Wunsch . . . . . 84
4.3.3 Example . . . . . 86
4.4 Functional PCA Templates . . . . . 87
4.5 Classifiers . . . . . 91
4.5.1 Decision Tree . . . . . 91
4.5.2 Linear Discriminant Analysis . . . . . 93
4.5.3 Comparing the DT and LDA classifiers . . . . . 95
4.6 Testing on Trent 1000 engines . . . . . 98
4.7 Testing on XWB engines . . . . . 99
4.8 Heatmap . . . . . 103

4.9 Conclusion . . . . . 103

**5 Manoeuvre Clustering in Cyclic tests 106**

5.1 Introduction . . . . . 106

5.2 Density Based Clustering . . . . . 108

5.3 Dynamic Time Warping . . . . . 109

5.4 Cluster Evaluation . . . . . 111

5.5 Clusters in Cyclic Test Data . . . . . 113

5.6 Clusters in Trent 1000 Pass-Off tests . . . . . 117

5.7 Clusters in XWB Pass-Off tests . . . . . 119

5.8 Discussion . . . . . 125

**6 Robust Functional Linear Regression 127**

6.1 Introduction . . . . . 127

6.2 Robust Functional Linear Regression . . . . . 129

6.2.1 Robust Bayesian Information Criterion for FLR . . . . . 132

6.3 Asymptotic Results . . . . . 133

6.3.1 Consistency of the Robust FLR . . . . . 135

6.3.2 Consistency of RBIC . . . . . 138

6.4 Simulation Study . . . . . 140

6.4.1 Scenarios . . . . . 141

6.5 Conclusion . . . . . 146

**7 Outlier Detection using Functional Regression 148**

7.1 Introduction . . . . . 148
7.2 Outlier Detection using RFLR . . . . . 150
7.3 Simulation Study . . . . . 152
7.4 Jet Engine data . . . . . 157
7.4.1 Vibration Surveys in Trent 1000 engines . . . . . 158
7.4.2 Vibration Surveys in XWB engines . . . . . 159
7.5 Conclusion . . . . . 164

**8 Prediction of Vibration Survey repeats 169**

8.1 Introduction . . . . . 169
8.2 Centroid classifier . . . . . 172
8.3 Depth Depth-Classifier . . . . . 174
8.4 Logistic Functional Linear Regression . . . . . 175
8.5 Results . . . . . 179

**9 Conclusion and Further work 183**

**Bibliography 186**

# List of Figures

1.1.1 Summary of LP, IP and HP. www.slideshare.net/egajunior/trent-1000-presentation . . . . . 2

1.1.2 Station Locations. https://speechfoodie.com/jet-engine-diagram-n1-n2/ 3

1.1.3 Perfect Pass-off test with samples taken every 40th of a second. . . . 5

1.1.4 N1 speed for Pass-Off test 1 (left) and 2 (right). . . . . 6

1.1.5 Plots of N1 (blue), N2 (orange) and N3 (red) speed time series for Dataset 1. . . . . 6

1.1.6 Plots of P30 (black), P42 (red) and P44 (cyan) which are plotted alongside the N1 speed (orange) time series for Dataset 1. . . . . 7

1.1.7 Plots of T30 (blue) and N1 speed (orange) time series for a perfect test. 8

1.1.8 Plots of the LPV and HPV in blue alongside the N1 speed in orange. 9

1.1.9 Plot of the N1 speed time series generated for the Cyclic test, with samples taken every second. . . . . 10

1.1.10 Plots of two sections of the Cyclic test. . . . . 10

3.1.1 Plots of Canadian Temperature dataset containing temperature reading over a year from 35 Canadian cities and Lip dataset of measurements of the lower lip of 20 people during the pronunciation of the word 'bob'. 28

3.4.1 Scatter plot of samples from a multivariate normal distribution, with point in red closer to the centre than the green point. . . . . 55

4.1.1 Labelled N1 speed plots for Pass-Off test 18 (left) and 21 (right). . . 72

4.2.1 Plot of a section of the Performance curve in Pass-Off test 1 with changepoints found using PELT with RSS cost function (red) and a BIC penalty. . . . . 77

4.3.1 Example of a matrix $Z$ aligning sequences GATTACA and GCATGCU, with scores $a = 1$, $b = -2$ and $c = -1$. . . . . 82

4.3.2 Plot of a a Performance curve with labelled fixed speed levels. . . . . 87

4.4.1 Plots of 29 Vibration surveys (V) and Fast acc/dec (F) manoeuvres. . 89

4.4.2 Plots of 29 mean corrected Vibration surveys (V) and Fast acc/dec (F) manoeuvres. . . . . 89

4.4.3 Plot of FPCA reconstruction of a Vibration Survey (left) and a Performance Curve (right), using FPCA representations of V (pink) and F (green). 90

4.5.1 Pruned Trees using NW (left) and Probabilistic NW (right) scores and applying 10-fold Cross Validation. . . . . 93

4.5.2 Pass-Off test 46, labelled using DT classifier. . . . . 96

4.7.1 Plot of first 20 F and V manoeuvres . . . . . 101

4.7.2 Plot of first 20 Fast acc/dec (F) and Vibration Survey (V) manoeuvres aligned at the deceleration point. . . . . 102

4.7.3 Labelled N1 speed plots for XWB Pass-Off test 25 (left) and 54 (right). 102

4.8.1 A Heat Map of the number of manoeuvres in the first 10 Pass-Off tests 105

5.5.1 Ordered log 10-nearest neighbour distances with red line at 1000. . . 114

5.5.2 Cyclic test plot with manoeuvres coloured in with respect to the four clusters and the the noise manoeuvres are coloured in red, using DBSCAN with $\epsilon = 1000$. . . . . 115

5.5.3 Plots of aligned manoeuvre in each of the 4 clusters found using $\epsilon = 1000$ in the DBSCAN algorithm. . . . . 116

5.5.4 tSNE mapping for each manoeuvre in the Cyclic test with the four clusters coloured, including noise points in red . . . . . 116

5.6.1 Ordered log $k$-nearest neighbour distances with line at $\epsilon = \log(4000)$ for Trent 1000 manoeuvres. . . . . 119

5.6.2 Plots of manoeuvre clusters found using $\epsilon = 4000$ in the DBSCAN algorithm for Trent 1000 manoeuvres. . . . . 121

5.6.3 tSNE mapping of the manoeuvre in the Trent 1000 Pass-Off tests using cluster labels (left) and using true labels (right). . . . . 122

5.7.1 Ordered log $k$-nearest neighbour distances with line at $\epsilon = 4000$ for XWB manoeuvres. . . . . 122

5.7.2 tSNE mapping of the manoeuvre in the XWB Pass-Off tests using cluster labels (left) and using true labels (right). . . . . 123

5.7.3 Plots of XWB manoeuvre in 5 clusters found using $\epsilon = 4000$ in the DBSCAN algorithm. . . . . 124

6.1.1 Plots of 30 TPR (left) and TGT (right) time series. . . . . 129

6.4.1 *Left:* Plots of the predictor curves $x_i(t)$, response curves $y_i^{(1)}(t)$ and residuals curves $r_i^{(1)}(t)$ for Scenario 1. *Right:* Plots of the predictor curves $x_i(t)$, response curves $y_i^{(2)}(t)$ and residuals curves $r_i^{(2)}(t)$ for Scenario 2. The residual curves are generated using the true regression function and mean functions. In each scenario there are 5 outliers each in a distinctive colour. . . . . 147

7.3.1 ROC curve for one instance of Scenario 1 and 2 with 20% of the samples contaminated. . . . . 153

7.3.2 Plots of the Functional Boxplots, the Outliergrams and the Functional Outlier Map (FOM) for the residuals using CFLR (left) and RFLR (right) for one instance of Simulation 1 with 20% of the data contaminated. In the Functional Boxplot the median function is in black, the 0.5-central region $C_{0.5}$ is in purple with the fences in blue, the outliers are coloured in red. In the Outliergrams the thresholds are the dotted lines and outliers lie outside the thresholds. In the FOM plots have a parabolic threshold given by dotted line. . . . . 156

7.4.1 Plots of the TPR, T25, T30, TGT, TCAR and TCAF time series from Vibration Surveys performed on Trent 1000 engines with outliers using robust FLR in red; those using the curves directly in green and those for both in purple. . . . . 160

7.4.2 Plots of the residuals of the T25, T30, TGT, TCAR and TCAF with outliers using classical FLR in blue. . . . . 161

7.4.3 Plots of the residuals of the T25, T30, TGT, TCAR and TCAF with outliers using robust FLR in red. . . . . 162

7.4.4 Plots of the TPR, T25, T30, TGT, TCAR and TCAF time series for Vibration Surveys performed on XWB engines with outliers using robust FLR in red; those using the curves directly in green and those for both in purple. . . . . 165

7.4.5 Plots of the residuals of the T25, T30, TGT, TCAR and TCAF for Vibration Surveys in XWB tests with outliers using classical FLR in blue. . . . . 166

7.4.6 Plots of the residuals of the T25, T30, TGT, TCAR and TCAF for Vibration Surveys in XWB tests with outliers using robust FLR in red. 167

8.1.1 Plot of 30 LPV, IPV and HPV curves during acceleration and deceleration of the Vibration Survey. . . . . 171

8.2.1 Density plot of score values from Centroid classifier applied to LPV deceleration curves. . . . . 173

8.3.1 Scatter plot of the depth value labelled by non-repeated (0) and repeated (1) manoeuvres (left). Density plot of depth values with respect to non-repeated manoeuvres (depth0) (right). The depth values are obtained from the LPV deceleration time series. . . . . 176

8.3.2 Scatter plot of the multivariate depth values labelled by non-repeated (0) and repeated (1) manoeuvres (left). Density plot of depth values with respect to non-repeated manoeuvres (depth0) (right). . . . . 176

8.4.1 Density plot of probability values obtained used FPCA basis and depth values (left) and using B-spline basis with depth and lasso (right). . . 178

8.5.1 The ROC curve for the five classifiers given in Table 8.5.1. . . . . 181

8.5.2 Plots of the regression functions for the hpv accel (left) and hpv decel (right) curves, using a B-spline basis with lasso penalty. . . . . 182

# List of Tables

4.5.1 The number of each manoeuvre in the training set. . . . . 92

4.5.2 Table of labels given for Test 46, shown in Figure 4.5.2, with colours matching the manoeuvre classes. We have the true labels, and the labels using the Decision Tree (DT) and Linear Discriminant Analysis (LDA). We also have the Mahalanobis distance with respect to the manoeuvre class given by LDA. . . . . 97

4.6.1 The number of each manoeuvre in the 93 Pass-Off tests. . . . . 98

4.7.1 The number of each manoeuvre in the 54 XWB Pass-Off tests. . . . . 101

5.6.1 The number of each manoeuvre in Trent 1000 dataset identified as noise by DBSCAN. . . . . 118

5.7.1 The number of each manoeuvre in XWB dataset identified as noise by DBSCAN. . . . . 123

6.4.1 Average fitting errors (FE) for 100 replications for Scenario 1, using classic FPCA and robust FPCA with different amount of trimming in the MLTS estimator and using models selected by BIC and RBIC. . . 144

6.4.2 Average fitting errors (FE) for 100 replications for Scenario 1, using classic FPCA and robust FPCA with different amount of trimming in the MLTS estimator and using models selected by BIC and RBIC. . . 145

7.3.1 Average AUC values over 100 replications for Scenario 1, using Direct compared to classic FPCA with BIC, and using robust FPCA with RBIC. We will use trimming levels $\alpha = 0.1, 0.2, 0.3$ and contaminate different proportions of the samples $a = 0.1, 0.2, 0.3$. . . . . 154

7.3.2 Average AUC values over 100 replications for Scenario 2, using Direct compared to classic FPCA with BIC, and using robust FPCA with RBIC and trimming levels $\alpha = 0.1, 0.2$ and 0.3. . . . . 155

7.4.1 Outliers detected for temperature features (Temp) using outlier detection on the temperature features directly (Direct), and the outliers found using CFLR and RFLR. . . . . 163

7.4.2 Outliers detected for temperature features (Temp) using outlier detection on the temperature features directly (Direct), and the outliers found using CFLR and RFLR for Vibration Surveys in XWB tests. . . . . 168

8.5.1 The AUC values for the version of each classifier that gave maximum AUC values. The Centroid classifier used the lpv decel time series. The DD classifier in the univariate case used the LPV acceleration time series and in the multivariate case used all the time series. The LFLR classifier used a FPCA basis with depth values and used a B-spline basis with depth and Lasso penalty. . . . . 180

# Chapter 1

# Introduction and Objectives

## 1.1 Introduction

Jet engines must pass a number of tests to ensure the engines comply to rigorous certification requirements, mostly associated with safety, as outlined by Walsh and Fletcher (2008). Before a jet engine is released from the factory it must go through a Pass-off test. Each test involves a series of engine manoeuvres (e.g. acceleration, deceleration cycles and holds at fixed speed points) where several hundred engine parameters are recorded at various sample rates. Key points in the test are manually analysed, but the majority of the data is not currently assessed at all. In this thesis we have developed a range of analytical methods to automatically process the entire engine test dataset and provide suitable labels that adequately summarise segments of engine running. We have then built methods that highlight novel behaviour in the jet data, which may be of further interest for analysis by an engineer.

In this chapter we give a general description of the mechanics of a jet engine and

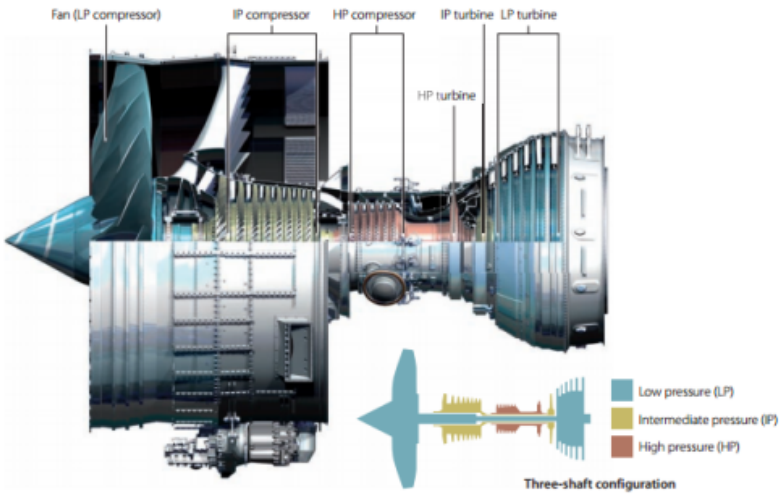



Figure 1.1.1: Summary of LP, IP and HP. www.slideshare.net/egajunior/trent-1000-presentation

the Pass-Off test. We will then outline the contributions of this thesis.

### 1.1.1 Jet Engines

A jet engine is composed of a fan that pumps air into the engine, the air goes through various chambers in which it is compressed thereby increasing the air temperature. The air then enters the combustion chamber in which fuel is injected, creating thrust. To ensure the engine is performing efficiently at different engine speeds there is an Engine Monitoring System (EMS). A jet engine can be split into three zones shown in Figure 1.1.1. There is a low pressure (LP) compressor at the front, which drives air into the turbine. Then there is intermediate pressure (IP) compressor that is composed of alternating static and turning fan blades to compress the air. Finally the high pressure (HP) compressor in the middle, further compresses the air.

There are hundreds of sensors in the engine with measurements taken at a rate of

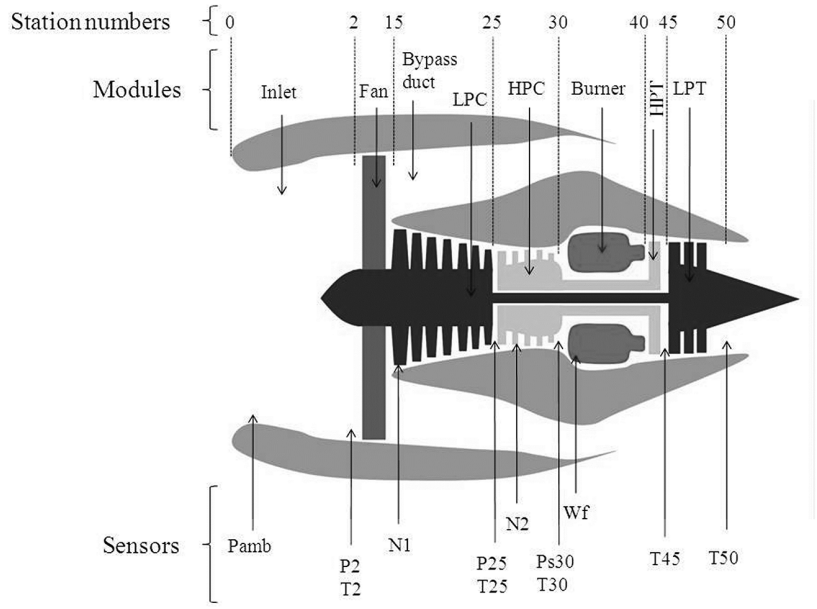



Figure 1.1.2: Station Locations. https://speechfoodie.com/jet-engine-diagram-n1-n2/

40 Hz, measuring different engine features. First we have the engine features N1, N2 and N3, which give the rotating speed of the LP, IP, HP shafts respectively. These can be used as proxies for thrust, and are reported as a percentage of a predefined maximum speed. Second, there are temperature and pressure features measured at different stations in the engine as shown in Figure 1.1.2. Finally, there are three vibration features LPV/IPV/HPV corresponding to vibration values in each of the LP, IP and HP zones respectively. The values are inferred from a single accelerometer at the stiffest part of the engine (Clifton, 2009).

### 1.1.2 Engine tests

We have been given three engine datasets. The first dataset contains sensor data from 93 Pass-Off tests performed on new Trent 1000 engines. Each Pass-off test was

conducted on a single test bed at the Satoo test facility. Note that the 93 tests don't necessarily correspond to 93 different engines, as an engine may be retested after alterations are performed. In each Pass-off dataset there are 22 sensor time series measurements for various engine features. The second dataset contains sensor data from 51 Pass-Off tests performed on XWB engines also performed at the Satoo test facility. The sensor data from the XWB engine Pass-off tests are very similar to those from the Trent 1000, so we will focus on the Trent 1000 data in this section. The third dataset is a Cyclic test performed on a single XWB engine. The focus of this thesis in on the Pass-Off test data however we will do some analysis on the Cyclic test dataset.

**Pass-Off test**

A Pass-Off test is performed by a human controller who pushes the throttle to accelerate and decelerate the engine. In the test the engine starts at a set idle speed, then a manoeuvre is performed in which the engine can be accelerated, decelerated and kept at fixed speeds before returning to idle speed. There are a predefined list of manoeuvres performed in a test. During the test the engineers check the manoeuvres at certain key points to ensure that the engine is complying with the regulatory requirements. If they notice something unusual during a manoeuvre, they can repeat the manoeuvre or they can stop the test. Once the engine is stopped they can make an adjustment to the engine and then restart the test. The manoeuvre is then repeated. For the Performance Curve (P) manoeuvre they sometimes perform only part of the manoeuvre where an issue was identified.

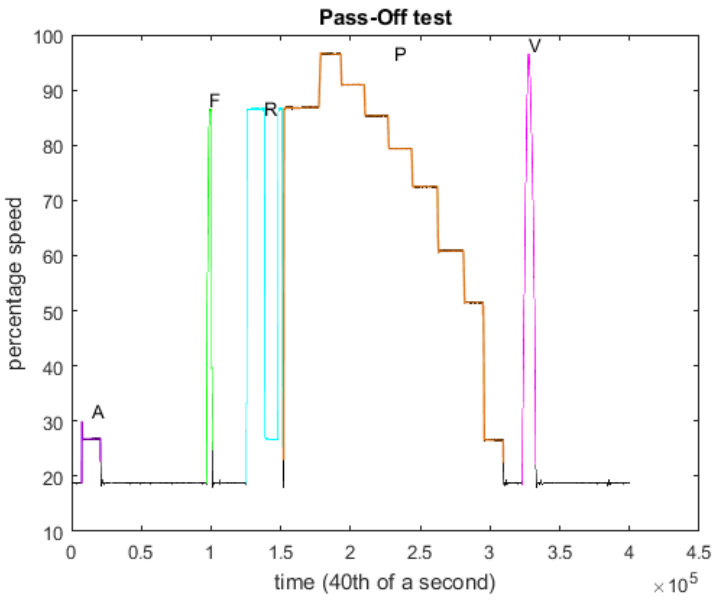
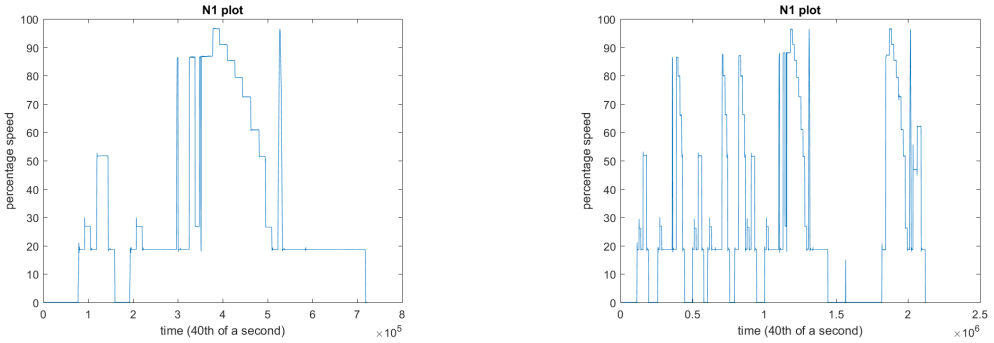



Figure 1.1.3: Perfect Pass-off test with samples taken every 40th of a second.

In Figure 1.1.3 we have plotted the N1 speed time series of a "perfect" test run, where each manoeuvre has been performed exactly once, in the correct order, with no stops during the test. The different manoeuvres are labelled on the time series. We can see that the manoeuvres start and finish at idle speed 18%. The N1 speed time series for two different Pass-Off tests are shown in Figure 1.1.4. The engine has been stopped and manoeuvres have been repeated, so neither of the tests are perfect. However in the two examples there is a section of the test that resembles Figure 1.1.3 i.e. where a perfect test run has been performed.

In Figure 1.1.5 we have a plot of the N1, N2 and N3 time series for Dataset 1. The three time series follow a similar pattern but have different speed ranges. In the data we have multiple pressure sensors located alongside the temperature sensors, we have P20, P30, P42 and P44 (location references can be found in Figure 1.1.2). In Figure 1.1.6, we have a plot of the different pressure measurements alongside the N1




(a) Pass-Off Test 1 (b) Pass-Off Test 2

Figure 1.1.4: N1 speed for Pass-Off test 1 (left) and 2 (right).

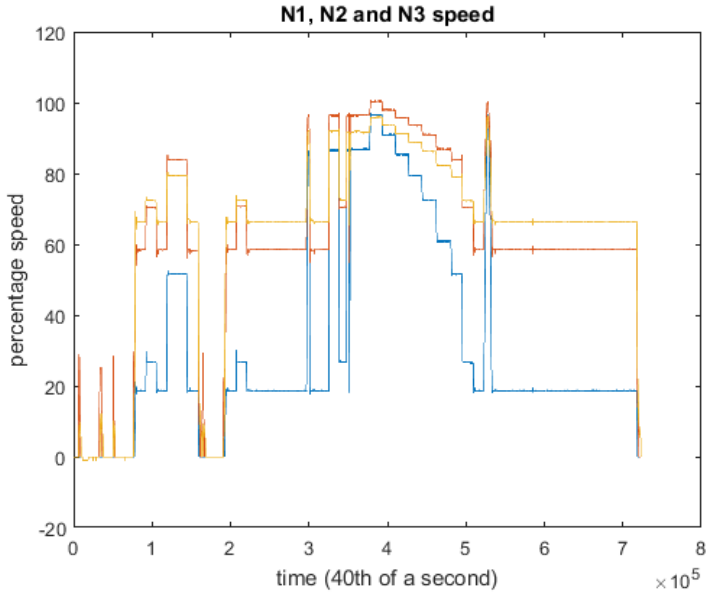



Figure 1.1.5: Plots of N1 (blue), N2 (orange) and N3 (red) speed time series for Dataset 1.

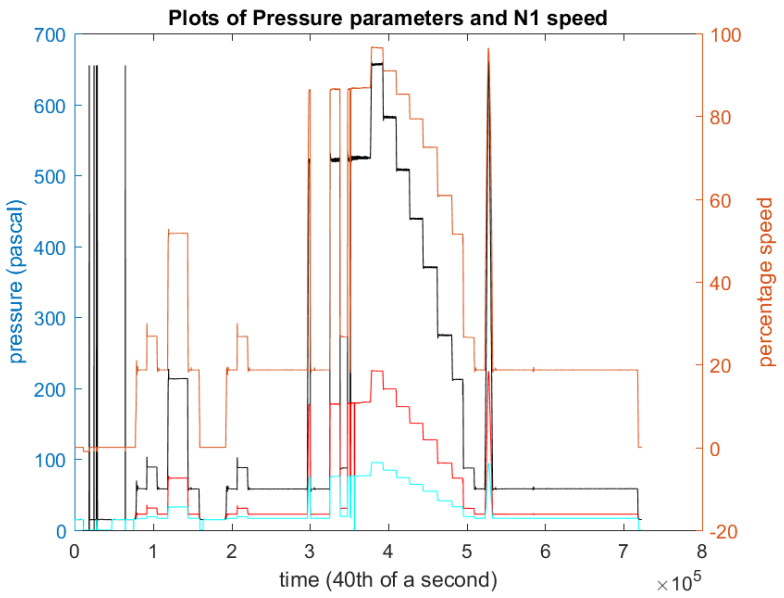



Figure 1.1.6: Plots of P30 (black), P42 (red) and P44 (cyan) which are plotted alongside the N1 speed (orange) time series for Dataset 1.

speed. The pressure time series follow the shape of the N1 time series though they are on different scales. This plot highlights the well know fact that pressure reacts immediately to changes in speed.

There are multiple temperature sensors located along the turbine. The T20 sensor measures the ambient temperature outside the engine, which typically remains constant. In our dataset we have five temperature features. We have temperature readings T25 and T30 at stations shown in Figure 1.1.2. We have the turbine gas temperature (TGT) and also temperature readings of the cooling air at the rear/front of the engine (TCAR/TCAF). In Figure 1.1.7 we have a plot of the the T30 temperature time series alongside the N1 speed. We can see that there is a delayed temperature response with respect to the engine accelerations and decelerations, which is also the same for the other temperature features.

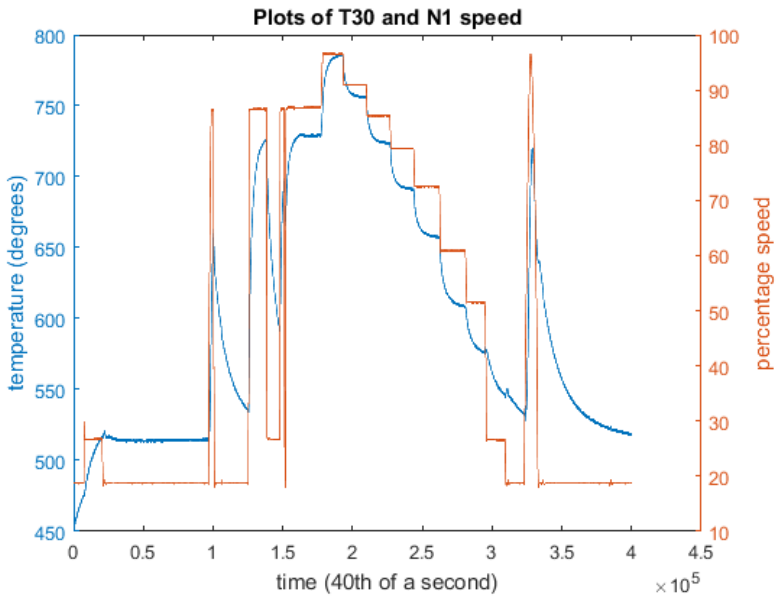
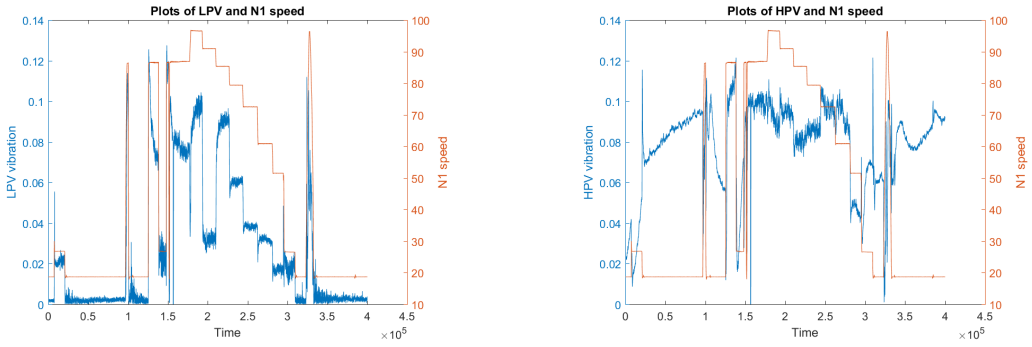



Figure 1.1.7: Plots of T30 (blue) and N1 speed (orange) time series for a perfect test.

The engine vibration values are important diagnostic engine features. The vibration data is acquired from a vibration transducer. As stated before there are the LPV/IPV/HPV vibration features. In Figure 1.1.8 we have a plot of the LPV and HPV time series, alongside the N1 speed time series. There is greater noise in the vibration in comparison to the pressure and temperature readings. When there is a change in N1 speed there is a direct change in the vibration, this illustrates vibration reacts quickly to changes in speed. The relationship between vibration and N1 speed is non-linear as illustrated by the drop in vibration in the middle of manoeuvre P, which is caused by resonance. The LPV and the HPV behaviour is very different. The LPV in general stays at a fixed vibration value when the engine is running at a fixed speed whereas the HPV displays drift, which is clearly not a product of the engine speed. In the engine tests one of the regulatory conditions is that the peak vibration values are below certain thresholds.




(a) LPV (b) HPV

Figure 1.1.8: Plots of the LPV and HPV in blue alongside the N1 speed in orange.

### Cyclic Test

A Cyclic test is performed to accumulate evidence to show that the engine build meets certain criteria, which is passed onto the required regulatory bodies. In a Cyclic test they start by performing a 'Shake-down' test to ensure they are satisfied with the engine build. In the second part of the test they perform cycles of repeated manoeuvres. The Cyclic tests have a planned schedule however deviations can be made. In Figure 1.1.9 we have the N1 speed plotted for the Cyclic test. The initial 'Shake-down' test can be seen by the spread out and seemingly random manoeuvres, then there are short highly repeated manoeuvres signalling the start of the engine cycles. The data is down-sampled due to storage limitations.

In Figure 1.1.10 we have a plot of two segments of the Cyclic test. Segment 1 is from the 'shake-down' phase where a range of manoeuvres are performed. Segment 2 contains clearly repeated cycles with the same N1 speed profiles. Note that there are no defined list of manoeuvres in the Cyclic test as in the Pass-Off test.

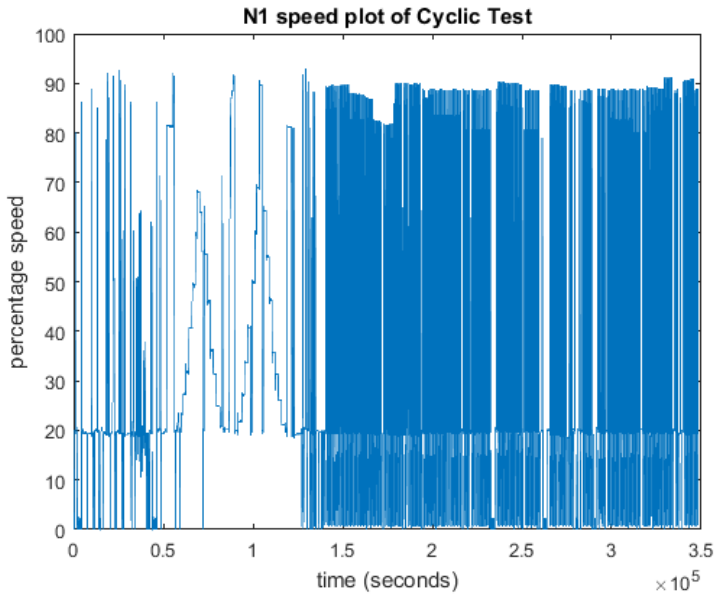
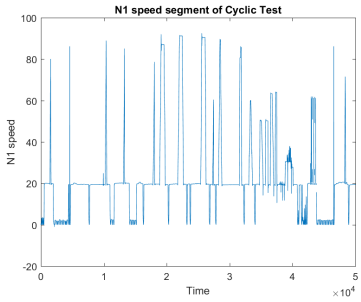
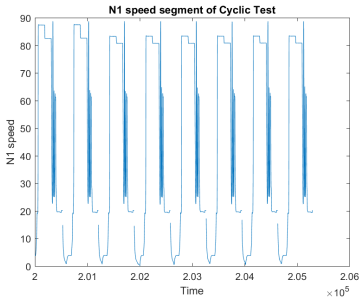



Figure 1.1.9: Plot of the N1 speed time series generated for the Cyclic test, with samples taken every second.




(a) Segment 1




(b) Segment 2

Figure 1.1.10: Plots of two sections of the Cyclic test.

### 1.1.3 Contributions and Thesis Outline

In this thesis we will describe the analytical tools that we have developed for the sensor data generated during Pass-Off and Cyclic tests. The first contribution is a classification algorithm for extracting and labelling the manoeuvres performed during a Pass-Off test. This algorithm uses a number of different statistical techniques to obtain informative features that are used to give effective classifications. The algorithm is computationally efficient and can deal with data sampled at different rates. The algorithm has been tested on various engine datasets and has been implemented into the Rolls Royce system. We have also developed an unsupervised approach to identify the manoeuvre classes in a Cyclic test. Our second contribution is a robust regression model that we have developed to model the engine temperature behaviour with respect to the engine speed. The model uses a number of functional data analysis techniques. We derive asymptotic results and perform a simulation study to illustrate the effectiveness of the model. Using this model we have built an outlier detection algorithm for the jet engine data.

In Chapter 2 we give a review of previous statistical techniques developed for engine data. In Chapter 3 we outline Functional Data Analysis techniques which we will use extensively in the algorithms we have developed. Chapters 4-8 contain new research, which we will outline briefly.

**Chapter 2: Methodology developed for Jet engine data**

This chapter contains a review of methodology developed for jet engine data. We focus on engine health monitoring, which typically involves using sensor data to give

early warning of potential engine issues. Early engine warnings can ensure the safety of the engine and enable relatively cheap maintenance to be performed. There are three main approaches to this problem. We shall also describe visualisation tools used to identify clusters and outliers in the data. Finally we will give a brief comparison between these methods and our approach.

**Chapter 3: Functional Data Analysis**

We shall give a review of four important areas of Functional Data Analysis: Functional Principal Component Analysis (FPCA), Functional Linear Regression (FLR), Functional Depth (FD) and Outlier detection for Functional Data. We will focus largely on FPCA, which is an extension of principal component analysis (PCA) for functional data. PCA is a technique that takes a set of multivariate points each of which come from the same underlying vector of random variables, and projects the data into a new feature space consisting of a smaller number of random variables. The new feature space still captures a significant proportion of the variance in the original data set, as correlated random variables can give redundant overlapping information. As expected there is a nice symmetry between PCA and FPCA. In particular both methods have two interesting derivations. By first looking at PCA then FPCA, the formulation and intuition can be shown to follow naturally; making it easier to understand the ideas behind FPCA. In this chapter we will include the formulation of FPCA, stating the classical results and proofs. We will then briefly discuss various modifications and extensions. We shall give a brief description of FLR and some of the popular estimates used. We give a short review of FD, which ranks a set of curves. The ordering from

FD can be used for a number of problems including outlier detection, classification and clustering. We shall describe the methodology and properties that a FD measure should satisfy. We give examples of some of the most popular FD choices. Finally, we outline various outlier detection approaches for Functional Data.

**Chapter 4: Classification of manoeuvres in a Pass-Off test**

*This chapter outlines an algorithm that is being used by the Rolls-Royce Control, Monitoring & Systems UTC at the University of Sheffield and within the Rolls Royce systems.*

This chapter outlines the classification algorithm developed to extract and label manoeuvres in a Pass-Off test. The Pass-Off test sensor data does not come with labelled manoeuvres. We therefore built a classification algorithm that can extract and label manoeuvres computationally efficiently and is able to achieve near perfect classification. The algorithm can support the engineers at Rolls Royce to make engine diagnostics for the Pass-Off tests. We have built templates for each of the seven pre-defined manoeuvres, with respect to the N1 speed. We can also have manoeuvres that do not match any of the pre-defined manoeuvres, which we will label as Unknown (U). To extract the manoeuvres we use the changepoint algorithm: Pruned Exact Linear Time (PELT) (Killick et al., 2012). We then use a modification of the Needleman-Wunsch (NW) algorithm (Needleman and Wunsch, 1970) for continuous data alongside Functional Principal Component Analysis (FPCA) (Ramsay and Silverman, 2005) to score the similarity between an unlabelled manoeuvre and the templates of the pre-defined manoeuvres. This gives us a vector of scores. We then consider using a

Decision Tree (DT) or a Linear Discriminant Analysis (LDA) classifier to label the manoeuvre using the vector of scores. The scores generated are very informative, making the resulting classification very accurate. The framework was originally built for Trent 1000 engine tests, however is has also been applied to XWB engine tests.

**Chapter 5: Manoeuvre Clustering in Cyclic tests**

This chapter outlines a clustering algorithm to identify manoeuvres in a Cyclic test. Unlike the Pass-Off test we do not have labels for the manoeuvres performed in a Cyclic test. We will therefore cluster the manoeuvres to identify the different manoeuvre classes. In the test the engineers can perform manoeuvres that do not match the manoeuvre classes, which can affect the clustering performance of many standard methods. We therefore consider a density based approach known as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which is capable of identifying outliers and estimating the number of clusters present. We chose to use a Dynamic Time Warping (DTW) distance as manoeuvres can vary slightly in length and shape. DTW aligns two time series and then takes the squared difference between the aligned time series. The DTW distances are used as inputs for the DBSCAN algorithm. Applying the algorithm on the manoeuvres in the Cyclic tests we obtained meaningful clusters. We test the algorithm on the manoeuvres in the Trent 1000 and XWB Pass-Off tests, for which we have labels.

**Chapter 6: Robust Functional Linear Regression**

*This chapter contains content from a journal contribution with co-authors David S. Leslie, Nicos G. Pavlidis and Steve King. The manuscript has been submitted to "Technometrics".*

In the Pass-Off test dataset the Vibration Survey (V) manoeuvre has been repeated multiple times, which suggests that something unusual may be occurring during this manoeuvre. We want to use the temperature engine parameters to identify any abnormal behaviour. However, because these manoeuvres are performed by a human controller, there is a variability that can mask the outliers. Therefore we have built a model to capture the relationship between the engine speed and engine temperature in the presence of possible outliers. The engine temperature has a lag effect with respect to the engine speed, which needs to be incorporated into the model. We will use Functional Linear Regression, which is a widely used approach to model functional responses with respect to functional inputs. However classical Functional Linear Regression models can be severely affected by outliers. We therefore introduce a Fisher-consistent robust Functional Linear Regression model that is able to effectively fit data in the presence of outliers. The model is built using robust Functional Principal Component and Least Squares regression estimators. The performance of the Robust Functional Linear Regression (RFLR) model depends on the number of principal components used, which will be chosen using a consistent robust model selection procedure. We give consistency results for both the RFLR model and the model selection procedure. A simulation study shows our method is able to effectively

capture the regression behaviour in the presence of outliers.

**Chapter 7: Outlier Detection using Functional Regression**

*This chapter contains content from a conference contribution with co-authors David S. Leslie, Nicos G. Pavlidis and Steve King. The manuscript has been accepted at the "Workshop on Advanced Analytics and Learning on Temporal data" at The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2019.*

We propose an outlier detection algorithm for temperature sensor data from jet engine tests using robust functional regression. Effective identification of outliers would enable engine problems to be examined and resolved efficiently. Outlier detection in this data is challenging because a human controller determines the speed of the engine during each manoeuvre. This introduces variability which can mask abnormal behaviour in the engine response. We therefore use the robust Functional Linear Regression model given in Chapter 6 to identify 'normal' behaviour, then use Functional Depth to identify the outliers. The framework is tested on simulated and real engine data.

**Chapter 8: Predict Repeated Vibration Surveys**

In a Pass-Off test manoeuvres can be repeated by an engineer during the test. Typically a manoeuvre is repeated if it does not fulfil the conditions required. We have found the Vibration Survey manoeuvre is repeated significantly more than the other manoeuvres. For this manoeuvre the engineers check certain vibration conditions

are satisfied. An automated approach to determine whether a manoeuvre should be repeated has a number of benefits. We implement three different functional classification methods, and compare the methods using ROC curves. We have found that these approaches can give reasonably accurate predictions.

# Chapter 2

# Methodology developed for Jet engine data

## 2.1 Introduction

Engine health monitoring (EHM) systems store sensor output throughout an engine test. The availability of this rich data source has prompted a number of early warning detection methods, enabling appropriate maintenance to be performed before detrimental engine damage. During engine design certain modes of failure are identified and either the engine design is altered to mitigate against these failures, or otherwise an on-line monitoring system is put in place to ensure these failures are detected early. The engineers follow a framework called Failure Mode Effect and Critical Analysis (FMECA) (Rausand and Høyland, 2004). The framework also considers the likelihood and impact of each of the failures and sets a guideline of actions that should be taken for the various types of failures. Fault-specific detection schemes have

therefore been built which use expert-knowledge (Merrington, 1994; Patton et al., 2000). A more detailed survey of expert-based monitoring techniques is given by Hanachi et al. (2018).

Statistical Process Control methods have also been deployed for jet engine monitoring. These methods typically give a warning when engine parameters exceed certain predefined thresholds. The thresholds are typically set using expert opinion, which may not pick up subtle abnormalities (King et al., 2009).

The abundance of normal engine data examples has prompted novelty detection approaches to be considered. Novelty detection models use only normal engine running instances to build a model of normal behaviour. The model can then be validated using abnormal engine examples. The approaches can be broken down into four key areas. First the data is pre-processed, next visualisation tools are used to explore the data, then a normality model is constructed, and finally a novelty threshold is set.

Visualisation tools are important in giving the engineers a tool for understanding the data structure and the potential outliers. Clifton (2009) outline a few projection methods that have been used to map engine data to a low dimensional space. These projections aim to preserve the structure in the higher dimensional space. We will describe various approaches and highlight the essential ideas between them in Section 2.2.

There are three main novelty detection approaches. The first approach transforms the data and then applies $k$-means clustering to capture different types of normal behaviour. A threshold is then set around each cluster (Nairac et al., 1999). The second approach uses a one-class Support Vector Machine (SVM) (Hayton et al.,

2007), which estimates a hyperplane that aims to give the best split between the normal data and potential outliers. The third approach fits a probability density to the data either using Kernel density (King et al., 2009) or Gaussian mixture models (Clifton, 2009). A threshold is then set using Extreme Value Theory methods (Clifton, 2009).

The three novelty detection approaches use vibration parameters as described in Chapter 1. Many of the approaches use Tracked Order Response (TOR) curves, which are defined as the vibration amplitude at fundamental frequencies. For example if the engine rotates at $h$ Hz, then the peak vibration energy occurs at $h$ Hz, with corresponding harmonics at multiple of $h$ Hz.

In this chapter we will describe the three novelty detection approaches currently developed for engine monitoring. We shall also outline our approach to identify abnormal engine behaviour. A brief discussion will be given on the projections used to obtain visualisation of the data.

## 2.2 Data Visualisation

The Pass-Off data is high dimensional with multiple engine parameters at various engine speeds. The data can be preprocessed and features can be extracted but these can also be in more than three dimensions. Therefore projection methods have been outlined to visualise the data. Visualisation approaches have been used by Clifton (2009) and King et al. (2009) to visualise the outliers. We will use visualisation techniques in Chapter 5 to highlight cluster structures.

There are linear approaches that project the data using a linear transformation. The most popular example is Principal Component Analysis (PCA) (described in Section 3.2.1). They map the data onto the first two principal components, which capture the largest proportion of the variance. This is a linear mapping and we can easily incorporate new data into the projection. However if the first two components do not capture a significant proportion of the variance this approach becomes unreliable. Alternatively, there are topographical approaches that aim to preserve the pairwise distances, for example Sammon's mapping. Let $x_1, ..., x_n \in \mathbb{R}^q$, then two points $x_i, x_j$ in the original space have distance $d_{ij} = d(x_i, x_j)$. The projected points $y_i, y_j \in \mathbb{R}^2$ have distance $d^*_{ij} = d(y_i, y_j)$. The mapping is chosen that minimises the Sammon stress metric

$$E_{sam} = \frac{1}{\sum_{i=1}^{n} \sum_{j>i}^{n} d^*_{ij}} \sum_{i=1}^{n} \sum_{j>i}^{n} \frac{(d_{ij} - d^*_{ij})}{d^*_{ij}}.$$

Typically the Euclidean distance is used and the optimisation is performed using gradient descent (Nabney, 2002). New samples can not be incorporated into the mapping.

NeuroScale (Lowe and Tipping, 1997) aims to minimise the Sammon stress metric $E_{sam}$ using a neural network with a single layer of $H$ hidden nodes. Each of the hidden nodes correspond to a radial basis function (RBF). The algorithm follows a two stage process, first the parameters of the radial basis functions are estimated so they approximate the probability density of the training set. Then the output weights are estimated. Unlike Sammon's new samples can be projected using the

neural network.

The t-Distributed Stochastic Neighbor Embedding (tSNE) by Maaten and Hinton (2008) aims to group points using a probabilistic framework. The algorithm works in two steps. In the first step, they estimate the probability of points being similar. Then they look for a projection such that the probability is preserved in the low dimensional points. The similarity of $x_j$ to $x_i$ is given by the conditional probability:

$$p(x_j|x_i) = \frac{\exp(-||x_i - x_j||^2/2h_i^2)}{\sum_{k\neq i}\exp(-||x_i - x_k||^2/2h_i^2)},$$

where $h_i$ is the bandwidth of the Gaussian kernels. Let $y_1, ..., y_n$ be the projected points with similarity measure

$$p(y_j|y_i) = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k\neq i}\exp(-||y_i - y_k||^2)}.$$

tSNE tries to find $y_i$ that minimises the difference between $p(x_i|x_j)$ and $p(y_i|y_j)$. They define the cost function to be the Kullback-Leibler (KL) distance

$$\sum_{i=1}^{n} KL(P_i|Q_i) = \sum_{i=1}^{n}\sum_{j=1}^{n} p(x_j|x_i)\frac{p(x_j|x_i)}{p(y_j|y_i)},$$

where $P_i$ and $Q_i$ are the conditional probability distributions over all $x_i$ and $y_i$ respectively. We will use tSNE in Chapter 5 to visualise the clusters.

## 2.3 Novelty Detection Approaches

### 2.3.1 K-means Model

Nairac et al. (1999) uses vibration parameters LPV, IPV and HPV. They split the vibrations into 6 equispaced speed ranges, and take an average in each range to obtain a vector of size 18. Given $n$ samples $x_1, ..., x_n \in \mathbb{R}^{18}$, they apply a whitening transformation that maps points $x_i$ to $x_i' = \Lambda^{-\frac{1}{2}} V^T (x_i - \mu)$ where $\mu$ is the mean vector; $\Lambda$ is a diagonal matrix of eigenvalues for the covariance matrix $\Sigma$ and $V$ is the corresponding matrix of eigenvectors.

The distribution of the feature vectors $x_i'$ is approximated by four spherical clusters found using $k$-means. To determine a threshold they define the cluster radius $\rho_k$ given by the average distance of points in cluster $k$ to cluster centre $c_k$. For a new point $x^*$ the normalised distance is given by $\delta(x^*) = \min_k \frac{1}{\rho_k} |x^* - c_k|$. The distance $\delta(x^*)$ essentially gives the number of standard deviations $x^*$ is from the closest cluster centre.

Nairac et al. (1999) uses the $k$-means model to capture different types of normal engine behaviour, and chooses $k = 4$ by visual inspection of a two dimensional projection. One significant limitation of the $k$-means model highlighted by Hayton et al. (2007) is that the engines cannot be ranked by the novelty score $\delta(x^*)$ as the distances may be evaluated with respect to different cluster centres.

### 2.3.2 Support Vector Machines

Support Vector Machines (SVM) estimate hyperplanes or decision boundaries that give the largest separation of the different classes. Using the hyperplane we can

classify points depending on which segment of the space the points appear. Typically the data is projected into a higher dimensional space, which increases the distance between the points.

A one-class support vector machine (SVM) is used by Hayton et al. (2007) to build a novelty detection model. They use the fundamental TOR and then take a weighted average in 10 equidistant speed bins. A probabilistic support vector machine approach was given by Clifton et al. (2014), which enables uncertainty values to be given which can improve decision making.

Matthaiou et al. (2017) also use a one-class SVM to perform novelty detection on jet engine data. However they use different feature to those by Hayton et al. (2007). They suggest applying a wavelet decomposition to the TOR curves (defined in Chapter 1) and then applying Kernel Principal Component Analysis on the coefficients from the wavelet decomposition. This procedure is similar to Functional PCA, which we will discuss in Section 3.2.

### 2.3.3 Probabilistic Model

Clifton (2009) apply a two stage pre-processing of the vibration data. First, note that the Pass-Off test stays a large portion of the time at certain fixed speed levels therefore the vibration values at these speeds will be overrepresented. To obtain a balanced dataset a filtering process is performed. Given vibration value $v_t$ and speed $s_t$ at time $t$, they discard $v_t$ if $|s_t - s_{t-1}| < w$ where $w$ is a pre-chosen threshold. In the second step they split the vibration values into equispaced bins as performed in the k-means and SVM approaches.

King et al. (2009) uses a Gaussian kernel $H(x) = (2\pi)^{-\frac{d}{2}} \exp\{-\frac{1}{2}x^2\}$ to estimate the probability density of the $d$-dimensional data in each speed bin:

$$p(x) = \frac{1}{nh^d} \sum_{i=1}^{n} h\left(\frac{x - \mu_i}{h}\right).$$

Alternatively a Gaussian Mixture Model could be used (Clifton, 2009) where

$$p(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} \sum_{k=1}^{K} \frac{P_k}{|\Sigma_k|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right\}$$

where $P_k$ are the weights associated to each of the $d$-dimensional Gaussian components with parameters $(\mu_k, \Sigma_k)$.

A new sample $x^*$ has a probability $p(x^*)$ of coming from the same distribution as the training set. Clifton (2009) choose a threshold using Extreme Value Theory (EVT) methods. To obtain the threshold they assume the data is distributed according to a one-sided Gaussian distribution. Given this assumption we could obtain a threshold by setting a quantile for the probability density, however for sufficiently large quantiles there are numerical issues estimating these thresholds. Therefore using EVT they avoid these numerical issues.

## 2.4 Our Approach

In this chapter we have discussed three novelty detection approaches applied to jet engine data. The three approaches use vibration data, which is preprocessed and grouped into speed bins. These approaches have two notable limitations: they all require labelled data and second, the preprocessing of the data loses important

temporal information.

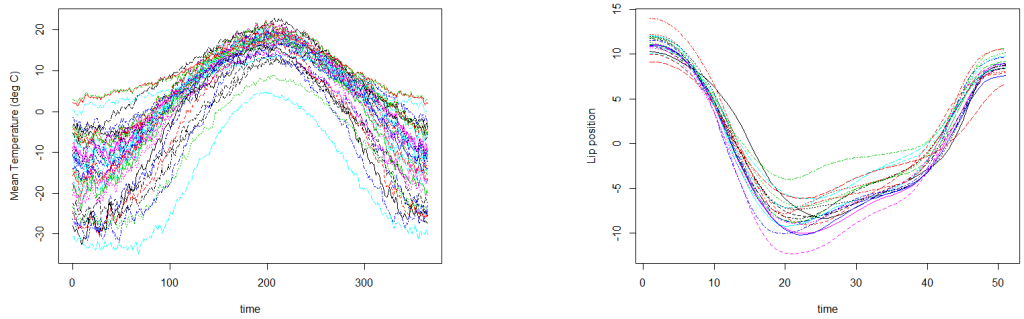
In this section we will give a brief description of our approach to identifying abnormal engine behaviour. We will adopt an outlier detection approach. In this paradigm we do not assume that the samples are labelled as normal, instead we assume there are outliers present in our data. We will therefore adopt robust statistical methods (Huber, 2011) to model the engine data. We will focus on the Vibration Survey manoeuvre, which we will extract using the classification algorithm given in Chapter 4. By comparing across the Vibration Survey manoeuvres instead of the Pass-Off tests, we should obtain more consistent results. We will use functional data analysis techniques (Ramsay and Silverman, 2005) to identify abnormal temperature behaviour with respect to the engine speed. We do not pre-process the data, instead we aim to use the temporal information to identify outliers.

# Chapter 3

# Functional Data Analysis

## 3.1 Introduction

Functional data analysis (FDA) is a popular tool for modelling and analysing time series data. The area has grown rapidly over the last 20 years due to the increase in sensor data collection. The sensor data is called *functional* if it is believed to arise from an underlying process. For example in Figure 3.1.1 we have two functional data examples. First we have temperature measurements from 35 cities in Canada over a year. We can see that there is a clear process where temperatures increase over summer and decrease over winter. The second example contains measurements of the lower lip of 20 people during the pronunciation of the word 'bob'. Again there is an underlying process of saying the word 'bob'. The FDA methodology is well suited to these types of data as we treat the time series as discrete observations from a single function rather than a sequence of observations. More details about the types of functional data and applications is given by Ramsay and Silverman (2005).




(a) Canadian Temperature dataset (b) Lip dataset

Figure 3.1.1: Plots of Canadian Temperature dataset containing temperature reading over a year from 35 Canadian cities and Lip dataset of measurements of the lower lip of 20 people during the pronunciation of the word 'bob'.

In this chapter we will discuss four important areas of FDA: Functional Principal Component Analysis (FPCA), Functional Linear Regression (FLR), Functional Depth and Functional Outlier Detection. FPCA is an extension of classical Principal Component Analysis (PCA) for functional data. FPCA can give a low-dimensional representation for a set of curves. We will use FPCA representations in the classification algorithm in Chapter 4, and a robust FPCA model in Chapter 6. In Section 3.2 we shall introduce PCA and the extension to FPCA.

Functional Linear Regression (FLR) is a popular regression model for functions. In the model one or both predictor and response variables can be functions. We will show using a double basis expansion approach by Ramsay and Dalzell (1991) that the FLR problem can be reduced to a multivariate regression problem. We shall also describe some extensions to the model. A robust extension of FLR will be given in Chapter 6 in which the predictor and response are both functions.

We will introduce Functional Depth (FD) and describe a few of the depth functions in the literature. The notion of depth was originally developed as a way of ordering multivariate data, but has been extended to functional data. Functional depth can be used in a variety of ways including outlier detection and classification (Wang et al., 2016). We will use FD to identify outliers in Chapter 7 and as a classification tool in Chapter 8. Lastly, we will discuss outlier detection approaches for functional data. A majority of these approaches use functional depth. We will compare these approaches to our outlier detection model in Chapter 7.

## 3.2 Functional Principal Component Analysis

Functional Principal Component Analysis (FPCA) is one of the most popular methods for understanding and exploring functional data. The first main application of FPCA, is dimensionality reduction; mitigating against the curse of dimensionality. The second application is to highlight modes of variation, which can be investigated further to uncover useful patterns in the data. We will give an introduction into FPCA including the formulation of FPCA, and the classical results. There will also be a discussion on how FPCA can be applied in practice using the Basis method (Ramsay and Silverman, 2005).

We will focus on classical FPCA and briefly discuss a few extensions. The literature in this area is vast and varied (Shang, 2014), so to simplify matters we will focus on the case of parametric methods for regularly sampled data. There are non-parametric approaches such as those discussed by Ferraty et al. (2012) and methods

for longitudinal data as discussed by Yao et al. (2005).

We will start by looking at Principal Component Analysis (PCA), which is a popular dimensionality reduction tool for multivariate data. PCA is a data-driven projection method that transforms a set of variables (possibly correlated) to a smaller set of variables that are uncorrelated. The uncorrelated random variables formed using PCA can retain a large amount of the information in the original variables. FPCA is the functional extension of PCA. As expected there is a nice symmetry between PCA and FPCA. In particular both methods have two interesting derivations. By first looking at PCA then FPCA, the formulation and intuition can be shown to follow naturally; making it easier to understand the ideas behind FPCA.

### 3.2.1 Principal component analysis

Let $X = (X_1, ..., X_p)$ be a vector of $p$ zero-mean random variables with covariance matrix $\Sigma$. Let $x = (x_1, ..., x_n)$ be $n$ observations from $X$, where $x_i = (x_{i1}, ..., x_{ip})$ for $i = 1, ..., n$. PCA finds a new set of independent random variables $(Z_1, ..., Z_p)$ where $Z_k = \sum_{j=1}^{p} \alpha_{kj} X_j$ is the $k$-th projection, and $\alpha_k = (\alpha_{k1}, ..., \alpha_{kp})$ is the $k$-th principal component (PC). The PC $\alpha_1$ is chosen such that $Z_1$ has the highest variance. Subsequently PCs $\alpha_k$ are chosen to maximise the variance of the projections $Z_k$ under the condition that $\alpha_k$ and $\alpha_l$ are orthogonal for $k \neq l$.

We will refer to $\hat{\alpha}_k$ as the estimated $k$-th principal component. The principal component $\hat{\alpha}_1$ is then chosen such that the sample variance of the projections $z_{1j} = \hat{\alpha}_1^T x_j$ are maximised for $j = 1, ..., n$. This can be condensed into matrix form $z_1 = (z_{11}, ..., z_{1n})$ so $z_1 = \hat{\alpha}_1^T x$, where $\hat{\alpha}_1^T$ denotes the transpose of $\hat{\alpha}_1$. More formally, the

PCs $\hat{\alpha}_k$ are chosen so

$$\hat{\alpha}_k = \underset{\alpha_k^T \alpha_k = 1}{\arg\max}\, \alpha_k^T \hat{\Sigma} \alpha_k \quad \text{such that} \quad \alpha_k^T \alpha_l = 0 \quad \forall l \neq k,$$

where $\hat{\Sigma} = \frac{1}{n} x^T x$ is the sample covariance matrix.

**Lemma 3.2.1.** *Let $x = (x_1, ..., x_n)$ be $n$ independent realisations of a $p$ dimensional vector $X$ corresponding to random variables $(X_1, ..., X_p)$, with sample covariance matrix $\hat{\Sigma}$. Denote $z_{11}, ..., z_{1n}$ as the projection vectors of the points $x_1, ..., x_n$ with respect to the first principal component $\hat{\alpha}_1$, and the normalisation condition that $\hat{\alpha}_1^T \hat{\alpha}_1 = 1$. Then the first principal component $\hat{\alpha}_1$ corresponds to the eigenvector of $\hat{\Sigma}$ with the largest eigenvalue.*

*Proof.* Let $\hat{\alpha}_1$ be the vector that maximises the variance: $var[\hat{\alpha}_1^T x] = \hat{\alpha}_1^T \hat{\Sigma} \hat{\alpha}_1$. Using a Lagrange multiplier $\lambda$ on the normalisation condition, we want to maximise the objective function

$$L = \hat{\alpha}_1^T \hat{\Sigma} \hat{\alpha}_1 - \lambda(\hat{\alpha}_1^T \hat{\alpha}_1 - 1).$$

Differentiating with respect to $\hat{\alpha}_1$ gives

$$\hat{\Sigma} \hat{\alpha}_1 - \lambda \hat{\alpha}_1 = 0 \rightarrow (\hat{\Sigma} - \lambda I) \hat{\alpha}_1 = 0,$$

where $I$ is a $p \times p$ identity matrix, so $\hat{\alpha}_1$ is an eigenvector of $\hat{\Sigma}$ with eigenvalue $\lambda$.

Next we will show that $\lambda$ is the largest eigenvalue of $\hat{\Sigma}$. In other words the eigenvector with the largest eigenvalue maximises the sample variance of the projected points $z_1$. This can be shown as follows:

$$var(z_1) = \hat{\alpha}_1^T \hat{\Sigma} \hat{\alpha}_1 = \hat{\alpha}_1^T \lambda \hat{\alpha}_1 = \lambda \hat{\alpha}_1^T \hat{\alpha}_1 = \lambda.$$

We have therefore found that $\hat{\alpha}_1$ is equal to the eigenvector of the sample covariance matrix $\hat{\Sigma}$ with the largest eigenvalue $\lambda$. □

We can extend this result to show that the $k$-th principal component $\hat{\alpha}_k$ corresponds to the eigenvector of $\hat{\Sigma}$ with the $i$-th largest eigenvalue $\lambda_i$, the proof follows a similar argument, which again uses Lagrange multipliers. Note that as $\hat{\alpha}_k$ has been normalised for all $i = 1, ..., p$, the variance of $z_i$ is $var(z_i) = \lambda_i$. The total variance is given by $\sum_{i=1}^{p} \lambda_i$.

The main aim of PCA is for dimensionality reduction: to determine a new set of random variables that captures a large proportion of the variance in the original set of random variables. Taking the first $M$ principal components where $M \ll p$, can be sufficient in capturing the majority of the variance in the data. There are various ad-hoc methods for choosing $M$, for example find $M$ such that the first $M$ principal components captures 90% of the variation, calculated using

$$\frac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{p} \lambda_i} > 0.9.$$

**Minimising the Squared Error**

The PCs have been shown to maximise the sample variance of the projections $z_i$. However the PCs can also be shown to form a basis representation, which gives minimal squared error between the observations and the basis representations. Let

$u = (u_1, ..., u_p)$ be an orthonormal basis such that

$$x_i = \sum_{j=1}^{p} c_{ij} u_j,$$

with constants $c_{ij} \in \mathbb{R}$. We want to find the best possible $M$-term estimate $\hat{x}_i^{(M)}$ for each of the $x_i$, where $\hat{x}_i^{(M)}$ is formed by taking a linear combination from a subset of the orthonormal basis $u_1, ..., u_M$, for some $M < p$:

$$\hat{x}_i^{(M)} = \sum_{j=1}^{M} c_{ij} u_j.$$

For a fixed choice of orthonormal vectors $u_j$, the choice of vector $c_i = (c_{i1}, ..., c_{ip})$ that minimises the reconstruction error can be shown to be $c_i = u^T x_i$.

We want to minimise the reconstruction error

$$\frac{1}{n} \sum_{i=1}^{n} ||x_i - \hat{x}_i^{(M)}||^2. \tag{3.2.1}$$

The reconstruction can be expanded:

$$\begin{aligned}
\frac{1}{n} \sum_{i=1}^{n} ||x_i - \hat{x}_i^{M}||^2 &= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=M+1}^{p} ||c_{ij} u_j||^2 \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=M+1}^{p} c_{ij}^2 \\
&= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=M+1}^{p} u_j^T x_i x_i^T u_j \\
&= \sum_{j=M+1}^{p} u_j^T \hat{\Sigma} u_j.
\end{aligned}$$

where $\hat{\Sigma}$ is the sample covariance matrix defined earlier.

To minimise $u_j^T \hat{\Sigma} u_j$ we need to choose $u_j$ to be the eigenvectors of the sample covariance matrix $\hat{\Sigma}$ with the smallest eigenvalues, this can be easily proven in a similar fashion to Lemma 3.2.1. Therefore the two derivations are equivalent.

### 3.2.2 Functional Principal Component Analysis

FPCA is the functional extension of PCA, and the formulations are very similar. The notion of FPCA was first envisaged by Tucker (1958) and Rao (1958), and has been popularised by Ramsay and Silverman (2005). The aim of FPCA is to capture the variance between functions rather than between points. In this section we will derive the FPCA formulation and show that the functional principal components are equal to the eigenfunctions of the covariance operator. We will then prove two important properties of FPCA. The first property is that the eigenfunctions give the best representation of the data in regards to maximising the variance captured. The second property is that the $M$ eigenfunctions (those with the largest eigenvalues) give the best reconstruction of the observed curves over all possible $M$ dimensional mappings with regards to squared error. These properties highlight the dimensionality reduction capabilities of FPCA. We shall then outline the estimation of the functional principal components using the Basis approach Ramsay and Silverman (2005) and we will briefly describe three extensions to the classical model.

We will assume throughout that the mean of the underlying process is zero. This simplifies computation however in reality the mean function also needs to be estimated. There are consistent estimators for the mean, for example Li and Hsing (2010). The quality of the estimators will naturally effect the resulting analysis and is

an area that has an impact in almost all areas of functional data analysis. However, since standard practice in the literature is to assume the processes have mean zero, we will continue that tradition.

### Deriving Functional PCA

In the following sections we will assume the observed curves are defined on the vector space $L^2(I)$, which is the Hilbert space of square integrable functions on the compact interval $I$ with the inner product $\langle f, g\rangle = \int_I f(t)g(t)dt$ for functions $f, g \in L^2(I)$.

Let $X(t)$ be a square integrable stochastic process on a compact interval $I$, with covariance function $C(s,t) = cov\{X(s), X(t)\}$ for all $s, t \in I$. We are then given $n$ observed curves $x_1(t), ..., x_n(t)$ which we assume to follow the stochastic process $X(t)$, with sample covariance

$$\hat{C}(s,t) = \frac{1}{n}\sum_{i=1}^{n} x_i(s)x_i(t), \tag{3.2.2}$$

and sample covariance operator

$$(\hat{C}f)(s) = \int_I \hat{C}(s,t)f(t)dt, \quad \text{for} \quad f \in L^2(I). \tag{3.2.3}$$

In the following sections we will assume that the estimated covariance function, eigenvalues and eigenfunctions converge almost surely to the true versions. There is a vast literature to measure the quality of estimators within FPCA, with Dauxois et al. (1982) showing that under regulatory conditions the estimated eigenfunctions converge to the true eigenfunctions as the number of sample time series increases.

The first $M$ Functional Principal components (FPCs) $\phi_m$ for $m = 1, ..., M$ maximise

the average variance captured from the observed curves:

$$\frac{1}{n}\sum_{i=1}^{n} < \phi_m, x_i >^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\int_I \phi_m x_i dt\right)^2.$$

subject to $||\phi_m||^2 = 1$ and $< \phi_m, \phi_k >= 0$ for all $k < m$. In Lemma 3.2.2 we will show that this is equivalent to maximising $\langle \phi, \hat{C}\phi \rangle$.

**Lemma 3.2.2.** *Let $x_1(t), ..., x_n(t)$ be independent realisations of the stochastic process $X(t)$ with sample covariance operator $\hat{C}$. Then the first functional PC $\phi_1$ maximises both $\langle \phi, \hat{C}\phi \rangle$ and $\frac{1}{n}\sum_{i=1}^{n} < \phi_1, x_i >^2$.*

*Proof.* To prove the lemma we simply need to show that the two expressions are equal.

$$\begin{aligned}
< \phi, \hat{C}\phi > &= \int_I \phi(t)\hat{C}\phi(t)dt \\
&= \int_I \phi(t)\left(\int_I \hat{C}(t,s)\phi(s)ds\right)dt \\
&= \int_I\int_I \phi(t)\hat{C}(t,s)\phi(s)dsdt \\
&= \int_I\int_I \phi(t)\left(\frac{1}{n}\sum_{i=1}^{n} x_i(t)x_i(s)\right)\phi(s)dsdt \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_I \phi(t)x_i(t)dt\int_I x_i(s)\phi(s)ds \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\int_I \phi(t)x_i(t)dt\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} < \phi, x_i >^2 .
\end{aligned}$$

We can take the sum outside of the integrals using Fubini's theorem, which holds as we assume $\phi \in L^2(I)$ so is continuous on the interval $I$. The expressions are therefore the same so using either statement will give the same FPCs. □

Next we will show that the first FPC $\phi_1$ is the eigenfunction of the empirical covariance function (3.2.2) with the largest eigenvalue $\lambda_1$. The result is given in Lemma 3.2.3.

**Lemma 3.2.3.** *Let $x_1(t), ..., x_n(t)$ be independent realisations of the stochastic process $X(t)$, then the first FPC $\phi_1$ is the eigenfunction of the covariance operator $\hat{C}$ with the largest eigenvalue.*

*Proof.* Using the Lagragian multipler $\rho$ on the normalisation condition, we want to find the first FPC $\phi_1$ that maximises the objective function $J$:

$$\begin{aligned} J =< \phi, \hat{C}\phi > +\rho(1- < \phi, \phi >) &= \int_I \phi(t)\hat{C}\phi(t)dt + \rho(1 - \int_I \phi(t)^2 dt) \\ &= \int_I (\phi(t)\hat{C}\phi(t) - \rho\phi(t)^2)dt + \rho \\ &= \int_I \left(\phi(t) \int_I \hat{C}(t,s)\phi(s)ds - \rho\phi(t)^2\right) dt + \rho \\ &= \int_I \int_I \phi(t)\hat{C}(t,s)\phi(s)dsdt - \rho \int_I \phi(t)^2 dt + \rho. \end{aligned}$$

Differentiating $J$ with with respect to $\phi$ and equating to zero, will give the function $\phi$ that maximises $J$. To do so we need to use functional derivatives, details of which can be found in (Bliss, 1925).

Write $J = J_1 - \rho J_2 + \rho$ where

$$J_1 = \int_I \int_I \phi(t)\hat{C}(t,s)\phi(s)dsdt \quad \text{and} \quad J_2 = \int_I \phi(t)^2 dt.$$

If we add an arbitrarily small perturbation $\delta\phi$ to a functional $J_i$ we can expand $J_i[\phi + \delta\phi]$ using a Taylor expansion in powers of $\delta\phi$

$$J_i[\phi + \delta\phi] = J_i[\phi] + \int_I \Gamma_{1i}(t)\delta\phi(t)dt + \cdots .$$

where $\Gamma_{1i}$ represents the Taylor expansion coefficients for the first order term of $J_i$. In fact $\Gamma_{1i}$ is the first functional derivative of $J_i$ with respect to $\phi$

$$\frac{\delta J_i}{\delta\phi(t)} = \Gamma_{1i}(t).$$

For $J_2$ it is clear that $\Gamma_{12} = 2\phi(t)$. To find the functional derivatives of $J_1$ we note that $\hat{C}(s,t)$ is a symmetric kernel so $\hat{C}(s,t) = \hat{C}(t,s)$ we can therefore show that

$$\Gamma_{11} = \frac{\delta J_1}{\delta\phi(t)} = 2\int_I \hat{C}(s,t)\phi(s)ds.$$

Combining the two results we get that

$$\begin{aligned}\frac{\delta J}{\delta\phi(t)} &= \frac{\delta J_1}{\delta\phi(t)} - \rho\frac{\delta J_2}{\delta\phi(t)} \\ &= 2\int_I \hat{C}(s,t)\phi(s)ds - 2\rho\phi(t) = 0.\end{aligned}$$

Dividing out the 2, we see the functional derivative of $J$ is an eigenequation and therefore $\phi$ must be an eigenfunction with eigenvalue $\rho$.

Next we need to show that $\phi_1$ corresponds to the eigenfunction with the largest eigenvalue

$$< \phi_1, \hat{C}\phi_1 >=< \phi_1, \lambda_1\phi_1 >= \lambda_1 < \phi_1, \phi_1 >= \lambda_1$$

where $\lambda_1$ is the eigenvalue corresponding to eigenfunction $\phi_1$. By Lemma 3.2.2 $\phi_1$ has the largest eigenvalue $\lambda_1$.

□

**Dimensionality Reduction using Functional PCA**

We will next show that the expansion of the first $M$ FPCs $\phi_i$ for $i = 1, ..., M$ gives the best approximation of the observed curves in terms of $L^2$ error. This property makes Functional PCA a powerful dimensionality reduction tool. First, we need to show that the eigenfunctions of the covariance function form a basis for the stochastic process $X(t)$. To do so we will use the Karhunen-Loéve theorem which states that the observed curves can be written as a linear combination of the eigenfunctions.

**Theorem 3.2.4** (Karhunen-Loéve)**.** *Let $(\Omega, F, P)$ be a probability space, where $\Omega$ is the sample space, with $F$ being a $\sigma$ algebra on $\Omega$ and probability measure $P$. Let $X : I \times \Omega \to \mathbb{R}$ be a centred mean-square continuous stochastic process with $X \in L^2(I \times \Omega)$. Then the eigenfunctions $\{\phi_k : k = 1, 2, ...\}$ of the covariance function $C$ of $X$ forms an orthonormal basis of $L^2(I)$, so $X$ can be decomposed into a sum of eigenfunctions*

$$X(t) = \sum_{k=1}^{\infty} W_k \phi_k(t) \tag{3.2.4}$$

*where $W_1, W_2, ...$ are uncorrelated random variables, where $W_k =< X_t, \phi_k >$ and $var(W_k) = \lambda_k$.*

The Karhunen-Loéve theorem shows that $X$ can be decomposed into a linear combination of eigenfunction of the covariance function $C$. We can therefore write the observed curves as

$$x_i(t) = \sum_{j=1}^{\infty} f_{ij}\phi_j(t)$$

where $f_{ij}$ is the principal component score $\int_I x_i(t)\phi_j(t)dt$.

Next we will show that the first $M$ eigenfunctions give the best $M$-basis approximation to the observed curves. The $M$-basis approximation is given by

$$\hat{x}_i^{(M)}(t) = \sum_{j=1}^{M} f_{ij}\phi_j(t). \tag{3.2.5}$$

The fitting criterion which is sometimes known as the error criterion, is given by:

$$\frac{1}{n}\sum_{i=1}^{n} ||x_i - \hat{x}_i^{(M)}||^2 = \frac{1}{n}\sum_{i=1}^{n}\int_I [x_i(t) - \hat{x}_i^{(M)}(t)]^2 dt. \tag{3.2.6}$$

**Lemma 3.2.5.** *Let $x_1, ..., x_n$ be $n$ independent realisations of a stochastic process $X$ defined over a compact interval $I$, with covariance operator $C$. Then the basis of eigenfunctions of the covariance operator $C$ minimises the fitting criterion.*

*Proof.*

$$\begin{aligned}
||x_i - \hat{x}_i^{(M)}||^2 &= \int_I [x(t) - \hat{x}_t^{(M)}]^2 dt \\
&= \int_I \left[\sum_{j=1}^{\infty} f_{ij}\phi_j(t) - \sum_{j=1}^{M} f_{ij}\phi_j(t)\right]^2 dt \\
&= \int_I \left[\sum_{j=M+1}^{\infty} f_{ij}\phi_j(t)\right]^2 dt \\
&= \int \sum_{j=M+1}^{\infty} (f_{ij}\phi_j(t))^2 dt \quad \text{by orthogonality} \\
&= \sum_{j=M+1}^{\infty} f_{ij}^2 \int_I \phi_j(t)^2 dt \\
&= \sum_{j=M+1}^{\infty} f_{ij}^2
\end{aligned}$$

the $f_{ij}$'s are minimised by taking the eigenfunctions with the smallest eigenvalues, so we pick the $M$ eigenfunctions with the largest eigenvalues to minimise the fitting criterion (3.2.6).

□

**Estimating the Functional Principal Components**

There are two main parametric methods for estimating the FPCs (Ramsay and Silverman, 2005). The discretisation approach uses PCA on the time series to find eigenvectors and then apply some smoothing to get an approximation of the FPCs. The basis approach uses some pre-defined basis to define the eigenfunctions and the observed curves, reducing the eigenfunction problem into an eigenvector problem. We will focus on the basis function approach. To find the eigenfunctions of the covariance operator, we can choose some basis functions $\{\theta_k\}_{k=1}^K$ where $K$ is a pre-set number of basis functions. We can then write each of the observed curves $x_i$ as

$$x_i(t) = \sum_{k=1}^{K} a_{ik}\theta_k(t).$$

Define the matrix $x(t) = (x_1(t), ..., x_n(t))$ and the vector of orthogonal basis functions $\theta(t) = (\theta_1(t), ..., \theta_K(t))$. We can then write $x(t) = A\theta(t)$ where $A$ is a $n \times K$ matrix. The covariance function is then

$$\hat{C}(s,t) = \frac{1}{n}x^T(s)x(t) = \frac{1}{n}\theta(s)^T A^T A\theta(t)$$

We next define the order $K$ symmetric matrix $W$ such that $W_{ij} = \int_I \theta_i(t)\theta_j^T(t)dt$ where $I$ is the interval the functions are defined on. Note if we choose the basis

functions to be orthogonal then $W$ is equal to the identity matrix.

Now suppose we can write the eigenfunctions $\phi$ of $\hat{C}$ as a linear combination of $\{\theta_k\}_{k=1}^K$:

$$\phi(s) = \sum_{k=1}^{K} b_k \theta_k(s) = \theta(s)^T b \tag{3.2.7}$$

for constants $b_k \in \mathbb{R}$ and $b = (b_1, ..., b_K)$.

We can then rewrite the eigenequation using decomposition (3.2.7)

$$\int_I C(s,t)\phi(t)dt = \lambda\phi(s) = \lambda\theta(s)^T b. \tag{3.2.8}$$

We can expand the LHS of (3.2.8) to obtain

$$\int_I C(s,t)\phi(t)dt = \int_I \frac{1}{n}\theta(s)^T A^T A\theta(t)\theta(t)^T b dt = \theta(s)^T \frac{1}{n} A^T AWb. \tag{3.2.9}$$

Equating (3.2.8) and (3.2.9) and cancelling out $\theta(s)^T$ we get the following equality

$$\frac{1}{n} A^T AWb = \lambda b. \tag{3.2.10}$$

We also have the condition that $||\phi||^2 = 1$ so

$$1 = ||\phi||^2 = \int_I (b^T\theta(s))(\theta(s)^T b)ds = b^T \left( \int_I \theta(s)\theta(s)^T ds \right) b = b^T W b.$$

Likewise for two distinct eigenfunctions $\phi_i$ and $\phi_j$ they are orthogonal

$$< \phi_i, \phi_j >= 0 \quad \text{iff} \quad b_i^T W b_j = 0,$$

where $b_i$ and $b_j$ are the coefficients of the basis expansion of $\phi_i$ and $\phi_j$.

In (3.2.10) we have an eigenequation with a nonsymmetric matrix. Therefore we will apply a transformation to form an eigenequation that has a symmetric matrix, which simplifies the calculation of the eigenfunctions.

Note that $W$ is a diagonal matrix as the basis functions are orthogonal. We can set $U = W^{\frac{1}{2}}b$ then we rewrite (3.2.10) to obtain

$$\frac{1}{n}W^{\frac{1}{2}}A^TAW^{\frac{1}{2}}U = \lambda U. \tag{3.2.11}$$

Solving the eigenequation (3.2.11) we can find $U$ and then calculate $b = W^{-\frac{1}{2}}U$.

To apply the basis method we first need to choose a basis. The choice of basis will have an affect on the analysis. We will focus on two of the most popular bases; the Fourier basis and the B-Spline basis. A Fourier basis consists of sines and cosines of increasing frequencies:

$$1, \sin(\omega t), \cos(\omega t), ..., \sin(m\omega t), \cos(m\omega t), ...$$

where $\omega = \frac{2\pi}{P}$ for period $P$.

There are a few useful properties of using a Fourier basis. First, it has great computational properties when the observations are equally spaced, as Fast Fourier Transforms (FFT) are of order $O(N\log(N))$, where $N$ is the length of the time series. More details on FFT can be found in (Brigham, 1988). It is a natural choice for modelling periodic data, but can perform badly for non-periodic data.

A B-spline basis consists of polynomial segments joined at points known as knots;

the segments are optimised to ensure smoothness at the knots. In a B-Spline basis we can control the order of the polynomials, with order 3 being sufficient in most real world applications (de Boor, 2001). We can also choose the location of the knots but as the data is already discretised it makes sense to set the knots as the time points of observations.

### 3.2.3 Extensions to FPCA

In this section we discuss three extensions of the FPCA model.

#### Smooth FPCA

In classical FPCA we assume that we observe time series $x_{1:T} = [x(t_1), ..., x(t_T)]$ at time points $0 \leq t_1 < \cdots < t_T \leq 1$. However if the time series contains noise this can affect the FPCA estimates. Typically we assume the observed time series $y_{1:T} = [y(t_1), ..., y(t_T)]$ contains Gaussian noise therefore $y(t_j) = x(t_j) + \epsilon_j$ where $\epsilon_j$ is random noise with $E(\epsilon_j) = 0$ and $var(\epsilon_j) = \sigma^2$. The noise in the data effects the estimation of the covariance function, and the subsequent eigenfunctions calculated. To overcome this issue the FPCs are typically smoothed using a roughness penalty. The ridge regression approach (Rice and Silverman, 1991) uses a roughness penalty $||D^2\phi_j||$ where $D$ is the differential operator. An alternative approach by Silverman (1996), incorporates the penalty into the norm, which has been proven to be consistent and contains a number of useful properties as shown by Qi and Zhao (2011).

### Multivariate FPCA

Multivariate FPCA is an extension of FPCA for multivariate functional data. Each observation is believed to come from a multivariate stochastic process. Applying univariate FPCA for each random function doesn't capture the cross correlation between the random functions. Multivariate FPCA methods that capture this cross correlation should give better estimates of the eigenfunctions and give smaller dimensional representations.

One approach by Ramsay and Silverman (2005) concatenates the multiple functions into one function and then applies univariate FPCA, this approach assumes the variability of the different functions are similar and that they have measurements on the same units. However this approach can give poor estimates if the functions have different scales of variability. Chiou et al. (2014) calculate normalisation constants that aim to captures the cross-correlation between functions, and ensure the functions are defined on the same scale. Happ and Greven (2018) outline a multivariate Karhunen-Loéve theorem. They define a relationship between the multivariate and the univariate eigenfunctions, enabling the multivariate FPCs to be estimated easily.

### Robust FPCA

Classical estimators assume the data arises from a certain distribution or model. However if the distribution is misspecified these estimators can give poor estimates. The motivation behind robust estimators is to obtain reasonable estimates under the assumed distribution, whilst being 'robust' to small deviations from this model. Additionally, large deviation should not cause arbitrarily large errors.

There are two concepts commonly used to assess a robust estimator. First, is the efficiency which can be defined in terms of *relative efficiency* with respect to a classical counterpart, or *absolute efficiency* with respect to an underlying distribution. Second, is the breakdown point which assesses the proportion of the data that can be arbitrarily corrupted before the estimator gives arbitrarily large values.

**Definition 3.2.6** (Efficiency)**.** *Let $T_R$ and $T_C$ be unbiased robust and classical estimators respectively for the same parameter $\theta$ then the relative efficiency is given by:*

$$e(T_R, T_C) = \frac{E[(T_R - \theta)^2]}{E[(T_C - \theta)^2]} = \frac{var(T_R)}{var(T_C)}.$$

*The relative efficiency gives the ratio of variance between two estimators. The absolute efficiency is given by:*

$$e(T_R) = \frac{1/I(\theta)}{var(T_R)},$$

*where $I(\theta)$ is the Fisher Information. The absolute efficiency can be shown to be less than or equal to 1 using the Cramér-Rao bound. The absolute efficiency is simply the minimum possible variance for an unbiased estimator divided by the variance of the estimator $T_R$.*

**Definition 3.2.7** (Breakdown point)**.** *Let $x_1, ..., x_n$ be samples in the set $Z$ and $T(Z)$ is an estimator. If $m < n$ samples are corrupted, giving a corrupted set $Z'$, we can define*

$$bias(m; T, Z) = sup_{Z'}(||T(Z') - T(Z)||),$$

*where the supremum is over all possible collections $Z'$. Then the breakdown point of $T$ at $Z$ is given by*

$$\nu(T, Z) = \min\{m/n; bias(m; T, Z) = \infty\}. \tag{3.2.12}$$

*The breakdown point is used to determine the sensitivity of an estimator in the presence of partially corrupted data. It determines the maximum proportion of the data that can be corrupted before the estimator gives an arbitrarily large error.*

We defined Functional principal component analysis (FPCA), which gives the $M$-dimensional projection of the data that maximises the sample variance. The objective function of FPCA uses a square loss function, which is known to be highly influenced by outliers (Huber, 2011). In recent years robust approaches have been developed to minimise the influence of outliers. There are two approaches. The first is to use robust estimates of the covariance function, then taking the eigenfunctions of the robust covariance function (Locantore et al., 1999). An alternative approach is to use Projection Pursuit (PP) (Hyndman and Ullah, 2007; Sawant et al., 2012; Bali et al., 2011; Boente and Salibian-Barrera, 2015). The PP approach aims to find low dimensional projections of high-dimensional points which maximises a certain objective function. This approach avoids the curse of dimensionality and is able to ignore irrelevant features. However it requires a high amount of computing time. A special case of PP is PCA, which aims to find projections that maximise the variance. We will use the PP approach by Bali et al. (2011) in Chapter 6. A description of the PP approach is given below.

The objective of FPCA is to find projections that maximise the variance. These projections are shown to be the eigenfunctions of the covariance operator. Bali et al. (2011) replaces the variance with an M-estimator of scale $\tilde{\sigma}_n$. To estimate this scale value they use the Bi-square loss function:

$$\chi_c(y) = \min\{3(y/c)^2 - 3(y/c)^4 + (y/c)^6, 1\},$$

where $c$ is a tuning parameter. The M-estimator of scale $\tilde{\sigma}_n$ is then a solution to

$$\frac{1}{n}\sum_{i=1}^{n} \chi_c\left(\frac{x_i - \tilde{\mu}_n}{\tilde{\sigma}_n}\right) = \delta,$$

where $\tilde{\mu}_n$ is a robust estimator of location and $c = 1.56$ and $\delta = 0.5$ are tuning constants, to ensure Fisher-consistency at the Normal distribution with a 50% breakdown point. A re-weighting algorithm can be used to estimate $\tilde{\sigma}_n$:

$$\tilde{\sigma}_n^{(k+1)} = \frac{1}{m\delta}\sum_{i=1}^{n} w\left(\frac{x_i - \tilde{\mu}_n}{\tilde{\sigma}_n^{(k)}}\right)(x_i - \tilde{\mu}_n)^2,$$

where $w(x) = \chi_c(x)/x^2$ for $x \neq 0$.

To apply PP they use the CR algorithm by Croux and Ruiz-Gazen (1996), which applies PP for multivariate data. Bali et al. (2011) take $N$ equidistant points on each curve $x_i$ to obtain vector $\vec{x}_i$ and then apply the CR algorithm on the $\vec{x}_i$ vectors. Let $\vec{x}_i$ be location centred then at step $k-1$ the CR algorithm returns $(k-1)$-th direction $\hat{\alpha}_{k-1}$ and then update

$$\vec{x}_i^{(k)} = \vec{x}_i^{(k-1)} - (\hat{\alpha}_{(k-1)}^T \vec{x}_i^{(k-1)})\hat{\alpha}_{(k-1)},$$

for $1 \leq i \leq n$ and $k > 1$. The CR algorithm searches for the $k$-th direction considering $n$ trial directions in the set

$$A_{n,k} = \left\{ \frac{\vec{x}_1^k}{||\vec{x}_1^k||}, ..., \frac{\vec{x}_n^k}{||\vec{x}_n^k||} \right\}.$$

Then the $k$-th direction is given by

$$\hat{\alpha}_k = \underset{a \in A_{n,k}}{\arg\max}\, \tilde{\sigma}_n \left( a^T \vec{x}_1^{(k)}, ..., a^T \vec{x}_n^{(k)} \right).$$

It has been shown by Croux and Ruiz-Gazen (1996) that the M-estimator of scale has a 50% breakdown point and can obtain high levels of efficiency by decreasing the parameter $\delta$. Note that the CR algorithm can fail when the sample size $n$ is low relative to number of measurement points $N$, prompting a modified algorithm called GRID (Croux et al., 2007).

## 3.3 Functional Linear Regression

There are three types of functional linear regression models: *Scalar-on-function* - for scalar response and functional predictors, *function-on-scalar* - for functional response and scalar predictors and *function-on-function* - response and predictor are functions. In this section we will focus on the function-on-function models. A comprehensive review of each of these areas is given in Morris (2015).

In this section we will introduce the classical Functional Linear Regression model for functional responses. The classical FLR model by Ramsay and Dalzell (1991) models the relationship between predictor $x_i(t)$ and response $y_i(t)$ as:

$$y_i(t) = \alpha(t) + \int_I x_i(s)\beta(s,t)ds + \epsilon_i(t), \tag{3.3.1}$$

where $\alpha(t)$ is the intercept function, $\beta(s,t)$ is the regression function and $\epsilon_i(t)$ is the error process. For a fixed $t$, we can think of $\beta(s,t)$ as the relative weight placed on $x_i(s)$ to predict $y_i(t)$. For simplicity we will assume the mean functions $\mu^X(t) = 0$ and $\mu^Y(t) = 0$ which thereby means $\alpha(t) = 0$.

FLR in the function-on-function case can be modelled parametrically (Yao et al., 2005; Chiou et al., 2016) or nonparametrically (Ferraty et al., 2012; Ivanescu et al., 2015; Scheipl et al., 2015). The nonparametric model uses a kernel estimator. In this section we will focus on the parametric approach, which models the regression function in terms of pre-defined basis functions.

We will represent $x_i(t)$ and $y_i(t)$ in terms of $(M, K)$ pre-chosen basis functions $\phi_j^X(t), \phi_j^Y(t)$ respectively:

$$x_i^{(M)}(t) = \sum_{m=1}^{M} z_{im}\phi_m^X(t) \text{ and } y_i^{(M)}(t) = \sum_{k=1}^{K} w_{ik}\phi_k^Y(t).$$

where $z_{im}, w_{ik} \in \mathbb{R}$.

We define $\phi^X(t) = [\phi_1^X(t), ..., \phi_M^X(t)]$, $\phi^Y(s) = [\phi_1^Y(s), ..., \phi_K^Y(s)]$, $z_i^{(M)} = [z_{i1}, ..., z_{iM}]$ and $w_i^{(K)} = [w_{i1}, ..., w_{iK}]$. We will then model the regression surface using a double basis expansion (Ramsay and Silverman, 2005):

$$\beta(s,t) = \sum_{m=1}^{M}\sum_{k=1}^{K} B_{km}\phi_m^X(s)\phi_k^Y(t) = \phi^X(s)^T B^{MK}\phi^Y(t), \tag{3.3.2}$$

for an $M \times K$ regression matrix $B^{MK}$. We can then write:

$$y_i(t) = z_i^{(M)} B^{MK} \phi^Y(t) + \epsilon_i(t). \tag{3.3.3}$$

Letting $\epsilon_i(t) = q_i \phi^Y(t)$ we can reduce Equation (3.3.3) to:

$$w_i^{(K)} = z_i^{(M)} B^{MK} + q_i. \tag{3.3.4}$$

This parametrisation of the residual function is also used by Chiou et al. (2016). We can then estimate $B^{MK}$ using standard multivariate regression methods typically assuming Gaussian $q_i$.

We have shown the FLR problem can be reduced into a LR problem with multiple responses. Typically the FPCA basis for $X$ and $Y$ is chosen in the FLR problem. This ensures only a small number of basis functions are required and can help obtain consistency results. Chiou et al. (2016) use a standard Least Squares estimator, which they prove to be consistent.

### 3.3.1 Historical FLR

In the classical FLR model (3.3.1) we integrate over all time points. However we may want to make predictions using only past time points. For example in an engine test the current engine behaviour should only depend on the previous engine behaviour. The historical FLR model by Malfait and Ramsay (2003) looks at this problem in the general setting. The model incorporates a lag threshold $\delta$, which imposes that values more than $\delta$ time units back will have no effect in the regression model. Let $s_0(t) = \max\{0, t - \delta\}$ then the historical FLR model is given by:

$$y_i(t) = \int_{s_0(t)}^{t} x_i(s)\beta(s,t)ds + \epsilon_i(t), \text{ for } t \in [0,1], \tag{3.3.5}$$

Let $\theta(s,t) = (\theta_1(s,t), ..., \theta_K(s,t))$ be $K$ basis functions, which we will use to approximate the regression function $\beta(s,t)$:

$$\hat{\beta}(s,t) = B\theta(s,t), \tag{3.3.6}$$

where $B$ is a $K$-dimensional vector of coefficients. We can then define

$$\Psi(t) = \int_{s_0(t)}^{t} x(s)\theta(s,t)ds \tag{3.3.7}$$

where $x(s) = (x_1(s), ..., x_n(s))$. We can then formulate the problem as

$$y_i(t) = \sum_{k=1}^{K} B_{ik} \int_{s_0(t)}^{t} x_i(s)\theta_k(s,t)ds + \int_{s_0(t)}^{t} x_i(s)\epsilon_a(s,t)ds + \epsilon_i(t) = \sum_{k=1}^{K} B_{ik}\Psi_{ik}(t) + \epsilon_i'(t), \tag{3.3.8}$$

where $\epsilon_a(s,t) = \beta(s,t) - \hat{\beta}(s,t)$ is the approximation error and $\epsilon_i'(t)$ is the residual error. Optimal $B$ will be a solution to

$$\int_0^1 \Psi^T(t)\Psi(t)dt \cdot B = \int_0^1 \Psi^T(t)y(t)dt \tag{3.3.9}$$

which is evident from Equation (3.3.8). Malfait and Ramsay (2003) find an approximate solution to Equation (3.3.9) by using a finite elements method over a finite grid of points.

This model requires a certain type of basis function making it less flexible than the classical FLR model. Furthermore the classical FLR model can be reduced to a

LR problem and uses a potentially small number of basis functions. On the other hand the historical FLR model requires finite elements methods to be used, which scale with the size of the data. We wanted to use this model in our regression model in Chapter 6, but were unable to work around these limitations.

A special case of the Historical FLR model, is the Concurrent Functional dependent variable model (CFDV) (Ramsay and Silverman, 2005), which considers function on function dependence, where the response function at time $t$ only depends on the predictor functions at time $t$. Under the CFDV model the functions are assumed to have the following relationship:

$$y_i(t) = \beta(t)x_i(t) + \epsilon_i(t). \tag{3.3.10}$$

This model is more general than a linear regression model as the regression function $\beta(t)$ is a function of time. However the model is unable to capture temporal relationships across time unlike the classical FLR model.

### 3.3.2 Model Selection for FLR

The FLR model relies on parameters $M$ and $K$, there are a number of ways to choose these terms when we use FPCA bases. Chiou et al. (2016) choose the number of components that capture 95% of the variance. This is a commonly used rule of thumb in the FPCA literature (Shang, 2014). However the estimation of $\beta(s,t)$ also depends on $M$ and $K$ and therefore should be incorporated into the choice of these terms. Yao et al. (2005) outline two ways of estimating $M, K$. The first is a leave-

one-curve-out cross validation approach. The second suggestion is an AIC criterion. However both methods focus on $X$ and $Y$ individually. Matsui (2017) suggests a Bayesian Information Criterion (BIC) to choose $M, K$. Matsui (2017) outlines a BIC procedure for the Quadratic FLR, which is an extension of the FLR model containing an additional quadratic term. In Section 6.2 we use the formulation by Matsui (2017) to determine a BIC model selection procedure for the FLR problem and give a robust BIC extension.

## 3.4 Functional Depth

Depth is a non-parametric tool for making inferences of multivariate data (Zuo and Serfling, 2000). Depth functions order a set of data, which can be used to determine quantiles. The idea has been extended to order functional data (Nieto-Reyes and Battey, 2016). We will use depth in Chapter 7 to identify outliers and as a classification tool in Chapter 8.

Depth functions order a set of data points with respect to the underlying probability distribution. The depth function gives a *centre-outward* sorting. Points close to the centre of the data distribution are given a higher depth, and points farther away are given a lower depth. However this ordering does not consider the direction, so two points equidistant from the centre but in opposite directions are given the same depth value.

The first and most intuitive depth function for multivariate data was the Halfspace depth (HD), introduced by Tukey (1974). The HD assigns a depth value to a point $z$

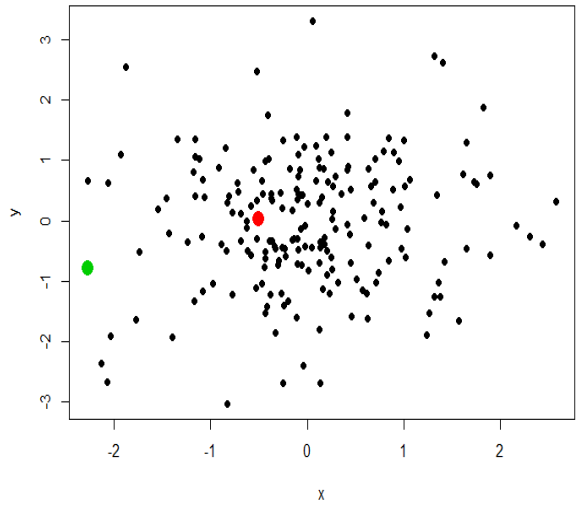



Figure 3.4.1: Scatter plot of samples from a multivariate normal distribution, with point in red closer to the centre than the green point.

with respect to samples $x = (x_1, ..., x_n)$ by determining a hyperplane that splits the point $z$ from the majority of samples $x$. The depth is then given by the number of points that lie within the halfspace containing $z$. More formally, let $x_1, ..., x_n \in \mathbb{R}^k$ be samples of a random variable $X$ with cumulative distribution function $F_x$ then the Halfspace depth for a sample $z \in \mathbb{R}^k$ is given by

$$HD(z, F_x) = \frac{1}{n} \min_{u \in \mathbb{R}^k, ||u||=1} \#\{x_i, i = 1, ..., n : u^T x_i \geq u^T z\}. \tag{3.4.1}$$

In Figure 3.4.1, we have an example of data samples from a bivariate Gaussian distribution. We can see that the sample highlighted in green will have a small depth value and the sample in red is closer to the centre so will have larger depth. In this scenario the idea of depth is very intuitive.

Zuo and Serfling (2000) outline 4 properties for a multivariate depth function,

which have been extended by Nieto-Reyes and Battey (2016) to give a statistical definition of depth for functional data. They state a functional depth should satisfy 6 properties.

**Definition 3.4.1** (Functional Depth)**.** *Let $B$ be a Borel $\sigma$-algebra of a measure space $H$ over sample space $\Omega$. We assume there exists a metric $d$ such that $(H, d)$ is a separable metric space. As in Section 4.4 we will work in the Hilbert space defined on the unit interval: $H = L^2([0,1])$. The random variable $X : (\Omega, B) \to H$ has a corresponding probability measure $P_X$. Let $P$ be the space of all probability measures on $B$, then for $z \in H$ the function $D(\cdot, \cdot) : H \times P \to \mathbb{R}$ is a statistical functional depth if*

$$z \mapsto D(z, P_X) \in \mathbb{R},$$

*satisfy the following 6 properties:*

1. (Distance Invariance) $D(f(x), P_{f(X)}) = D(x, P_X)$ for any $x \in H$ and $f : H \to H$ such that $d(f(x), f(y)) = a_f d(x, y)$ for any $y \in H$ and $a_f \in \mathbb{R}$. - This property states that depth does not change up to a scaling factor. For example if the functions are in Degrees Fahrenheit rather than Celsius, the depth values should remain the same.

2. (Maximality at centre) For any $p \in P$ which contains a unique centre of symmetry $\theta \in H$. This property states there exists a deepest point.

3. (Strictly decreasing with respect to the deepest point) For any $p \in P$ such that $D(z, p) = \max_{x \in H} D(x, p)$ exists, $D(x, p) < D(y, p) < D(z, p)$ holds for any

$x, y \in H$ such that $\min\{d(y,z), d(y,x)\} > 0$ and $\max\{d(y,z), d(y,x)\} < d(x,z)$. This condition ensures that samples that belong to successively larger envelopes around the deepest point, are assigned smaller depth values.

4. (Upper semi-continuity in $x$) $D(x,p)$ is upper semi-continuous as a function of $x$. In other words, for all $x \in H$ and $\epsilon > 0$ there exists a $\delta > 0$ such that $\sup_y D(y,p) \leq D(x,p) + \epsilon$ where $y$ satisfies the condition $d(x,y) < \delta$. This is a technical condition, based on the fact that each depth is linked to a cumulative distribution function.

5. (Receptivity to convex hull width across the domain) $D(x, P_X) < D(f(x), P_{f(X)})$ for any $x \in C(H,p)$ with $D(x,p) < \sup_y D(y,p)$ and $f : H \to H$ such that $f(y(v)) = \alpha(v)y(v)$ for a certain $\alpha(v) \in (0,1)$, where $C(H,p)$ is the convex hull of $h$ with respect to $p$ defined in Nieto-Reyes and Battey (2016). There may be subsets of the interval $I$, where the functions exhibit little variability. This can lead to different ranking arising from measurement error. The condition is that the depth function gives more weight to regions of $I$ with more variability when assigning depth.

6. (Continuity in $p$) For all $x \in H$, $p \in P$ and $\epsilon > 0$ there exists $\delta(\epsilon) > 0$ such that $|D(x,q) - D(x,p)| < \epsilon$ $p$-almost surely for all $q \in P$ with $d_P(q,p) < \delta$ $p$-almost surely, where $d_P(\cdot,\cdot)$ metricises the topology of weak convergence. This condition ensures that asymptotically the empirical depth converges to the population depth.

Nieto-Reyes and Battey (2016) suggest using these properties to choose the depth

functions we use. Gijbels and Nagy (2017) highlight that these conditions can be restrictive and unattainable for depth functions in practice and offer alternative conditions. Next we give some examples of functional depths from $L^2([0,1])$ to $\mathbb{R}$.

### 3.4.1 Fraiman Muniz depth

The Fraiman Muniz (FM) depth by Fraiman and Muniz (2001), takes the empirical distribution $F_{n,t}$ for sample $x_1(t), ..., x_n(t)$ and calculates the depth at time $t$ as $D(z(t)|x(t)) = 1 - |0.5 - F_{n,t}(z(t))|$. Then the overall depth for $z$ is given by

$$I = \int_0^1 D(z(t)|x(t))dt. \tag{3.4.2}$$

### 3.4.2 Random Projection depth

Cuevas et al. (2007) outlines a random projection (RP) approach. In the RP approach a random function $a$ is used to project the functions $x_i$:

$$< a, x_i >= \int_0^1 a(t)x_i(t)dt.$$

The projected values can be sorted using order statistics, which gives the depth value with respect to projection $a$. They apply multiple projections then suggest averaging over the depth values from each of the projections, to obtain the RP depth. Random projections are an effective dimensionality reduction approach, which has been used effectively in many applications. However in this context it is unclear whether the RP depth satisfies the properties of a functional depth.

### 3.4.3 h-modal depth

Cuevas et al. (2007), outlines the $h$-modal depth. For a Gaussian kernel $G$ with bandwidth $h$, the $h$-modal depth $D(z|x,h)$ is given by

$$D(z|x,h) = E(G||z-x||) \approx \frac{1}{n}\sum_{i=1}^{n} G(||z-x_i||). \tag{3.4.3}$$

We will define $||\cdot||$ as the standard norm in $L^2$. They suggest taking the bandwidth $h$ to be the 15th percentile of the empirical distribution of $\{||x_i - x_j||, i,j = 1,...,n\}$. Note that we are not trying to estimate the density, but the support so could use a range of values of $h$ as long as they are not too small. Nagy (2015) has proven consistency results for the $h$-modal depth in the general case of Banach-valued data. Nieto-Reyes and Battey (2016), shows that the h-modal depth satisfies condition 2 to 6 but not condition 1.

### 3.4.4 Band Depth

The Band depth (BD) was introduced by López-Pintado and Romo (2009), which intuitively states a function $z$ is central with respect to $P$ if $z$ is contained with high probability inside the envelope of $j$ copies of $X$.

Let the band:

$$B(x_1,...,x_n) = \left\{(t,y) : t \in I, \min_{i=1,...,n} x_i(t) \leq y \leq \max_{i=1,...,n} x_i(t)\right\}.$$

We define $S_j = \{w : w \subset \{x_1,...,x_n\}, |w| = j\}$ as the set of all subsets of $\{x_1,...,x_n\}$ of size $j$. Then the Band depth is given by:

$$BD(z|x_1, ..., x_n) = \sum_{j=2}^{J} \binom{n}{j}^{-1} \sum_{w \in S_j} 1\{z \subset B(w)\}. \tag{3.4.4}$$

This essentially counts the number of times $z(t)$ crosses each possible set of bands. The number of bands $J$ is preselected and is typically taken to be 2 or 3 to minimise the computational cost. For frequently crossing data, the BD values will be low. Therefore a Modified Band Depth (MBD) was outlined, which uses a count function:

$$A(z, x_1, ..., x_n) = \left\{ t \in I : \min_{i=1,...,n} x_i(t) \leq z(t) \leq \max_{i=1,...,n} x_i(t) \right\}, \tag{3.4.5}$$

to give the MBD:

$$MBD(z|x_1, ..., x_n) = \sum_{j=2}^{J} \binom{n}{j}^{-1} \sum_{w \in S_j} \frac{\lambda(A(z, w))}{\lambda(I)}. \tag{3.4.6}$$

where $\lambda$ is a Lebesgue measure. The MBD gives the proportion of times $z(t)$ is outside the bands. Nieto-Reyes and Battey (2016) shows that the BD and MBD do not satisfy conditions 3 and 5.

### 3.4.5 Multivariate Functional Depth

The multivariate Functional Depth developed by Claeskens et al. (2014), uses the Tukey halfspace depth to build a depth function for multivariate functional data. Let $D(\cdot)$ be the Halfspace depth function (3.4.1) defined in $\mathbb{R}^k$. Then the multivariate functional depth for $z$ with respect to the observed curves $x_1, ..., x_n$ is defined as:

$$MFD(z|x_1, ..., x_n) = \int D(x_1(t), ..., x_n(t))w(t)dt, \tag{3.4.7}$$

where $w(t)$ is a weight function. The weight function can be chosen using prior knowledge about the data or can be chosen using the depth values. In practice we observe the curves at discrete time points $t_1, ..., t_N$. Therefore the depth values are calculated independently at each time point.

### 3.4.6 Other Depth functions

There are a number of other depth functions including the Random Tukey Depth (RTD) (Cuesta-Albertos and Nieto-Reyes, 2007), Spatial depth (Chakraborty and Chaudhuri, 2014), Halfregion depth (López-Pintado and Romo, 2011), Extremal depth (Narisetty and Nair, 2016) and the functional Tukey depth (Dutta et al., 2011).

## 3.5 Outlier Detection for Functional Data

A number of approaches have been developed to identify outliers for functional data. The problem is challenging due to the range of outliers that can arise. Hubert et al. (2015) define five types of outliers in functional data. The first are *isolated outliers* that are abnormal in a small region of the function and second there are *persistent outliers* that effect the function over a large region. *Shift outliers* have a similar shape but have been shifted along the time-axis. *Shape outliers* are not necessarily abnormal at each time point but seen collectively, can be highlighted as abnormal. Finally, there are *amplitude outliers* have the same shape but a shift in scale.

Most outlier detection methods for functional data use functional depth. Febrero-Bande et al. (2008) use functional depth directly and identify outliers by identifying

samples with depth value below a threshold. We will describe this approach in Section 3.5.1. An alternative method is to build a Functional Boxplot (FB) (Sun and Genton, 2011) using the Band depth (López-Pintado and Romo, 2009), then as in classical boxplot samples that lie outside 1.5 times the quantiles are labelled as outliers. We describe this approach in Section 3.5.2. Alternatively we can use methods based on outlyingness measures such as the Outliergram by Arribas-Gil and Romo (2014) described in Section 3.5.3. An outlyingness measure can be extended to multivariate functional data (Dai and Genton, 2018a). The Functional Outlier Map (FOM) by Rousseeuw et al. (2018) forms a scatter plot of two outlyingness measures, which we will describe in Section 3.5.4.

In Chapter 7 we will introduce an outlier detection framework for functional data. We will compare our framework to these standard approaches.

### 3.5.1 Direct approach

Febrero-Bande et al. (2008) use functional depth (described in Section 3.4) to identify outliers in functional data. The approach assigns a depth value to samples $r_i(t)$. Samples with small depth values lie far away from the other samples. Curves with a depth value below a certain threshold are then labelled as outliers. They then discard the outliers and using the rest of the curves they recalculate the depth values excluding the outliers, this deals with possible masking effects. The threshold $C$ is chosen such that $P(D(r_i|r,h) \leq C) = \delta$, where $\delta$ is a pre-chosen percentile typically taken to be 0.01. To estimate the threshold $C$ they use a bootstrapping approach, which estimates a value of $C$ for different random sets of samples and then aggregates

these estimates. We call this approach Direct as it uses a threshold directly on the depth values. We will compare our outlier detection approach outlined in Chapter 7 to the Direct approach given in Algorithm 1.

**Algorithm 1** Direct Approach

1: **INPUTS:** Curves $r = \{r_1, ..., r_n\}$, number of bootstraps $v$ of size $k < n$ and percentile $\delta$,
2: **for** $i = 1 : n$ **do**
3: Calculate depth value $d_i = D(r_i|r)$
4: **end for**
5: Set bandwidth $h$ be 15% percentile of depth values $d$
6: **for** $j = 1 : v$ **do**
7: Take a subset of $k$ samples $V_j$ from $\{r_1, ..., r_n\}$
8: Calculate depths for samples in $V_j$ then take $C_j$ to be equal to $\delta$ percentile
9: **end for**
10: Estimate $C = \frac{1}{v}\sum_{j=1}^{v} C_j$
11: Set $r^* = r$ and $counter = 0$
12: **for** $r_i$ in $r^*$ **do**
13: **if** $D(r_i|r^*, h) < C$ **then**
14: Sample $i$ is labelled as an outlier.
15: $r^* = r^* \backslash r_i$ and $counter = counter + 1$.
16: **end if**
17: **end for**
18: **if** $counter > 0$ **then**
19: **go to** Step 11
20: **end if**
21: **RETURN:** List of outliers and depth values $d$.

### 3.5.2 Functional Boxplot

Sun and Genton (2011) outline a Functional Boxplot (FB), which uses the Band depth described in Section 3.4. The median function is taken to be the "deepest" curve i.e. the sample that has the largest depth value. To determine the quantiles we will first define the $\alpha$-central region of data $C_\alpha(X)$ i.e. the region containing the $\alpha$% most

central observations:

$$C_\alpha(X) = \left\{ (t, z(t)) : \min_{l=1,\ldots,\lceil \alpha n \rceil} x_{(l)} \leq z(t) \leq \max_{r=1,\ldots,\lceil \alpha n \rceil} x_{(r)}(t) \right\}.$$

For the functional boxplot we compute the region $C_{0.5}$, which contains 50% of the most central curves. The quantile curves of the region $C_{0.5}$ can be found using the depth values. To identify outliers they define fences by inflating the quantile curves of $C_{0.5}$ by a factor of 1.5. Observations that cross or lie outside the fences are then labelled as outliers. The Functional Boxplot gives a good visualisation of the data but is not effective in identifying isolated or shape outliers as shown by Dai and Genton (2018b). Examples of Functional Boxplots are given in Figure 7.3.2.

### 3.5.3 Outliergram

The outliergram by Arribas-Gil and Romo (2014) uses two measures. The first is the Modified Band Depth (MBD) defined in Section 3.4. The MBD for a curve $x_k$ with respect to a set of curves $x_1, ..., x_n$ will be denoted by $b_k = MBD(x_k|x_1, .., x_n)$. The second score is the Modified Edigraph Index (MEI), which for a sample $x_k$ with respect to $x_1, ..., x_n$ is given by:

$$e_k = MEI(x_k|x_1, ..., x_n) = \frac{1}{n} \sum_{i=1}^{n} \frac{\lambda(\{t \in I | x_i(t) \geq x_k(t)\})}{\lambda(I)},$$

for a Lebesgue measure $\lambda$ on $\mathbb{R}$. The $e_k$ gives the mean proportion of time $x_k$ lies below all the other sample curves.

If sample $x_k$ has an MEI value $e_k$ close to 0.5 then the curve is located in the centre. However if the MBD value $b_k$ is small this would indicate $x_k$ is a shape outlier

as the sample is only contained in a small number of bands.

Arribas-Gil and Romo (2014) show that

$$b_k = a_0 + a_1 e_k + a_2 \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\lambda(E_{ik} \cup E_{jk})}{\lambda(I)},$$

where $E_{ik} = \{t \in I | x_i(t) \geq x_k(t)\}$, for certain values of $a_0, a_1, a_2 \in \mathbb{R}$. This relationship shows the points $(b_i, e_i)$ should lie on a parabola. Using the distance $d_i = a_0 + a_1 e_i + n^2 a_2 e_i^2 - b_i$ they use a univariate boxplot rule to determine lower thresholds $D_1 - 1.5 \times IQR$ and upper threshold $D_3 + 1.5 \times IQR$ where $D_1$ and $D_3$ are the first and third quantiles respectively and $IQR$ is the interquartile range of the distances $d_1, ..., d_n$. The points $(b_i, e_i)$ are shifted down by the threshold $D_3 + 1.5 \times IQR$, and the scatter plot of the shifted values forms the outliergram. Examples of outliergrams is given in Figure 7.3.2.

### 3.5.4 Functional Outlier Map

The Functional Outlier Map (FOM) by Rousseeuw et al. (2018) uses directional outlyingness measures to identify outliers. The FOM map tries to identify the 'average' and 'variance' outlyingness of a sample with respect to a set of data. They suggest a scatter plot of two measures to identify outliers. They use the Functional Directional Outlyingness and the variability of the Directional Outlyingness.

The Directional Outlyingness (DO) at time $t$ is given by:

$$DO(x_i(t), x(t)) = \begin{cases} \frac{x_i(t) - med(x(t))}{s_a(x)}, \\ \\ \frac{med(x(t)) - x_i(t)}{s_b(x)}, \end{cases} \tag{3.5.1}$$

where $s_a(x)$ and $s_b(x)$ are M-estimators of scale above and below the median $med(x(t))$ respectively.

The Functional DO (FDO) is given by:

$$FDO(x_i, x) = \int_I DO(x_i(t), x(t)) w(t) dt \tag{3.5.2}$$

where $w(\cdot)$ is a weight function with the condition $\int_I w(t) = 1$. The FDO of a function $x_i$ can be considered the 'average outlyingness' of its functional values. In practice the function $x_i$ is observed at time points $t_1, ..., t_T$, then the discrete version of FDO is given by:

$$FDO_T(x_i, x) = \sum_{j=1}^{T} DO(x_i(t_j), x(t_j)) w(t_j). \tag{3.5.3}$$

The variability of the DO values is then given by:

$$VDO(x_i, x) = \frac{stdev_j(\{DO(x_i(t_j), x(t_j)), j = 1, ..., T\}}{1 + FDO_T(x_i, x)}). \tag{3.5.4}$$

The Functional Outlier Map (FOM) is the scatter plot of the points $(VDO_d(x_i, x), VDO(x_i, x))$ for $i = 1, ..., n$.

Defining the combined functional outlyingness (CFO) of $x_i$ as

$$CFO_i = CFO(x_i, x) = \sqrt{(FDO(x_i, x)/med(FDO_T))^2 + (VDO(x_i, x)/med(VDO))^2}, \tag{3.5.5}$$

where

$$med(FDO_T) = med(FDO_T(x_1, x), ..., FDO_T(x_n, x)),$$

$$med(VDO) = med(VDO(x_1, x), ..., VDO(x_n, x)).$$

Let $LCFO_i = \log(0.1 + CFO_i)$ then the function $x_i$ is flagged as an outlier if

$$\frac{LCFO_i - med(LCFO)}{MAD(LCFO)} > \Phi^{-1}(0.995). \tag{3.5.6}$$

This threshold can be seen as the functional version of the threshold used for multivariate data. Examples of Functional Outlier Maps are given in Figure 7.3.2.

# Chapter 4

# Classification of manoeuvres in a Pass-Off test

## 4.1 Introduction

In Chapter 1 we have described the Pass-Off test and the manoeuvres that are performed in the test. The N1 speed profiles (described in Chapter 1) for two Pass-Off tests are given in Figure 4.1.1, with labelled manoeuvres. We can see tests can differ due to engine stops and manoeuvre repeats. Surprisingly the manoeuvres are not labelled. We have therefore built a classification algorithm that is able to extract and label the manoeuvres with almost perfect accuracy. The algorithm is computationally efficient given the large volume of sensor data generated during the engine tests. The labels can be used to highlight problematic tests, for example where a large number of manoeuvre repeats have been performed. These tests can be investigated further by the engineers. We also noted that the novelty detection algorithms outlined in

Chapter 2 use Pass-Off test data without consideration of the large differences between the tests. Therefore comparisons between the tests can be unreliable. On the other hand the manoeuvres are generally consistent between tests meaning novelty detection for specific manoeuvres can give more reliable models. The classification algorithm can be split into three main parts: manoeuvre extraction; feature extraction using Needleman-Wunsch and Functional Principal Component Analysis and classification using either a decision tree or Linear Discrimant Analysis classifier.

In the classification algorithm we will use the N1 speed time series as the manoeuvres have distinctive speed profiles. The N1 speed is primarily piecewise linear. We can therefore use the Pruned Exact Linear Time (PELT) changepoint algorithm (Killick et al., 2012) to identify changes in speed. Using the fact that a manoeuvre starts and ends at idle speed, changepoints preceded or acceded by an idle speed segment can be used as indicators for the start and end of a manoeuvre. In Section 4.2.1 we will describe the PELT algorithm and explain how we can use the algorithm to extract the manoeuvres.

The labelling of the Pass-Off test manoeuvres is a time series classification problem, in which there are two standard approaches (Susto et al., 2018). First, there are *Feature-based* methods where features are calculated and used as inputs into a classifier. Second, we can use *distance-based* methods which typically use a distance measure such as Dynamic Time Warping (DTW) (Senin, 2008). A standard approach is to compare an unlabelled time series to some pre-labelled time series and then classify using 1-nearest neighbour. DTW is computationally inefficient as it is of the order $O(MN)$ for two time series of length $M$ and $N$. We adopt the first approach, which

focuses on constructing informative features.

In Chapter 1 we have outlined templates for the piecewise linear manoeuvres. The templates are the fixed speed levels the manoeuvre must reach. The Performance Curve (P) manoeuvre occasionally does not match its template, because engineers may sometimes add or miss out steps. For example in Figure 4.1.1 we have the N1 speed plots for Pass-Off tests 18 and 21, which contain P manoeuvres that do not match the template. We want a distance measure to compare an extracted manoeuvre against each of the templates, and we require that it copes with missing steps. We therefore use the Needleman Wunsch (NW) algorithm, which finds the optimal alignment between two sequences that may contain potential gaps. The NW algorithm gives a similarity score corresponding to the alignment. The standard NW algorithm and a probabilistic alternative will be described in Section 4.3.

We have defined a manoeuvre as a segment of engine running that starts and ends at idle speed. However sometimes the 'Running and Handling' manoeuvre labelled as (R) in Figure 1.1.3, does not end at idle speed. We will therefore create another manoeuvre that combines the 'Running and Handling' with the 'Performance curve' labelled as (RP). Now all manoeuvres start and end at idle speed.

There are two manoeuvres, called the Fast Acceleration/Deceleration (F) and the Vibration Survey (V), which do not have fixed speed levels. For these two manoeuvres we use Functional Principal Component Analysis (FPCA) to build templates as described in Section 3.2. A manoeuvre can then be modelled with respect to the FPCA representations. We will use the difference between the manoeuvre profile and the FPCA representations, as features in the classification algorithm.

Below we have listed the various manoeuvres in a Pass-Off test, with the corresponding colouring, labels and templates we will use in the classification algorithm.

A (26)

B (51)

C (86, 80, 66, 52)

R Running & Handling (86, 26, 86)

P Performance Curve (96, 90, 86, 79, 72, 60, 51, 38, 27)

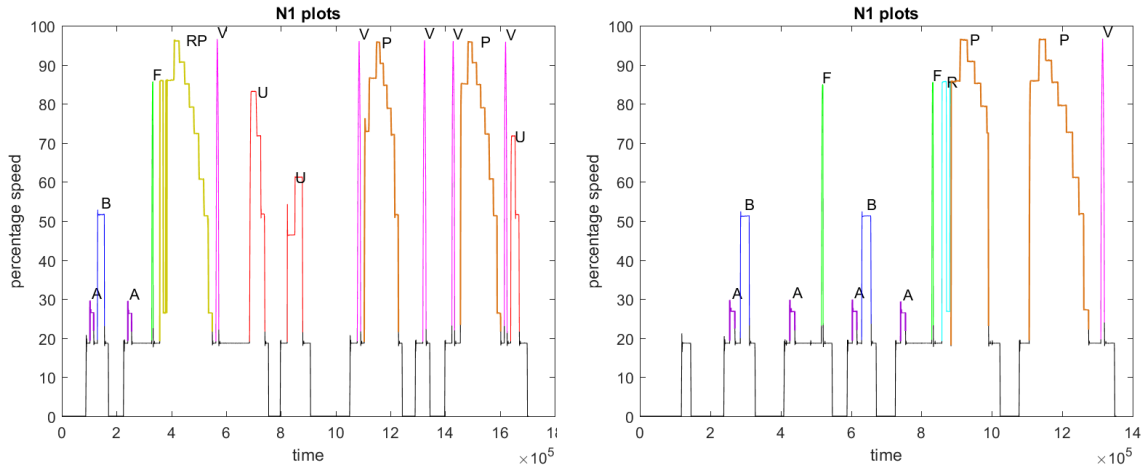
RP Running & Handling/Performance Curve (86, 26, 86, 96, 90, 86, 79, 72, 60, 51, 38, 27)

F Fast Acceleration/Deceleration

V Vibration Survey

U Unknown.

For each manoeuvre we will obtain NW scores with respect to each of the piecewise linear manoeuvres and FPCA scores with respect to manoeuvres F and V. These scores will be used as inputs for a classifier. We need a training set to build the FPCA representations and train the classifiers. The true classifications have been obtained by manually labelling manoeuvres. We will consider two classifiers, the first is a standard decision tree (Rokach and Maimon, 2005). The second classifier uses Linear Discriminant Analysis to fit a Gaussian model for each class. For the Unknown manoeuvres we set an uninformative prior Gaussian distribution with a significantly




(a) Pass-Off test 18

(b) Pass-Off test 21

Figure 4.1.1: Labelled N1 speed plots for Pass-Off test 18 (left) and 21 (right).

larger variance than in the other classes. The large variance ensures manoeuvres that don't match any of the pre-defined manoeuvres will be labelled as Unknown.

To train and test the classification algorithm we will use the 93 Pass-Off tests we have been given from Trent 1000 engines. Using $k$-fold cross-validation we will assess the classification performance of the models. We shall also highlight insights that can be made using the labels. This approach is general enough to be applied to other engine types as we will demonstrate on XWB engine Pass-Off tests in Section 4.7.

## 4.2 Manoeuvre Extraction

### 4.2.1 Changepoints

A changepoint is defined as a time point where the statistical properties of the time series before and after this time point are different. We will describe the Pruned

Exact Linear Time (PELT) changepoint algorithm in Section 4.2.1, which we will use to find changes in the piecewise linear structure of the N1 speed time series. Using the changepoints we can extract the manoeuvre segments and filter out the fixed speed segments within each manoeuvre.

Let $y_{1:T} = (y(t_1), ..., y(t_T))$ be a time series, which contains $m$ changepoints $\tau_{1:m} = (\tau_1, ..., \tau_m)$ where $\tau_0 = 0$ and $\tau_{m+1} = T$. We have $m+1$ segments, where each segment $i$ contains points $y_{(\tau_{i-1}+1):\tau_i}$. We assume the points in each segment are sampled from different distributions. For each stationary segment of the time series (between consecutive changepoints), we want to estimate a statistical model. The problem is we don't know the location of the changepoints. We can estimate the number and location of the changepoints as a solution to the following optimisation problem:

$$\min_{m} \min_{1 \leq \tau_1 < ... < \tau_m \leq T-1} \sum_{i=1}^{m+1} [C(y_{(\tau_{i-1}+1):\tau_i})] + \beta m, \tag{4.2.1}$$

where $C$ is the negative log-likelihood function associated with the statistical model we want to estimate and $\beta$ is a penalty to stop overfitting. This penalty determines the trade off between model accuracy and complexity. There are two main model selection tools. The first is the Akaike Information Criterion (AIC) penalty (Akaike, 1998), which sets $\beta = 2p$ where $p$ is the number of parameters to estimate in the cost function $C$. The AIC penalty is the same irrespective of the length of the time series, and is known to overestimate the number of changepoints. The Bayesian Information Criterion (BIC) penalty (Schwarz, 1978) which sets $\beta = p \log(T)$, considers both the number of parameters $p$ and the length of the time series $T$. The BIC penalty can

underfit the data, but is typically the preferred choice in the changepoints literature, and is used by Killick et al. (2012).

**Pruned Exact Linear Time algorithm**

We can use dynamic programming to solve the problem in (4.2.1). The Pruned Exact Linear time (PELT) (Killick et al., 2012) approach, is a modification of the Optimal Partitioning algorithm by Jackson et al. (2005), in which they have added a pruning step to improve computational efficiency.

Define $F(s, m')$ as the minimum of (4.2.1) with respect to the changepoints $\tau_{1:m'}$ for data $y_{1:s}$ with a fixed number of changepoints $m'$. We define

$$F(s) = \min_{m'} F(s, m') \tag{4.2.2}$$

then for $t < s$ we have the following recursive relationship

$$\begin{aligned} F(s) &= \min_{1<\tau_1<\ldots<\tau_{m'}<\tau_{m'+1}=s} \left\{ \sum_{i=1}^{m'+1} [C(y_{(\tau_{i-1}+1):\tau_i}) + \beta] \right\} \\ &= \min_{t} \left\{ \min_{1<\tau_1<\ldots<\tau_{m'}=t} \sum_{i=1}^{m'} [C(y_{(\tau_{i-1}+1):\tau_i}) + \beta] + C(y_{(t+1):s}) + \beta \right\} \\ &= \min_{t} \left\{ F(t) + C(y_{(t+1):s}) + \beta \right\} \end{aligned} \tag{4.2.3}$$

In (4.2.3) we have defined $F(s)$ with respect to $F(t)$, conditional on the fact that $t$ is the optimal location of the last changepoint in the time series $y_{1:s}$. Optimal Partitioning uses the recursion (4.2.3) to build a dynamic programming algorithm to find $F(s)$ for $s = 1, ..., T$. The overall computational complexity of Optimal Partitioning is $\mathcal{O}(T^2)$. The PELT algorithm adds a pruning step to Optimal Partitioning, which can reduce the computational complexity to $\mathcal{O}(T)$. Rather than minimising

over all $t$ in (4.2.3), we minimise over a subset of time points, chosen using a pruning condition. Assuming there exists some constant $K$ such that for all $t < s < t^*$,

$$C(y_{(t+1):s}) + C(y_{(s+1):t^*}) + K \leq C(y_{(t+1):t^*}),$$

If

$$F(t) + C(y_{(t+1):s}) + K \geq F(s).$$

then at a future time $t^* > s$, $t$ will never be the optimal last changepoint prior to $t^*$. Using this condition we can introduce a pruning step, which enables us to optimise over a subset of points, as we know certain points cannot be changepoints by this condition. In the worst case there is no pruning and we get Optimal Partitioning. Pruning will obviously decrease computations as the number of terms to minimise over decreases. It has been shown to get linear computational complexity, when using a cost function $C$ equal to the negative log-likelihood, where the constant $K = 0$.

**Changes in Regression**

In this section we will show how PELT can be used to find the changepoints in the N1 speed time series. We will use these changepoints to extract manoeuvre segments, and the fixed speed segments within the manoeuvres. The N1 speed time series is piecewise linear. To apply PELT on a piecewise linear time series we need a suitable cost function $C$. Assume we have a time series $y_{1:T}$ with time index $t_{1:T}$ and changepoints $\tau_{1:m}$. In each segment $i$ we have a pair of coefficients $\alpha_0^{(i)}, \alpha_1^{(i)} \in \mathbb{R}$ such that

$$y_j \sim N(\alpha_0^{(i)} + \alpha_1^{(i)} t_j, \sigma^2), \quad \text{if } y_j \text{ is in segment } i.$$

Under this model we consider two possible cost functions. Assuming we have a constant variance $\sigma^2$, the maximum log-likelihood cost function is given by the Residual Sum of Squares (RSS) (4.2.4). If we assume the variance $\sigma^2$ can change, and therefore needs to be estimated, we get the maximum log-likelihood cost function given in Equation (4.2.9).

For a segment $y_{1:s}$, we apply Ordinary Least Squares with respect to the index $t_{1:s}$, we then get an estimate of the intercept $\hat{\alpha}_0$ (4.2.7) and the slope $\hat{\alpha}_1$ (4.2.6), we can then write the RSS cost function as

$$C(y_{1:s}) = \sum_{i=1}^{s} \{y_i - (\hat{\alpha}_0 + \hat{\alpha}_1 t_i)\}^2. \tag{4.2.4}$$

If we assume the variance $\sigma^2$ can change, we get the second cost function, given by the log-likelihood

$$l(\alpha_0, \alpha_1, \sigma^2) = -\frac{s}{2}\log(2\pi) - s\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{s}\{y_i - (\alpha_0 + \alpha_1 t_i)\}^2 \tag{4.2.5}$$

the maximum likelihood estimators (MLEs) can be shown to be

$$\hat{\alpha}_1 = \frac{\sum_{i=1}^{s}(t_i - \bar{t})(y_i - \bar{y})}{\sum_{i=1}^{s}(t_i - \bar{t})^2} \tag{4.2.6}$$

$$\hat{\alpha}_0 = \bar{y} - \hat{\alpha}_1 \bar{t} \tag{4.2.7}$$

$$\hat{\sigma}^2 = \frac{1}{s}\sum_{i=1}^{s}(y_i - (\hat{\alpha}_0 + \hat{\alpha}_1 t_i))^2 \tag{4.2.8}$$

where $\bar{t}$ and $\bar{y}$ are the means of $t_{1:s}$ and $y_{1:s}$ respectively. Applying the MLEs we get a maximum log likelihood

$$l(\hat{\alpha}_0, \hat{\alpha}_1, \hat{\sigma}^2) = -\frac{s}{2}\log(2\pi) - s\log(\hat{\sigma}) - \frac{s}{2} \tag{4.2.9}$$

For the penalty $\beta$ we use a BIC penalty which is equal to $p\log(T)$ where $p$ is the number of parameters estimated when we set a changepoint, and $T$ is the length of

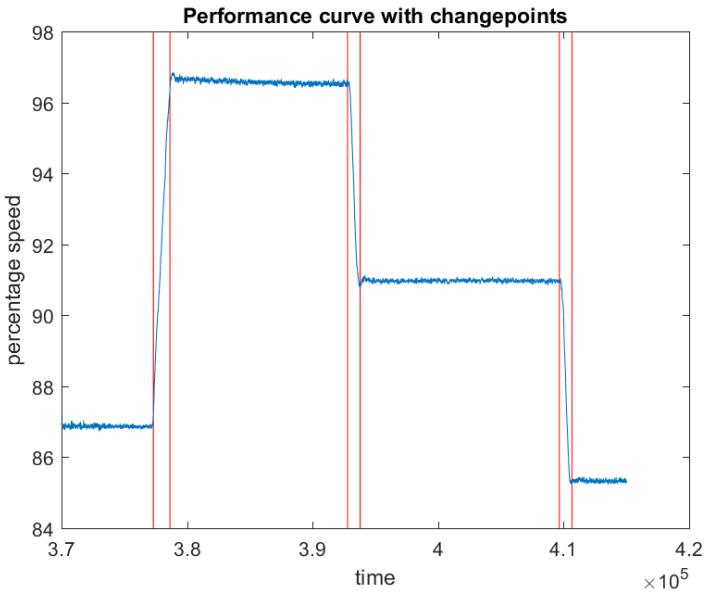



Figure 4.2.1: Plot of a section of the Performance curve in Pass-Off test 1 with changepoints found using PELT with RSS cost function (red) and a BIC penalty.

the time series. For the Pass-Off tests we have found using the cost function (4.2.9) typically under fits the number of true changes in the N1 speed time series. We therefore use the RSS cost function (4.2.4) to identify changes in the piecewise linear structure.

In Figure 4.2.1, we have a plot of a section of a Performance curve with the changepoints plotted in red, the changepoints have been calculated using PELT with the RSS cost function (4.2.4) and a BIC penalty. We can see that the least squares cost function is able to pick up the changes in slope effectively. Our implementation of the changepoints algorithm can be found in the R-package (Killick et al., 2018).

### 4.2.2 Extracting Fixed Speed segments

Given a N1 speed time series $y$ from a Pass-off test, we apply the PELT changepoint method outlined in Section 4.2.1. The PELT algorithm outputs the changepoints, which we use to create a vector of the mean and length of each of the segments. To label fixed speed segments we apply linear regression to each segment, and using the slope coefficient $\alpha_1$, we label a segment as fixed speed if $|\alpha_1| < 0.3$. The choice of threshold 0.3 is made empirically, from looking at the fixed speed extraction in the first few Pass-Off tests. The threshold works well in practice.

Looping through the vector of means for the fixed speed segments $u = (u_1, ..., u_l)$, if $u_{i-1} = [18 \pm 2]\%$ and $u_i > 21$, we start a manoeuvre vector $M = (u_j)$. We have the manoeuvre start time $t_{start}$. We can keep concatenating values to the time series till we get to $u_{k+1} = [18 \pm 2]\%$. For the manoeuvre we get a sequence representation $M = (u_j, ..., u_k)$ where $u_{j-1} = [18 \pm 2]\%$ and $u_{k+1} = [18 \pm 2]\%$ and $u_i > 21\%$ for all $i = j, ..., k$. We now have the manoeuvre vector $M$, and the end time of the manoeuvre $t_{end}$.

## 4.3 Needleman Wunsch

We apply PELT to extract fixed speed segments of a manoeuvre. Taking the mean of each segment we obtain a sequence of the fixed speed levels reached. We can then classify a manoeuvre by matching the sequence against different template sequences. Each template sequence corresponds to the fixed speeds that a defined manoeuvre should reach. However in some manoeuvres fixed speed segments may be missing.

We therefore need an algorithm that can correctly label a manoeuvre with respect to the list of templates even if some sections are missing. To address this challenge we have used the Needleman Wunsch algorithm. In this section we will discuss the Needleman Wunsch algorithm for a fixed alphabet. Later we will discuss extensions for continuous values.

The Needleman and Wunsch (1970) (NW) algorithm was the first computationally efficient sequence alignment algorithm that is guaranteed to find the optimal alignment between sequences from a fixed alphabet. NW is a dynamic programming algorithm with the capability of placing gaps in places where there may have been an insertion or deletion.

Let $G = (G_1, ..., G_p)$ and $H = (H_1, ..., H_q)$ be two sequences with elements $G_i, H_j \in L$ for some alphabet $L$. For example in DNA sequencing $L = \{A, T, C, G\}$. The NW algorithm finds the alignment between $G$ and $H$ that maximises the NW score, defined as:

$$s = Aa + Bb + Cc \tag{4.3.1}$$

where $A$ is the number of matches with scores $a$, $B$ is the number of mismatches with scores $b$ and $C$ is the number of gaps with scores $c$. The score $s$ is therefore the objective function we are trying to maximise. To get a meaningful alignment we need to choose the values $a$, $b$ and $c$ appropriately. There is no consensus in the sequence alignment literature on how the scores $a$, $b$ and $c$ should be chosen. This choice ultimately depends on the characteristics of the application. The score values

$a$, $b$ and $c$ are chosen to be constant as matches and mismatches are clearly defined in the discrete case.

To align the two sequences we generate a $(p+1) \times (q+1)$ similarity matrix $Z$. The matrix considers all possible alignments of the two sequences including gaps. To fill in the matrix we use a gap score $c < 0$ and a similarity measure:

$$S(G_i, H_j) = \left\{ \begin{array}{ll} a, & \text{if } G_i = H_j \\ \\ b, & \text{if } G_i \neq H_j \end{array} \right\}$$

where $a > 0$, $b < 0$. The matrix $Z$ contains elements:

$$Z_{r0} = -r \cdot c \quad \text{for } r = 0, ..., p$$

$$Z_{0k} = -k \cdot c \quad \text{for } k = 0, ..., q$$

$$Z_{i,j} = \max\{Z_{i-1,j-1} + S(G_i, H_j), \quad Z_{i,j-1} + c, \quad Z_{i-1,j} + c\}$$

$$\text{for } i = 1, ..., p \text{ and } j = 1, ..., q.$$

Note the last entry $Z_{p+1,q+1}$ gives the NW score $s$ (4.3.1) for the two sequences.

- If $Z_{i,j} = Z_{i-1,j-1} + S(G_i, H_j)$ we have made a diagonal move and have chosen to align $G_i$ with $H_j$ however the score depends on whether they match or mismatch
- If $Z_{i,j} = Z_{i,j-1} + c$ then we have made a horizontal move and aligned $H_{j-1}$ to a gap.
- If $Z_{i,j} = Z_{i-1,j} + c$ then we have made a downwards move and aligned $G_{i-1}$ to a gap.

To get the alignment we can trace-back along the matrix $Z$. We start in the bottom right-hand corner, and we create a path to the top left. We make a diagonal

move if a match or mismatch was made to get to that value when we constructed the matrix $Z$. We make movements left or up corresponding to gaps, i.e. if a gap penalty was made to get to that value. This is easier to understand by looking at an example. In Figure 4.3.1 we have the $Z$ matrix for sequences GATTACA and GCATGCU, with arrows indicating the trace-back. Note that there can be multiple alignments that give the same optimal score. The coloured arrows indicate the route that was used to get to the score in the bottom corner of $Z$. We have chosen $a = 1$, $b = -2$ and $c = -1$. If two elements mismatch we give a score $-2$, alternatively we can align elements to gaps giving a score of $-2$. These two possibilities are equally weighted as $2c = b$. The optimal paths can be formed using either mismatches or two gaps. By changing the scores we can give preference to mismatches or gaps. Using ':' to denote a gap, one of the optimal paths gives the alignment

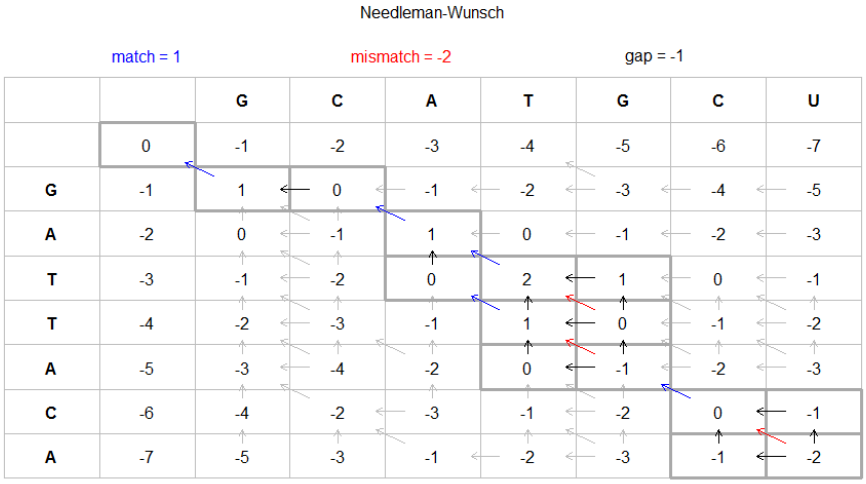
$$GCATG : CU$$

$$G : ATTACA$$

In Theorem (4.3.1) we will show that the NW algorithm gives an optimal alignment.

**Theorem 4.3.1.** *Let $G = (G_1, ..., G_p)$ and $H = (H_1, ..., H_q)$ be two sequences with values from the alphabet $L$. Applying the NW algorithm, with similarity measure $S(\cdot, \cdot)$ where matches are given a score $a$, mismatches are given a score $b$ and there is a gap penalty $c$, with $a > b$ and $a > c$. Then the NW alignment maximises the score* (4.3.1).

*Proof.* We will use proof by induction, let $p = 1$ and $q = 2$ then the similarity matrix




| | | G | C | A | T | G | C | U |
|---|---|---|---|---|---|---|---|---|
| | 0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 |
| G | -1 | 1 | 0 | -1 | -2 | -3 | -4 | -5 |
| A | -2 | 0 | -1 | 1 | 0 | -1 | -2 | -3 |
| T | -3 | -1 | -2 | 0 | 2 | 1 | 0 | -1 |
| T | -4 | -2 | -3 | -1 | 1 | 0 | -1 | -2 |
| A | -5 | -3 | -4 | -2 | 0 | -1 | -2 | -3 |
| C | -6 | -4 | -2 | -3 | -1 | -2 | 0 | -1 |
| A | -7 | -5 | -3 | -1 | -2 | -3 | -1 | -2 |

Figure 4.3.1: Example of a matrix $Z$ aligning sequences GATTACA and GCATGCU, with scores $a = 1$, $b = -2$ and $c = -1$.

$Z$ is easily calculated with

$$Z_{1,1} = S(G_1, H_1) = \left\{ \begin{array}{ll} a, & \text{if } G_1 = H_1 \\ b, & \text{if } G_1 \neq H_1 \end{array} \right\}$$

and

$$Z_{2,1} = \left\{ \begin{array}{ll} a + c, & \text{if } G_1 = H_1 \text{ or } H_2 \\ b + c, & \text{if } G_1 \neq H_1 \text{ and } H_2. \end{array} \right\}$$

Tracing back through the matrix we will get an optimal alignment, maximising the score (4.3.1). So we know the optimal path for $Z_{1,1}$ and $Z_{2,1}$. It follows similarly when $p = 2$ and $q = 1$, so we have an optimal path for $Z_{1,2}$.

Assume that the paths found using NW from $Z_{p-1,q-1}$, $Z_{p,q-1}$ and $Z_{p-1,q}$ are optimal with regards to the score (4.3.1), we can then calculate $Z_{p,q}$ using

$$Z_{p,q} = \max\{Z_{p-1,q-1} + S(X_p, Y_q), \quad Z_{p-1,q} + c, \quad Z_{p,q-1} + c\}.$$

We have extended each of the three paths from $Z_{p-1,q-1}$, $Z_{p,q-1}$ and $Z_{p-1,q}$, and taken the path that maximises the score (4.3.1). As the three previous paths were optimal the extended path must also be optimal. By the law of induction the result follows. □

### 4.3.1 Thresholding Needleman Wunsch

The Needleman Wunsch algorithm is effective at finding alignments between sequences from an alphabet. However we will be using means of fixed speed segments which are continuous. We will therefore need to adapt the similarity measure to account for this variability. From inspection of the fixed speed segments we have found that the means of the fixed speed segments can fluctuate. Typically the means differ by ±2% from those in the templates. Different fixed speed levels always differ by more than 5%.

From Section 4.2 we have shown how a manoeuvre is extracted. From the extraction we have the sequence representation $M = (u_j, ..., u_k)$ where $u_{j-1} = [18 \pm 2]\%$ and $u_{k+1} = [18 \pm 2]\%$ and $u_i > 21\%$ for all $i = j, ..., k$. We also have the extracted manoeuvre time series $y_{t_{start}:t_{end}}$, where $t_{start}$ and $t_{end}$ are the start and end times of the manoeuvre.

We have a defined sequence of fixed speeds for the piecewise linear manoeuvres, for example manoeuvre 'A' has a template sequence $\Upsilon_A = (26)$. Applying the Needleman Wunsch algorithm on the extracted sequence $M$ and the different templates $\Upsilon_A, \Upsilon_B, ..., \Upsilon_{RP}$ gives scores $s_A, s_B, ..., s_{RP}$ respectively. To apply NW we use a threshold similarity measure

$$S(\Upsilon_i, M_j) = \left\{ \begin{array}{ll} 1, & \text{if} \quad |\Upsilon_i - M_j| < \delta \\ -1, & \text{otherwise} \end{array} \right\},$$

to compare the template $\Upsilon$ and the extracted vector of fixed speeds $M$. We have used a gap penalty of $c = -1$. To ensure we correctly match fixed speed levels we have added a tolerance $\delta = 3$.

### 4.3.2 Probabilistic Needleman Wunsch

The NW algorithm outlined in Section 4.3.1 is highly dependent on the choice of the similarity measure and gap penalty, which are not very interpretable. The parameters $a$, $b$ and $c$ have been chosen arbitrarily. We would ideally want to tune the parameters, yet it is unclear how this can be done. Secondly we have chosen a matching threshold $\delta$ again without any tuning. This is a big weakness in the Thresholding NW approach. We have therefore built a probabilistic Needleman Wunsch algorithm, where we make some assumptions on the underlying model generating the sequences. We can therefore choose parameters for the generative model, instead of choosing parameters for NW directly. To make the NW algorithm more interpretable we fit a likelihood model to the scores, motivated by the ideas of Holmes and Durbin (1998).

In our case we have continuous values for the means of the fixed speed segments in the N1 speed time series. We therefore want to build a model where the values are drawn from a continuous distribution. The distribution these values come from depends on whether we are in a match, mismatch or gap state. We choose a simple sequence generation model, with certain probabilities of entering the three states. For

simplicity we have assumed the probability of being in a particular state is independent of the previous state.

Using a template sequence $\Upsilon$ we build a model for how the means of a fixed speed sequence of a manoeuvre can be generated. To ensure clear subscripting we introduce a function $\eta(t)$ which is the highest index in the template assigned up to point $t$. We take the extracted manoeuvre sequence $M$, at time $t$ there are 4 possibilities

- With probability $a$ we are in the state match, then $M_t \sim N(\Upsilon_{\eta(t-1)+1}, 1)$.
- With probability $b$ we are in the state mismatch, then $M_t \sim N(\omega, \psi^2)$.
- With probability $c_1$, we are in the state $M$ insertion, where an extra $M_t$ is emitted, then $M_t \sim N(\omega, \psi^2)$.
- With probability $c_2$, we are in the state $M$ gap, where $M_t$ is not emitted.

$$f(M_t, \Upsilon_{\eta(t-1)+1}|\text{match}) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(M_t - \Upsilon_{\eta(t-1)+1})^2\right\} \tag{4.3.2}$$

$$f(M_t, \Upsilon_{\eta(t-1)+1}|\text{mismatch}) = \frac{1}{\psi\sqrt{2\pi}} \exp\left\{-\frac{1}{2\psi^2}(M_t - \omega)^2\right\} \tag{4.3.3}$$

$$f(M_t|M \text{ insertion}) = \frac{1}{\psi\sqrt{2\pi}} \exp\left\{-\frac{1}{2\psi^2}(M_t - \omega)^2\right\} \tag{4.3.4}$$

$$f(\Upsilon_t|M \text{ gap}) = 1 \tag{4.3.5}$$

where $f$ is the probability density function.

In the probabilistic model we set $a = 0.7$, $b = 0.1$, $c_1 = 0.1$ and $c_2 = 0.1$. We have therefore assumed there is a higher probability of being in a match state than a mismatch or gap state. We have set the hyper-parameters $\omega = 50$ and $\psi = 10$, where

$\omega$ is the midpoint as the percentage speed ranges from $[0, 100]$. Whilst $\psi$ captures the large variance in speeds.

The score $s$ gives the best alignment of the two sequences assuming the manoeuvres sequences are generated by this model. For this model the similarity measure is given by

$$S(\Upsilon_i, M_j) = \max\{af(M_j, \Upsilon_i|\text{match}), bf(M_j, \Upsilon_i|\text{mismatch}), c_1 f(M_j|M \text{ insertion}), c_2 f(\Upsilon_i|M \text{ gap})\}$$

We can fill the similarity matrix $Z$ by going along the diagonal if the probability of matching or mismatching is maximal in the similarity measure (**??**). Likewise we place gaps if the gap probabilities are maximal.

Applying Probabilistic Needleman Wunsch on the extracted sequence $M$ and the different templates $\Upsilon_A, \Upsilon_B, ..., \Upsilon_{RP}$ gives scores $p_A, p_B, ..., p_{RP}$ respectively. Thresholding NW and Probabilistic NW both give the same alignments, we therefore don't get more information by using both scores.

### 4.3.3 Example

In this section we give an example of the alignment using Needleman-Wunsch (NW). We will use a Performance Curve (P) shown in Figure 4.3.2. The P manoeuvre was extracted using the PELT changepoint algorithm discussed in Section 4.2.2. The P manoeuvre has a sequence of fixed speed levels $M = [87, 97, 91, 85, 79, 73, 61, 52, 27]$. We will show the alignment given by NW for sequence $M$ and the template for manoeuvre $P$: $\Upsilon_P = [96, 90, 86, 79, 72, 60, 51, 38, 27]$. We set the scores $a = 1$, $b = -1$

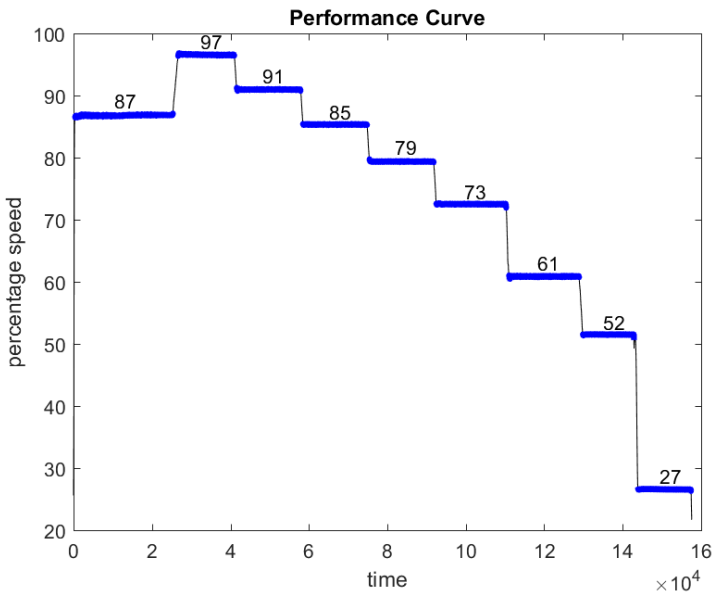



Figure 4.3.2: Plot of a a Performance curve with labelled fixed speed levels.

and $c = -1$ for the Thresholding NW, which gives a score of $s_P = 6$ and alignment

$$M = [87, 97, 91, 85, 79, 73, 61, 52, \quad , 27]$$

$$\Upsilon_P = [\quad , 96, 90, 86, 79, 72, 60, 51, 38, 27]$$

## 4.4 Functional PCA Templates

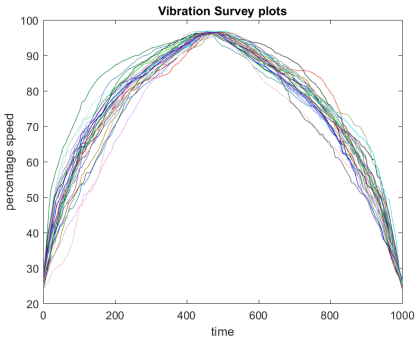
In Section 4.3 we obtained scores for the manoeuvres which are based on fixed speed levels. Manoeuvres F and V do not contain fixed speed levels. We will therefore build templates for these manoeuvres in a different way. We will use Functional PCA as outlined in Chapter 3.2 to build the templates. We have discussed how Functional PCA can be applied using the Basis method. In reality we don't have the functions $x_i(t)$ instead we have realisations of the functions, giving a time series for

each function $x_i(t)$. The realisations can be made at different times with different number of observations for each function. We can fit a basis to these time series to obtain an approximation of the curves $\tilde{x}_i(t)$. Using the Basis method outlined in Chapter 3.2, we can calculate the eigenfunctions.

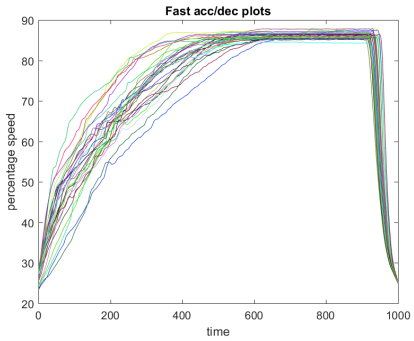
We will use 30 samples of manoeuvres F and V to build the FPCA models. The 30 samples are shown in Figure 4.4.1. The two manoeuvres have a clear shape which will be represented by a mean function. Taking out the mean we get the mean corrected time series shown in Figures 4.4.2. There is a lot of variation that will be picked up by the eigenfunctions. For the F manoeuvres the quick deceleration causes a huge amount of variance in the residuals, as the change from acceleration to deceleration can occur in slightly different places.

We used a Fourier basis formed of 201 functions. This basis worked well in modelling the F and V manoeuvres however other bases could have been used. Next we needed to choose the number of principal components we wanted to model the curves. We chose the first $K$ eigenfunctions that ensure 95% of the variance is captured. For both the F and V manoeuvres we found that four eigenfunctions was sufficient. The templates are formed of mean functions $\mu^F(t)$, $\mu^V(t)$ and eigenfunctions $\phi_i^F(t)$, $\phi_i^V(t)$ for manoeuvres F and V respectively, where $t \in [0, 1]$ and $i = 1, ..., 4$.

Next we will show how we will generate the scores $s_F$, $s_V$ using the templates for manoeuvres F and V. For an unlabelled manoeuvre function $z(t)$, we generate a representation of $z(t)$ with respect to manoeuvre V:

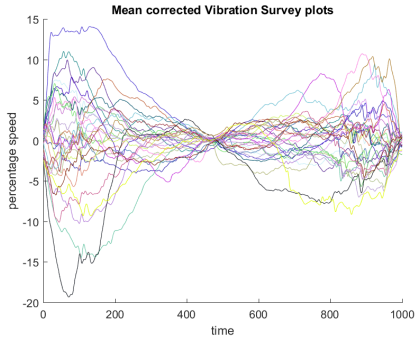



(a) Vibration Surveys

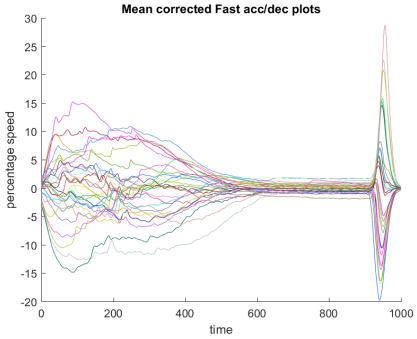



(b) Fast acc/dec

Figure 4.4.1: Plots of 29 Vibration surveys (V) and Fast acc/dec (F) manoeuvres.

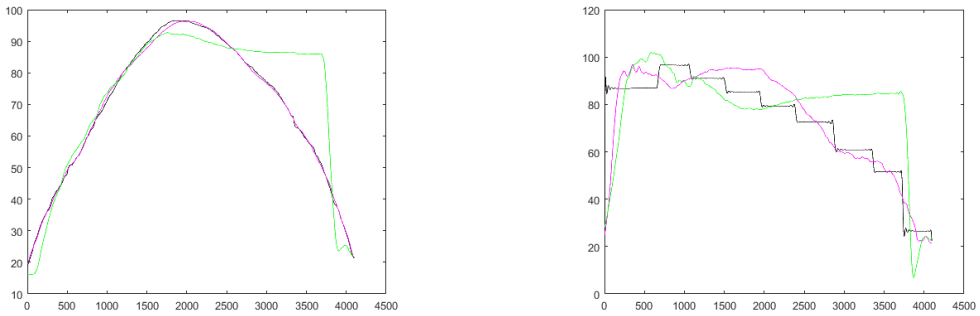



(a) Vibration Surveys




(b) Fast acc/dec

Figure 4.4.2: Plots of 29 mean corrected Vibration surveys (V) and Fast acc/dec (F) manoeuvres.



(a) Reconstruct Vibration Survey (b) Reconstruct Performance Curve

Figure 4.4.3: Plot of FPCA reconstruction of a Vibration Survey (left) and a Performance Curve (right), using FPCA representations of V (pink) and F (green).

$$\hat{z}^V(t) = \mu^V(t) + \sum_{i=1}^{4} g_i \phi_i^V(t)$$

where $g_i = \int_0^1 (z(t) - \mu^V(t))\phi_i^V(t)dt$. The reconstruction error gives the required score $s_V = \int_0^1 [\hat{z}^V(t) - z(t)]^2 dt$. If $z(t)$ is a V manoeuvre it should be well represented by the mean and eigenfunctions, giving a small reconstruction error. This feature makes the reconstruction error a good score to help identify a manoeuvre as a V. We can do the same analysis to measure the fit $z(t)$ to a F manoeuvre. We now have two scores $s_F$ and $s_V$ corresponding to the representations for manoeuvres F and V.

In Figure 4.4.3 we have a plot of the reconstruction of a manoeuvre V and P using the FPCA representations for F and V. We can see that manoeuvre V is well represented by its FPCA model, but not by the FPCA representation for F. For manoeuvre P, neither FPCA representations give a good fit. From these figures we can see that the FPCA fit is an informative metric.

## 4.5 Classifiers

In Section 4.3 we have outlined the Needleman-Wunsch (NW) algorithm. For an unlabelled manoeuvre $z(t)$ we extract the fixed speed segments and take the mean in each segment to obtain a vector $u$. We apply the Needleman-Wunsch algorithm to align the vector $u$ with the templates $\Upsilon_A, \Upsilon_B, ..., \Upsilon_{RP}$ to obtain a corresponding vector of NW scores $s_A, s_B, ..., s_{RP}$. For manoeuvre F and V we use the Functional PCA templates to obtain representations $z^F(t)$ and $z^V(t)$. Taking the squared difference of the representations with respect to $z(t)$ we obtain the scores $s_F$ and $s_V$ as outlined in Section 4.4. The vector of scores $(s_A, s_B, ..., s_{RP}, s_F, s_V)$ is used as an input vector for a classifier. We will outline two classifiers.

### 4.5.1 Decision Tree

In this section we will apply a Decision tree classifier (Rokach and Maimon, 2005) to label the manoeuvres. Decision trees are a popular method for classifying samples using a given set of features. A decision tree is comprised of a root node which splits into test nodes. The leaves of the tree are called the decision nodes which set the classification of the sample. A decision tree once built will take a vector of features and using a set of decision rules will output a classification.

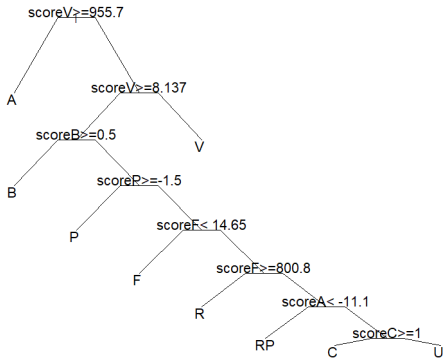
To build a decision tree we need a training set comprised of labelled samples with the corresponding feature vectors. We will build a decision tree using the Classification And Regression Trees (CART) method (Breiman et al., 1984). To illustrate the general structure of these decision trees we use a training set of the first 40 Pass-Off

Table 4.5.1: The number of each manoeuvre in the training set.

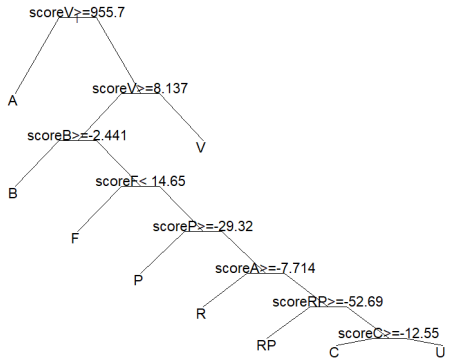
| Manoeuvre | A | B | C | R | P | RP | F | V | U | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Number in training Set | 108 | 53 | 8 | 38 | 52 | 36 | 51 | 86 | 49 | 481 |

tests. The training set is comprised of 481 manoeuvres, with each manoeuvre having a vector of scores. The number of occurrences for each manoeuvre is given in Table 4.5.1. Note that there is only one instance of an RP manoeuvre, which is insufficient to classify the manoeuvre. We have therefore combined 35 R and P manoeuvres that occurred sequentially in a test.

Taking the training set of labelled manoeuvres and scores, we use the CART algorithm to build a decision tree. We chose the decision tree using a 10-fold cross-validation approach. In k-fold cross-validation we split the data into k folds of equal size. We leave out one fold and train the classification algorithm on the remaining (k-1)-folds. We then test on the left out fold. We perform the same procedure, leaving out a different fold each time. We obtain k scores, which we can average over to obtain the average classification accuracy. We can then estimate the mean number of misclassifications, and the standard error. We have used the Rpart package in R (Therneau et al., 2011), to form these decision trees. We then picked a tree using the 1-Standard Error approach (Breiman et al., 1984), which chooses the smallest tree that is within one standard error of the tree with the minimum number of misclassifications. The resulting decision trees using Thresholding and Probabilistic NW are shown in Figure 4.5.1.




(a) Tree using NW scores




(b) Tree using Probabilistic NW scores

Figure 4.5.1: Pruned Trees using NW (left) and Probabilistic NW (right) scores and applying 10-fold Cross Validation.

The CART algorithm identifies that features for the Unknown manoeuvres differ from the other manoeuvres, which is why they are classified by the last decision node. The decision node's typically use the scores generated for that particular manoeuvre. For example manoeuvre V is classified using the score $s_V$. Both the decision trees in Figure 4.5.1 give 3 misclassifications for the manoeuvres in Test 41 to 93. Two of these misclassifications occur in Test 46, which we will discuss in Section 4.5.3.

### 4.5.2 Linear Discriminant Analysis

The decision tree classifier outlined in Section 4.5.1 is an effective classifier. However the classifier has a notable weakness. The Unknown manoeuvres are not explicitly modelled, which can make the DT liable to misclassify Unknown manoeuvres that do not match those in the training set. We therefore consider Linear Discriminant Analysis (LDA) to classify the manoeuvres, which gives an associated probability for each classification. In this model we explicitly model the Unknown manoeuvres. We

will show for Test 46 that misclassifications by the Decision tree approach, can be identified by the LDA method. A more thorough comparison will be given in Sections 4.6 and 4.7.

Let $n_i$ be the number of samples of class $i$ in a training set, for $i \in \Theta$ where $\Theta = \{A, B, C, R, P, RP, F, V\}$. We assume the score vectors in each class follows a multivariate normal distribution. We estimate the parameters of these distributions using maximum likelihood estimation (MLE) obtaining estimated mean vector $\hat{\mu}_i$ and covariance matrix $\hat{\Sigma}_i$ for class $i$. We set the prior probability of class $i$

$$P(\text{class } i) = \frac{n_i}{\sum_{k\in\Theta} n_k} \tag{4.5.1}$$

Then using Bayes theorem we can calculate the probability an unlabelled manoeuvre with score vector $x^*$ coming from class $i$:

$$P(\text{class } i|x^*, \hat{\mu}_i, \hat{\Sigma}_i) = \frac{P(x^*|\hat{\mu}_i, \hat{\Sigma}_i)P(\text{class } i)}{\sum_k P(x^*|\hat{\mu}_k, \hat{\Sigma}_k)P(\text{class } k)}. \tag{4.5.2}$$

We have seen that the Unknown manoeuvres can have different shapes and lengths. It therefore doesn't make sense to fit a distribution to the scores of the Unknown manoeuvres, as in practice an Unknown manoeuvre may be performed that wasn't observed in the Training phase. We want to classify a manoeuvre as Unknown if $P(\text{class } i|x^*)$ is very small for all manoeuvres $i$. We therefore model the Unknown manoeuvres using a Gaussian distribution with very high variance, which gives it a flat density. The covariance matrix $\Sigma_U = 1000 * I$ has sufficiently large variance. The mean is inconsequential so is taken to be the zero vector.

### 4.5.3 Comparing the DT and LDA classifiers

Using the manoeuvres from the first 40 Pass-Off tests, we will train the DT and LDA classifiers. We will use Pass-Off test 46 shown in Figure 4.5.2 to illustrate how the LDA classifier gives improved classification performance. The LDA classifier gives a vector of probabilities for each type of manoeuvre. We have found that the classifier gives probability 0 to all manoeuvres except one. The manoeuvre with probability 1 is in almost all cases the truth. The degeneracy in the probability values seem to arise due to the vector of scores generated for each type of manoeuvre being so distinct. There is effectively no overlap in the probability densities.

To identify unusual samples, we can use the Mahalanobis distance (MD). To calculate MD we use the means $\hat{\mu}_i$ and covariance matrices $\hat{\Sigma}_i$ for $i \in \Theta$. These terms have already been calculated for the LDA classifier, meaning there is no additional computational cost. The MD is given by:

$$MD(x|\hat{\mu}_i, \hat{\Sigma}_i) = \sqrt{(x - \hat{\mu}_i)^T \hat{\Sigma}_i^{-1} (x - \hat{\mu}_i)} \tag{4.5.3}$$

The MD gives a score which can be used to identify unusual samples. It uses the covariance information, so considers the spread of the distribution when giving a score. It is also unitless and scale-invariant, which is particularly useful in our case as we can compare the MD directly for manoeuvres in different classes.

In Table 4.5.2, we have the true labels for the manoeuvres in Test 46, alongside the labels using the DT and LDA classifiers. We have also included the MD of the manoeuvres with respect to the class assigned by LDA. We can see that there are

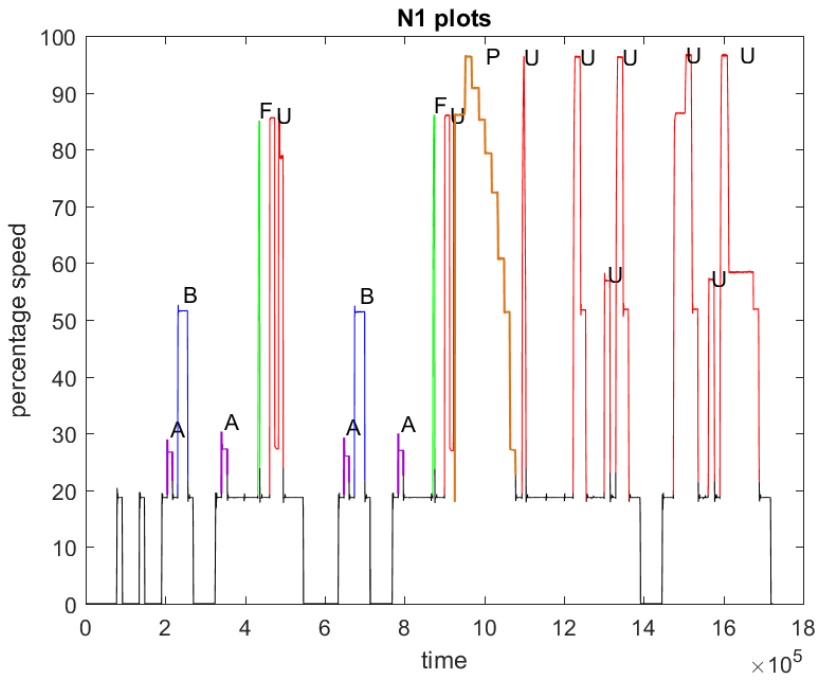



Figure 4.5.2: Pass-Off test 46, labelled using DT classifier.

three misclassifications by DT, and one by LDA. Both classifiers mislabelled the first Running and Handling (R) manoeuvre, which is not surprising given it has a different profile to a normal R manoeuvre. Looking at the probabilities in Table 4.5.2, there are two cases that have zero probabilities. These are the manoeuvres that have been misclassified by the DT, but correctly identified by the LDA. One manoeuvre is a Vibration Survey, but was mislabelled as it gave a score of 8.66, which was above the 8.1372 threshold in the DT. Likewise there is a misclassification of the R manoeuvre before the P manoeuvre, which occurs for the same reason. However the LDA classifier is able to correctly classify these manoeuvres. The MD highlights unusual cases for example one of the R manoeuvres has a significantly large MD value, which highlights it is worth further inspection even though the LDA is able to correctly classify the manoeuvre.

Table 4.5.2: Table of labels given for Test 46, shown in Figure 4.5.2, with colours matching the manoeuvre classes. We have the true labels, and the labels using the Decision Tree (DT) and Linear Discriminant Analysis (LDA). We also have the Mahalanobis distance with respect to the manoeuvre class given by LDA.

| True Labels | DT Labels | LDA Labels | Mahalanobis |
|---|---|---|---|
| Stops | Stops | Stops | - |
| Stops | Stops | Stops | - |
| A | A | A | 0.438 |
| B | B | B | 0.986 |
| Stops | Stops | Stops | - |
| A | A | A | 3.719 |
| F | F | F | 2.283 |
| R | U | U | 361.567 |
| Stops | Stops | Stops | - |
| A | A | A | 3.167 |
| B | B | B | 1.266 |
| Stops | Stops | Stops | - |
| A | A | A | 1.094 |
| F | F | F | 0.654 |
| R | U | R | 732.006 |
| P | P | P | 0.261 |
| V | U | V | 55.322 |
| U | U | U | 12.186 |
| U | U | U | 4.905 |
| U | U | U | 12.398 |
| Stops | Stops | Stops | - |
| U | U | U | 11.914 |
| U | U | U | 5.387 |
| U | U | U | 9.381 |

Table 4.6.1: The number of each manoeuvre in the 93 Pass-Off tests.

| Manoeuvre | A | B | C | R | P | RP | F | V | U | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Number of Instances | 231 | 110 | 10 | 89 | 123 | 85 | 111 | 199 | 107 | 1030 |

## 4.6 Testing on Trent 1000 engines

We have outlined a classification algorithm for manoeuvres in a Trent 1000 Pass-Off test. In this section we will assess the classification accuracy of the model. We will use k-fold cross-validation to assess the classification accuracy.

In Table 4.6.1 we have the total number of each manoeuvre in the 93 Pass-Off tests. We can see that manoeuvres A and V are performed significantly more than the other manoeuvres. We can also see a large number of manoeuvres can be categorised as Unknown. The RP manoeuvre only occurs once, which is insufficient for the DT and LDA classifiers. We therefore create an additional 84 instances of RP manoeuvres by combining R and P manoeuvres performed together in the tests.

We will use standard 10-fold cross-validation. The mean percentage of misclassifications of the test data is (0.184%, 0.0464%) with variances (0.00045%,0.000066%) for the DT and LDA classifiers respectively. We can see that both classifiers give high classification accuracy, however the LDA classifier does significantly outperform the DT classifier. The DT in particular struggles classifying manoeuvre C, as there only a few instances.

## 4.7 Testing on XWB engines

In Section 4.6, we have shown the classification algorithm is effective in classifying the manoeuvres in a Trent 1000 Pass-Off test. Naturally we ask whether this algorithm can be used for Pass-Off tests for different engines. We therefore considered testing the classification algorithm on XWB Pass-Off tests.

We found that using the classification algorithm directly on the XWB Pass-Off test data gives poor classification performance. The manoeuvres have different speed ranges and slightly different shapes. We therefore need to create new templates as done before. There also a few other subtle details to outline. First the manoeuvre 'B' is not performed but a new manoeuvre, which we have labelled as 'D' is performed. Second the Running and Handling (R) manoeuvre does not return to idle speed so will not be treated as a manoeuvre. Third the F and V manoeuvres have a different profile to those performed in the Trent 100 engines, as shown in Figure 4.7.2.

The list of manoeuvres for the XWB Pass-Off tests are given below.

A (22)

D (85)

P Performance Curve (96, 92, 90, 86, 79, 72, 66, 57, 22)

RP Running & Handling/Performance Curve (92, 22, 92, 96, 92, 90, 86, 79, 72, 66, 57, 22)

F Fast Acceleration/Deceleration

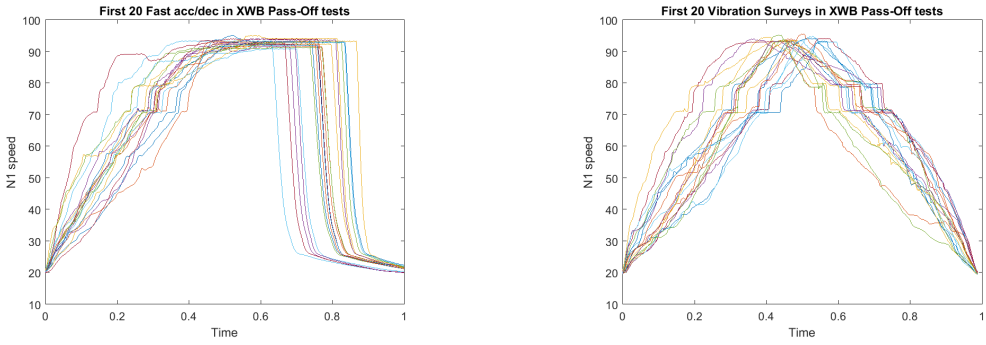
V Vibration Survey

U Unknown.

Using the templates given above for the piecewise linear manoeuvres and those we constructed using FPCA for manoeuvres F and V we can classify the manoeuvres in an XWB Pass-Off test. We built a LDA classifier using all the data and obtain 43 misclassifications. There are two main causes of these misclassifications the first are the spike manoeuvres, which can be seen in Test 25 shown in Figure 4.7.3. These spike manoeuvres do not appear in the Trent 1000 engine tests, but appear in some of the XWB engine tests. Second, the fixed speed levels can be very small and therefore difficult to extract as shown in Test 54 in Figure 4.7.3. However the vast majority of the manoeuvres are effectively classified. Third, looking at the F and V manoeuvres in Figure 4.7.1 we can see that there is a higher variance in comparison to the F and V manoeuvres in the Trent 1000 Pass-Off tests shown in Figure 4.4.1. We can

Table 4.7.1: The number of each manoeuvre in the 54 XWB Pass-Off tests.

| Manoeuvre | A | D | P | RP | F | V | U | Total |
|---|---|---|---|---|---|---|---|---|
| Number of Instances | 86 | 73 | 24 | 55 | 58 | 92 | 42 | 430 |

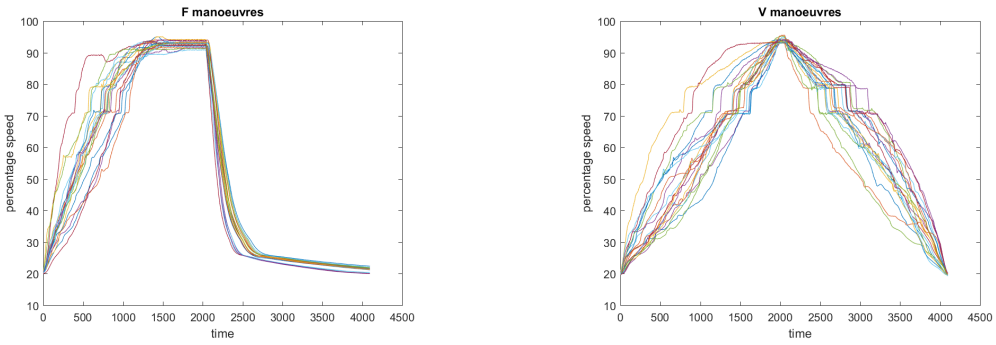



(a) Fast acc/dec

(b) Vibration Survey

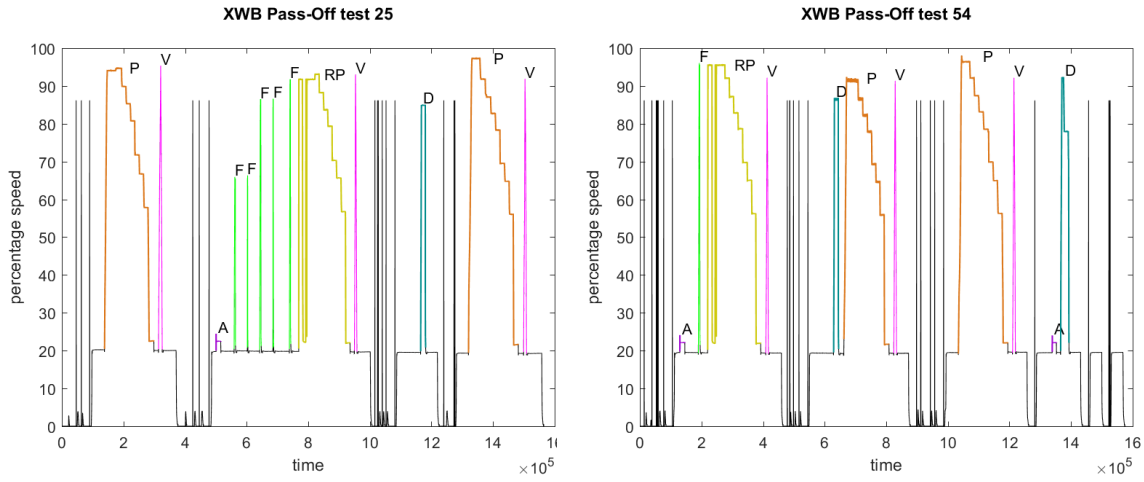
Figure 4.7.1: Plot of first 20 F and V manoeuvres

remedy this issue by aligning the curves using curve registration approaches (Ramsay and Silverman, 2005). There are two standard approaches, the first uses warping functions to find an alignment however this can be computationally expensive. The second approach is to align using landmarks or features of the curves. For the F and V manoeuvres there is a distinctive point of deceleration, these points can be identified using the PELT algorithm outlined in Section 4.2.1. In Figure 4.7.2 we have a plot of the F and V manoeuvres aligned at the deceleration points, which are clearly easier to model using Functional PCA.




(a) Fast acc/dec (b) Vibration Survey

Figure 4.7.2: Plot of first 20 Fast acc/dec (F) and Vibration Survey (V) manoeuvres aligned at the deceleration point.




(a) Pass-Off test 25 (b) Pass-Off test 54

Figure 4.7.3: Labelled N1 speed plots for XWB Pass-Off test 25 (left) and 54 (right).

## 4.8 Heatmap

The classification algorithm we have developed can be used to highlight problematic engine tests, which we will illustrate using a heatmap. We count the number of each manoeuvre performed in each Pass-Off test, and then put these values into a heatmap. The heatmap highlight instances of manoeuvres that have been performed a large number of times. The colouring gives a quick visual comparison and highlights tests in which a large number of manoeuvre repeats occurred. An example heatmap is given in Figure 4.8.1 for the first 10 Pass-Off tests. We can see that Pass-Off test 4 has been stopped 13 times and Vibration Survey (V) has been repeated 12 times. This is clearly a problematic test relative to the other tests. Manoeuvre V has been repeated in the majority of the Pass-Off tests, highlighting possible engine issues were detected during this manoeuvre. The number of stops performed can also be a good indicator of problems that have arisen during the test. For example Pass-Off test 2 has been stopped 6 times, which indicates multiple engine tweaks were performed.

## 4.9 Conclusion

We have built a classification algorithm to extract and label the manoeuvres in a Pass-Off test. The PELT changepoint algorithm is used to extract the manoeuvre segments from the N1 speed time series. Using templates for each manoeuvre class, we have calculated Needleman-Wunsch (NW) and FPCA scores. We have also built a Probabilistic NW algorithm that can align two real-valued sequences. The scores are treated as features that can be input into a classifier. Two classifiers are considered: an

off-the-shelf Decision Tree (DT) and a Linear Discriminant Analysis (LDA) classifier. Using 10-fold cross validation, we have shown that the LDA classifier has higher classification accuracy than the DT classifier. We can also use the Mahalanobis distance to highlight manoeuvres that may be performed in an unusual way. The labels from the classification algorithm can determine problematic Pass-Off tests, which can be visualised via a heatmap. We have tested the classification approach on Trent 1000 engine and have found the algorithm gives exceptional classification performance. We also tested the approach on XWB engines, which gives good classification performance however the manoeuvres in the XWB engines introduce further difficulties. In summary, the classification algorithm is fast, exploiting the efficiency of PELT, NW and FPCA and gives near perfect classification. However the algorithm requires prior information to build the templates and needs a training set for the classifiers.

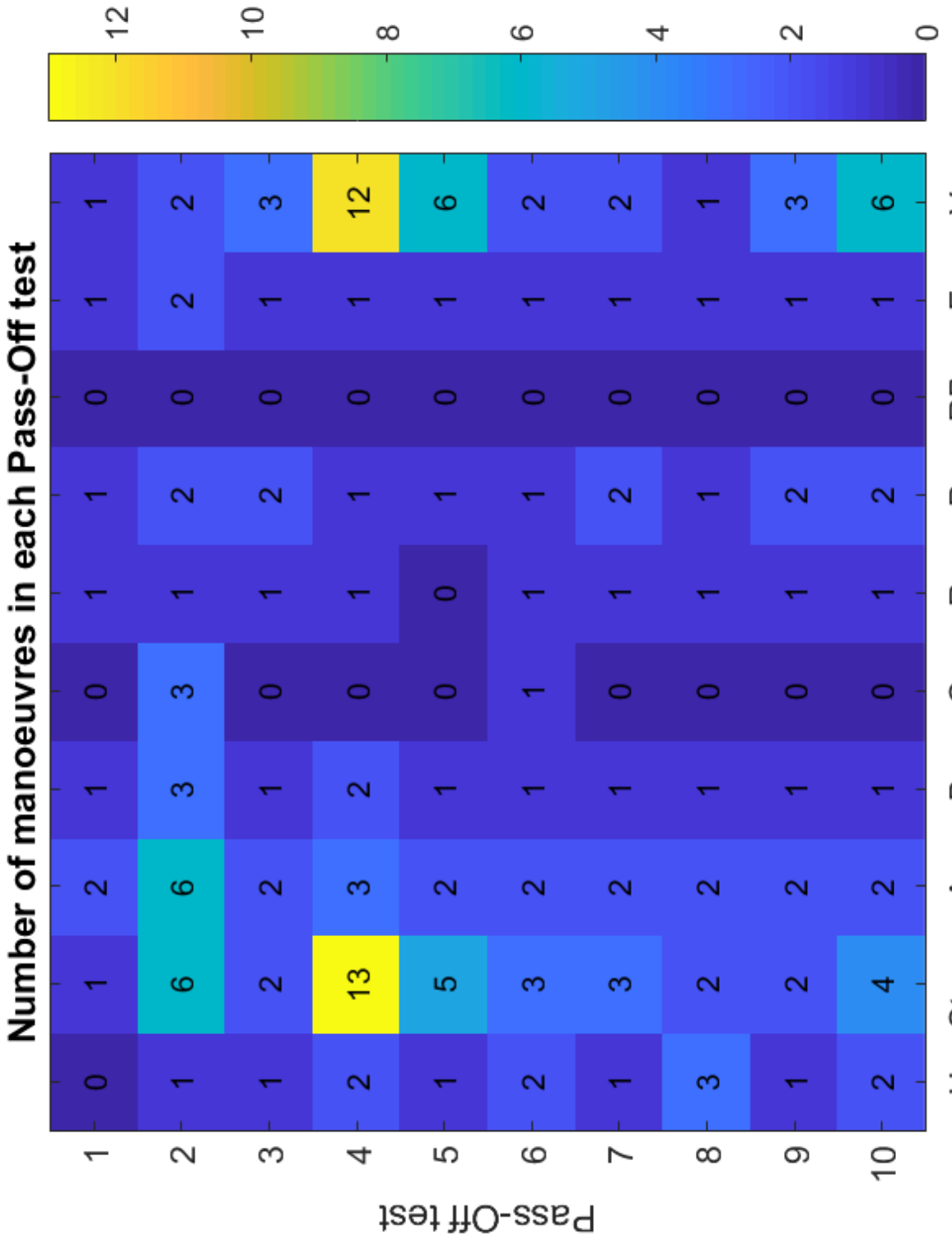



Figure 4.8.1: A Heat Map of the number of manoeuvres in the first 10 Pass-Off tests

# Chapter 5

# Manoeuvre Clustering in Cyclic tests

## 5.1 Introduction

In a Pass-Off test engineers perform a sequence of pre-defined manoeuvres, whereas in the Cyclic engine tests (described in Chapter 1) the manoeuvres are not pre-defined. In both tests the engineers follow a schedule plan but deviations can occur. In particular manoeuvres can be performed that do not match those in the schedule. We have previously referred to these manoeuvres as Unknown. We do not know the manoeuvres classes for the Cyclic test. We therefore propose a clustering algorithm. There is potential to use the output of the clustering algorithm to build templates. We could then create a classification algorithm as we did for the Pass-Off test manoeuvres in Chapter 4.

The Cyclic test is performed to assess the degradation of the engine performance

over time. We would ideally want an online monitoring system to flag signs of engine degradation. The different manoeuvres performed makes it difficult to identify engine degradation. Therefore the aim is to use the clusters to identify engine deterioration by comparing the behaviour within each cluster over time.

We will split a Cyclic test into manoeuvres, each of which is a time series starting and ending at idle speed. We then calculate the pairwise distances between each pair of manoeuvres, but to do so we need to deal with the varying lengths. We will therefore use Dynamic Time Warping (DTW), which is capable of comparing two time series of different lengths. We then cluster using the pairwise distances.

In Section 5.2 we discuss density based clustering algorithms. In particular we outline the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm that is able to cluster data in the presence of outliers. There are alternative approaches such as robust $k$-means (García Escudero et al., 2015), hierarchical (Balcan et al., 2014) and spectral clustering (Bojchevski et al., 2017) methods that aim to mitigate the effect of outliers. However these approaches do not explicitly identify the outliers and are unable to determine the number of clusters endogenously. In Section 5.3 we describe Dynamic Time Warping (DTW), which is capable of giving the distance between two time series of different lengths. The clustering approach is outlined in Algorithm 3, which uses Dynamic Time Warping with a density based clustering algorithm to determine manoeuvre classes in an engine test. In Section 5.5 we apply the clustering algorithm on the Cyclic test data. We also apply the visualisation tool: tSNE, described in Chapter 2 to see the cluster structures. We have chosen tSNE as it only requires the pairwise distances and gives good visualisations in

practice. In Sections 5.6 and 5.7 we test the clustering algorithm on the manoeuvres from the Trent 1000 and XWB engine Pass-Off tests, which were previously analysed in Chapter 4. We can then assess the effectiveness of the clustering algorithm using the true labels. Finally we discuss the results and possible extensions in Section 5.8.

## 5.2 Density Based Clustering

There are a number of density based clustering algorithms (Kriegel et al., 2011). Typically we assume that the data is sampled from an unknown probability density $p(x)$. Density based clustering approaches are non-parametric, where clusters are assigned by regions where the density of points is above a threshold. The most famous is the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (Ester et al., 1996), which uses a distance parameter $\epsilon$ and a minimum cluster size $m$. The algorithm finds clusters in which the points are mutually *density-connected*, i.e. that every point in the cluster is within $\epsilon$ of another point in the cluster and the points are *density-reachable* i.e. a point can be connected to another point in the cluster by a chain of points where each link is less than $\epsilon$. DBSCAN approximates the density of each cluster using uniform kernel distributions. A point is an outlier if there are less than $m$ points in this $\epsilon$-neighbourhood.

The DBSCAN procedure is outlined in Algorithm 2. Note that DBSCAN does not require the number of clusters to be known unlike $k$-means. DBSCAN can find clusters of arbitrary shape and can effectively identify outliers for a well chosen $\epsilon$. However there are some notable weaknesses: DBSCAN results depend heavily on the

choice of $\epsilon$. The standard approach is to calculate the $m$-nearest neighbour distances and by ordering and plotting the distances, we form an elbow plot to determine $\epsilon$. Alternatively we can take a 95% percentile of the $m$-nearest neighbour distances as suggested by Daszykowski et al. (2001). Another potential weakness is that a single-link can cause two potentially disjoint clusters to merge together.

**Algorithm 2** DBSCAN

**INPUTS:** Data points $x_1, ..., x_n$, distance $\epsilon$ and minimum cluster size $m$,
**Initialisations:**
Set $S = \{x_1, ..., x_n\}$ and $cluster = 0$
**while** While $|S| > 0$ **do**
    Take a random point $x_i \in S$
    Find all points in $S$ that are density-reachable to $x_i$ and put into a set $H$
    **if** $|H| < m$ **then**
        Label all points in $H$ as noise (-1)
        $S = S \setminus H$
    **else**
        $cluster = cluster + 1$
        Assign all points in $H$ to $cluster$
        $S = S \setminus H$
    **end if**
**end while**
**RETURN:** cluster assignment.

## 5.3 Dynamic Time Warping

In this section we will describe the Dynamic Time Warping (DTW) distance (Senin, 2008). We will use this distance measure to obtain the pairwise distances between the manoeuvre samples. Given two sequences $a = (a_1, ..., a_N)$ and $b = (b_1, ..., b_M)$, we build a distance matrix $C \in \mathbb{R}^{N \times M}$ of all pairwise distances between $a$ and $b$. An alignment path can be defined as $s = (s_1, .., s_L)$ where $s_l = (N_l, M_l)$ and $M \leq L \leq N$, which satisfies the following conditions:

1. Boundary: $p_1 = (1, 1)$ and $p_L = (N, M)$

2. Monotonicity: $1 = N_1 \leq \cdots \leq N_L = N$ and $1 = M_1 \leq \cdots \leq M_L = M$

3. Step size: $N_{l+1} - N_l < \delta$ and $M_{l+1} - M_l < \delta$ for a threshold $\delta$.

We define the cost of an alignment $s$ as

$$c_p(a, b) = \sum_{l=1}^{L} c(a_{N_l}, b_{M_l}) \text{ where } c(a_{N_l}, b_{M_l}) = |a_{N_l} - b_{M_l}|.$$

The optimal path is given by:

$$DTW(a, b) = \min_{s \in S}\{c_p(a, b)\}$$

where $S$ is the set of all alignment paths that satisfy the conditions given above.

To find the optimal path we use a dynamic programming procedure similar to Needleman-Wunsch discussed in Section 4.3. We build a global cost matrix $D$ where

- $D(1, j) = \sum_{l=1}^{j} c(a_1, b_l)$ for $j = 1, ..., M$
- $D(i, 1) = \sum_{l=1}^{i} c(a_l, b_1)$ for $i = 1, ..., N$
- $D(i, j) = \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + c(a_i, b_j)\}$ for $i = 1, ..., N$ and $j = 1, ..., M$.

The computational cost of DTW is O($NM$) due to the construction of the global cost matrix. The optimal path is found by backtracking from $(N, M)$.

Additional constraints can be added:

- Step function - the alignment path can only move up to $w$ consecutive times in a certain direction.

- Weighting - we can penalize horizontal or vertical directions in $D$, this is equivalent to penalising gaps in Needleman and Wunsch (1970).

- Global path constraints - allow alignments only in a band or a parallelogram region, this will notably reduce the computational cost.

The DBSCAN model relies on two parameters $m$ and $\epsilon$, which are codependent. We follow the standard procedure of selecting $m$, which is relatively intuitive. Then we select $\epsilon$ using an elbow plot of the $m$-nearest neighbour distances. The clustering algorithm is outlined in Algorithm 3.

**Algorithm 3** Manoeuvre Clustering algorithm

**INPUTS:** Time series $x_1, ..., x_n$ and minimum cluster size $m$,
**Initialisations:**
Set empty matrix $W$.
**for** $i = 1 : n$ **do**
    **for** $j = i + 1 : n$ **do**
        Obtain distance $W_{ij} = DTW(x_i, x_j)$.
    **end for**
**end for**
**for** $l = 1 : n$ **do**
    Calculate $m$-nearest neighbour distances for $x_l : d_l$.
**end for**
Order distances $d_l$ and concatenate to form a vector $d$
Plot $d$ to obtain an elbow plot and choose parameter $\epsilon$ at elbow
Apply DBSCAN$(\epsilon, m)$ using distance matrix $W$
**RETURN:** cluster assignment.

## 5.4 Cluster Evaluation

For the Trent 1000 and XWB Pass-Off tests we have labels for the manoeuvres, which we can use to assess the effectiveness of the clustering algorithm given in Algorithm 3. There are a number of evaluation techniques to assess the clustering performance

with respect to the true classes (Manning et al., 2008). The most intuitive is the *purity*, where each cluster is assigned to the class, which appears most frequently in the cluster. Then the purity is given by counting the number of correctly assigned samples divided by the total number of samples $n$. Let $w = (w_1, ..., w_K)$ be the true class groupings and $c = (c_1, ..., c_J)$ be the cluster groupings, then the purity is given by:

$$purity = \frac{1}{n} \sum_{k=1}^{K} \max_{j} |w_k \cap c_j|.$$

Purity does not penalise for increasing number of clusters, i.e. the purity is equal to 1 if every point is assigned to a unique cluster.

Alternatively there are information-metrics such as the Mutual Information (MI), which quantifies the amount of information gained about the classes when we are told the cluster assignments. For classes $w$ and cluster assignment $c$ the MI is given by:

$$MI(w; c) = H(w) - H(w|c),$$

where $H(\cdot)$ is the entropy. However the MI like the purity measure does not penalise large number of clusters. Therefore we use the Normalised Mutual Information (NMI):

$$NMI(w, c) = \frac{2MI(w; c)}{H(w) + H(c)}.$$

The normalisation term $[H(w) + H(c)]/2$ tends to increase as the number of clusters increases. Using NMI we can compare across different cluster assignments. We can show $0 \leq NMI(w, c) \leq 1$, where a value of 1 corresponds to the cluster assignment

being identical to the class assignment. We will use the NMI measure to evaluate the clustering performance for the Trent 100 and XWB engines.

## 5.5 Clusters in Cyclic Test Data

There are 281 manoeuvres in the Cyclic test. We do not need to extract the manoeuvres as in the Pass-Off test as there is a marker to identify the start and end of each manoeuvre. We will set the minimum cluster size $m = 10$. We apply Algorithm 3 to identify clusters in the Cyclic test manoeuvres. In Figure 5.5.1 we have an elbow plot of the ordered log 10-nearest neighbourhood distances. There is an evident elbow at a distance of log(1000). Therefore the choice of $\epsilon = 1000$ seems reasonable for the DBSCAN algorithm. Using $\epsilon = 1000$ the algorithm identifies four clusters and labels 42 of the manoeuvres as noise. The manoeuvres labelled as noise typically appear at the beginning of the test as shown in Figure 5.5.2, which contains the N1 speed plot for the whole Cyclic test. The various clusters are coloured, with the noise manoeuvres in red. The noise manoeuvres at the start of the test are part of the 'shake-down' test performed before the cycles are performed. It's also worth noting that the manoeuvre classes tend to occur in groups.

In Figure 5.5.3 we have a plot of the aligned time series in each cluster. All the clustered manoeuvres have two fixed speed levels and a spike. The manoeuvres have distinctive speed levels, which suggest the classification approach outlined in Chapter 4 would be effective.

We can visualise the clusters using the tSNE mapping discussed in Chapter 2. A

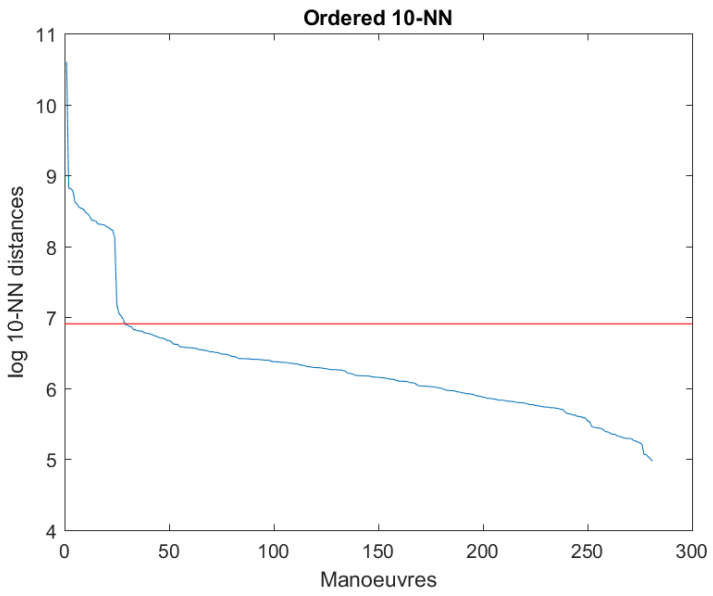



Figure 5.5.1: Ordered log 10-nearest neighbour distances with red line at 1000.

tSNE mapping plot of the clusters is given in Figure 5.5.4, with the points coloured based on the clusters from Algorithm 3. We can see the clusters are well separated, although there are some points that are near the clusters that may have been labelled as noise. We did a sensitivity analysis of the clustering results for different choices of $\epsilon$. For large $\epsilon$ then clusters merged together and noise manoeuvres were mislabelled, whilst for small $\epsilon$ we overestimated the number of noise manoeuvres. Choosing $\epsilon = 1000 \pm 100$ gave the same clustering results.

The manoeuvres are performed sequentially. To capture the time-dependent nature of the data we use a video showing the points arising over time, which can be found online (Hullait, 2019).

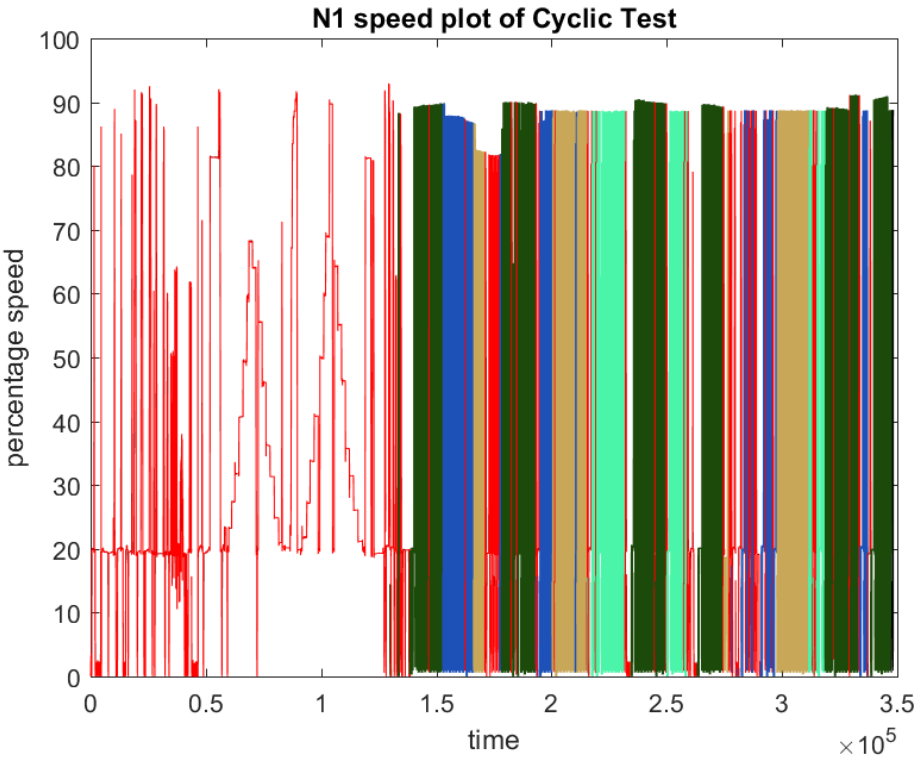
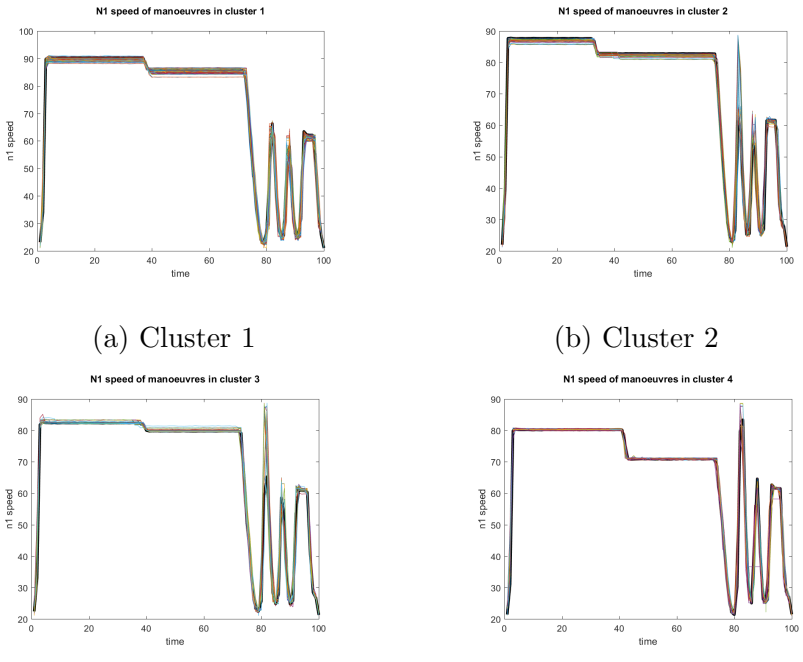



Figure 5.5.2: Cyclic test plot with manoeuvres coloured in with respect to the four clusters and the the noise manoeuvres are coloured in red, using DBSCAN with $\epsilon = 1000$.




(a) Cluster 1 (b) Cluster 2

(c) Cluster 3 (d) Cluster 4

Figure 5.5.3: Plots of aligned manoeuvre in each of the 4 clusters found using $\epsilon = 1000$ in the DBSCAN algorithm.

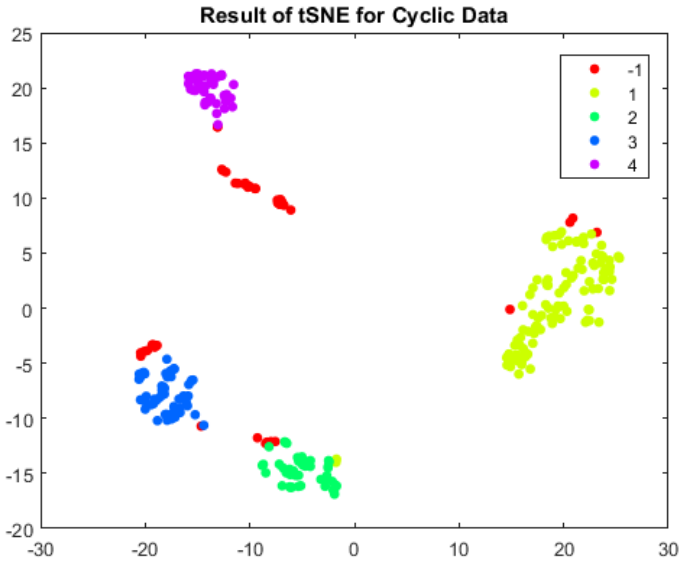



Figure 5.5.4: tSNE mapping for each manoeuvre in the Cyclic test with the four clusters coloured, including noise points in red

## 5.6 Clusters in Trent 1000 Pass-Off tests

The clustering procedure described in Algorithm 3 is effective in identifying manoeuvre classes in the Cyclic test. In this section we will test the clustering algorithm on manoeuvres in the Trent 1000 Pass-Off tests. In the test the engineers can perform 8 predefined manoeuvres: A,B,C,R,P,RP,F,V and occasionally perform an unspecified manoeuvre U. We can use the true manoeuvre labels to assess the clustering performance.

We have 981 manoeuvres in the 93 Pass-Off test datasets. The manoeuvres lengths are significantly longer than those in the Cyclic tests, and therefore using Dynamic Time Warping would be computationally impractical. We therefore shrink the manoeuvre time series by down-sampling by taking observations at every 200 points. This reduction is size maintains the general shape of the manoeuvres.

The elbow plot of the log 10-nearest neighbour distances is given in Figure 5.6.1, and by inspection we choose $\epsilon = \log(4000)$. Then applying DBSCAN we obtain 9 clusters shown in Figures 5.6.2. These clusters pick up the different classes defined earlier. However it splits P manoeuvres into two clusters, and likewise for the R manoeuvres. Cluster 7 is a manoeuvre we would have labelled as Unknown, whilst cluster 'C' defined in Chapter 4 does not appear as a cluster. The output from the clustering algorithm suggests that cluster 7 should perhaps be included in the classification algorithm as there are 17 instances of the manoeuvre.

In Figure 5.6.3 we have two plots of the tSNE mapping of the manoeuvres labelled with respect to the cluster assignment and with respect to the true labels. We can see that most of the clusters are distinct. Clusters 4 and 6 both contain P manoeuvres

Table 5.6.1: The number of each manoeuvre in Trent 1000 dataset identified as noise by DBSCAN.

| Manoeuvre | A | B | C | R | P | RP | F | V | U | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Number Noise | 4 | 2 | 10 | 26 | 8 | 1 | 25 | 3 | 90 | 169 |

however we can see the groups are distinctive, which explains the samples being split into two clusters. Clusters 8 and 9, look to be overlapping, which we would expect given they are both examples of R manoeuvres.

We have the true labels of the manoeuvres in the Pass-Off test, which we can use to assess the clustering performance. We will use the Normalised Mutual Information (NMI), outlined in Section 5.4. The NMI value is 0.8453, which shows that the clustering algorithm is able to distinguish the different classes effectively. The classification algorithm outlined in Chapter 4 achieves a NMI value of 0.9921, which is notably higher.

The algorithm overestimates the number of Unknown manoeuvres, labelling 169 manoeuvres as noise, when there are in fact 108 Unknown manoeuvres. The overestimation is likely due to the choice of $\epsilon$. In Table 5.6.1 we have a breakdown of the number of each manoeuvre type that was labelled as noise. We can see manoeuvre R and F are the most troublesome to cluster. The mislabelled RP manoeuvre is expected given there is only one instance. The P manoeuvres that are labelled as Unknown arise due to missing steps.

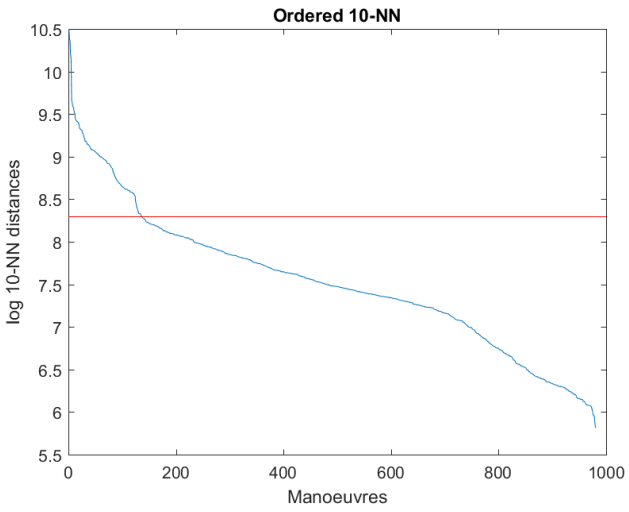
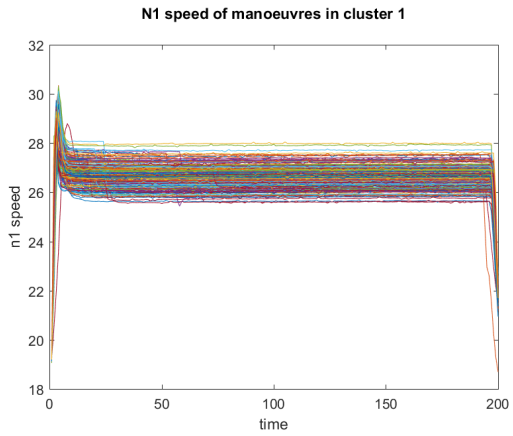



Figure 5.6.1: Ordered log $k$-nearest neighbour distances with line at $\epsilon = \log(4000)$ for Trent 1000 manoeuvres.

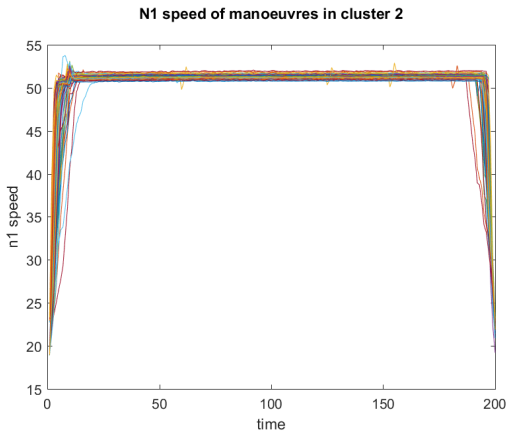
## 5.7 Clusters in XWB Pass-Off tests

In this section we apply the clustering algorithm on the manoeuvres performed in the XWB Pass-Off tests. We follow the same process as for the Trent 1000 manoeuvres in Section 5.6. We have 430 manoeuvres, that come from 7 classes: A,D,P,RP,F,V and U. In Algorithm 3 we choose the $\epsilon$ parameter using an elbow plot of the log 10-nearest neighbour distances. Looking at the elbow plot in Figure 5.7.1 there is not a clear 'elbow' point, however $\epsilon = \log(4000)$ is reasonable and is consistent with the choice of $\epsilon$ for the Trent 1000 engines. Applying DBSCAN we obtain 5 clusters shown in Figures 5.7.3. The classes are in general well identified, however the algorithm has merged the F and V manoeuvres together.

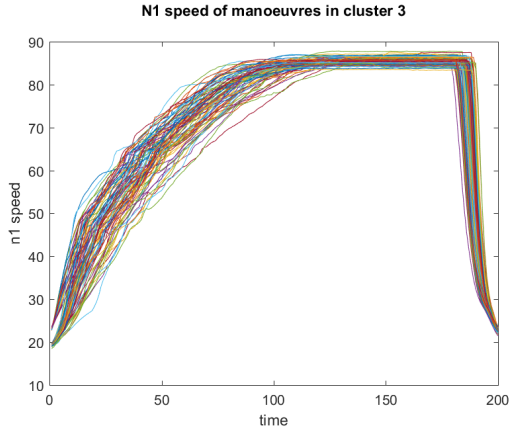
In Figure 5.7.2 we have a plot of the tSNE mapping of the manoeuvres coloured using the cluster labels and using the true labels. We can see that most of the classes



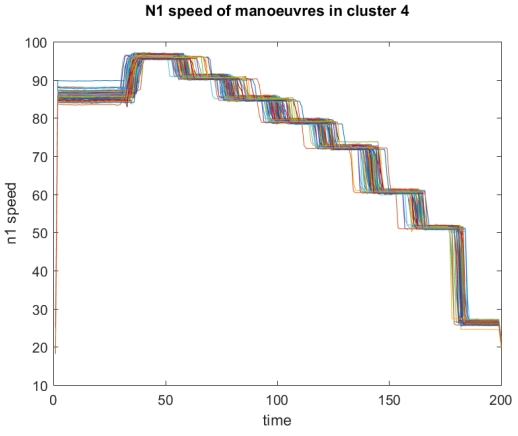

(a) Cluster 1



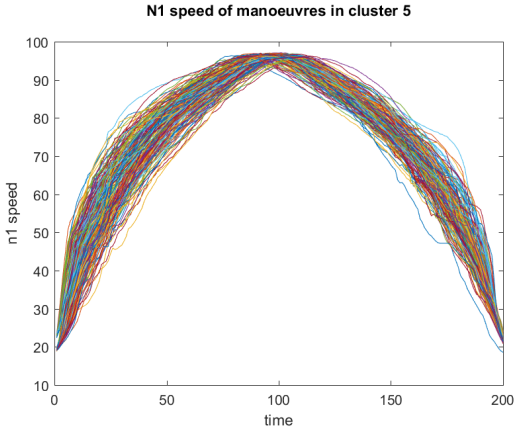

(b) Cluster 2



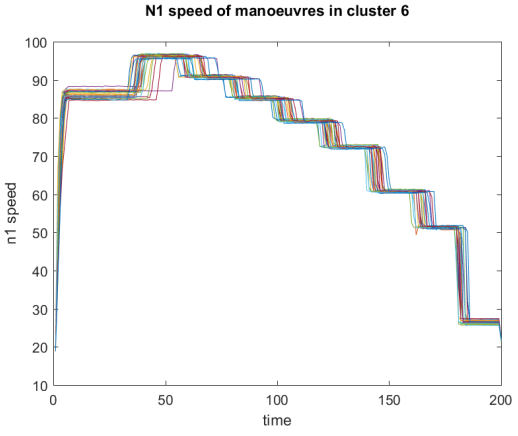

(c) Cluster 3



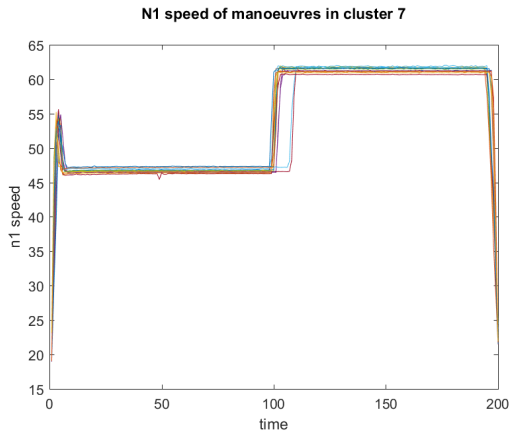

(d) Cluster 4



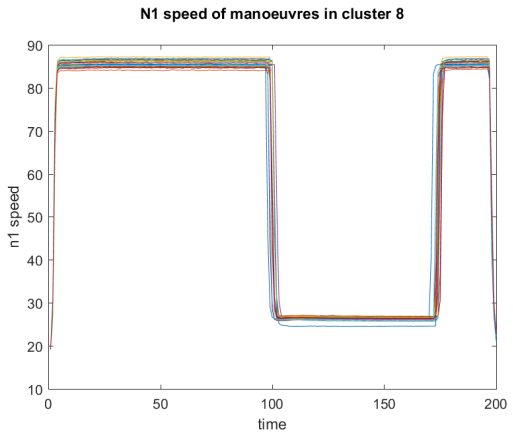

(e) Cluster 5



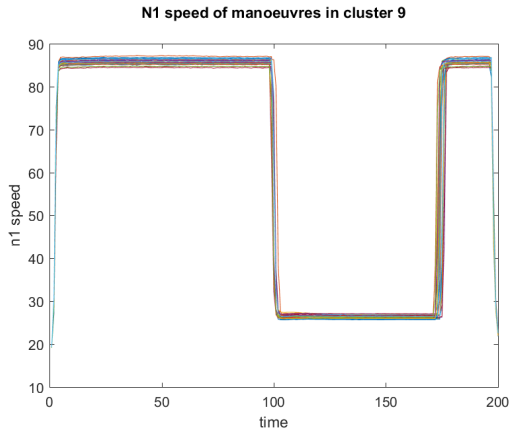

(f) Cluster 6




(g) Cluster 7



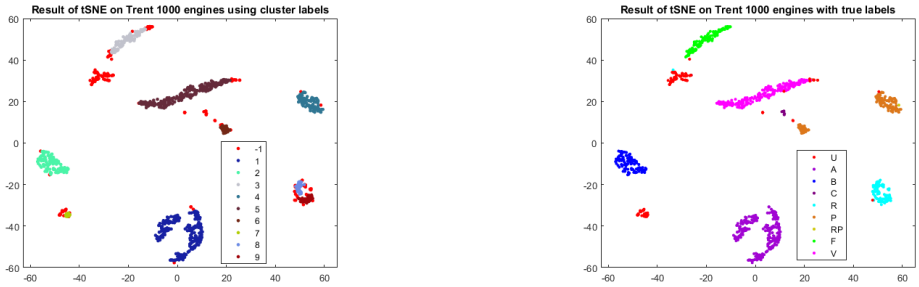

(h) Cluster 8




(i) Cluster 9

Figure 5.6.2: Plots of manoeuvre clusters found using $\epsilon = 4000$ in the DBSCAN algorithm for Trent 1000 manoeuvres.

are distinct, however the classes containing manoeuvres F and V are very close, which explains why they have been grouped together in Cluster 2. We have 42 Unknown manoeuvres however DBSCAN identifies 62 cases. In Table 5.7.1 we have a breakdown of the number of each manoeuvre class which were labelled as noise. As in the Trent 1000 Pass-Off tests the F manoeuvres are often mislabelled. Next we will use the Normalised Mutual Information (NMI) given in Section 5.4 to evaluate the cluster




(a) Cluster Labels (b) True Labels.

Figure 5.6.3: tSNE mapping of the manoeuvre in the Trent 1000 Pass-Off tests using cluster labels (left) and using true labels (right).

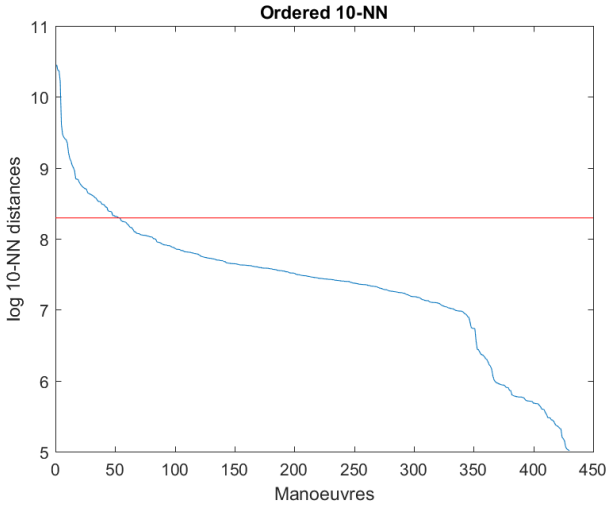
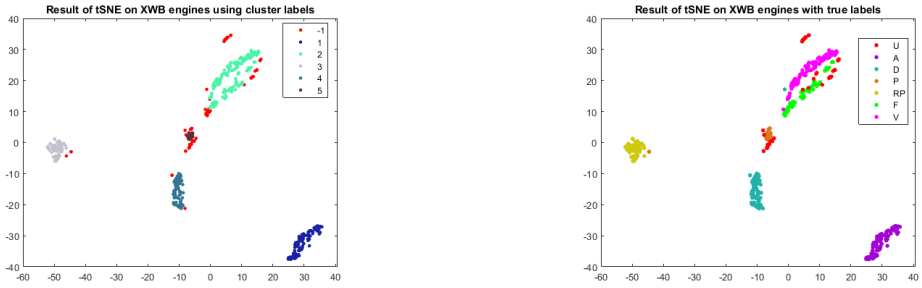



Figure 5.7.1: Ordered log $k$-nearest neighbour distances with line at $\epsilon = 4000$ for XWB manoeuvres.

performance. We have obtained a NMI value of 0.8137 that is notably smaller than the NMI value from the classification algorithm labels: 0.9012. The NMI value is also notably smaller than for the Trent 1000 manoeuvres.

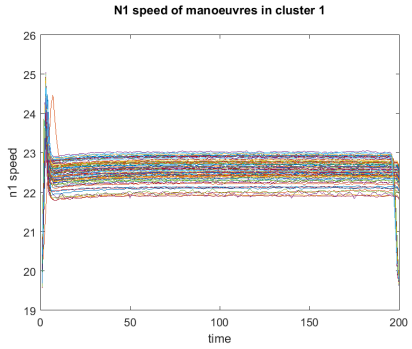



(a) Cluster Labels (b) True Labels.

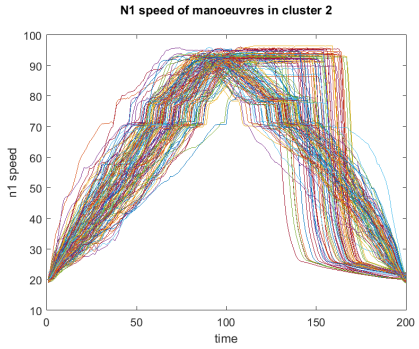
Figure 5.7.2: tSNE mapping of the manoeuvre in the XWB Pass-Off tests using cluster labels (left) and using true labels (right).

Table 5.7.1: The number of each manoeuvre in XWB dataset identified as noise by DBSCAN.

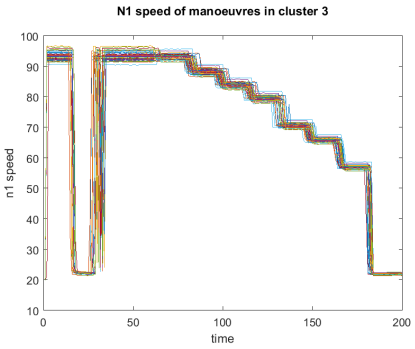
| Manoeuvre | A | D | P | RP | F | V | U | Total |
|---|---|---|---|---|---|---|---|---|
| Number Noise | 1 | 6 | 5 | 1 | 10 | 2 | 37 | 62 |



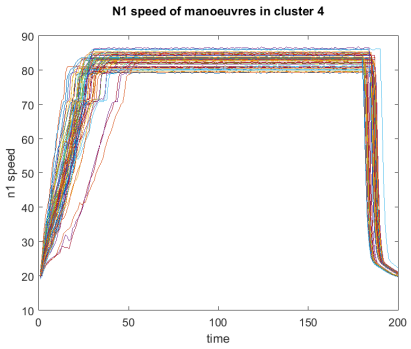

(a) Cluster 1



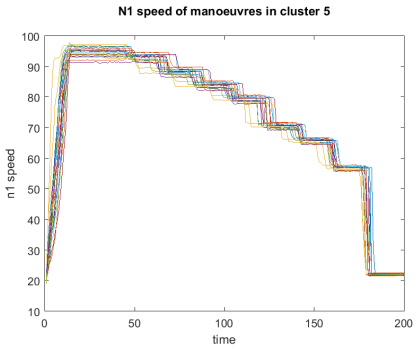

(b) Cluster 2




(c) Cluster 3




(d) Cluster 4




(e) Cluster 5

Figure 5.7.3: Plots of XWB manoeuvre in 5 clusters found using $\epsilon = 4000$ in the DBSCAN algorithm.

## 5.8 Discussion

We have built a general clustering procedure to identify manoeuvres types in a Cyclic test, which can also be applied to Pass-Off tests. The algorithm is able to effectively cluster the manoeuvres in the presence of outliers. The algorithm is relatively simple using a standard distance function and a classical clustering algorithm. The approach does not require prior information or a training set unlike the classification algorithm in Chapter 4. The clusters highlight high serial correlation within the Cyclic test, as manoeuvre types typically occur in groups, which we have not taken into account in the clustering algorithm. We can use the clusters identified to define the manoeuvre classes in a Cyclic test.

We have used the DTW to obtain distances between samples. This distance measure can deal with time series of different lengths and slight differences in shape. We then use a density based clustering algorithm: DBSCAN, to identify the clusters. DBSCAN gives effective clustering results whilst being able to identify outliers, which we know to be present in the test datasets. However DBSCAN relies on a parameter $\epsilon$. We choose $\epsilon$ using an elbow plot, however a more rigorous approach is required, highlighted by the overestimation of the number of Unknown manoeuvres.

The clustering algorithm was applied on manoeuvres in a Cyclic test. The clusters look reasonable, with samples in the same cluster typically following the same structure. The tSNE mapping shows the clusters are separable mitigating the potential issue of a single-link effect. Applying the algorithm on the manoeuvres in the Pass-Off tests, we have found the different class structure are identified. However in some cases the

classes merge or multiple clusters are formed for the same manoeuvre class. One way of mitigating this issue is to have an associated uncertainty measure for the cluster assignments. Ideally we would have a probability for a point being assigned to each cluster. Using the probability of a manoeuvre being in each cluster can aid in splitting or merging potential clusters.

# Chapter 6

# Robust Functional Linear Regression

*The material in this chapter is under submission at Technometics journal.*

## 6.1 Introduction

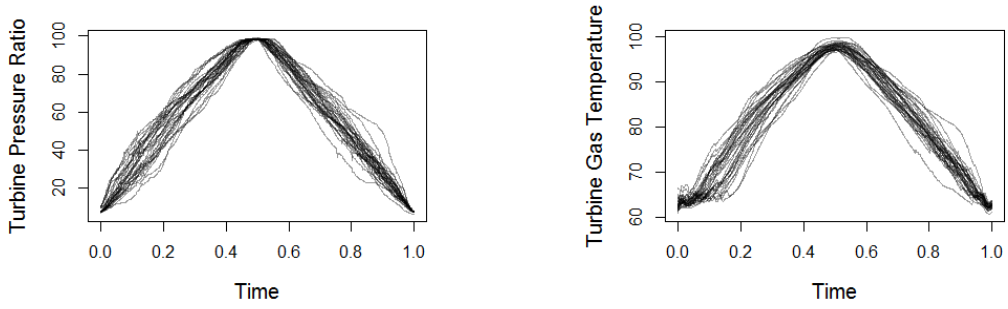
Functional Linear Regression (FLR) in the function-on-function case (Ramsay and Dalzell, 1991) is a widely used technique for modelling functional responses with respect to functional inputs. The FLR model is able to capture complex dependency structures as it uses information across time (Morris, 2015). However classical FLR models can be severely affected by outliers as we will demonstrate via a simulation study in Section 6.4. We therefore develop a robust FLR (RFLR) model, which is able to effectively fit the data in the presence of outliers. The model is built using the robust Functional Principal Component model by Bali et al. (2011) and

the multivariate Least Trimmed Squares (MLTS) estimator by Agulló et al. (2008). The RFLR model can be used to identify abnormal functional responses, i.e. samples in which the functional behaviour between the predictor and response curve deviates from normal.

Our study of FLR is motivated by a need to identify unusual temperature behaviour in jet engine sensor data collected during Pass-Off tests. In Chapter 1 we described the Pass-Off test and in Chapter 4 we built an algorithm to extract and label manoeuvres performed in the test. One of the key manoeuvres in a Pass-Off test is the Vibration Survey (V). In this manoeuvre the engine is accelerated slowly to a certain speed then slowly decelerated. We have 199 Vibration Survey datasets for the Trent 1000 engines and 92 for the XWB engines. The datasets include speed parameters such as the N1 speed and the turbine pressure ratio (TPR), and various temperature features including the turbine gas temperature (TGT). In Figure 6.1.1 we have plots of the TPR and TGT for 30 V manoeuvres in the Trent 1000 Pass-Off tests. To anonymise the data we have transformed the time index onto the interval $[0, 1]$ and the sensor measurements to the range $[0, 100]$.

For the classification algorithm in Chapter 4 we used the N1 speed parameter. The N1 speed gives stable measurements as it only relies on the shaft speed of the fan. However the N1 speed does not give a direct measurement of thrust. We will therefore instead use the TPR, which gives the actual thrust produced by the engine.

The V manoeuvres are performed by a human controller, which causes variability in the TPR curves as can be seen in Figure 6.1.1. This variability will naturally affect the TGT curves and may mask the unusual behaviour produced by the engine.




(a) TPR (b) TGT

Figure 6.1.1: Plots of 30 TPR (left) and TGT (right) time series.

We therefore require a method of detecting outliers in the presence of the controller induced variability. We expect that the relationship between the engine speed and engine temperature for different V manoeuvres should be the same irrespective of the way the manoeuvre is performed. For example given a certain engine acceleration we would expect a certain temperature response. If however the response differs from expectation this could be indicative of an engine issue. In Chapter 7 we will show how RFLR can be used for outlier detection, which we use to identify abnormal temperature behaviour in the jet engine datasets.

## 6.2 Robust Functional Linear Regression

In Section 3.3 we have defined the FLR model, which can be estimated using a pre-chosen basis. In particular we can use FPCA bases to estimate parameters of the model. In this section we will use robust FDA techniques to build a robust FLR model. This will allow us to fit a normality model even in the presence of outliers.

We shall also propose a robust BIC procedure for model selection. We will replace classical FPCA with robust FPCA estimates by Bali et al. (2011) which ensure that outliers do not unduly affect the FPCA estimates.

Let $x_i(t)$ and $y_i(t)$ be pairs of predictor and response functions respectively in $L^2(I)$ for $i = 1, ..., n$. We define the robust FPCs $\tilde{\phi}_m^X(t)$ $(m = 1, ..., M)$ and $\tilde{\phi}_k^Y(t)$ $(k = 1, ..., K)$ for $x_i$ and $y_i$ respectively. These orthonormal functions form a basis such that

$$x_i(t) \approx \sum_{m=1}^{M} \tilde{z}_{im}\tilde{\phi}_m^X(t), \quad y_i(t) \approx \sum_{k=1}^{K} \tilde{w}_{ik}\tilde{\phi}_k^Y(t),$$

are good approximations for $x_i(t)$ and $y_i(t)$.

We define $\tilde{y}_i(t) = \tilde{w}_i\tilde{\phi}^Y(t)$ and assume that $\epsilon_i = \tilde{q}_i\tilde{\phi}^Y(s)$. We can now write

$$\tilde{w}_i = \tilde{z}_i\tilde{B} + \tilde{q}_i. \tag{6.2.1}$$

To obtain a robust estimate of the regression matrix $\tilde{B}$, we will use the Multivariate Least Trimmed Squares (MLTS) estimator by Agulló et al. (2008), to mitigate the affect of outliers with respect to the regression relationship. Given $\alpha \in [0, 1]$ we can define $r = [\alpha n]$ as the $\alpha$ proportion of samples rounded to the nearest integer, and the set $\mathcal{S} = \{S \subset \{1, ..., n\}, |S| = r\}$. The objective of MLTS is to find a subset $S$ such that

$$S = \arg\min_{S \in \mathcal{S}} \sum_{i \in S} ||\tilde{w}_i - \tilde{z}_i\tilde{B}||^2.$$

This is robust as outliers will not be in the subset by definition so shall not affect the model estimation. We will choose a subset of size $r = [0.8n]$.

**Bayesian Information Criterion**

In this section we formulate a Bayesian Information Criterion (BIC) to determine the basis size $M$ and $K$, similarly to Matsui (2017). We will outline a robust extension of the BIC in Section 6.2.1. A component of the BIC is the log likelihood, often expressed as a squared error term. It is tempting to use the squared error resulting from Equation (3.3.4). However the objective is to fit the data $y_i$ which comes in the form of a discrete time series, so we should use a likelihood of this data instead of a squared error term of basis coefficients.

We have a set of models $J = \{(M, K)|M = 1, ..., M_{\max}, K = 1, ..., K_{\max}\}$, where $M_{\max}$ and $K_{\max}$ are pre-set maximum number of FPCs that will be considered in the model. Let vector $\vec{y_i}$ be the values of $y_i(t)$ evaluated at discrete time points: $\vec{y_i} = [y_i(t_1), ..., y_i(t_T)]$. Let $z_i^{(M)}$ be the first $M$ principal scores of $x_i(t)$ with respect to the FPCs $\phi^X(t)$ and let $\phi^{(K)}$ be the matrix with $(k, i)$ entry $\phi_k^Y(t_i)$. We assume there exists a true model $(M_0, K_0)$ with associated $M_0 \times K_0$ matrix $B^{M_0K_0}$ such that

$$\vec{y_i} = (z_i^{(M_0)})^T B^{M_0,K_0} \phi^{(K_0)} + \epsilon_i, \tag{6.2.2}$$

where the error $\epsilon_i = [\epsilon_i(t_1), ..., \epsilon_i(t_T)]$ is assumed for simplicity to be sampled from $N(0, v^2 I_T)$, where $I_T$ is the identity matrix of size $T$.

For Model $(M, K)$ we define $\theta^{M,K} = (B^{M,K}, v^{M,K})$ and the prediction $\hat{y}_i^{M,K} = (z_i^{(M)})^T B^{M,K} \phi^{(K)}$. We want to identify this true model $(M_0, K_0)$, which we can use to obtain consistent estimates of $\theta^{M_0,K_0}$.

For Model $(M, K)$ we can define the likelihood for sample $i$ as

$$f(\vec{y}_i|\theta^{M,K}) = \frac{1}{(2\pi)^{\frac{T}{2}}(v^{M,K})^T} \exp\left\{-\frac{[\vec{y}_i - \hat{y}_i^{M,K}]^T[\vec{y}_i - \hat{y}_i^{M,K}]}{2(v^{M,K})^2}\right\}, \tag{6.2.3}$$

and the log-likelihood $l(\theta^{M,K}) = \sum_{i=1}^{n} \log(f(\vec{y}_i|\theta^{M,K}))$. As in Eilers and Marx (1996)

$$BIC_n(M,K) = -2l(\theta^{M,K}) + w(M,K)\log(n) \tag{6.2.4}$$

where the penalty $\omega(M,K) = MK+1$, in which $MK$ is the number of free parameters in the model and the 1 comes from $v$. We will denote $(M^*,K^*)_n = \arg\min_{(M,K)\in J} BIC_n(M,K)$, which is dependent on the sample size $n$.

To summarise, we estimate the FPCs for $X$ and $Y$ and solve the FLR model for different models $(M,K)$. We then choose model $(M^*,K^*)_n$ that minimises the BIC criterion.

### 6.2.1 Robust Bayesian Information Criterion for FLR

The BIC model selection method is known to be non-robust (Machado, 1993). In particular outliers can significantly affect the loglikelihood estimation. We therefore outline a robust BIC (RBIC) model, which, similar to MLTS, maximises over a subset of samples $S$. RBIC can therefore give good model selection performance in the presence of outliers.

We will define $\tilde{\theta}^{M,K} = (\tilde{B}^{M,K}, \tilde{v}^{M,K})$ as robust estimated parameters for model $(M,K)$ and the robust prediction $\tilde{y}_i^{M,K} = (\tilde{z}_i^{(M)})^T\tilde{B}^{M,K}\tilde{\phi}^{(K)}$. We define the trimmed likelihood for model $(M,K)$ and set $S$ as

$$\tilde{l}(\tilde{\theta}^{M,K}, S) = \sum_{i\in S}\left(\frac{[\vec{y}_i - \tilde{y}_i^{M,K}]^T[\vec{y}_i - \tilde{y}_i^{M,K}]}{(\tilde{v}^{M,K})^2}\right) + rT\log(2\pi) + 2rT\log(\tilde{v}^{M,K}). \tag{6.2.5}$$

We will define $S^{M,K} = \arg\min_{S\in\mathcal{S}} \tilde{l}(\tilde{\theta}^{M,K}, S)$, where $\mathcal{S} = \{S \subset \{1, ..., n\}, |S| = r\}$ for $r = [0.8n]$. Then

$$RBIC_n(M, K) = -2 \min_{S\in\mathcal{S}} \tilde{l}(\tilde{\theta}^{M,K}, S) + \omega(M, K) \log(r) \tag{6.2.6}$$

$$= -2\tilde{l}(\tilde{\theta}^{M,K}, S^{M,K}) + w(M, K) \log(r) \tag{6.2.7}$$

We will denote $(\tilde{M}, \tilde{K})_n = \arg\min_{(M,K)\in J} RBIC_n(M, K)$, and we will assume that this minimum is unique.

In Algorithm 4 we outline the calculation of the robust FLR model, which incorporates the RBIC procedure. In the algorithm we estimate the model for different values of $(M, K)$ and choose the model with the minimum RBIC value. We consider $M = 1, \ldots, M_{\max}$ and $l = 1, ..., K_{\max}$ where $M_{\max}, K_{\max}$ are chosen to ensure that 99.99% of the variance in the raw data is captured.

**Algorithm 4** Robust FLR procedure

**INPUTS:** Centred time series $(x_i, y_i)$ of length $T$ for $i = 1, ..., n$,
Estimate $\{\tilde{\phi}_1^X(t), ..., \tilde{\phi}_{M_{\max}}^X(t)\}$, $\{\tilde{\phi}_1^Y(t), ..., \tilde{\phi}_{K_{\max}}^Y(t)\}$ (Bali et al., 2011).
**for** $M = 1, ..., M_{\max}$ **do**
    **for** $K = 1, ..., K_{\max}$ **do**
        Estimate the regression matrix $B^{M,K}$ using MLTS (Agulló et al., 2008).
        Calculate $RBIC_n(M, K) = \arg\min_{(M,K)\in J} RBIC_n(M, K)$ (6.2.6)
    **end for**
**end for**
Select model $(\tilde{M}, \tilde{K})_n$.
**RETURN:** $\tilde{B}$ from model $(\tilde{M}, \tilde{K})_n$ and $\{\tilde{\phi}_1^X(t), ..., \tilde{\phi}_{\tilde{M}}^X(t)\}$, $\{\tilde{\phi}_1^Y(t), ..., \tilde{\phi}_{\tilde{K}}^Y(t)\}$.

## 6.3 Asymptotic Results

In Section 6.2 we proposed a Robust FLR model for the function-on-function problem. A minimum criteria for a good model is consistency, i.e. that given an ideal scenario of

unlimited data that the estimator will be equal or arbitrarily close to the truth. In this section we shall prove consistency and Fisher-consistency for the robust FLR model. We shall follow a similar approach to Kalogridis and Aelst (2019) who developed a robust FLR model for the scalar-on-function problem. We shall also prove the consistency of the RBIC model selection method outlined in Section 6.2.

**Definition 6.3.1.** *Let $X_1, X_2, ..., X_n$ be a sequence of real-valued random variables. An estimator $T_n := T(X_1, X_2, ..., X_n)$ of a parameter $\theta$ is said to be (asymptotically)* ***consistent*** *if for all $\epsilon > 0$*

$$\lim_{n\to\infty} P(|T_n - \theta| > \epsilon) = 0.$$

**Definition 6.3.2.** *Let $X_1, X_2, ..., X_n$ be a sequence of real-valued random variables with an associated cumulative distribution function $F_\theta$, which depends on an unknown parameter $\theta$. Let the estimator $T_n := T(F_n)$ of a parameter $\theta$, be a function of the empirical distribution function $F_n$. We say this estimator is* ***Fisher-consistent*** *for the parameter $\theta$ if*

$$T(F_\theta) = \theta$$

**Remark 6.3.3.** *Fisher consistency is equivalent to (asymptotic) consistency if the empirical distribution function $F_n$ converges pointwise to the true distribution function $F_\theta$. This can be shown to be the case for iid real multivariate random variables using the Glivenko-Cantelli theorem (Pollard, 2012).*

### 6.3.1 Consistency of the Robust FLR

To prove Fisher-consistency we need to define appropriate probability measures on the predictor $X(t)$, response $Y(t)$ and the residual $\epsilon(t)$. We will then define conditions by which the robust FPCA and MLTS regression are Fisher-consistent, which will then ensure the Fisher-consistency of $\tilde{\beta}(s,t)$. We shall also prove consistency of $\tilde{\beta}(s,t)$ using Remark 6.3.3. Following the ideas set by Kalogridis and Aelst (2019), we make 6 assumptions:

(C1) $X$ has a finite-dimensional Karhunen-Loéve decomposition: $\lambda_m^X = 0$ for $m > M_0$.

(C2) $Y$ has a finite-dimensional Karhunen-Loéve decomposition: $\lambda_k^Y = 0$ for $k > K_0$.

(C3) The residual $\epsilon(t) = \tilde{q}\tilde{\phi}^Y(t)$ where $\tilde{q}$ is a Gaussian random variable with mean 0 and covariance matrix $\Sigma$.

(C4) $\beta(s,t)$ lies in a linear subspace spanned by $\{\tilde{\phi}_m^X\}_{m=1}^{M_0}$ and $\{\tilde{\phi}_k^Y\}_{k=1}^{K_0}$.

(C5) The random variables $\{\tilde{\xi}_j^X\}_{j=1}^{M_0}$ are absolutely continuous and have joint density $g_1(x)$ satisfying $g_1(x) = h_1(||x||_E)$ for $x \in \mathbb{R}^{M_0}$ and some measurable function $h_1 : \mathbb{R} \to \mathbb{R}_+$.

(C6) The random variables $\{\tilde{\xi}_j^Y\}_{j=1}^{K_0}$ are absolutely continuous and have joint density $g_2(y)$ satisfying $g_2(y) = h_2(||y||_E)$ for $y \in \mathbb{R}^{K_0}$ and some measurable function $h_2 : \mathbb{R} \to \mathbb{R}_+$.

We define $||\cdot||_E$ as the Euclidean norm.

Let $P_X$ be the image measure of $X$ i.e. $P_X(U) = P(X \in U)$ for a Borel set $U$, and likewise for $P_Y$. We can define the cumulative distribution functions

$$F_X(a_1, ..., a_{M_0}) := P_X(\tilde{\xi}_1^X \leq a_1, ..., \tilde{\xi}_{M_0}^X \leq a_{M_0}),$$
$$F_Y(b_1, ..., b_{K_0}) := P_Y(\tilde{\xi}_1^Y \leq b_1, ..., \tilde{\xi}_{K_0}^Y \leq b_{K_0}).$$

Let $F_\epsilon$ denote the distribution function of $\epsilon(t)$, which can be defined in the same way as $P_X$ and $P_Y$. We can write the functional of the robust estimator $\tilde{\beta}(s,t)$ as:

$$\tilde{\beta}(F_\epsilon, F_X, F_Y)(s,t) = \sum_{k=1}^{K_0}\sum_{m=1}^{M_0} \hat{B}_{km}(F_\epsilon, F_X, F_Y)\tilde{\phi}_m^X(F_X)(s)\tilde{\phi}_k^Y(F_Y)(t). \tag{6.3.1}$$

The functional is Fisher-consistent if $\tilde{\beta}(F_\epsilon, F_X, F_Y)(s,t) = \beta(s,t)$ for $s,t \in I$, which in turn follows from $\tilde{B}_{km}(F_\epsilon, F_X, F_Y) = B_{km}$, $\hat{\phi}_k^Y(F_Y)(t) = \phi_k^Y(t)$ and $\hat{\phi}_m^X(F_X)(t) = \phi_m^X(s)$. Conditions C1-C4 are to ensure the FLR problem can be defined by a finite number of terms. Kalogridis and Aelst (2019) show that Conditions C5 and C6 are sufficient for the Fisher-consistency of the robust FPCA estimators by Bali et al. (2011).

**Lemma 6.3.4.** *Assume C1-C6 holds then $\tilde{\beta}(F_\epsilon, F_X, F_Y)(s,t)$ is Fisher-consistent.*

*Proof.* Conditions C1-C2 and C5-C6 ensure Fisher-consistency of the robust FPCA estimators as shown by Bali et al. (2011), so $\tilde{\phi}^Y(F_Y)(t) = \phi^Y(t)$ and $\tilde{\phi}^X(F_X)(t) = \phi^X(t)$. By conditions C1-C2 we can write

$$Y(t) = c\tilde{\phi}^Y(F_Y)(t), \quad X(t) = Z\tilde{\phi}^X(F_X)(t)$$

Then

$$\int_I X(s)\tilde{\beta}(F_\epsilon, F_X, F_Y)(s,t)ds = \int_I Z\tilde{\phi}^X(F_X)(s)\tilde{\phi}^X(F_X)(s)^T\tilde{B}(F_\epsilon, F_X, F_Y)\tilde{\phi}^Y(F_Y)(t)ds \text{ using C4}$$
$$= Z\tilde{B}(F_\epsilon, F_X, F_Y)\tilde{\phi}^Y(F_Y)(t).$$

Using condition C3 we can write $\epsilon(t) = \tilde{q}\tilde{\phi}^Y(t)$ therefore

$$Z\tilde{B}(F_\epsilon, F_X, F_Y)\tilde{\phi}^Y(F_Y)(t) + \epsilon(t) = Z\tilde{B}(F_\epsilon, F_X, F_Y)\tilde{\phi}^Y(F_Y)(t) + \tilde{q}\tilde{\phi}^Y(F_Y)(t),$$

multiplying by $\tilde{\phi}^Y(F_Y)(t)$ and integrating over $t$ we obtain

$$Z\tilde{B}(F_\epsilon, F_X, F_Y) + \tilde{q}.$$

Agulló et al. (2008) show that Condition C3 implies the MLTS estimator is Fisher-consistent so $\tilde{B}(F_\epsilon, F_X, F_Y) = B$. Therefore $\tilde{\beta}(F_\epsilon, F_X, F_Y)(s,t)ds = \beta(s,t)$.

□

**Corollary 6.3.5.** *If $\{x_1(t), y_1(t)\}, ..., \{x_n(t), y_n(t)\}$ are iid samples with cumulative distribution function $(F_X, F_Y)$. Then, assuming C1-C6 holds, $\tilde{\beta}(s,t)$ is consistent.*

Note that $x_i(t)$ and $y_i(t)$ are defined on a finite number of eigenfunctions, so are defined by finite score vectors. Therefore Corollary 6.3.5 follows from Lemma 6.3.4 and Remark 6.3.3, which states almost sure convergence of the empirical distribution for iid multivariate random variables. In this case Fisher-consistency is equivalent to consistency.

### 6.3.2 Consistency of RBIC

We defined RBIC for the FLR problem in Section 6.2.1. In this section we will prove consistency of RBIC for the FLR problem. We will assume there is a true model, which we previously defined as $(M_0, K_0)$. We can then define overspecified and underspecified models in reference to this true model. We make some assumptions on the behaviour of the likelihood for these two model classes to prove consistency. We also denoted $(\tilde{M}, \tilde{K})_n = \min_{(M,K)\in J} RBIC_n(M, K)$, which we will assume is unique.

We will split the candidate models in $J$ into two sets, one is the overspecified models that include the true model $J_+ = \{(M, K) \in J | M \geq M_0 \text{ and } K \geq K_0\}$ and underspecified models $J_- = J_+^c \cap J$. Recall that $r = [\alpha n]$ for some $\alpha \in (0, 1)$, and the likelihood $\tilde{l}$ in (6.2.5) depends on $r$ terms.

**Assumption 1** For $(M, K) \in J_-$, there exists some $\varepsilon^{M,K} > 0$ such that

$$\lim_{n\to\infty} P\left[\frac{1}{r}(\tilde{l}(\tilde{\theta}^{M_0,K_0}, S^{M_0,K_0}) - \tilde{l}(\tilde{\theta}^{M,K}, S^{M,K})) > \varepsilon^{M,K}\right] = 1.$$

This is a reasonable assumption as the underspecified models should give a poorer fit to $y_i$ than the true model.

**Assumption 2** For $(M, K) \in J_+$, there exists some $\gamma^{M,K} > 0$ such that

$$\lim_{n\to\infty} P\left[\tilde{l}(\tilde{\theta}^{M,K}, S^{M,K}) - \tilde{l}(\tilde{\theta}^{M_0,K_0}, S^{M_0,K_0}) > \gamma^{M,K}\right] = 0.$$

This assumption states that the difference in the trimmed loglikelihood is less than a finite $\gamma$. The likelihood for the overspecified models and the true model should be close, given the true model is contained within the overspecified models, so the difference in the penalty terms will outweigh the difference in the likelihoods for large

enough $n$.

Note that in Assumption 1 we consider the average difference between the log-likelihoods, whereas in Assumption 2 we look at the total difference.

**Theorem 6.3.6.** *Given Assumptions 1 and 2 hold, and the true model $(M_0, K_0) \in J$ then $(\tilde{M}, \tilde{K})_n$ is a consistent estimator of $(M_0, K_0)$.*

*Proof.* For $j \in J_-$, we will show

$$\lim_{n\to\infty} P(\{RBIC_n(M,K) - RBIC_n(M_0,K_0)\} > 0) = 1. \tag{6.3.2}$$

By definition we can show that:

$$\begin{aligned}
&\lim_{n\to\infty} P\left(RBIC_n(M,K) - RBIC_n(M_0,K_0) > 0\right) \\
&= \lim_{n\to\infty} P\left(-2\left(\frac{\tilde{l}(\tilde{\theta}^{M,K}, S^{M,K}) - \tilde{l}(\tilde{\theta}^{M_0,K_0}, S^{M_0,K_0})}{r}\right) > -\frac{(\omega(M,K) - \omega(M_0,K_0))\log(r)}{r}\right).
\end{aligned}$$

We will label $H_r = -2\left(\frac{\tilde{l}(\tilde{\theta}^{M,K}, S^{M,K}) - \tilde{l}(\tilde{\theta}^{M_0,K_0}, S^{M_0,K_0})}{r}\right)$ and $G_r = \frac{(\omega(M,K) - \omega(M_0,K_0))\log(r)}{r}$.

Using $\varepsilon^{M,K}$ from Assumption 1, we can see that $-G_r < 2\varepsilon^{M,K}$ for sufficiently large $r$.

Using this and Assumption 1 we can show

$$\lim_{n\to\infty} P(H_r > -G_r) \geq \lim_{n\to\infty} P(H_r > 2\varepsilon^{M,K}) = 1.$$

Therefore $\lim_{n\to\infty} P\left(RBIC_n(M,K) - RBIC_n(M_0,K_0) > 0\right) = 1$ for $(M,K) \in J_-$.

For $(M,K) \in J_+ \backslash \{(M_0,K_0)\}$, we know that $\frac{1}{2}(\omega(M,K) - \omega(M_0,K_0))\log(r) > 0$ and is monotonically increasing. Therefore there exists $N$ such that for $r \geq N$

$$\frac{1}{2}(w(M,K) - w(M_0,K_0))\log(r) > \gamma^{M,K}. \tag{6.3.3}$$

We can show that

$$\begin{aligned}
&\lim_{n\to\infty} P\left(RBIC_n(M,K) - RBIC_n(M_0,K_0) < 0\right) \\
&= \lim_{n\to\infty} P\left(\left[\tilde{l}(\tilde{\theta}^{M,K}, S^{M,K}) - \tilde{l}(\tilde{\theta}^{M_0,K_0}, S^{M_0,K_0})\right] > \frac{1}{2}(\omega(M,K) - \omega(M_0,K_0))\log(r)\right) \\
&\leq \lim_{n\to\infty} P\left(\left[\tilde{l}(\tilde{\theta}^{M,K}, S^{M,K}) - \tilde{l}(\tilde{\theta}^{M_0,K_0}, S^{M_0,K_0})\right] > \gamma^{M,K}\right) = 0 \text{ by Assumption 2.}
\end{aligned}$$

□

Note that BIC is a special case of RBIC where $r = n$, so is also consistent by Theorem 6.3.6.

## 6.4 Simulation Study

In this section we will provide a simulation study to investigate the finite sample properties of RBIC and robust FLR (RFLR) in comparison to BIC and classical FLR (CFLR). In the simulation study we will generate data using a FLR process and corrupt a certain number of samples, which will be the outliers. The outliers have been designed to be undetectable, if the response curves are considered independently of the predictor curves. Therefore standard functional data outlier detection algorithms such as those we will discuss in Section 3.5 will perform poorly.

The main motivation for the RFLR model is to obtain good model fitting in the presence of outliers. In this simulation study we compare the fitting error (FE) given in (6.4.1), for the non-outlier samples using the robust model, which uses RFLR and RBIC with the classical approach using CFLR and BIC. We define the indicator

variable $u_i = 1$ if sample $i$ is an outlier and 0 otherwise. Letting $\hat{y}_i(t)$ be the estimation of $y_i(t)$ and given that proportion $a$ of the samples have been contaminated then FE is given by:

$$FE = \frac{1}{(1-a)n} \sum_{i=1}^{n} (1 - u_i)||y_i - \hat{y}_i||^2. \tag{6.4.1}$$

Next we compare the outlier detection capabilities of robust and classical approaches using the receiver operating characteristic (ROC) curve to determine the sensitivity/specificity trade-off for different thresholds. If we have perfect outlier detection for all thresholds then the area under the curve (AUC) of the ROC curve would be 1. We can therefore use the AUC value as a measure of outlier detection accuracy regardless of threshold.

FPCA is performed by taking the principal components of a 200 cubic B-spline representation of each of the predictor and response curves (Ramsay and Silverman, 2005). The robust FPCA approach outlined in Section 6.2 is performed using the CR algorithm proposed by Croux and Ruiz-Gazen (1996) on the same B-spline coefficients. The MLTS estimator is calculated using the heuristic given by Agulló et al. (2008) using different trimming proportions $(1 - \alpha)$ for $\alpha \in [0, 1]$.

### 6.4.1 Scenarios

We will generate samples $x(t)$ using a FPCA based model with mean function $\mu_X(t) = -10(t - 0.5)^2 + 2$ for $t \in [0, 1]$ and eigenfunctions:

$$\phi_1^X = \sqrt{2}\sin(\pi t), \;\; \phi_2^X = \sqrt{2}\sin(7\pi t), \;\; \phi_3^X = \sqrt{2}\cos(7\pi t).$$

The principal scores are sampled from Gaussian distributions with mean 0 and variances 40, 10 and 1 for the eigenfunctions respectively. Note that we do not create any outliers in the FPCA decompositions of the predictor curves. We generate 400 predictor curves $x_1(t), ..., x_{400}(t)$, which are observed at $T = 500$ equidistant points in the interval $[0, 1]$.

The samples $y(t)$ will have eigenfunctions:

$$\phi_1^Y = \sqrt{2}\sin(12\pi t), \;\; \phi_2^Y = \sqrt{2}\sin(5\pi t), \;\; \phi_3^Y = \sqrt{2}\cos(2\pi t),$$

and mean function $\mu_Y(t) = 60\exp(-(t-1)^2)$. We will generate $\beta(s,t) = \phi^X(s)^T B\phi^Y(t)$ where $B$ will have random entries between $[-3, 3]$. We generate non-outlier curves:

$$y_i(t) = \mu_Y(t) + \int_I \beta(s,t)(x_i(s) - \mu_X(s))ds + \epsilon_i(t),$$

where the residual function $\epsilon_i(t) = q_i\phi^Y(t) + d_i$ where $q_i$ and $d_i$ are sampled iid from $N(0, 0.1)$. We will consider three cases when the proportion of outliers are $a = 0.1, 0.2$ and 0.3.

In **Scenario 1** outliers will be generated by replacing $B$ with $B_1 = B + R$ where $R$ has random entries sampled from $N(0, 0.5)$ giving $\beta_1(s,t) = \phi^X(s)^T B_1\phi^Y(t)$. Outliers $y_i'(t)$ are given by

$$y_i'(t) = \mu_Y(t) + \int_I \beta_1(s,t)(x_i(s) - \mu_X(s))ds + \epsilon_i(t).$$

In **Scenario 2** we generate outliers by adding a random B-spline function $p(t)$ defined on an interval of length 1/10. Letting $\beta_2(s,t) = \phi^X(s)^T B_2[\phi^Y(t), p(t)]$, for $3 \times 4$ matrix $B_2 = [B, l]$ for $l \sim N(2, 1)$, then the outliers $y_i''(t)$ are given by

$$y_i''(t) = \mu_Y(t) + \int_I \beta_2(s,t)(x_i(s) - \mu_X(s))ds + \epsilon_i(t).$$

Note that the outliers in Scenario 1 affect the regression function across the entire interval whereas the outliers in Scenario 2 only affect a small interval of the curves.

In Figure 6.4.1 we have a plot of the predictor curves $x_i(t)$ and response curves $y_i(t)$ with outliers from Scenario 1 and Scenario 2. The figure shows the outliers are masked by the variability in the curves and therefore cannot by identified using standard outlier detection algorithms. To make the outliers clearer we have plotted the residuals of the response curves using the true regression function and mean functions. In the bottom row of Figure 6.4.1 we can see that the outliers in Scenario 2 are localised to a fixed interval whereas in Scenario 1 the outliers affect the response curve at all time points.

The RFLR model depends on the proportion of trimming $\alpha$. To investigate the effect of the trimming we will consider trimming proportions $\alpha = 0.1, 0.2$ and 0.3. We shall also investigate the performance using BIC and RBIC with fixed trimmed sample size of $r = [0.8n]$.

We sample 400 predictor and response curve datasets and generate classical and robust models to calculate the average FE (6.4.1). In Tables 6.4.1 and 6.4.2 we present the results for Scenario 1 and 2 respectively. The CFLR model gives a smaller FE value in the case of no-outliers $a = 0$, however the robust model still gives good model fits. If we compare the FE using BIC and RBIC, we can see that BIC gives better model choices when $a = 0$. This is due to BIC using all the data and in particular

Table 6.4.1: Average fitting errors (FE) for 100 replications for Scenario 1, using classic FPCA and robust FPCA with different amount of trimming in the MLTS estimator and using models selected by BIC and RBIC.

| | Trim | Model | a=0 | a=0.1 | a=0.2 | a=0.3 |
|---|---|---|---|---|---|---|
| Classic | $\alpha = 0.0$ | BIC | **5.326** | 18.441 | 48.771 | 101.320 |
| Robust | $\alpha = 0.1$ | BIC | 8.283 | 14.166 | 21.118 | 33.907 |
| | $\alpha = 0.1$ | RBIC | 9.285 | **9.179** | 10.674 | 28.393 |
| | $\alpha = 0.2$ | BIC | 8.288 | 14.178 | 15.750 | 16.623 |
| | $\alpha = 0.2$ | RBIC | 9.292 | 9.207 | **9.535** | 13.436 |
| | $\alpha = 0.3$ | BIC | 8.294 | 14.199 | 15.815 | 16.518 |
| | $\alpha = 0.3$ | RBIC | 9.301 | 9.214 | 9.544 | **12.334** |

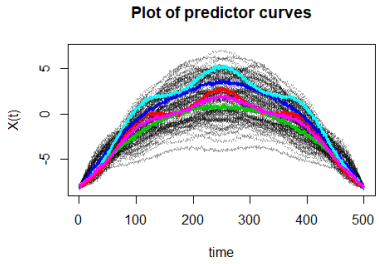
using samples in the tails of the distribution. In the presence of outliers the robust model outperforms the classical model, and as expected the difference in FE increases as the number of outliers increases. We should also note that RBIC is giving better model choices than BIC when outliers are present. Next, we can see using trimming proportion $\alpha = 0.1$ we obtain significantly large FE values when $a = 0.3$. However the FE values for $\alpha = 0.2$ and 0.3 are very similar in the case of $a = 0.3$. The outliers generated can have different sizes, therefore in the $\alpha = 0.2$ robust model only small outliers are present, which only affect the model fitting slightly .

Table 6.4.2: Average fitting errors (FE) for 100 replications for Scenario 1, using classic FPCA and robust FPCA with different amount of trimming in the MLTS estimator and using models selected by BIC and RBIC.

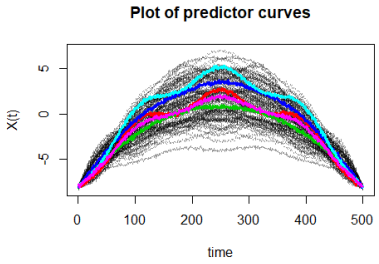
| | Trim | Model | a=0 | a=0.1 | a=0.2 | a=0.3 |
|---|---|---|---|---|---|---|
| Classic | $\alpha = 0.0$ | BIC | **5.326** | 17.252 | 48.906 | 85.063 |
| Robust | $\alpha = 0.1$ | BIC | 8.283 | 15.242 | 21.524 | 28.758 |
| | $\alpha = 0.1$ | RBIC | 9.285 | **9.074** | 9.919 | 18.546 |
| | $\alpha = 0.2$ | BIC | 8.288 | 16.745 | 20.652 | 21.928 |
| | $\alpha = 0.2$ | RBIC | 9.292 | 9.191 | **8.997** | 13.628 |
| | $\alpha = 0.3$ | BIC | 8.294 | 16.808 | 20.695 | 21.750 |
| | $\alpha = 0.3$ | RBIC | 9.301 | 9.233 | 9.018 | **11.439** |

## 6.5 Conclusion

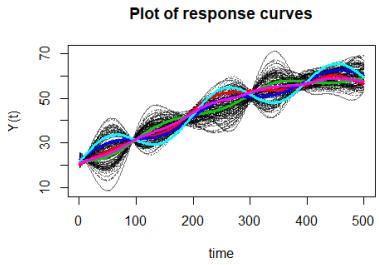
We have built a robust Functional Linear Regression (FLR) model for functional responses and introduced a robust model selection procedure. The robust procedure has been shown to be Fisher and asymptotically consistent. Then using a simulation study we have shown that the robust model significantly outperforms the classical model in the presence of outliers. In Chapter 7 we will show the residuals from the robust FLR model can be used to identify outliers.



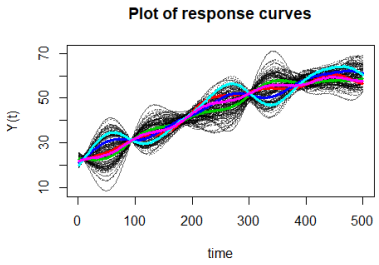

(a) $x_i(t)$



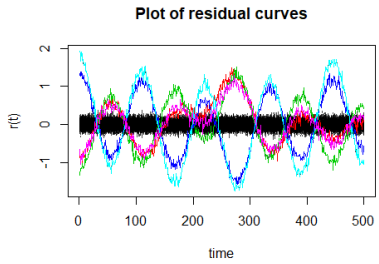

(b) $x_i(t)$



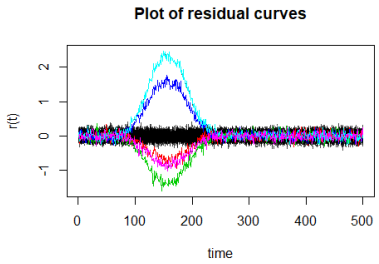

(c) $y_i^{(1)}(t)$




(d) $y_i^{(2)}(t)$




(e) $r_i^{(1)}(t)$




(f) $r_i^{(2)}(t)$

Figure 6.4.1: *Left:* Plots of the predictor curves $x_i(t)$, response curves $y_i^{(1)}(t)$ and residuals curves $r_i^{(1)}(t)$ for Scenario 1. *Right:* Plots of the predictor curves $x_i(t)$, response curves $y_i^{(2)}(t)$ and residuals curves $r_i^{(2)}(t)$ for Scenario 2. The residual curves are generated using the true regression function and mean functions. In each scenario there are 5 outliers each in a distinctive colour.

# Chapter 7

# Outlier Detection using Functional Regression

*The material in this chapter has been presented at the "Workshop on Advanced Analytics and Learning on Temporal data" at The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2019.*

## 7.1 Introduction

In Chapter 6 we have outlined a function-on-function robust Functional Linear Regression (RFLR) model. One of the motivations for the model was to identify outliers in the temperature behaviour in the jet engines. We will use the RFLR model to define "normal" engine behaviour. We can then use the residuals from this model to identify outlying behaviour. To identify outliers we will apply functional depth, which we have defined in Section 3.4. The depth values give an ordering of the samples. We will show

in conjunction with the RFLR model that the depth values give a good separation of the normal and abnormal samples.

In Chapter 2 we have outlined the novelty detection approaches used for jet engine data. The standard approaches require a training set of 'normal' samples to build a normality model. They then apply novelty detection using an appropriate distance measure and threshold. We instead use Functional Data Analysis (FDA) methods to identify Vibration Survey (V) manoeuvres that display unusual temperature behaviour in response to the variable (human-controlled) TPR time series. We will robustly build a normality model thereby not requiring a set of 'normal' samples. FDA techniques have been used effectively to model sensor data (Morris, 2015), as they combine information across samples and exploit the underlying behavioural structure. However this is to the best of our knowledge the first time these techniques are being used for modelling jet engine data.

In Section 3.5 we discussed various outlier detection approaches for functional data. None of the outlier detection approaches are able to model the dependency between the functional response and functional input, and may therefore miss important outliers. RFLR can model this dependency structure, which can improve the detection of outliers. We therefore suggest an outlier detection algorithm which uses RFLR to model the dependency structure. Using residuals from the model we can apply standard outlier detection approaches. The outliers in the residuals will be samples that display abnormal temperature behaviour with respect to engine speed.

We shall outline our outlier detection approach in Section 7.2. We will use the same simulation setup given in Chapter 6, which focused on the fit of the RFLR model to

the normal samples. In Section 6.4 the simulations will focus on outlier detection. In Section 7.4 we apply the outlier detection algorithm on jet engine data from Pass-Off tests performed on Trent 1000 and XWB engines. We focus on outlier detection of the V manoeuvres extracted using the classification algorithm in Chapter 4. Manoeuvre V is a natural choice given the smooth trajectories and the large number of samples.

## 7.2 Outlier Detection using RFLR

The RFLR model produces estimates of the responses $\tilde{y}_i(t) = \tilde{z}_i \tilde{B} \tilde{\phi}^Y(t)$ for $i = 1, ..., n$. For an outlier we expect the residual curve $r_i(t) = y_i(t) - \tilde{y}_i(t)$ to deviate in behaviour from the other residuals. Traditionally, we would use the integrated square error to identify outliers. However using functional depth is more effective in identifying shape outliers. We apply the outlier detection approach by Febrero-Bande et al. (2008) to the residuals from the RFLR model. We describe the outlier detection algorithm in Algorithm 5.

We need to choose a depth function for the outlier detection algorithm. We have chosen to use the $h$-modal depth (Cuevas et al., 2007) to rank samples $r_i$, as it satisfies most of the desirable properties of a functional depth defined in Section 3.4. The $h$-modal depth also captures distance i.e. a sample that is twice as far from the centre as another sample will have a proportionally lower depth value. For a given kernel $G_h$ (typically Gaussian with bandwidth $h$), the $h$-modal depth of $r_i$ with respect to $r = \{r_1, ..., r_n\}$ is given by:

$$D(r_i|r,h) = E(G_h(||r_i - r||)) \approx \frac{1}{n}\sum_{l=1}^{n} G\left(\frac{||r_i - r_l||}{h}\right). \tag{7.2.1}$$

Further details are given in Section 3.4. The $h$-modal depth has two useful properties. First, it uses a distance metric therefore samples further away from the centre will be given a smaller depth value. Second, in the case of multiple "normal" types behaviour, the $h$-modal depth works effectively as it doesn't assume there is one centre. Febrero-Bande et al. (2008) also show in their simulation studies that the $h$-modal depth outperforms the FM and random projection depth functions in regards to false outlier detection rate.

The $h$-modal depth has one further appealing property. Suppose $z = a^T\theta(t)$ and $x_i = b_i^T\theta(t)$. Then

$$\begin{aligned}
||z - x_i||_2^2 &= \int_I [z(t) - x_i(t)]^2\, dt \\
&= \int_I \left[a^T\theta(t) - b_i^T\theta(t)\right]^2 dt \\
&= \int_I \left[(a - b_i)^T\theta(t)\right]^2 dt \\
&= [a - b_i]^T \int_I \theta(t)\theta(t)^T dt\, [a - b_i] \\
&= [a - b_i]^T\, I\, [a - b_i] \\
&= [a - b_i]^T\, [a - b_i] \\
&= ||a - b_i||^2
\end{aligned}$$

where $||\cdot||$ without the suffix is the finite dimensional Euclidean norm. The $h$-modal

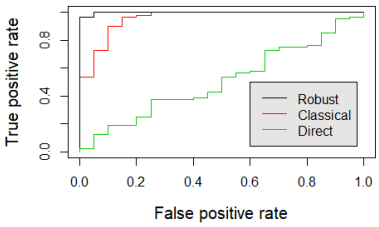
depth in Equation (7.2.1) becomes a standard multivariate Kernel density estimation with respect to the basis coefficients. This means that we can calculate the depth using only the basis coefficients.

**Algorithm 5** Outlier Detection using Robust FLR

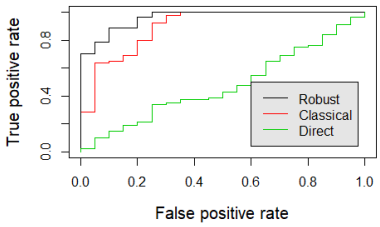
**INPUTS:** Centred curves $\{x_i(t), y_i(t)\}$ for $i = 1, ..., n$ and percentile $\delta$,
Use Algorithm 4 to obtain $\tilde{\phi}_k^Y(t)$, $\tilde{z}_m$ and $\tilde{B}$.
**for** $i = 1 : n$ **do**
    Calculate residual curves $r_i(t)$.
**end for**
Calculate depth values $d$ for $(r_1(t), ..., r_n(t))$
Set bandwidth $h$ be 15% percentile of depth values $d$
**for** $i = 1 : n$ **do**
    **if** $D(r_i|r, h) < C$ **then**
        Sample $i$ is labelled as an outlier.
    **end if**
**end for**
**RETURN:** List of outliers and depth values $d$.

## 7.3 Simulation Study

We will use the same simulation study given in Chapter 6. In Chapter 6 we focused on the model fit of the robust estimators. In this section we will test the outlier detection capabilities of the robust FLR. We will compare the depth based outlier detection (Direct) (Febrero-Bande et al., 2008) to the FLR models. In Figure 7.3.1 we have ROC curve generated for one of the repetitions in Scenario 1 and 2 in which we have contaminated 20% of the samples. In both scenarios the robust model outperforms the classical model. We can also see that using the Direct approach performs poorly. The ROC curves also show that the robust and classical models are more effective in identifying the outliers in Scenario 1 and 2. By only using the specificity and




(a) Scenario 1




(b) Scenario 2

Figure 7.3.1: ROC curve for one instance of Scenario 1 and 2 with 20% of the samples contaminated.

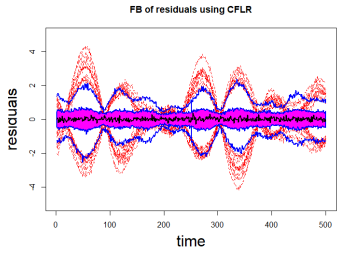
sensitivity for a fixed threshold a lot of information is being lost, therefore a better comparison would be the area under the curve (AUC). Using the AUC metric we can understand the model outlier detection capabilities overall, in particular how well are the outliers separated from the other samples. We have taken the average AUC values over the 100 iterations performed for Scenario 1, which are shown in Table 7.3.1. We have considered the average AUC values for trimming levels $\alpha = 0.1, 0.2$ and 0.3. The robust models give larger AUC values than the classical model. However the different trimming levels does not seem to have a significant effect on the AUC values. In Scenario 2 we have the results in Table 7.3.2. The same patterns appear as in Scenario 1 except the the AUC values are notably smaller.

Table 7.3.1: Average AUC values over 100 replications for Scenario 1, using Direct compared to classic FPCA with BIC, and using robust FPCA with RBIC. We will use trimming levels $\alpha = 0.1, 0.2, 0.3$ and contaminate different proportions of the samples $a = 0.1, 0.2, 0.3$.

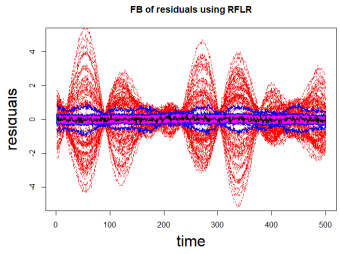
| | Trim | a=0.1 | a=0.2 | a=0.3 |
|---|---|---|---|---|
| Direct | - | 0.532 | 0.538 | 0.550 |
| Classic | $\alpha = 0.0$ | 0.960 | 0.898 | 0.797 |
| | $\alpha = 0.1$ | 0.995 | 0.991 | 0.953 |
| | $\alpha = 0.2$ | **0.996** | **0.996** | 0.987 |
| | $\alpha = 0.3$ | **0.996** | **0.996** | **0.990** |

Table 7.3.2: Average AUC values over 100 replications for Scenario 2, using Direct compared to classic FPCA with BIC, and using robust FPCA with RBIC and trimming levels $\alpha = 0.1, 0.2$ and 0.3.

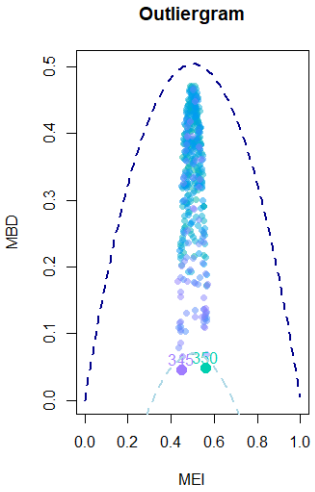
| | Trim | a=0.1 | a=0.2 | a=0.3 |
|---|---|---|---|---|
| Direct | - | 0.512 | 0.548 | 0.554 |
| Classic | $\alpha = 0.0$ | 0.922 | 0.838 | 0.734 |
| | $\alpha = 0.1$ | **0.985** | 0.964 | 0.932 |
| | $\alpha = 0.2$ | 0.980 | **0.980** | 0.966 |
| | $\alpha = 0.3$ | 0.980 | **0.980** | **0.968** |



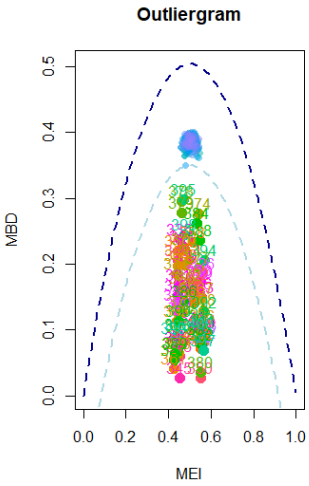

(a) Functional Boxplot



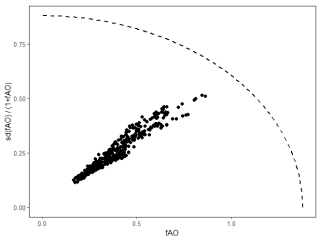

(b) Functional Boxplot



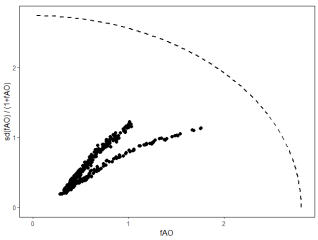

(c) Outliergram




(d) Outliergram



(e) FOM



(f) FOM

Figure 7.3.2: Plots of the Functional Boxplots, the Outliergrams and the Functional Outlier Map (FOM) for the residuals using CFLR (left) and RFLR (right) for one instance of Simulation 1 with 20% of the data contaminated. In the Functional Boxplot the median function is in black, the 0.5-central region $C_{0.5}$ is in purple with the fences in blue, the outliers are coloured in red. In the Outliergrams the thresholds are the dotted lines and outliers lie outside the thresholds. In the FOM plots have a parabolic threshold given by dotted line.

## 7.4 Jet Engine data

Our motivation behind the robust FLR model is to identify outliers in the temperature parameters for manoeuvres performed in a Pass-Off test. We have already extracted the manoeuvres from the Pass-Off test data using the classification algorithm given in Chapter 4. We will focus on the Vibration Survey (V) manoeuvre, which has a distinctive shape with a slow acceleration and a slow deceleration, with examples shown in Figure 6.1.1. For the Trent 1000 Pass-Off tests we have 199 V manoeuvres. For the XWB Pass-Off tests we have 92 V manoeuvres. We do not have labels for whether any of the individual engines have outliers but we do have log books from the engine test, which we can use obtain insights into the abnormal V manoeuvres. We have five temperature readings T25, T30, TGT, TCAR and TCAF, from sensors measuring temperature in different parts of the engine. All the temperature features for Trent 1000 engine are shown in Figure 7.4.1. The TCAR is particularly interesting as it has two distinct curve behaviours. It is also worth noting that the temperature values are distinctively higher at the end of the manoeuvre than at the beginning even though the engine speeds are the same. This highlights the trajectory-dependent behaviour that we seek to model. The V manoeuvres time series are of similar length. To standardise we have fitted a B-spline basis of 400 basis functions to each to ensure the time series are well approximated. Then we have taken 1000 equally spaced points on the B-spline representations to be our inputs $x_i(t)$ and $y_i(t)$.

We will be applying the outlier detection algorithm described in Algorithm 5, which uses RFLR. We will compare these outliers with those detected using CFLR

and BIC in Algorithm 5. We can look at the residuals curves to determine if the outliers do indeed look abnormal. We will apply the depth based outlier detection (Direct) (Febrero-Bande et al., 2008) directly on the temperature curves (with a default threshold of $\delta = 0.01$), and on the TPR speed curves. If abnormal speed profiles cause abnormal temperature profiles as we have conjectured then the outliers using the Direct approach should be the same for the TPR and the temperature parameters. In particular we want to show that our robust functional regression model is able to determine outliers that would otherwise be missed by investigating the temperature curves directly.

### 7.4.1 Vibration Surveys in Trent 1000 engines

In this section we will apply the outlier detection model using robust FLR on the V manoeuvres extracted from the Trent 1000 Pass-Off tests. In Table 7.4.1, we have the outliers detected using the Direct approach, using a classical approach with CFLR and BIC and finally using our outlier detection approach with robust FLR given in Algorithm 5. For each of the three approaches we determined a threshold using $\delta = 0.01$. We can see that the outliers in the TPR are the same as the outliers in the temperature features. This suggests the outliers being identified are arising from the controller induced variability. We therefore need to model the dependency between the control feature (TPR) and the temperature features.

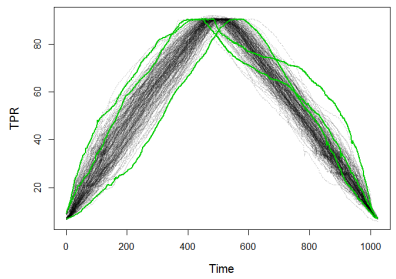
The residual curves from the classical approach are shown in Figure 7.4.2, with the outliers coloured in blue. It is not clear from this plot that the outliers are truly different from the other data. In Figure 7.4.3 we have the residual curves using RFLR.

We can see that the RFLR model fits the majority of the temperature curves well. The outliers that are picked up clearly look abnormal, with significant deviations from the general behaviour. The RFLR model is therefore able to identify interesting behaviour, which may otherwise have been undetected. Engineers have informed us that Sample 24 comes from an engine in which they detected damaged hardware. All the other outliers in the RFLR column of Table 7.4.1 were also noted to come from engines that displayed odd behaviour during the Pass-Off test. This is not the case for the outliers reported in the CFLR column.

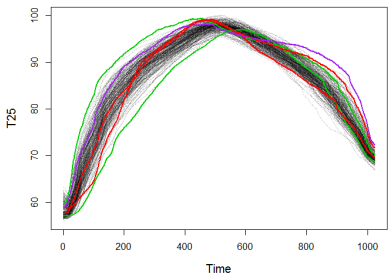
In Figure 7.4.1 we have a plot of the temperature parameters with the outliers identified using the curves directly in green, those using the RFLR model in red and those detected by both in purple. We can see that the outliers from the RFLR model do not necessarily appear as abnormal if we look at the temperature curves directly. Sample 106 is identified as an outlier by multiple temperature features and also when the depth based outlier detection is used on the temperature curves directly. Comparing the outliers identified using a classical approach, we can see Sample 24 is identified as an outlier multiple times using the classical and robust approaches. However most of the outliers from the classical approaches differ from the outliers detected using the robust approach. We can also see that the outliers using the RFLR are significantly more distinctive than the outliers using CFLR.

### 7.4.2 Vibration Surveys in XWB engines

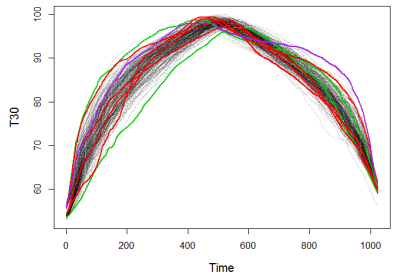
In this section we will give the results from the robust FLR model applied to V manoeuvres extracted from XWB Pass-Off tests. We will perform the same analysis



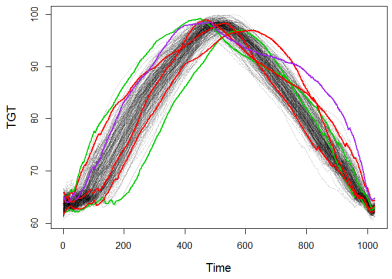

(a) TPR



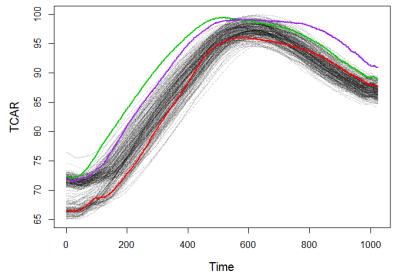

(b) T25



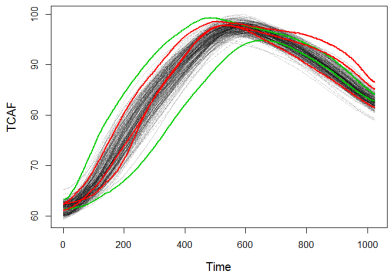

(c) T30




(d) TGT

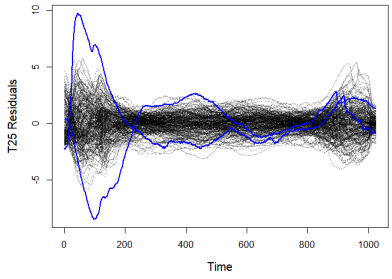



(e) TCAR

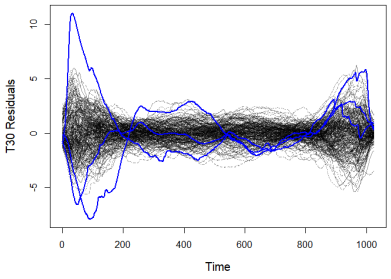



(f) TCAF

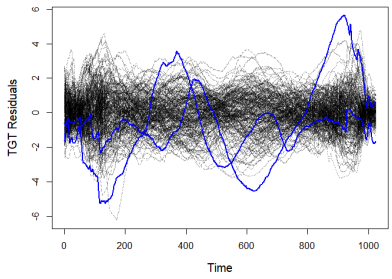
Figure 7.4.1: Plots of the TPR, T25, T30, TGT, TCAR and TCAF time series from Vibration Surveys performed on Trent 1000 engines with outliers using robust FLR in red; those using the curves directly in green and those for both in purple.



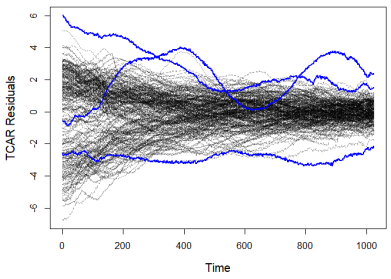

(a) T25



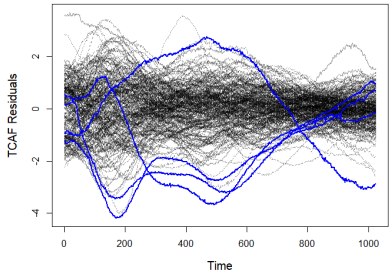

(b) T30




(c) TGT

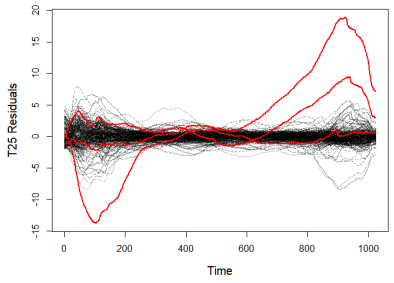



(d) TCAR

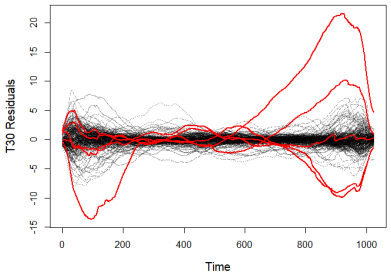



(e) TCAF

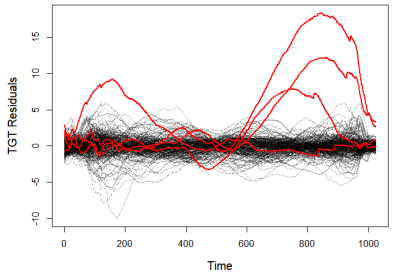
Figure 7.4.2: Plots of the residuals of the T25, T30, TGT, TCAR and TCAF with outliers using classical FLR in blue.



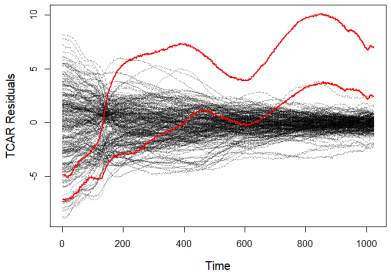

(a) T25



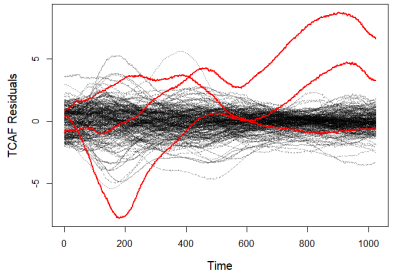

(b) T30




(c) TGT




(d) TCAR




(e) TCAF

Figure 7.4.3: Plots of the residuals of the T25, T30, TGT, TCAR and TCAF with outliers using robust FLR in red.

| **Temp** | **Direct** | **CFLR** | **RFLR** |
|---|---|---|---|
| TPR | 33, 106, 167 | - | - |
| T25 | 33, 106, 167 | 24, 182 | 24, 70, 106 |
| T30 | 33, 106, 167 | 24, 182, 192 | 24, 44, 70, 106, 196 |
| TGT | 33, 106, 167 | 119, 153 | 44, 70, 106, 117 |
| TCAR | 33, 106 | 36, 91, 106 | 70, 106 |
| TCAF | 33, 167 | 65, 167, 170, 171 | 24, 70, 106 |

Table 7.4.1: Outliers detected for temperature features (Temp) using outlier detection on the temperature features directly (Direct), and the outliers found using CFLR and RFLR.

as we did for the V manoeuvres in the Trent 1000 tests discussed in Section 7.4.1. In Table 7.4.2, we have the outliers detected using the Direct approach, using a classical approach with CFLR and BIC and finally using our robust FLR approach given in Algorithm 5. For each of the three approaches we determined a threshold using $\delta = 0.01$. We can see that the outliers in the TPR are the same as the outliers in the temperature features except for the TCAR parameter. We need to model the dependency between the engine speed and the temperature parameters, as we did for the Trent 1000 V manoeuvres.

The residuals curves from the classical approach are shown in Figure 7.4.5, with the outliers coloured in blue. In the Trent 1000 examples we saw a range of samples identified as outliers. However for the XWB V manoeuvres we consistently identify

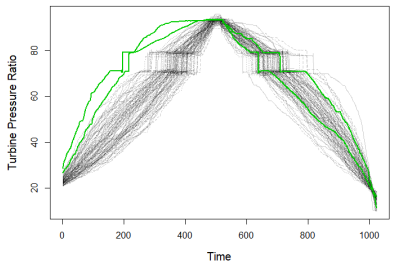
Sample 31 as an outlier, which suggests it requires further investigation.

In Figure 7.4.6 we have the residual curves using RFLR. We can see that the RFLR model identifies some very abnormal samples, with significant deviations from the general behaviour. We only have 92 samples from the XWB engine tests, which is significantly smaller than the 199 samples used in Section 7.4.1. Therefore using $\delta = 0.01$ will expectedly give fewer outliers.

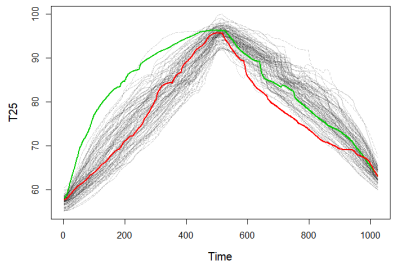
In Figure 7.4.4 we have a plot of the temperature parameters with the outliers identified using the Direct approach in green; those using the RFLR model in red and those detected by both in purple. We can see that the outliers from the RFLR model do not necessarily appear as abnormal if we look at the temperature curves directly. Samples 10 and 14 have an abnormal TPR profile, which has lead to a number of abnormal temperature profiles. There is little overlap in the outliers detected using the classical and robust approaches. There is agreement between the two approaches for the TCAR parameter. Samples 37 has significantly larger temperature values than the other samples, whilst Sample 31 has a decrease in temperature during an engine acceleration which is very abnormal.

## 7.5 Conclusion

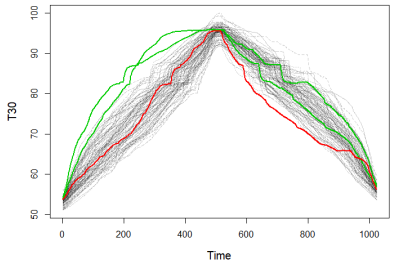
The robust Functional Linear Regression (RFLR) model we outlined in Chapter 6 has been used to identify outliers. Using the residuals of the RFLR model and functional depth we can identify abnormal response curves with respect to a predictor curve. We have shown via a simulation study that we are able to label isolated



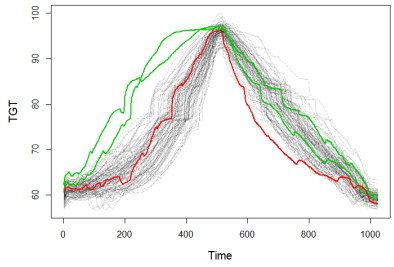

(a) TPR



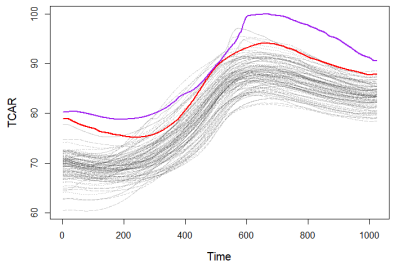

(b) T25



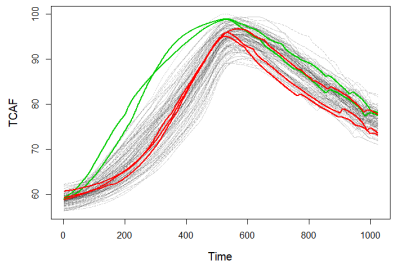

(c) T30




(d) TGT

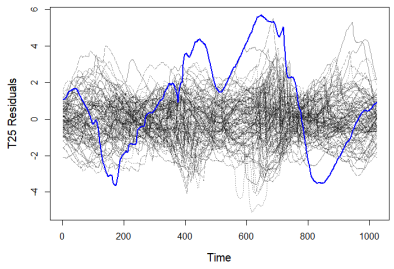



(e) TCAR

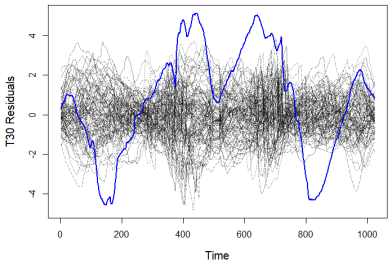



(f) TCAF

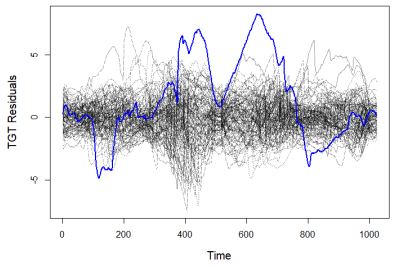
Figure 7.4.4: Plots of the TPR, T25, T30, TGT, TCAR and TCAF time series for Vibration Surveys performed on XWB engines with outliers using robust FLR in red; those using the curves directly in green and those for both in purple.



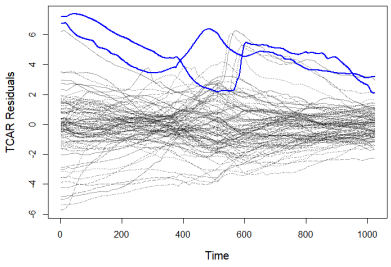

(a) T25



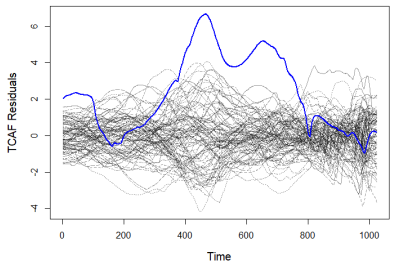

(b) T30




(c) TGT

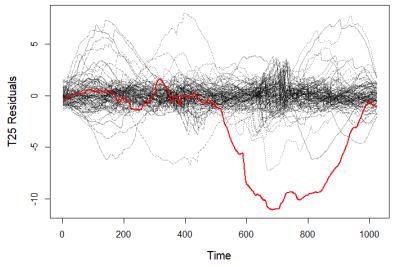



(d) TCAR

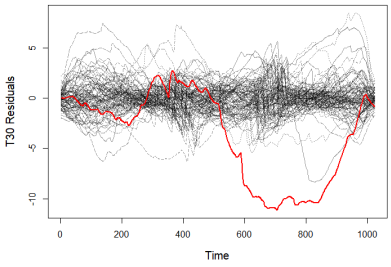



(e) TCAF

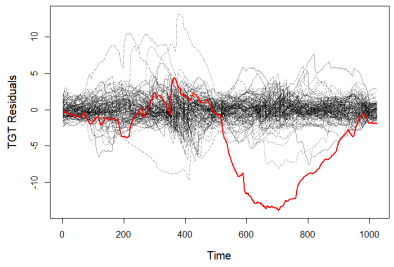
Figure 7.4.5: Plots of the residuals of the T25, T30, TGT, TCAR and TCAF for Vibration Surveys in XWB tests with outliers using classical FLR in blue.



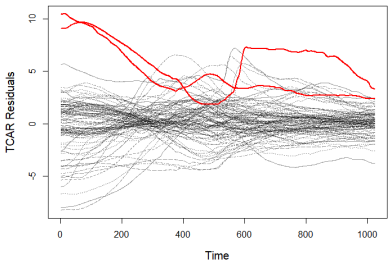

(a) T25



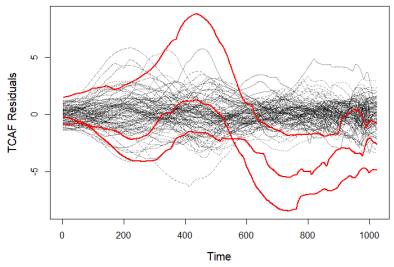

(b) T30




(c) TGT




(d) TCAR




(e) TCAF

Figure 7.4.6: Plots of the residuals of the T25, T30, TGT, TCAR and TCAF for Vibration Surveys in XWB tests with outliers using robust FLR in red.

| **Temp** | **Direct** | **CFLR** | **RFLR** |
|---|---|---|---|
| TPR | 10, 14 | - | - |
| T25 | 10 | 31 | 16 |
| T30 | 10,14 | 31 | 16 |
| TGT | 10, 14 | 31 | 16 |
| TCAR | 37 | 31, 37 | 31, 37 |
| TCAF | 10, 14 | 31 | 3, 16, 19 |

Table 7.4.2: Outliers detected for temperature features (Temp) using outlier detection on the temperature features directly (Direct), and the outliers found using CFLR and RFLR for Vibration Surveys in XWB tests.

and persistent shape outliers. The robust FLR model outperforms standard outlier detection procedures and classical FLR. Using jet engine sensor data as a motivating application for robust FLR we have identified unusual temperature behaviour. We have applied the outlier detection model on Vibration Survey manoeuvres from both the Trent 1000 and XWB Pass-Off tests. We highlighted that unusual speed profiles cause abnormal temperature profiles. Therefore the dependency of the temperature and speed behaviour needed to be modelled. We have identified interesting outliers that would not have been detected if we modelled the engine temperature independently of the engine speed.

# Chapter 8

# Prediction of Vibration Survey repeats

## 8.1 Introduction

In a Pass-Off test an engineer can choose to repeat a manoeuvre. They may repeat due to the manoeuvre not meeting certain specifications or perhaps they noticed something during the test. We consider a data driven approach to identify repeated Vibration Survey manoeuvres. We use the Vibration Surveys due to there being a large number of repeats. Given the large number of repeated and non-repeated cases, a classification approach is a natural choice. We know that the key diagnostic for a Vibration Survey being repeated is the vibration behaviour, therefore we will use the vibration parameters as predictors. We will consider three functional classification methods and highlight the strengths and weaknesses of each approach.

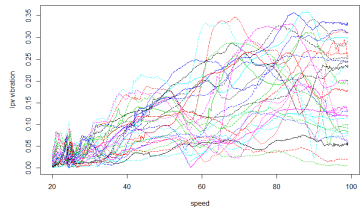
A tool that can identify whether a manoeuvre should or should not be repeated

can aid the engineers to make more informed decisions during the test, for example to highlight issues or to verify their concerns. We can also determine features that are meaningful to detect engine issues.

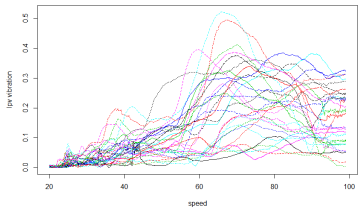
We have been given 93 Pass-Off tests from Trent 1000 engines tested in SATU, in which 199 Vibration Survey manoeuvres were performed. Each Vibration Survey is labelled as non-repeated, if another Vibration Survey is not performed later in the test, and repeated otherwise. Of the 199 Vibration Surveys, 86 are non-repeated and 113 are repeated. We have three vibration parameters, denoted as LPV, IPV and HPV (described in Chapter 1). We found treating the vibration values as a function of speed gives similar looking curves as seen in Figures 8.1.1. Capturing the behaviour between speed and vibration has also been suggested in previous jet engine models outlined in Chapter 2.

For each Vibration Survey we have six curves associated to the LPV, IPV and HPV during acceleration and deceleration. We will define the vibration with respect to the N1 speed. In Figure 8.1.1, we have 30 acceleration and deceleration curves for the vibration engine parameter.

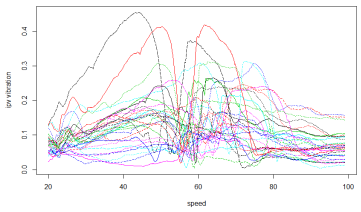
We have investigated three functional data classification methods (Ramsay and Silverman, 2005) for this problem. The first method is a Centroid-classifier, which aims to find a projection that has good theoretical classification accuracy. This model is simple and easy to apply, as discussed in Section 8.2. Second, we applied the DD-classifier in Section 8.3, which uses depth functions to create a scatter plot, enabling standard classification techniques to be applied, including $k$-nearest neighbour and support vector machines. Lastly, we applied a Logistic Functional Linear Regression



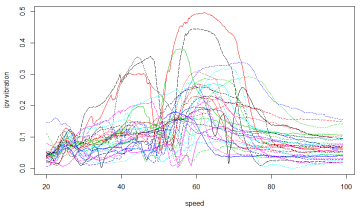

(a) LPV deceleration



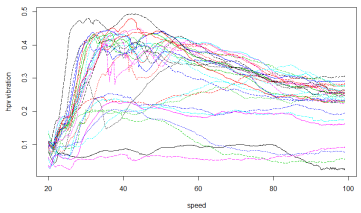

(b) LPV acceleration



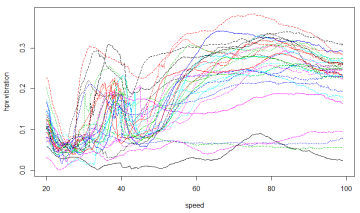

(c) IPV deceleration




(d) IPV acceleration




(e) HPV deceleration




(f) HPV acceleration

Figure 8.1.1: Plot of 30 LPV, IPV and HPV curves during acceleration and deceleration of the Vibration Survey.

(LFLR) model, which is an extension of the Functional Linear Regression model described in Section 3.3. Logistic regression is the standard method used for problems with binary outcomes (Mousavi and Sørensen, 2018). We also considered a lasso penalty on the LFLR model that can enable key features to be identified. The model has two nice features. First, it is fast as we can reduce the dimensionality by working with basis coefficients. Second, the model gives associated probabilities for the classifications, which gives a measure of uncertainty. Finally, we will compare the classification accuracy of the three models using ROC curves.

## 8.2 Centroid classifier

The first functional classifier we will consider is by Delaigle and Hall (2012). Their aim is to project the data function $X$ onto a one dimensional space. By choosing an appropriate projection function, they aim to minimise the classification error in the one-dimensional problem. They suggest a possible projection function and a distance measure to classify the one-dimensional projections. The idea is that if the two classes of data are projected into distinctive groups then it will be relatively easy to classify using an appropriate distance measure.

Let $(x_i, l_i)$ be data pairs, where $x_i$ is a function defined on the interval $I$ and $l_i$ is the corresponding label. They assume that the functions for non-repeated curves lie around a mean $\mu_0$ and functions for repeated curves lie around a mean $\mu_1$, which we estimate with sample means $\bar{x}_0$ and $\bar{x}_1$. They suggest the projection $\int_I x_i(t)\phi(t)dt$, where $\phi(t)$ is a projection function that needs to be chosen. They outline two estimates

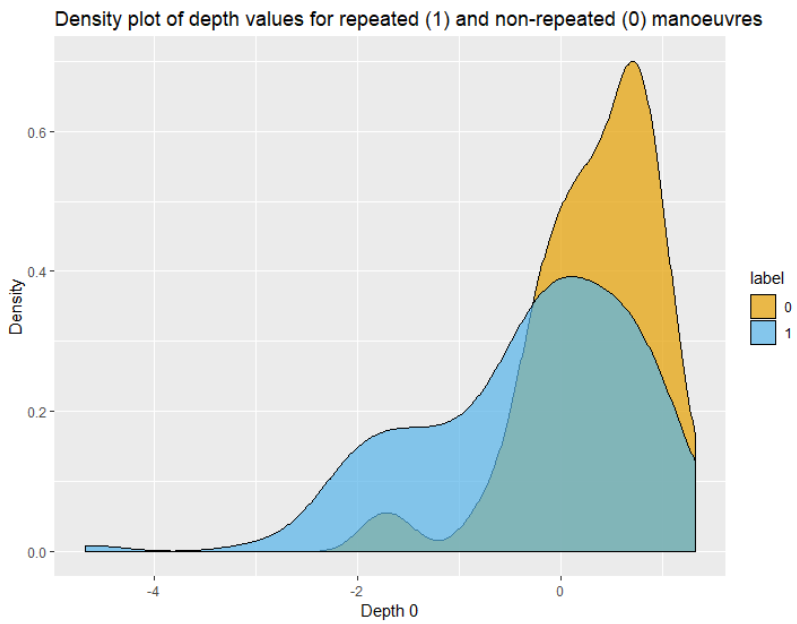



Figure 8.2.1: Density plot of score values from Centroid classifier applied to LPV deceleration curves.

of $\phi$ with good classification properties, with details available in Delaigle and Hall (2012). We shall use the first estimate, which is a weighted sum of Functional Principal Components (Ramsay and Silverman, 2005).

The centroid classifier takes an unlabelled function $x$ and gives label 0 or 1 if the test statistic

$$T(x) = \left[\int x(t)\phi(t)dt - \int x(t)\mu_1(t)dt\right]^2 - \left[\int x(t)\phi(t)dt - \int x(t)\mu_0(t)dt\right]^2. \quad (8.2.1)$$

is positive or negative respectively.

## 8.3 Depth Depth-Classifier

The second functional data classifier that we will be investigating, is called the Depth Depth classifier or DD-classifier (Li et al., 2012). This classifier does not assume that vibration functions for repeated and non-repeated manoeuvres lie around different mean functions as in the Centroid classifier in Section 8.2. Instead the classifier uses Functional depth, which was described in Section 3.4. The DD-classifier assumes that the curves in the two classes have different distributions. Therefore the depth values with respect to the repeated and non-repeated manoeuvres should be different.

The DD-classifier takes samples $z_1, ..., z_m$ with label 0 and $w_1, ..., w_k$ with label 1, for some $m, k \in \mathbb{N}$. We assume the samples $z_i$ come from a distribution $F_0$ and samples $w_i$ come from the distribution $F_1$. We obtain the depth values $d_0$ and $d_1$ with respect to samples $z_1, ..., z_m$ and sample $w_1, ..., w_k$ respectively. Each sample has two depth values, which gives a scatter plot. If $F_0$ and $F_1$ are the same distribution then the points on the scatter plot will lie along a line angled at 45 degrees. Once the scatter plot is made, we can use different classification techniques for multivariate data, including $k$-Nearest Neighbour (k-NN), Support Vector Machines (SVM) and kernel methods.

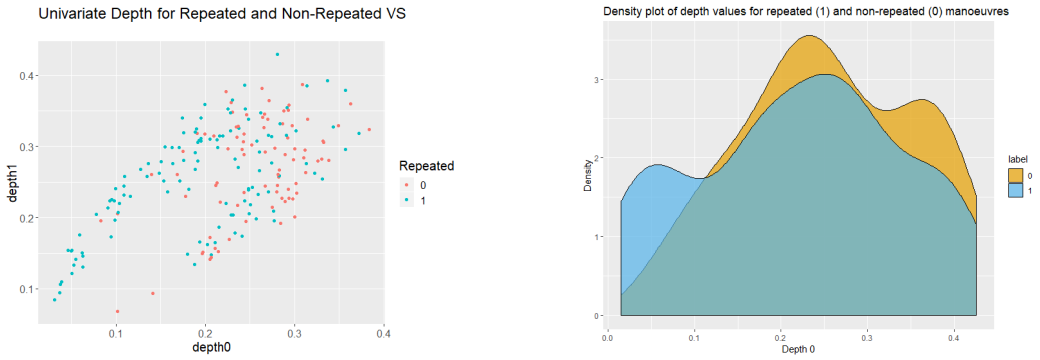
In Figure 8.3.1, we have a scatter plot using the Halfspace depth for the LPV deceleration curves. If the depth function for repeated and non-repeated manoeuvres were different the points would be away from the diagonal. We can therefore see that the depth functions for repeated and non-repeated manoeuvres are very similar. There are a large number of repeated manoeuvres on the left and then a mixture, but

mainly non-repeated manoeuvres on the right. Note that the repeated manoeuvre samples that lie near the origin, are the samples that are furthest away from the centre of the distribution of the curves. We would expect that the most unusual curves (smallest depth values) will arise from the repeated manoeuvres, which is indeed the case. There are a few non-repeated samples near the origin. These are cases where the model believes these manoeuvres should have been repeated. In Figure 8.3.1 we have density plots of the $d_0$ values for repeated and non-repeated cases. We can see that the non-repeated manoeuvres and the repeated manoeuvres have very similar distribution of depth values. There is no clear split between the two classes in terms of these depth values, which makes classification difficult.

Using a multivariate depth function we can use information across all 6 curves. We make a scatter plot of the depth values in Figure 8.3.2. We can see a better split of the groups than using individual vibration curves. Using information across the vibration curves evidently improves the separation of the curves. Looking at the density plot of the depth with respect to the non-repeated manoeuvres in Figure 8.3.2, we can see that there is less of an overlap between the depth values.

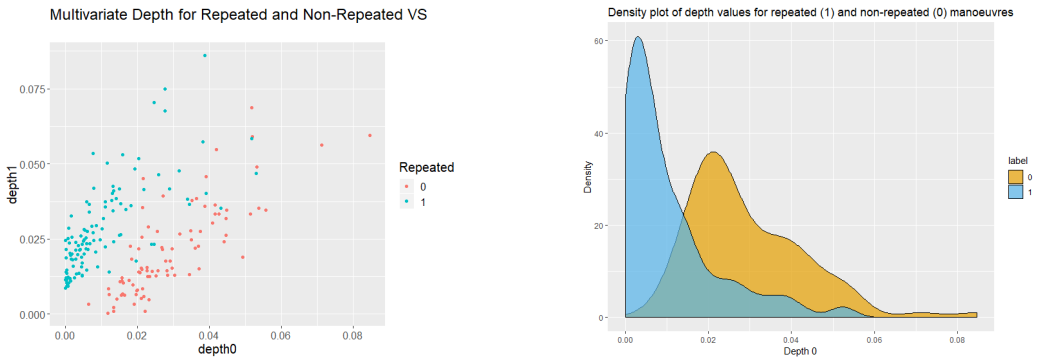
## 8.4 Logistic Functional Linear Regression

In Logistic Functional Linear Regression (LFLR) (Mousavi and Sørensen, 2018), we have a binary response $Y$, with predictor function $X(t)$. Let $y = (y_1, ..., y_n)^T$ be $n$ observations, with corresponding predictor functions $x(t) = (x_1(t), ..., x_n(t))^T$ for $t \in I$. Then the Logistic FLR model gives the conditional probability:




(a) Scatter Plot (b) Density Plot

Figure 8.3.1: Scatter plot of the depth value labelled by non-repeated (0) and repeated (1) manoeuvres (left). Density plot of depth values with respect to non-repeated manoeuvres (depth0) (right). The depth values are obtained from the LPV deceleration time series.




(a) Scatter Plot (b) Density Plot

Figure 8.3.2: Scatter plot of the multivariate depth values labelled by non-repeated (0) and repeated (1) manoeuvres (left). Density plot of depth values with respect to non-repeated manoeuvres (depth0) (right).

$$\pi(x) = P(Y = 1|X = x) = \frac{\exp\{\alpha + \int_I \beta(t)x(t)dt\}}{1 + \exp\{\alpha + \int_I \beta(t)x(t)dt\}} \tag{8.4.1}$$

with regression function $\beta(t)$. Using the logit transform, we have

$$\eta(x) = logit(\pi(x)) = \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \alpha + \int_I \beta(t)x(t)dt. \tag{8.4.2}$$

For a pre-defined basis $\theta$, we let $x(s) = W\theta(s)$ for coefficient matrix $W$, and $\beta(s) = \theta(s)^T b$ for coefficient vector $b$, then

$$\eta = \alpha + Wb. \tag{8.4.3}$$

We will consider using two basis classes for $\theta(s)$. The first basis is the Functional Principal Components of $x(s)$ and the second is a B-spline basis.

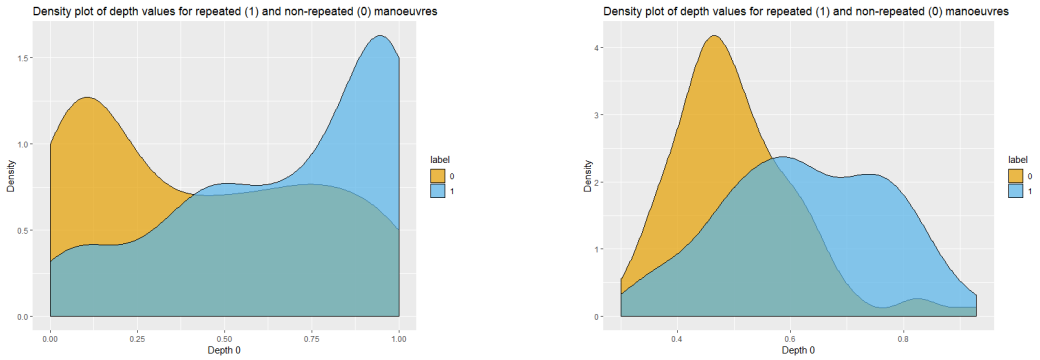
Given $n$ independent samples, we can write the likelihood as

$$L(\alpha, \beta) = \prod_{i=1}^{n} \pi_i^{y_i}(1 - \pi_i)^{1-y_i} = \prod_{i=1}^{n} \frac{\exp(y_i\{\alpha + \int_I \beta(t)x_i(t)dt\})}{1 + \exp\{\alpha + \int_I \beta(t)x_i(t)dt\}}. \tag{8.4.4}$$

This model can be easily extended to multiple predictors, by concatenating the basis coefficients.

The LFLR classifier can be modified in a number of ways. One possibility is to incorporate a regularisation term to stop the classifier overfitting, which we can perform using a lasso penalty. The lasso penalty can be incorporated into the regression equation (8.4.3):

$$\eta = \alpha + Wb + \lambda|b|,$$




(a) FPCA (b) B-spline

Figure 8.4.1: Density plot of probability values obtained used FPCA basis and depth values (left) and using B-spline basis with depth and lasso (right).

where $\lambda$ is a tuning parameter which penalises large values of $b$. The lasso model can also be seen as a model selection procedure as it shrinks a majority of the coefficient terms in $b$ to zero. If we use a B-spline basis with a lasso penalty we can perform domain selection to identify segments of the vibration curves that are informative.

We can easily add exogenous variables $d$ to the model (8.4.3):

$$\eta = \alpha + Wb + \gamma d,$$

where $\gamma$ is another regression term. We have seen in Section 8.3 that the depth values can be informative. We can therefore add the depth values into the model.

## 8.5 Results

We have outlined three different classification algorithms to label repeated and non-repeated Vibration Surveys. We also outlined a number of variants, in particular for the LFLR classifier in which we can incorporate depth value information and a lasso penalty. The LFLR model reduces to a standard logistic problem of the basis coefficients. We can therefore apply LFLR using standard logistic regression. We concatenate the basis coefficients for each vibration curve, enabling all the vibration curves to be used simultaneously. We consider two basis types: FPCA bases functions, using the first six eigenfunctions that capture 95% of the variance. We also considered a B-spline basis using 61 functions, which fits the vibration curves sufficiently well, and can highlight informative segments of the vibration data. We have found both basis choices give similar results, if we incorporate a Lasso penalty with the B-spline basis.

For the Centroid-classifier we will also use six eigenfunctions. For the DD-classifier we considered multiple depth functions including the Halfspace depth and the $h$-modal depth for the univariate curves. We found that the results were similar for different depth functions. We have chosen to use the Halfspace depth as it can be extended into a Multivariate Functional depth (Claeskens et al., 2014), enabling information to be used across all the vibration time series.

We will use a leave-one-out procedure to test each of these models. To compare the classification performance of the three algorithms, we will look at the ROC curves and the Area Under the Curve (AUC) as we did in Chapter 7.

Table 8.5.1: The AUC values for the version of each classifier that gave maximum AUC values. The Centroid classifier used the lpv decel time series. The DD classifier in the univariate case used the LPV acceleration time series and in the multivariate case used all the time series. The LFLR classifier used a FPCA basis with depth values and used a B-spline basis with depth and Lasso penalty.

| Model | Centroid | DD uni | DD multi | LFLR-FPCA | LFLR-Bspline |
|---|---|---|---|---|---|
| AUC | 0.6886 | 0.7364 | 0.8859 | 0.7208 | 0.753 |

The Centroid classifier requires univariate time series. We applied it to each vibration time series and found the LPV deceleration time series gives the maximum AUC value. For the DD classifier we applied it to the univariate curves and found the LPV acceleration curve gave the largest AUC value. We also applied the Multivariate Functional depth using all the vibration time series. The multivariate DD classifier significantly outperforms the univariate cases. The improvement in classification arises due to information being used across each of the vibration curves.

Finally we tested the LFLR classifier using both an FPCA and a B-spline basis. We considered two variants using depth value information and a lasso penalty. The FPCA based model with depth values was the best performing model. For the B-spline basis using depth also improved the model and gave significantly better results using a Lasso penalty. In Table 8.5.1 we have the results for the model cases that give the largest AUC value for each classifier. We can see that DD classifier using multivariate depth significantly outperforms the other models.

In Figure 8.5.1 we have the ROC curves for the three classifiers using the best

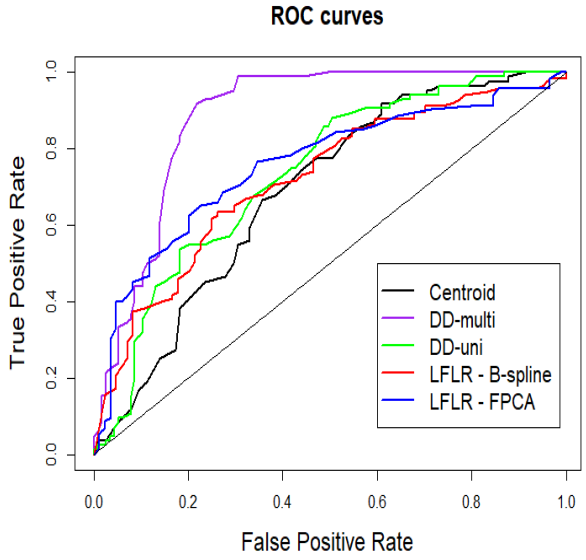
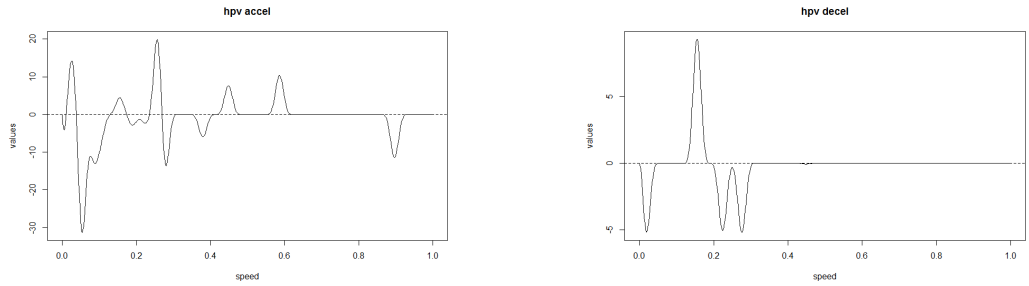



Figure 8.5.1: The ROC curve for the five classifiers given in Table 8.5.1.

case models. The ROC curves were formed using values given by the leave-one-out procedure. The four curves in Figure 8.5.1 are sufficiently far from the diagonal indicates that they are able to give meaningful classifications. We can see the DD classifier using multivariate depth significantly outperforms the other models. However the LFLR with a B-spline basis can highlight segments of the vibration curves that are informative. For example in Figure 8.5.2, we can see that the HPV deceleration regression function, has non-zero weight for vibration values at lower speeds. The HPV acceleration is more sporadic with spikes in various parts of the speed spectrum.




(a) Regression function for hpv accel (b) Regression function for hpv decel

Figure 8.5.2: Plots of the regression functions for the hpv accel (left) and hpv decel (right) curves, using a B-spline basis with lasso penalty.

# Chapter 9

# Conclusion and Further work

A large amount of sensor data is generated during engine testing. Currently only a small percentage of this data is being used by the engineers, which typically involves checking the engine behaviour at certain segments of the tests. In this thesis we have developed a range of statistical tools to make inferences from jet engine sensor data. These tools have been built to aid the engineers at Rolls Royce to make assessments on the engine health.

In a Pass-Off test engineers perform manoeuvres corresponding to various engine accelerations and decelerations. The manoeuvres must pass certain conditions. The manoeuvres can be repeated and the test can be stopped to enable changes to be made. These manoeuvres are not currently labelled. We therefore developed an automated classification algorithm that is able to extract and identify the different manoeuvre types. The algorithm has been shown to give high classification accuracy and has been tested on two different engine types. The labels can then be used to identify problematic engine tests, for example tests that were stopped multiple times

and manoeuvres were repeated. There is scope to use the classification information to obtain summary statistics. These statistics can be used to assess the effectiveness of the engine test, identify patterns that can be used to characterise different engine behaviours, and potentially use these models to make predictions.

In the Cyclic engine test engineers perform manoeuvres which are referred to as cycles. The purpose of a Cyclic test is to repeatedly perform manoeuvres on an engine to assess the engine degradation over time. Unlike the Pass-Off tests, there are no pre-defined manoeuvres in the Cyclic test. We therefore cannot apply a classification approach. Instead we have built a clustering algorithm to identify the different manoeuvre types. The algorithm gives distinct clusters that look reasonable from visual inspection. We tested the clustering approach on the Pass-Off test data, as we have labels for the true classes. In general the algorithm identifies the different classes effectively. Our main aim was to use the clustering results from the Cyclic test to identify degradation in the engine behaviour. Building an algorithm to model the engine degradation in a Cyclic test is a natural further step. We have attempted a few approaches, but were not able to identify any clear signs of decreased engine performance.

We have found that the Vibration Survey manoeuvre was the most repeated manoeuvre in the Pass-Off tests. We therefore suspect some of the repeated manoeuvres will display unusual engine behaviour. The manoeuvres are performed by a human-controller, which causes variability between manoeuvre profiles. This variability can mask abnormal behaviour. We therefore built a Robust Functional Linear Regression (RFLR) model to capture the relationship between engine temperature and speed. By

modelling the dependency between engine temperature and speed we can mitigate the variability introduced by the human-controller. Using the residuals from the RFLR model we identified distinct outliers that were not picked up by standard outlier detection approaches. The RFLR model was built for univariate curves however there is clearly correlation between different temperature parameters. Therefore a multivariate RFLR model that can capture the correlation between the different temperature parameters will be more effective in identifying outliers. This extension of the RFLR model would require a multivariate robust FPCA model.

Currently an engine test is performed by a group of engineers who decides whether a manoeuvre should be repeated and whether the test should be stopped. These decisions could be aided using data-driven statistical models. We focused on the Vibration Survey manoeuvre, which has been repeated a large number of times. We have modelled the prediction of a Vibration Survey manoeuvre being repeated in the test as a classification problem. The decision to repeat a Vibration Survey is typically made using the information from the vibration parameters. Therefore we used the vibration parameters as predictors. We considered three different approaches and found that one of the models was able to give high classification accuracy. One extension would be to build a decision tool for each manoeuvre type. Also rather than considering the two class case of manoeuvres being repeated and not-repeated, we could consider a third option for whether the test should be stopped after a manoeuvre.

# Bibliography

Jose Agulló, Christophe Croux, and Stefan Van Aelst. The multivariate least-trimmed squares estimator. *J. Multivar. Anal.*, 99(3):311–338, 2008. ISSN 0047-259X.

Hirotugu Akaike. A new look at the statistical model identification. In Emanuel Parzen, Kunio Tanabe, and Genshiro Kitagawa, editors, *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer New York, New York, NY, 1998.

Ana Arribas-Gil and Juan Romo. Shape outlier detection and visualization for functional data: the outliergram. *Biostatistics*, 15 4:603–19, 2014.

Maria-Florina Balcan, Yingyu Liang, and Pramod Gupta. Robust hierarchical clustering. *Journal of Machine Learning Research*, 15:4011–4051, 2014.

Juan Lucas Bali, Graciela Boente, David E. Tyler, and Jane-Ling Wang. Robust functional principal components: A projection-pursuit approach. *Ann. Statist.*, 39 (6):2852–2882, 2011.

Gilbert Ames Bliss. *Calculus of Variations*, volume 1. Mathematical Association of America, 1 edition, 1925.

Graciela Boente and Matías Salibian-Barrera. S-estimators for functional principal

component analysis. *Journal of the American Statistical Association*, 110(511): 1100–1111, 2015.

Aleksandar Bojchevski, Yves Matkovic, and Stephan Günnemann. Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 737–746, New York, NY, USA, 2017. ACM.

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

E. Oran Brigham. *The Fast Fourier Transform and Its Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

Anirvan Chakraborty and Probal Chaudhuri. The spatial distribution in infinite dimensional spaces and related quantiles and depths. *Ann. Statist.*, 42(3):1203–1231, 06 2014. doi: 10.1214/14-AOS1226.

Jeng-Min Chiou, Yu-Ting Chen, and Ya-Fang Yang. Multivariate functional principal component analysis: A normalization approach. *Statistica Sinica*, 24(4):1571–1596, 2014.

Jeng-Min Chiou, Ya-Fang Yang, and Yu-Ting Chen. Multivariate functional linear regression and prediction. *Journal of Multivariate Analysis*, 146:301 – 312, 2016.

Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces.

Gerda Claeskens, Mia Hubert, Leen Slaets, and Kaveh Vakili. Multivariate functional halfspace depth. *Journal of the American Statistical Association*, 109(505):411–423, 2014.

David A Clifton. *Novelty detection with extreme value theory in jet engine vibration data*. PhD thesis, University of Oxford, 2009.

L. Clifton, D. A. Clifton, Y. Zhang, P. Watkinson, L. Tarassenko, and H. Yin. Probabilistic novelty detection with support vector machines. *IEEE Transactions on Reliability*, 63(2):455–467, June 2014.

C. Croux and A. Ruiz-Gazen. A fast algorithm for robust principal components based on projection pursuit. In Albert Prat, editor, *COMPSTAT*, pages 211–216, Heidelberg, 1996. Physica-Verlag HD.

C. Croux, P. Filzmoser, and M.R. Oliveira. Algorithms for projection–pursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87(2):218 – 225, 2007. ISSN 0169-7439.

J. A. Cuesta-Albertos and A. Nieto-Reyes. The random Tukey depth. *ArXiv e-prints*, July 2007.

Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. Robust estimation and classification for functional data via projection-based depth notions. *Comput. Stat.*, 22(3):481–496, September 2007. ISSN 0943-4062.

Wenlin Dai and Marc G. Genton. Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics*, 27(4):923–934, 2018a.

Wenlin Dai and Marc G. Genton. Functional boxplots for multivariate curves. *Stat*, 7(1):e190, 2018b.

M Daszykowski, B Walczak, and D.L Massart. Looking for natural patterns in data: Part 1. density-based approach. *Chemometrics and Intelligent Laboratory Systems*, 56(2):83 – 92, 2001.

J. Dauxois, A. Pousse, and Y. Romain. Asymptotic theory for the principal component analysis of a vector random function: Some applications to statistical inference. *Journal of Multivariate Analysis*, 12(1):136 – 154, 1982.

C. de Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer New York, 2001.

Aurore Delaigle and Peter Hall. Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):267–286, 2012.

Subhajit Dutta, Anil K. Ghosh, and Probal Chaudhuri. Some intriguing properties of tukey's half-space depth. *Bernoulli*, 17(4):1420–1434, 11 2011.

Paul H. C. Eilers and Brian D. Marx. Flexible smoothing with b -splines and penalties. *Statist. Sci.*, 11(2):89–121, 1996.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

Manuel Febrero-Bande, Pedro Galeano, and Wenceslao Gonzãlez-Manteiga. Outlier detection in functional data by depth measures, with application to identify abnormal nox levels. *Environmetrics*, 19:331 – 345, 06 2008.

F. Ferraty, I. Van Keilegom, and P. Vieu. Regression when both response and predictor are functions. *Journal of Multivariate Analysis*, 109:10 – 28, 2012.

Ricardo Fraiman and Graciela Muniz. Trimmed means for functional data. *Test*, 10 (2):419–440, Dec 2001.

Luis Ángel García Escudero, Alfonso Gordaliza, Carlos Matrán Bea, Agustín Mayo Iscar, and Ch Hennig. Robustness and outliers. In *Handbook of cluster analysis*, pages 653–678. Chapman and Hall/CRC, 2015.

Irène Gijbels and Stanislav Nagy. On a general definition of depth for functional data. *Statist. Sci.*, 32(4):630–639, 11 2017.

H. Hanachi, C. Mechefske, J. Liu, A. Banerjee, and Y. Chen. Performance-based gas turbine health monitoring, diagnostics, and prognostics: A survey. *IEEE Transactions on Reliability*, 67(3):1340–1363, Sep. 2018.

Clara Happ and Sonja Greven. Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113(522):649–659, 2018.

Paul Hayton, Simukai Utete, Dennis King, Steve King, Paul Anuzis, and Lionel Tarassenko. Static and dynamic novelty detection methods for jet engine health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):493–514, 2007.

Ian Holmes and Richard Durbin. Dynamic programming alignment accuracy. In *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, RECOMB '98, pages 102–108, New York, NY, USA, 1998. ACM. ISBN 0-89791-976-9.

Peter J. Huber. *Robust Statistics*, pages 1248–1251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

Mia Hubert, Peter J. Rousseeuw, and Pieter Segaert. Multivariate functional outlier detection. *Statistical Methods & Applications*, 24(2):177–202, 2015.

Harjit Hullait. Cyclic test. `https://github.com/hullait`, 2019.

Rob J. Hyndman and Md. Shahid Ullah. Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics and Data Analysis*, 51:4942–4956, 2007.

Andrada E. Ivanescu, Ana-Maria Staicu, Fabian Scheipl, and Sonja Greven. Penalized function-on-function regression. *Computational Statistics*, 30(2):539–568, 2015.

B. Jackson, J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. San, L. Tan, and T. T. Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12:105–108, February 2005.

Ioannis Kalogridis and Stefan Van Aelst. Robust functional regression based on principal components. *Journal of Multivariate Analysis*, 173:393 – 415, 2019. ISSN 0047-259X.

R. Killick, P. Fearnhead, and I. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107 (500):1590–1598, 2012.

Rebecca Killick, Claudie Beaulieu, Simon Taylor, and Harjit Hullait. *EnvCpt: Detection of Structural Changes in Climate and Environment Time Series*, 2018. URL `http://CRAN.R-project.org/package=EnvCpt`.

S King, P R Bannister, D A Clifton, and L Tarassenko. Probabilistic approach to the condition monitoring of aerospace engines. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 223(5):533–541, 2009.

Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

Jun Li, Juan A. Cuesta-Albertos, and Regina Y. Liu. Dd-classifier: Nonparametric

classification procedure based on dd-plot. *Journal of the American Statistical Association*, 107(498):737–753, 2012.

Yehua Li and Tailen Hsing. Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *Ann. Statist.*, 38 (6):3321–3351, 12 2010.

N. Locantore, J. S. Marron, D. G. Simpson, N. Tripoli, J. T. Zhang, and K. L. Cohen. Robust principal component analysis for functional data. *Test*, 8(1):1–73, Jun 1999. ISSN 1863-8260.

David Lowe and Michael E Tipping. Neuroscale: Novel topographic feature extraction using rbf networks. In *Advances in neural information processing systems*, pages 543–549, 1997.

Sara López-Pintado and Juan Romo. On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486):718–734, 2009.

Sara López-Pintado and Juan Romo. A half-region depth for functional data. *Computational Statistics and Data Analysis*, 55(4):1679 – 1695, 2011.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

José A. F. Machado. Robust model selection and M-estimation. *Econometric Theory*, 9(3):478–493, 1993.

Nicole Malfait and James O. Ramsay. The historical functional linear model. *The*

*Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 31(2):115–128, 2003.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

Hidetoshi Matsui. Quadratic regression for functional response models. *arXiv e-prints*, art. arXiv:1702.02009, Feb 2017.

Ioannis Matthaiou, Bhupendra Khandelwal, and Ifigeneia Antoniadou. Vibration monitoring of gas turbine engines: Machine-learning approaches and their challenges. *Frontiers in Built Environment*, 3:54, 2017.

GL Merrington. Fault diagnosis in gas turbines using a model-based technique. *ASME*, 39:374–380, 1994.

Jeffrey S. Morris. Functional regression. *Annual Review of Statistics and Its Application*, 2(1):321–359, 2015.

Seyed Nourollah Mousavi and Helle Sørensen. Functional logistic regression: a comparison of three methods. *Journal of Statistical Computation and Simulation*, 88(2):250–268, 2018.

Ian Nabney. *NETLAB: algorithms for pattern recognition*. Springer Science & Business Media, 2002.

Stanislav Nagy. Consistency of h-mode depth. *Journal of Statistical Planning and Inference*, 165:91 – 103, 2015.

Alexandre Nairac, Neil Townsend, Roy Carr, Steve King, Peter Cowley, and Lionel Tarassenko. A system for the analysis of jet engine vibration data. *Integr. Comput.-Aided Eng.*, 6(1):53–66, January 1999.

Naveen N. Narisetty and Vijayan N. Nair. Extremal depth for functional data and applications. *Journal of the American Statistical Association*, 111(516):1705–1714, 2016.

Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970.

Alicia Nieto-Reyes and Heather Battey. A topologically valid definition of depth for functional data. *Statist. Sci.*, 31(1):61–79, 02 2016.

R.J. Patton, S. Simani, S. Daley, and A. Pike. Fault diagnosis of a simulated model of an industrial gas turbine prototype using identification techniques. *IFAC Proceedings Volumes*, 33(11):511 – 516, 2000. 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes 2000 (SAFEPROCESS 2000), Budapest, Hungary, 14-16 June 2000.

David Pollard. *Convergence of stochastic processes*. Springer Science & Business Media, 2012.

Xin Qi and Hongyu Zhao. Some theoretical properties of silverman's method for smoothed functional principal component analysis. *Journal of Multivariate Analysis*, 102(4):741 – 767, 2011.

J. O. Ramsay and C. J. Dalzell. Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):539–572, 1991.

J.O. Ramsay and B.W. W Silverman. *Functional Data Analysis (Springer Series in Statistics)*. Springer Publishing Company, Incorporated, 2005.

C. Radhakrishna Rao. Some statistical methods for comparison of growth curves. *Biometrics*, 14(1):1–17, 1958.

Marvin Rausand and Arnljot Høyland. *System Reliability Theory: Models, Statistical Methods and Applications*. Wiley-Interscience, Hoboken, NJ, 2004.

John A. Rice and B.W. W Silverman. Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(1):233–243, 1991.

Lior Rokach and Oded Maimon. *Decision Trees*, pages 165–192. Springer US, Boston, MA, 2005.

Peter J. Rousseeuw, Jakob Raymaekers, and Mia Hubert. A measure of directional outlyingness with applications to image data and video. *Journal of Computational and Graphical Statistics*, 27(2):345–359, 2018.

Pallavi Sawant, Nedret Billor, and Hyejin Shin. Functional outlier detection with robust functional principal component analysis. *Computational Statistics*, 27(1): 83–102, Mar 2012.

Fabian Scheipl, Ana-Maria Staicu, and Sonja Greven. Functional additive mixed models. *Journal of Computational and Graphical Statistics*, 24(2):477–501, 2015.

Gideon Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.

Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40, 2008.

Han Lin Shang. A survey of functional principal component analysis. *AStA Advances in Statistical Analysis*, 98(2):121–142, 2014.

Bernard W. Silverman. Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1–24, 02 1996.

Ying Sun and Marc G. Genton. Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334, 2011.

Gian Antonio Susto, Angelo Cenedese, and Matteo Terzi. *Time-Series Classification Methods: Review and Applications to Power Systems Data*, pages 179–220. 01 2018. ISBN 9780128119686.

Terry M. Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning*, 2011. URL `http://CRAN.R-project.org/package=rpart`.

Ledyard R. Tucker. Determination of parameters of a functional relation by factor analysis. *Psychometrika*, 23(1):19–23, 1958.

John W. Tukey. Mathematics and the Picturing of Data. In Ralph D. James, editor, *International Congress of Mathematicians 1974*, volume 2, pages 523–532, 1974.

Philip P. Walsh and Paul Fletcher. Engine performance testing. In *Gas Turbine Performance*, pages 519–563. Blackwell Science Ltd, 2008.

Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295, 2016.

Fang Yao, Hans-Georg Müller, Jane-Ling Wang, et al. Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, 33(6):2873–2903, 2005.

Yijun Zuo and Robert Serfling. General notions of statistical depth function. *Ann. Statist.*, 28(2):461–482, 04 2000.