

**GENERAL BAYESIAN APPROACH FOR MANUFACTURING
EQUIPMENT DIAGNOSTICS USING SENSOR FUSION**

A Thesis
Presented to
The Academic Faculty

by

Stephanie Isabel Locks

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Mechanical Engineering

Georgia Institute of Technology
May 2016

COPYRIGHT© BY STEPHANIE I. LOCKS

GENERAL BAYESIAN APPROACH FOR MANUFACTURING EQUIPMENT DIAGNOSTICS USING SENSOR FUSION

Approved by:

Dr. Thomas Kurfess, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Christopher Saldana
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Steven Liang
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Cassandra Telenko
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: April 25th 2016

To my wonderful brother Ian

ACKNOWLEDGEMENTS

I would like to thank Dr. Thomas Kurfess for his advice and commitment to providing an encouraging lab environment. I would also like to acknowledge my parents and friends for their support throughout this process. Lastly, I am grateful for the financial support provided by Moog Inc.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
SUMMARY	xiv
1 INTRODUCTION	1
2 BACKGROUND AND LITERATURE REVIEW	3
2.1 Artificial Neural Networks	3
2.2 Support Vector Machine	5
2.3 Neuro-Fuzzy Systems	6
2.4 Frequentist analysis	6
2.5 Control Charts	9
2.5.1 Student's t-distribution.....	11
2.5.2 Type I and Type II Error	13
2.6 Bayesian Analysis	15
2.6.1 Naïve Bayesian Models	18
2.6.2 Implementing Bayesian Models	20
2.6.3 Model Updating	23
3 METHODS	25

3.1	Method Assumptions.....	26
3.2	Mathematical Tools.....	27
3.2.1	Bayesian Model Components	27
3.2.2	Bayesian Model for Sensor Fusion.....	36
3.3	Method for Implementation of Bayesian Models	38
3.3.1	Phase 1	39
3.3.2	Phase 2	40
3.3.3	Model Updating	40
3.4	Metrics for Results	41
3.5	Experimental Implementation: Tool Diagnostics for a Three Insert Tool.....	42
3.5.1	Assumptions.....	42
3.5.2	Purpose.....	42
3.5.3	Equipment.....	44
3.5.4	Materials	44
3.5.5	Experimental Procedure.....	45
3.5.6	Phase 1	47
3.5.7	Phase 2	67
3.6	Simulated Implementation	67
3.6.1	Purpose.....	67
3.6.2	Inputs & Simulated Model Setup.....	68

3.6.3	Computational Tools.....	69
3.6.4	Simulated Procedure: Overlapping Distributions	71
3.6.5	Simulated Procedure: Model Updating.....	75
4	RESULTS AND DISCUSSION	76
4.1	Experimental Implementation	76
4.1.1	Effect of Overlapping Distributions in Experimental Case	78
4.2	Simulated Implementation	84
4.2.1	Effect of Overlapping Distributions.....	85
4.2.2	Effect of Model Updating	87
4.2.3	Additional Results.....	93
5	CONCLUSIONS AND RECOMMENDATIONS	97
5.1	Conclusions	98
5.2	Recommendations	101
	APPENDIX A.....	102
	APPENDIX B	114
	REFERENCES	120

LIST OF TABLES

Table 1: Nomenclature for example and equation (18)	16
Table 2: Probabilities for Multi-Condition Example	35
Table 3: Layout of Experiments	45
Table 4: Final Vector of Extracted Features for Experimental Implementation d	60
Table 5: Parameters for test of overlap effect.....	72
Table 6: Results for 3mm, 4mm, & 5mm Radial Depth of Cut Case with 10 instances of Data in Conditional Database for each condition.....	76
Table 7: Results for 3mm Radial Depth of Cut Case with 10 instances of Data in Conditional Database for each condition.....	77
Table 8: Three variable multivariate model, Dynamometer Y axis.....	81
Table 9: Effects of distribution Overlap for 3 Variable Bayesian Model.....	85

LIST OF FIGURES

Figure 1: Structure of an Artificial Neural Network.....	4
Figure 2: Normal distribution with standard deviations on the independent axis. (Psomas, 2003)	7
Figure 3: A larger variance signifies a larger spread or scatter of data. (The Basic Statistical Foundation, 2002).....	8
Figure 4: Control chart and sample distribution ‘in control’	9
Figure 5: Control chart with ‘out of control’ sample distributions.....	10
Figure 6: Off-line (Phase 1) and On-line (Phase 2) set up for diagnostics algorithm by Tobon-Mejia 2012	22
Figure 7: Example Histogram for Feature $F_j C_i$	28
Figure 8: Venn diagram representing $P(H)$ and $P(V)$	31
Figure 9: Venn diagram of Three Conditions and One Feature.....	35
Figure 10: The phase one and the phase two of proposed method	38
Figure 11: Illustration of a condition database with 5 sets of data in each condition.....	39
Figure 12: Milling tool and possible insert conditions	43
Figure 13: Illustration of Radial Depth of Cut vs. Axial Depth of Cut in End Milling.....	46
Figure 14: Experimental Setup	47
Figure 15: FFT of MEMS Accelerometer Data for a Healthy and Damaged Tool	49
Figure 16: FFT of Piezoelectric Accelerometer Data for a Healthy and Damaged Tool	49

Figure 17: Normalized FFT of MEMS Accelerometer Data for a Healthy and Damaged Tool ..	51
Figure 18: Normalized FFT of Piezoelectric Accelerometer Data for a Healthy and Damaged Tool.....	52
Figure 19: Illustration of Windowing by 50%	54
Figure 20: FFT of X, Y, & Z axis of ‘Healthy’ Dynamometer Data.....	55
Figure 21: FFT of X, Y, & Z axis of ‘Damaged’ Dynamometer Data	56
Figure 22: Initial Results of Power Meter Data for 3, 4, & 5 mm Radial Depth of Cut.....	57
Figure 23: Distributions of Power Data from the Healthy and Damaged Condition.....	58
Figure 24: Condition Database in Excel	62
Figure 25: Naïve Bayesian Network for Tool Health Model	65
Figure 26: Quantile-Quantile regression plot of simulated data.....	70
Figure 27: Simulated Distributions, 0% Area Overlap.....	73
Figure 28: Simulated Distributions, 8.01% Area Overlap	74
Figure 29: Simulated Distributions, 21.13% Area Overlap	74
Figure 30: Simulated Distributions, 45.33% Area Overlap.....	75
Figure 31: Distributions for Dynamometer in the Y axis	79
Figure 32: Distribution for healthy and damaged case, Dynamometer 3rd peak Y axis	80
Figure 33: Error as a function of number of samples for 3mm Radial Depth of Cut	82

Figure 34: Error as a Function of Number of Samples for 4mm Radial Depth of Cut.....	83
Figure 35: Error as a Function of Number of Samples for 5mm Radial Depth of Cut.....	84
Figure 36: Error vs. Percent of Distribution Range Overlap	86
Figure 37: Error as a Function of Number of Samples: 0.00% Distribution Area Overlap	87
Figure 38: Error as a Function of Number of Samples: 8.01% Distribution Area Overlap	88
Figure 39: Error as a Function of Number of Samples: 21.13% Distribution Area Overlap	89
Figure 40: Error as a Function of Number of Samples: 45.33% Distribution Area Overlap	90
Figure 41: Histogram of Error as a Function of Number of Samples and Percent Distribution Range Overlap	91
Figure 42: Mesh Plot of Error as a Function of Number of Samples and Percent Distribution Range Overlap	92
Figure 43: Histogram of Error as a Function of Number of Samples and Percent Distribution Range Overlap When Feature with Zero Overlap is Added to Model	93
Figure 44: Close up of Histogram of Error as a Function of Number of Samples and Percent Distribution Range Overlap When Feature with Zero Overlap is Added to Model	94
Figure 45: Effect of Number of Features and Distribution Overlap on Error	96
Figure 46: Simulated Distributions, 0% Area Overlap.....	102
Figure 47: Simulated Distributions, 0.60% Area Overlap.....	102
Figure 48: Close Up Simulated Distributions, 0.60% Area Overlap.....	103
Figure 49: Simulated Distributions, 1.24% Area Overlap.....	103
Figure 50: Simulated Distributions, 2.44% Area Overlap.....	104

Figure 51: Simulated Distributions, 4.55% Area Overlap	104
Figure 52: Simulated Distributions, 8.01% Area Overlap	105
Figure 53: Simulated Distributions, 13.36% Area Overlap	105
Figure 54: Simulated Distributions, 21.13% Area Overlap	106
Figure 55: Simulated Distributions, 31.73% Area Overlap	106
Figure 56: Simulated Distributions, 45.33% Area Overlap	107
Figure 57: Error as a Function of Number of Samples: 0.0% Distribution Area Overlap	107
Figure 58: Error as a Function of Number of Samples: 0.60% Distribution Area Overlap	108
Figure 59: Close Up - Error as a Function of Number of Samples: 0.60% Distribution Area Overlap	108
Figure 60: Error as a Function of Number of Samples: 1.24% Distribution Area Overlap	109
Figure 61: Close Up - Error as a Function of Number of Samples: 1.24% Distribution Area Overlap	109
Figure 62: Error as a Function of Number of Samples: 2.44% Distribution Area Overlap	110
Figure 63: Error as a Function of Number of Samples: 4.55% Distribution Area Overlap	110
Figure 64: Error as a Function of Number of Samples: 8.01% Distribution Area Overlap	111
Figure 65: Error as a Function of Number of Samples: 13.36% Distribution Area Overlap	111
Figure 66: Error as a Function of Number of Samples: 21.13% Distribution Area Overlap	112
Figure 67: Error as a Function of Number of Samples: 31.37% Distribution Area Overlap	112

Figure 68: Error as a Function of Number of Samples: 45.33% Distribution Area Overlap	113
Figure 69: MATLAB Code for FFT Peak Extraction.....	114
Figure 70: MATLAB Function for Bayesian Model for Experimental Case	115
Figure 71: Code for Kfold Cross Validation and Effect of Number of Points	116
Figure 72: General Code for Bayesian Model	117
Figure 73: Code Used to Create Condition Database Structures in MATLAB for Experimental Case.....	119

SUMMARY

Statistical analysis is used quite heavily in production operations. To use certain advanced statistical approaches such as Bayesian analysis, statistical models must be built. This thesis demonstrates the process of building the Bayesian models and addresses some of the classical limitations by presenting mathematical examples and proofs, by demonstrating the process with experimental and simulated implementations, and by completing basic analysis of the performance of the implemented models. From the analysis, it is shown that the performance of the Bayesian models is directly related to the amount of separation between the likelihood distributions that describe the behavior of the data features used to generate the multivariate Bayesian models. More specifically, the more features that had clear separation between the likelihood distributions for each possible condition, the more accurate the results were. This is shown to be true regardless of the quantity of data used to generate the model distributions during model building. In cases where distribution overlap is present, it is found that models performance become more consistent as the amount of data used to generate the models increases. In cases where distribution overlap is minimal, it is found that models performance become consistent within 4-6 data sets.

1 INTRODUCTION

In manufacturing, statistical analysis is used for a range of activities, from metrology to predictive based maintenance. Of all of the statistical approaches used in the field, Bayesian statistics are especially powerful and revealing. The use of Bayesian statistical models in the manufacturing field is not unseen; however, there is a lack of material explaining clear methods for implementing general Bayesian multivariate models or explaining general considerations for use of the models in the field. As such, this thesis addresses the process of building the Bayesian models and the classical limitations that are encountered during model building and use. To limit scope, this thesis discusses model building from the perspective of machine diagnostics even though the process presented to implement multivariate Bayesian models can be used for almost any implementation of statistical analysis on the manufacturing floor.

The term “Bayesian” is derived from Thomas Bayes, an 18th century statistician and philosopher who formulated a powerful theorem, Bayes’ Theorem. The key difference between Bayesian statistics and the more classical frequentist statistics, is that Bayesian statistics utilize prior ‘beliefs’ or prior probabilities. This allows Bayesian statistical approaches to be more effective than classical frequency statistical approaches because they incorporate the relationship between conditional and non-conditional probabilities. In the field of machine diagnostics, Bayesian models have been used for a range of activities, from bearing health diagnostics to updating parameters of alternative types of models. These other models range from statistical models to artificial intelligence models.

A challenge to diagnostics models is that the models representing the machine tool states must account for the noise, vibration, and other outside disturbances found in the manufacturing environment. Using multivariate models, often from the integration of data from several sources (sensor fusion), can help address this challenge as the noise in one sensor can be offset by information from other sensors. As such, the model presented here is a multivariate model that performs sensor fusion. As a further solution, the process for applying a Bayesian multivariate model presented includes a step for updating the Bayesian models with data collected in production conditions. This promotes the ability to update statistical models in their functional environment so that one can better capture the behavior of the model features while they are under the effect of the manufacturing environment or low frequency conditions.

The thesis is laid out as follows. First, a method to build multivariate Bayesian models is explained and supported with mathematical proofs and applied examples. Then, an experimental case using sensor fusion to identify the current state of a tool is provided to illustrate a simple implementation of the general multivariate Bayesian model. Finally, a simulated approach is taken to illustrate the classical limitations during model building and model use. In the simulated testing, the effects of three aspects of the models are explored: the quantity of data used to build the model, the amount of similarity between the distributions of the features for different conditions (distribution overlap), and the quantity of features used to build the model.

2 BACKGROUND AND LITERATURE REVIEW

The process presented to implement multivariate Bayesian models can be used for a wide range of statistical analysis. To limit scope, this discussion approaches model building from the perspective of machine diagnostics. The first goal of machine diagnostics is to use sensor information from the machine to reliably identify the current state of the machine tool. From a probabilistic or statistical perspective, this objective of machine diagnostics is to search for the most probable state of the tool, C_i , given the extracted signal features at a specific instance in time, vector $\text{data}(t_c)$ (Zhu, 2009). The state or condition of the tool could be severity of a worn tool, identification of a damaged bearing, or many other conditions of interest. How the conditions are represented in the models is dependent on the type of model used. Generally, to build each model, data from each condition is recorded and suitable sensor signals or data features that correlate to the machine state are extracted. These sensor signals or data features are then used to train artificial intelligence models, to build statistical models, or a mix of both. Once the models are built, they are ready to be used and commonly are not updated once in their functional state. In this background, first common artificial intelligence based models are reviewed and then statistical based models are reviewed. While discussing statistical based models, Bayesian analysis and a type of Bayesian model called Naïve Bayesian models, are examined in more detail.

2.1 ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANN) are models inspired by neural networks found in the animal brain. ANN can be used to diagnose and predict machine health behavior in cases

where there are many inputs and their relationships to the outputs are not well understood. ANNs are made up of “neurons” or “nodes” and are broken up into a minimum of three layers: the input layer, hidden layers, and the output layer. This is illustrated in Figure 1.

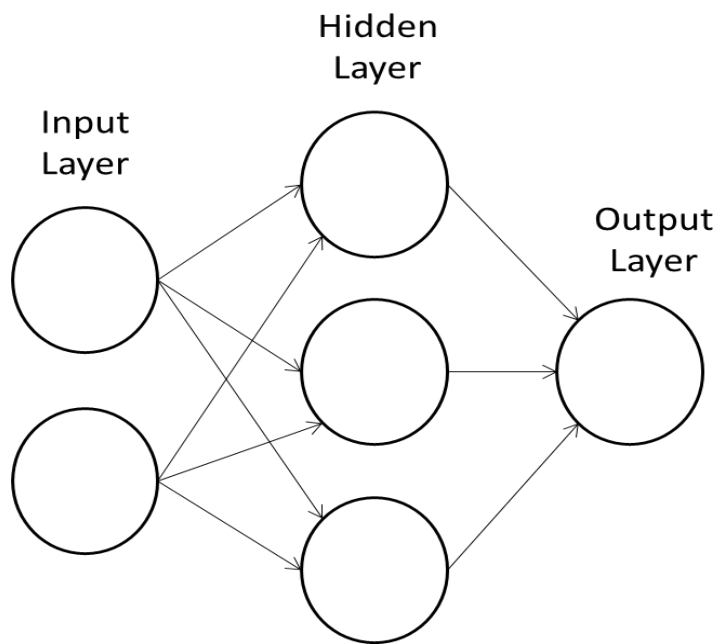


Figure 1: Structure of an Artificial Neural Network

Initially, the values of the input layer and hidden layers are randomly assigned connection weights, represented by arrows in Figure 1. These weights are ‘tuned’ by a learning algorithm until the outputs of the network correlate with expected outputs. The type of Neural Network and learning paradigm used determines how to tune the ANN. Examples

where ANN were used in machine health diagnostics are (B.A. Paya, 1997) (Jaouher Ben Ali, 2015) (Farzaneh Ahmadzadeh, 2013) (C. Sbarufatti, 2016).

2.2 SUPPORT VECTOR MACHINE

Support vector machines (SVM) are used for classification and regression analysis for machine learning. In either supervised or unsupervised mode, SVM analyzes data and recognizes patterns. In supervised (classification) mode, the SVM model is first built using a training set of data where each data point is labeled with the desired output category. In unsupervised (clustering) mode, no labeled training data is used. Instead, SVM is used on a data set to find natural clustering of data points to categories. In either case, after a model is built new input data are mapped to output categories developed when building the model. Examples where SVM was used in machine health diagnostics are (K.C. Gryllias, 2012) (Fafa Chen, 2013) (Khmais Bacha, 2012).

There have been instances where Bayesian and SVM techniques have been integrated together. For example, Saha & Gobel (2008) used Bayesian techniques to develop a method for uncertainty management for performing diagnostics and prognostics of batteries. Bayesian techniques were applied to regression and classification through Relevance Vector Machine (RVM). The Bayesian treatment of Support Vector Machine, in the form of RVM, was used for model recognition. Next, state estimation was performed through a Particle Filter (PF) framework that used the developed model and anticipated operational conditions to estimate the remaining useful life (RUL) of the battery. The estimate of the remaining useful life was in the form of a probability density function. (Saha, 2008)

2.3 NEURO-FUZZY SYSTEMS

A Neuro-fuzzy system is a hybrid of an Artificial Neural Network and Fuzzy System. Although Fuzzy Systems and ANNs have disadvantages, most disappear when the two systems are united together. For example, although both ANN and Fuzzy systems do not require mathematical models to create the final classification model, it is difficult to extract rules from the ANN where as it is easy to from a Fuzzy System. On the other hand, in the same situation, an ANN can be developed without any apriori knowledge where as it is essential for a Fuzzy System. Apriori knowledge is also necessary for a Bayesian system. Examples where Neuro Fuzzy systems are used for machine health diagnostics are (Chaochao Chen, 2012) (Ramasso, 2014) (Soualhi, 2014).

2.4 FREQUENTIST ANALYSIS

A classic statistical method for machine health is frequentist analysis. Frequentist analysis is used on Normally distributed data as is represented in Figure 2. The values on the independent axis represent standard deviations (σ) from the mean. The dependent axis represents percent of data within said standard deviation. The percent of data within $\pm 1\sigma$, $\pm 2\sigma$, and $\pm 3\sigma$ are also indicated in Figure 2.

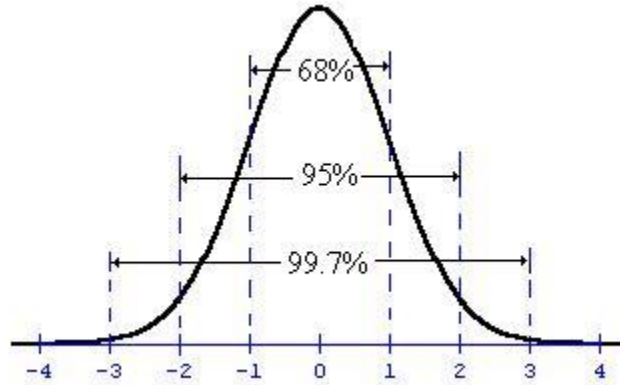


Figure 2: Normal distribution with standard deviations on the independent axis. (Psomas, 2003)

The equation representing Figure 2 is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \text{ for } -\infty < x < \infty \quad (1)$$

where μ and σ are the mean and standard deviation respectively. As shown in Figure 2, the graph of the pdf with a mean of 0 and standard deviation of 1 is also known as the standard Normal curve (Montgomery, 2009). Figure 2 illustrates that the farther a point is from the mean of a Normally distributed data set, the less likely it is to occur under normal conditions. A common manufacturing tool that uses frequentist methods is a control chart (e.g. \bar{x} and s charts).

Important properties of Normal distributions are the mean, μ , variance, σ^2 , and standard deviation, σ . The mean, μ , is an arithmetic average of the values in data set, X, and represents a measure of central tendency of the dataset. As such, the mean is also known as the expected value of X or E(X):

$$E(X) = \frac{1}{n} \sum_{j=1}^n x_j \quad (2)$$

Where n is the number of points in the data set. The variance and standard deviation are the measurement of the dispersion or scatter of the data about the mean μ . As shown in Figure 3, if the majority of values in X tend to be close to the mean, the variance and standard deviation are small; if the majority of values in X are more widely distributed around the mean, then the variance and standard deviation are larger.

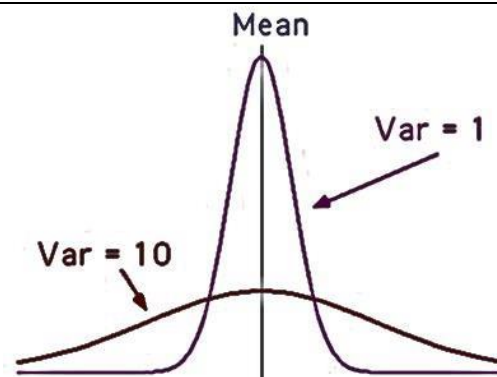


Figure 3: A larger variance signifies a larger spread or scatter of data. (The Basic Statistical Foundation, 2002)

The variance σ^2 can be calculated as:

$$Var(X) = \frac{1}{n} \sum_{j=1}^n (x_j - \mu)^2 \quad (3)$$

The standard deviation is the positive square root of the variance, thus:

$$\sigma = +\sqrt{Var(X)} = +\sqrt{\sigma^2} \quad (4)$$

Examples where frequentist analysis was used in machine health diagnostics are (Michael L. Fugate, 2001) (N. Baydar, 2001).

2.5 CONTROL CHARTS

Control charts use frequentist analysis to test if sample data sets are ‘in control’ or ‘out of control’ from population statistical parameters. When data sets are ‘in control’ they are within the specified limits that characterize desirable conditions. When data sets are ‘out of control’ they indicate a deviation from Normal conditions and that process investigation should take place. In manufacturing, control charts are often used to monitor production specifications when samples are collected at regular intervals from a production line. Figure 4 illustrates a sample distribution within control limits.

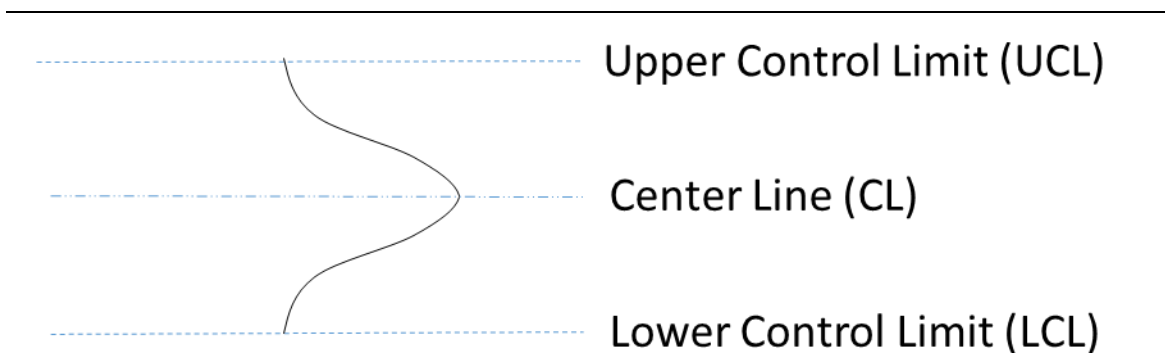


Figure 4: Control chart and sample distribution ‘in control’

In contrast, Figure 5 illustrates how sample distributions can be ‘out of control.’

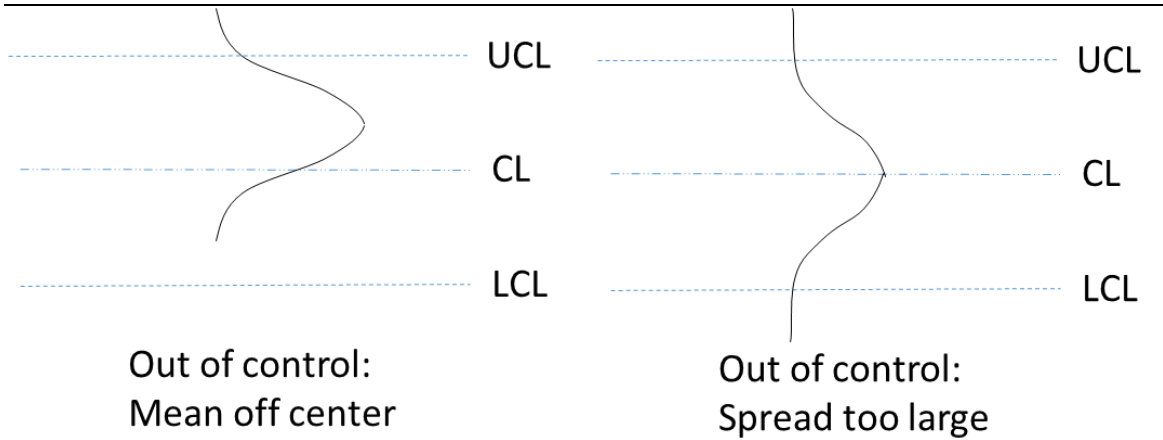


Figure 5: Control chart with ‘out of control’ sample distributions

There are a number of types of control charts with the most popular being the X-bar and the s-chart. The X-bar chart compares the current sample mean to the population mean and trends it across time. The s-chart compares the current sample standard deviation to the population standard deviation and charts it across time. The X-bar and s-chart are used together when one has sample sets with at least 10 data points. The center line (CL) for the X-bar chart is represented as $\bar{\bar{X}}$. To calculate the (CL) for the X-bar chart:

$$\bar{\bar{X}} = \frac{\sum_{i=1}^m n_i \bar{X}_i}{\sum_{i=1}^m n_i} \quad (5)$$

where m is the number of sample sets, n_i is the number of samples in the i^{th} sample set and \bar{X}_i is the sample mean of the i^{th} sample set (Banks, 1989). To calculate the 3σ upper and lower control limits for the X-bar chart, UCL_x and LCL_x respectively:

$$UCL_{\bar{X}} = \bar{\bar{X}} + \frac{3\bar{s}}{\sqrt{n}} \quad (6)$$

$$LCL_{\bar{X}} = \bar{\bar{X}} - \frac{3\bar{s}}{\sqrt{n}} \quad (7)$$

where

$$c_4 = \frac{\left[\frac{n-2}{2}\right]!}{\left[\frac{n-3}{2}\right]!} \sqrt{\frac{2}{n-1}} \quad (8)$$

And n is the number of samples in the set and \bar{s} is the sample standard deviation (Banks, 1989).

To calculate the center line, upper and lower control limit, CL_s , UCL_s and LCL_s respectively, for the s-chart:

$$CL_s = c_4\sigma \quad (9)$$

$$UCL_s = \left(c_4 + 3\sqrt{1 - c_4^2}\right)\sigma \quad (10)$$

$$LCL_s = \left(c_4 - 3\sqrt{1 - c_4^2}\right)\sigma \quad (11)$$

where σ is the population standard deviation (Banks, 1989). As an example where control charts were used for diagnostics, Fugate et al (2001) used control charts to determine when vibration data from a concrete structure were ‘out of control’ and entering a degradation state (Michael L. Fugate, 2001).

2.5.1 STUDENT’S T-DISTRIBUTION

When using frequentist analysis for machine health diagnostics often times student’s t distributions are used if machine testing only produces small amounts of data. The student’s t-distribution is standardized statistical distribution that is used when the

population standard deviation is unknown and/or when the number of sample values is less than 30. To use the student's t-distribution, the population mean should be known. One way to use the student's t-distribution is to use a t-score (similar to a z-score for the Normal distribution). The t-score is calculated as follows:

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} \quad (12)$$

where t is the t score, \bar{x} is the sample mean, μ is the population mean, s is the sample standard deviation, and n is the samples size.

The student's t-distribution is defined by its degrees of freedom. The degrees of freedom is

$$df = n - 1 \quad (13)$$

where df is the degrees of freedom and n is the sample size. There is a different student's t-distribution for each degree of freedom and as the degrees of freedom reach 30, the student's t-distribution converges to a Normal distribution.

The mean is always zero for a standard student's t distribution. To calculate the population variance of a standard t-distribution, σ^2 :

$$\sigma^2 = \frac{df}{(df - 2)} \quad (14)$$

The variance is always be more than one for the standard student's t-distribution unless the degrees of freedom are infinite. As such, the student's t-distribution is a more conservative form of the Normal distribution.

2.5.2 TYPE I AND TYPE II ERROR

Type I and Type II error represent the two types of error associated with using frequentist analysis. Type I and Type II error are related to testing a statistical null hypothesis. The null hypothesis is the hypothesis that there is no statistically significant difference between the desired distribution and the current distribution of a data set. Type I error is when the null hypothesis is incorrectly rejected. In other words, type I error is the case where a system reports that the current distribution has shifted from the desired when it actually has not. It is a false positive or false alarm. Type II error is when the alternative hypothesis is incorrectly rejected (also described as failing to reject the null hypothesis). This means that the current distribution has shifted from the desired distribution; however, it has not been detected. This is more commonly known as a false negative or consumer's risk. Both Type I and Type II error can be calculated and represented statistically. Type I error is represented as α and can be found in a t distribution table after calculating the t statistic. The t statistic is calculated with the following two Equations

$$t = \frac{\mu_1 + \mu_2}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \quad (15)$$

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \quad (16)$$

where t is the t statistic to use in the t-distribution look-up table, μ is the mean of each distribution, n is the number of samples in each distribution, s_p is the pooled standard deviation and s_1 and s_2 are the standard deviations of each distribution, respectively. From the t distribution table, the p-value associated with the t statistic is the probability of

type I error. This has become such a common calculation that Microsoft Excel has included a TDIST function that outputs the p-value for the t-distribution.

In manufacturing, Type I and Type II error are often used for quality engineering. Understanding Type I and Type II error allows engineers to calculate the probability of detecting a shift in process parameters that affect product specifications. By using the frequentist approach and understanding Type I and Type II error, one optimizes the number of quality inspections performed while ensuring produced parts match customer specification. For example, if a large batch of a product is supposed to have a mean diameter of 15mm, a process engineer would use Type I and Type II error to decide how many samples they should inspect from each batch and what should be the process limits of the sample mean and standard deviation. Consequently, when the mean and standard deviation of the sample are outside of the process limits during an inspection, the batch is rejected. This is often done using control charts. In this example, the Type I error would be the probability of rejecting a good batch and the Type II error would be the probability of accepting a bad batch. The challenge with Type I and Type II error is that changing the process limits to decrease the probability for Type I error increases the probability for Type II error. Likewise, changing the process limits to decrease the probability of Type II error increases the probability for Type I error. Increasing the number of samples taken is one way to decrease both Type I and Type II error, however, this is often costly and may delay detection of a process shift. Type I and Type II error is also used when using frequentist analysis in machine health diagnostics.

As an example of using frequentist testing for machine health diagnostics, Jun Ma & James Li (1995) developed a scheme for detecting localized defects in rolling element

bearings using hypothesis test theory. By detecting the distributions of vibrations when the rolling elements were in contact with the defect and the distributions of vibrations when the rolling elements were not in contact with the defect, they were able to classify two separate distributions that, when a defect was present, alternated at the rate that the rolling elements passed over the defect. Thus, by using a hypothesis test to classify the bearing vibration as one of the two distributions and checking the switching frequency, a defect was detected. They experimentally confirmed the method with different defects under different loads (Jun Ma, 1995).

2.6 BAYESIAN ANALYSIS

Bayesian statistical methods and models are derived from Bayes' Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (17)$$

where $P(A|B)$ is the probability of A given B, $P(B|A)$ is the likelihood of B given A, $P(A)$ is the probability of A, and $P(B)$ is the probability of B. The meaning of each probability is more easily comprehended through an example. For example, if Bayes Theorem is rewritten

$$P(H|V) = \frac{P(V|H)P(H)}{P(V)} \quad (18)$$

where H represents a condition “tool is Hhealthy,” V represents evidence that “there is Vibration over a set alarm threshold” and the probabilities are as follows:

Table 1: Nomenclature for example and equation (18)

Representation	Meaning	Type of probability
P(H)	Probability that the “tool is <u>H</u> healthy”	Prior, Unconditional
P(V)	Probability that “there is <u>V</u> ibration over a set alarm threshold”	Evidence, Unconditional
P(V H)	Likelihood that “there is <u>V</u> ibration over a set alarm threshold” given that the “tool is <u>H</u> healthy”	Likelihood, Conditional
P(H V)	Probability the “tool is <u>H</u> healthy” given that “there is <u>V</u> ibration over a set alarm threshold”	Posterior, Conditional

Then Bayes Theorem can be expressed as

The probability that the tool is healthy given that there is machine vibration over the preset alarm threshold =

The likelihood that there is vibration over the preset alarm threshold when the tool is healthy multiplied by the prior probability that the tool is healthy and divided by the probability there is vibration over the set alarm threshold.

(19)

or more generally

(Posterior) Probability of the condition given the evidence =

$$\frac{\textit{Likelihood of the evidence given the condition} \times \textit{Prior probability of the condition}}{\textit{Probability of the evidence}} \quad (20)$$

For compactness, equation (20) can be rewritten as:

$$\textit{Posterior} = \frac{\textit{Likelihood} \times \textit{Prior}}{\textit{Evidence}} \quad (21)$$

The likelihood (or likelihood probability) technically is not a probability density function (PDF). By definition, to be a PDF the summation or integral of all possible probabilities in the distribution must equal one (Drake, 1967). In the case of the likelihood represented in equation (21), this would be the summation of the ‘probability of the evidence given the condition’ for each condition. For technical clarity, the evidence (not the probability of evidence) is now defined as a random variable x and the condition defined as a hypothesis. Although the summation of probabilities of all possible values of a random variable x across a single hypothesis adds to one (by definition), the summation of probabilities for all hypotheses given a single value of x does not necessarily add up to one. Moving forward, the posterior and prior are true probabilities. The posterior probability is conditional probability as it is constrained to a single condition. The prior distribution is an unconditional distribution because it is independent of any other variables.

Bayes rule is a powerful tool because it allows one to derive conditional probabilities using probabilities that are more readily found through prior experience or experimentation. To illustrate, the prior probability that a tool is healthy and the evidence

probability that vibration is over a set threshold can be calculated from data from the shop floor or be found through experimentation. The likelihood that there is vibration over the alarm threshold given the tool is healthy can also be generated through experimentation or observation. Using these three pieces of information together, one can understand how to calculate the probability that a tool is healthy whenever the alarm “vibration has exceeded a set threshold” is triggered. Examples where Bayes approach was used in machine health diagnostics are (Jun Ma, 1995) (Mehta, 2015) (Mosallam, 2014) (Nagi Z. Gebraeel, 2005) (Saha, 2008) (Tobon-Mejia, 2012) (Wang, 2012).

As a more detailed example in the field, Gebraeel et. all (2005) provided Bayesian updating methods to develop a closed form remaining useful life distribution for a monitored manufacturing device. This remaining useful life distribution was used to update stochastic parameters of exponential degradation models. To build the models, they use data from a “population” of components in addition to data from the device for which they are predicting residual life. They further use a degradation model with a Brownian motion (Wiener) error process. Periodically, the system uses real time data and a Bayesian method to update the exponential degradation models used to predict residual life. From this study, it was seen how machine diagnostic models can address the random nature found on the shop floor by having stochastic parameters in the model update with real time conditions (Nagi Z. Gebraeel, 2005).

2.6.1 NAÏVE BAYESIAN MODELS

A challenge to diagnostic algorithms is that the models representing machine tool states must account for the noise, vibration, and other outside disturbances that are found in the manufacturing environment. Using multivariate models, often from sensor fusion, can

help address this challenge as the noise in one sensor can be offset by information from other sensors. Although using multivariate models is more robust, this can cause the models to become complicated and difficult to build. As such, using naïve Bayesian approaches to simplify multivariate Bayesian models is common.

Naïve Bayesian models are Bayesian models that exploit conditional independence assumptions to produce a compact representation of a high dimensional probability distribution. In other words, naïve Bayesian models use conditional independence to produce a simple compact representation of the otherwise complicated multivariate distribution. For example, a multivariate representation of Bayes rule is

$$P(C_i|\vec{d}) = \frac{P(\vec{d}|C_i)P(C_i)}{P(\vec{d})} \quad (22)$$

where

$$\vec{d} = [F_1, \dots, F_m] \quad (23)$$

and F_j represents a feature and C_i represents a class or condition. Without conditional independence, calculating $P(C_i|\vec{d})$ and $P(\vec{d})$ is complicated and sometimes impossible. On the other hand, if features are independent given class (in other words conditionally independent) and classes are mutually exclusive and exhaustive, then

$$P(\vec{d}|C_i) = \prod_{j=1}^m P(F_j|C_i) \text{ where } \vec{d} = [F_1, \dots, F_m] \quad (24)$$

and

$$P(\vec{d}) = \sum_{i=1}^n P(C_i) \prod_{j=1}^m P(F_j|C_i) \quad (25)$$

Although the assumption of conditional independence is usually unrealistic and often violated, the Naïve Bayesian model has proven often be robust to violations of this condition (Friedman, 1997) (Pearl, 1988) (M. Elangovan, 2010). (Rish, 2001) (Domingos, 1997) (Hilden, 1984) (Langley, 1992) (Francesco, 2012) (Zhang, 2004). For example, M. Elanovan et al (2010) found when using statistical features instead of histogram features for condition monitoring of single point carbide tipped tool, the difference in accuracy between the Bayesian network and the Naïve Bayesian model was about 1%. Also, Fancesco et al (2012) confirmed that a data driven stochastic approach relying on a Naïve Bayesian classifier was a feasible tool to estimate the remaining useful life of damaged thrust ball bearings. Furthermore, Friedman and Goldszmidt compared naïve Bayes to Bayesian networks (Pearl, 1988) and found that not only did the Bayesian network not perform better than the naïve Bayesian network; it actually reduced accuracy in some domains. Why naïve Bayesian networks perform well even when the assumption of conditional independence is violated has been studied and tested (Hand, 2001) (Rish, 2001)(Zhang, 2004).

2.6.2 IMPLEMENTING BAYESIAN MODELS

As a reminder, the objective of machine diagnostics is to search for the most probable state of the tool, C_i , given the extracted signal features at a specific instance in time, vector $\text{data}(t_c)$ (Zhu, 2009). As one would expect, the model that most closely matches the current sensor data is the most probable current condition of the tool. This is like a

form of pattern matching. As is found in pattern matching algorithms, many applications of Bayesian models for machine health diagnostics rely on a basic two phase structure for implementation (Tobon-Mejia, 2012), (Mosallam, 2014), (Gebrael, 2005). For example, when D.A Tbon-Mejia et. al (2011) used dynamic Bayesian Networks to perform prognostics and diagnostics for CNC machine tool health assessment, the developed algorithm was based on two main phases: an off-line (development) phase 1 and an on-line (useful) phase 2. In the off-line phase, raw data from experimentation was used to develop data analysis and feature extraction methods as well as build models that are used in the diagnostic and prognostic assessments. In the on-line phase, current data from a machine tool was input into the constructed models to classify the current health state and compute the remaining useful life and the associated confidence value (Tobon-Mejia, 2012). The structure from Tobon-Mejia et. al 2012 is shown in Figure 6. The algorithm identifies the current wear state of the tool and then predicts the remaining useful life (RUL). This model was also used for bearing health diagnostics.

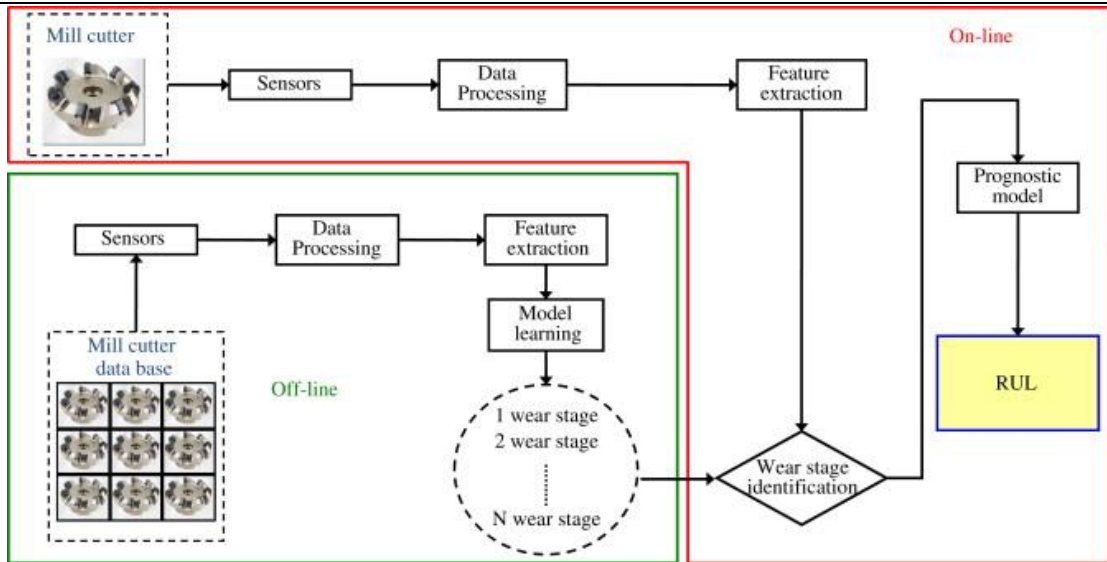


Figure 6: Off-line (Phase 1) and On-line (Phase 2) set up for diagnostics algorithm by Tobon-Mejia 2012

In Figure 6 it can be seen that the models are built in the off-line phase by the “model learning” block and the models are stored in the dashed circle. The comparison of the models to current data is shown by the “Wear stage identification” diamond in the on-line phase. The results of the comparison are the inputs to the prognostic model in the online phase.

The main reason why many algorithms rely on a two phase approach to implement Bayesian models is to separate the system development phase (phase 1), from the functional on-line phase (phase 2). In other words, since machine diagnostics is like a form of pattern matching, phase 1 is necessary to build the models, and phase 2 is necessary to use the developed models. Moreover, a separate initial phase is essential to find signal features that correlate to the condition of the tool and to standardize methods for extracting data before the models can be built or tested. Once this practice is

standardized, the methods can be used to ensure quality results in the functional online phase.

As there are two general phases for implementing a Bayesian model, there are also general tasks for each phase. Since phase 1 is for system development, usually data acquisition, signal processing, and feature extraction methods are established and model building performed in phase 1. Data acquisition is the process of acquiring electrical signals that measure physical conditions, and converting them into digital values as digital samples of data. Signal processing, in machine diagnostics, filters out operational and environmental disturbances to reduce the data to condition indicators. Feature extraction is extracting features from sensor signals that correlate with the machine condition states. Model building is a “learning” step where data driven, experience driven, etc. models are built up into a model database. Phase 2 utilizes the data acquisition, signal processing, and feature extraction methods established in phase 1 to process raw sensor data into an input for the models. The next step of phase 2 is dependent on if the system is deterministic or nondeterministic thus is not a general step.

2.6.3 MODEL UPDATING

Although sensor fusion helps machine tool state models be less affected by the noise, vibration, and other outside disturbances that are found in the manufacturing environment, it does not completely solve the problem of having the models address the stochastic nature of most cutting processes or the heavy disturbances caused by other machinery in the manufacturing environment. Theoretically, this challenge could be caused by the idealness of the environment in which the models are developed. Solving this problem is no easy feat because there is strong motivation to develop the models in

the ideal offline phase and to develop the models in the functional online phase. First, during selection of features for feature extraction, it is accepted that sensor signals should be as free from environmental noise as possible to confidently correlate the signals with machine states. Thus, feature extraction for the models should be performed in an ideal off-line state. On the other hand, when building up the models, the models should take into account the increased uncertainty and spread that occur due to the noise, vibration, and other environmental conditions due to the manufacturing environment, thus model building should be performed in an on-line state.

This thesis presents a general approach for applying Bayesian models for machine diagnostics using sensor fusion that addresses the aforementioned challenges by breaking model building into two sections. First, to preserve quality feature selection, selecting methods for feature extraction and initial model building remains in phase 1. Second, to better represent the stochastic nature of features in the models, the models can be updated in the online phase 2 where the machine is already in its functional environment. This is accomplished by having an ‘update’ feature in the implementation algorithm that adds newly classified data to the condition databases that are used to generate the models.

3 METHODS

The methods section satisfies three main purposes: to address the process to generally build and apply a multivariate Bayesian model for sensor fusion, to present the methods used to implement the models for an experimental case, and to present the methods used to test simulated models in order to illustrate the classical limitations during model building and model use. In more detail, the sections satisfying each purpose are as follows.

To address the process to generally build and apply the multivariate Bayesian model for sensor fusion, first the method assumptions for building the Bayesian model are stated and the reasoning for each given in the ‘Method Assumptions’ section. Second, in the ‘Mathematical Tools’ section, the mathematical proofs used to obtain each of the main Bayesian models components are provided and supported with applied examples. Third, in the ‘Bayesian Model for Sensor Fusion’ section, the final mathematical representation of the general multivariate Bayesian model for n conditions and m features is presented and a mathematical example provided. Next, the method for implementation and metrics for model results are stated in the ‘Method for Implementation of Bayesian Models’ section and ‘Metrics for Results’ section respectively. This concludes the general explanation of how to build and generally apply a multivariate Bayesian model for sensor fusion.

The next third of the methods section provides a demonstration of an experimental implementation of the general Bayesian method in the ‘Experimental Implementation:

Tool Diagnostics for a Three Insert Tool' section. In this section the purpose, materials, experimental procedure, and the process for building and using the models are presented.

In the final third of the methods section, the procedure for using simulated data to address the classical limitations that occur when building and using a multivariate Bayesian model is presented. This section includes the purpose for simulated testing, the specific inputs and computational tools used for testing, and the specific procedures for testing the effects of overlapping distributions and the effects of model updating (i.e., the effects of quantity of data used to build the models). The code used for testing is also discussed and provided in the computational tools section.

3.1 METHOD ASSUMPTIONS

1. Inspection is assumed to be perfect. Therefore, assume data used to build models do not contain incorrect values.
2. Observations from multiple sensors are random and conditionally dependent on the machine state.
3. Conditions, also known as the model classes, are mutually exclusive and collectively exhaustive.
4. Features are conditionally independent of each other

Assumption 1 is self-explanatory. Assumption 2 establishes the stochastic state between the observed features and the system state (Wang and Christer, 2000).

Assumption 3 is required for valid computation of the probability of evidence (equation (40)). Assumption 4 does not always hold; however, a number of papers have found that the Bayesian model performs well even when the features are not

conditionally independent (Friedman, 1997) (Pearl, 1988) (M. Elangovan, 2010) (Rish, 2001) (Domingos, 1997) (Hilden, 1984) (Langley, 1992) (Francesco, 2012) (Zhang, 2004).

3.2 MATHEMATICAL TOOLS

3.2.1 BAYESIAN MODEL COMPONENTS

As is seen from the equation representing the basic structure of Bayes Theorem (equation (21)), Bayesian Models are made from prior probabilities, likelihood probabilities, and the probability of evidence. This section provides an in depth discussion about each.

3.2.1.1 *Prior Probabilities*

In this method, the prior distributions are the classical probabilities that each tool condition would occur under normal conditions. Mathematically represented as $P(\text{Condition})$ or $P(C_i)$. In this generalized method, the prior probabilities are generated through observation, experimentation, or prior knowledge. For example, if there are two conditions, Healthy tool and Damaged tool, and it is known from gathered data on the manufacturing floor that the tool is damaged 5% of the time, the two prior probabilities are $P(\text{Healthy}) = 0.95$ and $P(\text{Damaged}) = 0.05$.

3.2.1.2 *Likelihood Probabilities*

Likelihood probabilities are the likelihood of observed data results given the machine tool is in a specific condition. For an example, view Table 1. In this method there are univariate likelihood probabilities and multivariate likelihood probabilities. The univariate likelihood probabilities are mathematically represented as

$P(\text{feature}|\text{Condition})$ or $P(F_j|C_i)$. This is the likelihood one would observe the value of F_j given the current state of the tool is C_i . The symbol "|" signifies "given." The multivariate likelihood probabilities are mathematically represented as, $P(\text{Data}|\text{Condition})$ or $P(\vec{d}|C_i)$. This is the probability one would observe the data \vec{d} given the current state of the tool is C_i .

There are multiple ways to calculate the univariate likelihood probability. The first method discussed is a histogram approach. Data for a feature belonging to condition C_i is used to generate a histogram. For clarity, a possible histogram and condition history is illustrated in Figure 7.

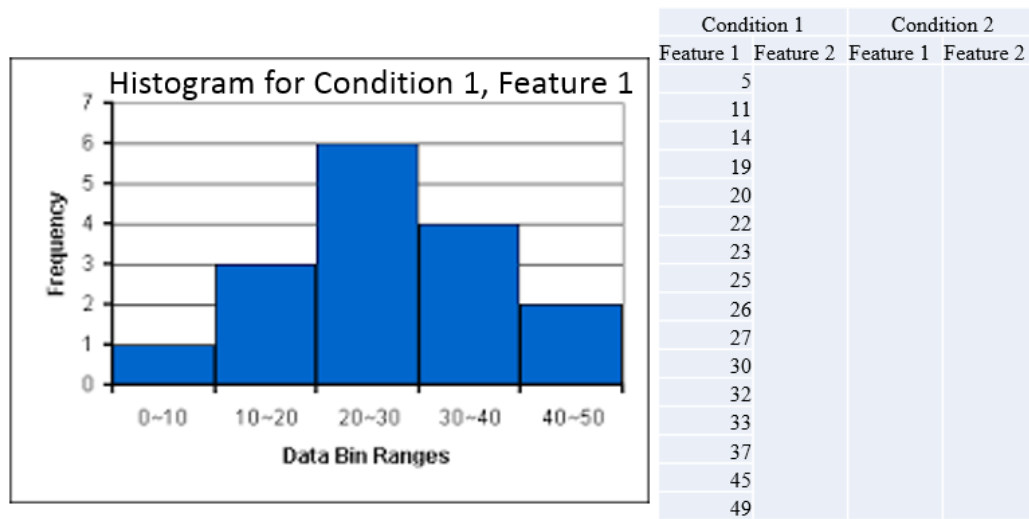


Figure 7: Example Histogram for Feature $F_j|C_i$

The frequency on the vertical represents the number of data points that were within the bin range indicated on the horizontal axis. The likelihood for each bin is calculated as

$$\text{likelihood of bin} = \frac{\text{frequency of bin}}{\text{total number of data points}} \quad (26)$$

The total number of data points is the total number of data points for feature F_j in condition C_i . The likelihood probability $P(F_j|C_i)$ is equal to the likelihood of the bin whose bin range encompasses the value of F_j . For example, if $F_j = 11$ it would be in the 2nd bin whose frequency is 3 and bin range is 10 to 20. If one adds up the frequencies of each bin, the total number of points is found to be 16. Therefore if $F_j = 11$, the likelihood would be 3/16 or 0.188. A shortcoming of this technique is that it is possible to have zero probabilities.

If the behavior of the feature for each condition is known, the probability density function (PDF) that represents the behavior for the feature can be used to calculate the likelihood probability instead of using a histogram. For example, if a feature is Normally distributed, the PDF is equation (1)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \text{ for } -\infty < x < \infty$$

By extracting the μ and σ for feature F_j in condition C_i , one can calculate $P(F_j|C_i)$ as $f(F_j)$ of equation (1) shown above. This approach can also be used for a variety of distributions including Lognormal, Gamma, and Poisson density functions.

As this approach builds Naïve Bayesian models, it is assumed that the features are conditionally independent (as discussed in section 2.6.1). In other words, the features are

dependent on the condition of the tool, but independent of each other. With the assumption of conditional independence, the multivariate likelihood probability, $P(\vec{d}|C_i)$, for condition C_i of all features $F_1 \dots F_m$ using the univariate likelihood probabilities $P(F_j|C_i)$ as follows.

$$P(\vec{d}|C_i) = \prod_{j=1}^m P(F_j|C_i) \text{ where } \vec{d} = [F_1, \dots, F_m] \quad (27)$$

Equation (27) demonstrates that the multivariate likelihood probability is simply the product of the univariate likelihood probabilities.

3.2.1.3 Probability of Evidence

In this thesis, the systematic approach uses the likelihood and prior probabilities to generate the evidence probability. This decision exploits that the likelihood and prior probabilities must already be available to calculate the numerator of the Bayesian model (equation (21)). Given that $P(E_v)$ represents the probability of evidence, C represents a condition of the machine tool, and “ \sim ” represents logical “not”, the general equation for the method used to calculate the probability of evidence is

$$P(E_v) = P(E_v|C)P(C) + P(E_v|\sim C)P(\sim C) \quad (28)$$

The following example demonstrates that equation (28) is a valid method to calculate the probability of evidence (using the example on page 15). The variables are defined in Table 1. In this example, the probability of evidence is $P(V)$.

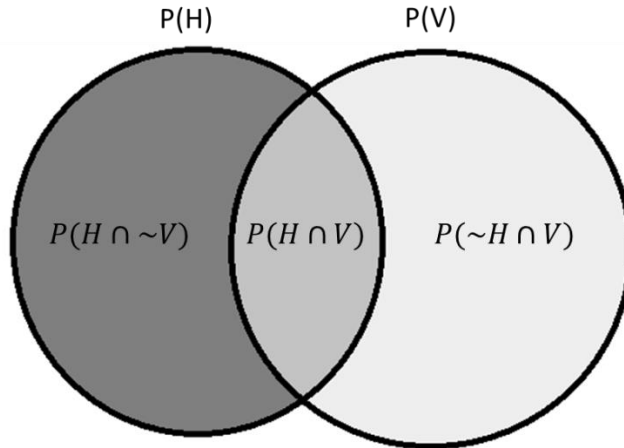


Figure 8: Venn diagram representing $P(H)$ and $P(V)$

Looking at the Venn diagram above, the circle on the left represents the probability the tool is Healthy, $P(H)$, and the circle on the right represents the probability there is vibration over a set alarm threshold, $P(V)$. The probability for each subspace of the circle is represented in the center of each subspace. Each subspace is indicated by a unique color.

From the Venn diagram, it is seen that the probability of evidence, $P(V)$

$$P(V) = P(H \cap V) + P(\sim H \cap V) \quad (29)$$

Using the commutative law these components can be changed to,

$$P(H \cap V) = P(V \cap H) \quad (30)$$

$$P(\sim H \cap V) = P(V \cap \sim H) \quad (31)$$

Then from the probability chain rule,

$$P(V \cap H) = P(V|H)P(H) \quad (32)$$

$$P(V \cap \sim H) = P(V|\sim H)P(\sim H) \quad (33)$$

Therefore, substituting (32) and (33) into (29)

$$P(V) = P(V|H)P(H) + P(V|\sim H)P(\sim H) \quad (34)$$

Thus, probability of evidence, $P(V)$, is calculated using the prior and likelihood probabilities as defined by (28).

Application examples aside, the general equation for the method used to calculate the probability of evidence (28) can be validated through a mathematical proof.

Proof of (28)

From the probability chain rule, (28) can be rewritten as

$$P(E_v) = P(E_v \cap C) + P(E_v \cap \sim C) \quad (35)$$

Next, from the distributive law this can be reorganized as,

$$P(E_v) = P(E_v)(P(C) + P(\sim C)) \quad (36)$$

By the axiom of the algebra of events, $P(A)+P(\sim A)=U$ where U is the universal set,

$$P(C) + P(\sim C) = U \quad (37)$$

Thus substituting (37) into (36)

$$P(E_v) = P(E_v)(U) \quad (38)$$

Finally, by the law $P(A)*U=P(A)$

$$P(E_v) = P(E_v) \quad (39)$$

From (34) and the proof above, it is seen that the method used to calculate the probability of evidence requires at least two conditions to exist. In (34) the two conditions are “the tool is healthy” (H), and “the tool is not healthy” (\sim H).

In reality, this approach is valid for more than just 2 conditions. This thesis now demonstrates and proves how the mathematical representation of the evidence probability for two conditions (28) can be expanded to build models for up to n conditions where n is an integer from 1 to ∞ . A condition is now represented as C_i , where i is a number between 1 to the total number of conditions in the model.

Demonstration and Proof

If C_1, \dots, C_n are mutually exclusive and collectively exhaustive conditions, then

$$P(C_1) + \dots + P(C_n) = U \quad (40)$$

Which is also written as

$$\sum_{i=1}^n P(C_i) = U \quad (41)$$

Thus, by the law $P(A) \cdot U = P(A)$

$$P(E_v) = P(E_v) \left(\sum_{i=1}^n P(C_i) \right) \quad (42)$$

Using the distributive law

$$P(E_v) = \sum_{i=1}^n P(E_v \cap C_i) \quad (43)$$

Finally, using the probability chain rule,

$$P(E_v) = \sum_{i=1}^n P(E_v | C_i) P(C_i) \quad (44)$$

Thus for n conditions, the probability of evidence is mathematically represented by (44).

The following example demonstrates the validity of the method to calculate the probability of evidence for multiple conditions using the example on page 15. The previous probabilities are defined in Table 1 and additional probabilities for two additional conditions are defined below. The two new conditions are, W for “tool is Worn” and Ch for the “tool is Chipped.”

Table 2: Probabilities for Multi-Condition Example

Representation	Meaning	Type of probability
$P(W)$	Probability that the “tool is <u>W</u> orn”	Prior, Unconditional
$P(Ch)$	Probability that the “tool is <u>Ch</u> ipped”	Prior, Unconditional
$P(V W)$	Probability (likelihood) that “there is <u>V</u> ibration over a set alarm threshold” given that the “tool is <u>W</u> orn”	Likelihood, Conditional
$P(V Ch)$	Probability (likelihood) that “there is <u>V</u> ibration over a set alarm threshold” given that the “tool is chipped”	Likelihood, Conditional
$P(W V)$	Probability that the “tool is <u>W</u> orn” given that “there is <u>V</u> ibration over a set alarm threshold”	Posterior, Conditional
$P(Ch V)$	Probability that the “tool is <u>Ch</u> ipped” given that “there is <u>V</u> ibration over a set alarm threshold”	Posterior, Conditional

In this example, the probability of evidence is $P(V)$. As in the example on page 15, H represents the condition “tool is healthy.”

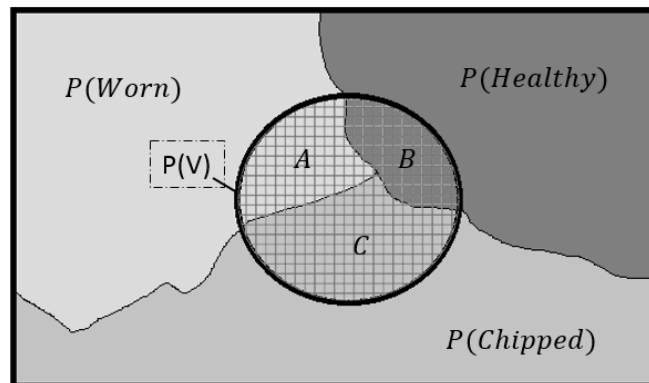


Figure 9: Venn diagram of Three Conditions and One Feature

The rectangle containing the Venn diagram represents U. The light grey area in the top left represents the condition “tool is worn.” The medium grey area on the bottom of the space represents the condition “tool is chipped.” The dark grey area at the top right of the space represents “Tool is healthy.” The circle in the center represents the set, “there is vibration over a set alarm threshold.” The conditions, “tool is worn” “tool is chipped” and “tool is healthy” are mutually exclusive and exhaustive, meaning they do not overlap each other and cover the entire universal set U. There are three overlap regions between the set V and the three conditions represented by A, B, and C where

$$A = P(V \cap W) = P(V|W)P(W) \quad (45)$$

$$B = P(V \cap H)P = P(V|H)P(H) \quad (46)$$

$$C = P(V \cap Ch) = P(V|Ch)P(Ch) \quad (47)$$

From the diagram, it can be seen that the probability of evidence P(V) equals

$$P(V) = A + B + C \quad (48)$$

Therefore

$$P(V) = P(V|W)P(W) + P(V|H)P(H) + P(V|Ch)P(Ch) \quad (49)$$

Through comparison, one can see that (49) would be generated by applying the equation for the probability of evidence of n conditions (44).

3.2.2 BAYESIAN MODEL FOR SENSOR FUSION

As stated previously, Bayesian Models are made from prior probabilities, likelihood probabilities, and the probability of evidence (21). Hence, to derive a general Bayesian model for n conditions and m features, the mathematical representation of the multivariate likelihood probability (27), the mathematical representation of the evidence

probability (44), and the classic prior probability $P(C_i)$, are implemented into the basic structure of Bayes Rule (21), to get

$$P(C_i|\vec{d}) = \frac{P(C_i) \prod_{j=1}^m P(F_j|C_i)}{\sum_{i=1}^n P(C_i) \prod_{j=1}^m P(F_j|C_i)} \quad (50)$$

where $P(C_i|\vec{d})$ is the posterior probability, C_i represents each condition or state of the manufacturing equipment, $[F_1, \dots, F_m]$ are the features extracted from a data set and $\vec{d} = [F_1, \dots, F_m]$ where \vec{d} is a vector of the extracted features.

For example, supposed there are 3 features and 3 conditions. The inputs to the model are the values of the 3 features and the outputs are be three posterior probabilities: $P(C_1|\vec{d})$, $P(C_2|\vec{d})$, $P(C_3|\vec{d})$. The inputs are current data feature values from the sensors on the machine and the outputs are the probability the machine is in condition 1, condition 2 and condition 3 respectively. Applying Bayes' Theorem, the models are written fully as defined by (50)

$$P(C_1|\vec{d}) = \frac{P(C_1)P(F_1|C_1)P(F_2|C_1)P(F_3|C_1)}{P(C_1)P(F_1|C_1)P(F_2|C_1)P(F_3|C_1) + P(C_2)P(F_1|C_2)P(F_2|C_2)P(F_3|C_2) + P(C_3)P(F_1|C_3)P(F_2|C_3)P(F_3|C_3)} \quad (51)$$

$$P(C_2|\vec{d}) = \frac{P(C_2)P(F_1|C_2)P(F_2|C_2)P(F_3|C_2)}{P(C_1)P(F_1|C_1)P(F_2|C_1)P(F_3|C_1) + P(C_2)P(F_1|C_2)P(F_2|C_2)P(F_3|C_2) + P(C_3)P(F_1|C_3)P(F_2|C_3)P(F_3|C_3)}. \quad (52)$$

$$P(C_3|\vec{d}) = \frac{P(C_3)P(F_1|C_3)P(F_2|C_3)P(F_3|C_3)}{P(C_1)P(F_1|C_1)P(F_2|C_1)P(F_3|C_1) + P(C_2)P(F_1|C_2)P(F_2|C_2)P(F_3|C_2) + P(C_3)P(F_1|C_3)P(F_2|C_3)P(F_3|C_3)}. \quad (53)$$

The model for each can be compactly written as

$$P(C_1|\vec{d}) = \frac{P(C_1)P(\vec{d}|C_1)}{P(C_1)P(\vec{d}|C_1) + P(C_2)P(\vec{d}|C_2) + P(C_3)P(\vec{d}|C_3)} \quad (54)$$

$$P(C_2|\vec{d}) = \frac{P(C_2)P(\vec{d}|C_2)}{P(C_1)P(\vec{d}|C_1) + P(C_2)P(\vec{d}|C_2) + P(C_3)P(\vec{d}|C_3)} \quad (55)$$

$$P(C_3|\vec{d}) = \frac{P(C_3)P(\vec{d}|C_3)}{P(C_1)P(\vec{d}|C_1) + P(C_2)P(\vec{d}|C_2) + P(C_3)P(\vec{d}|C_3)} \quad (56)$$

3.3 METHOD FOR IMPLEMENTATION OF BAYESIAN MODELS

The method is broken up into an initial offline phase, phase 1, and a functional online phase, phase 2. The method is illustrated in Figure 10.

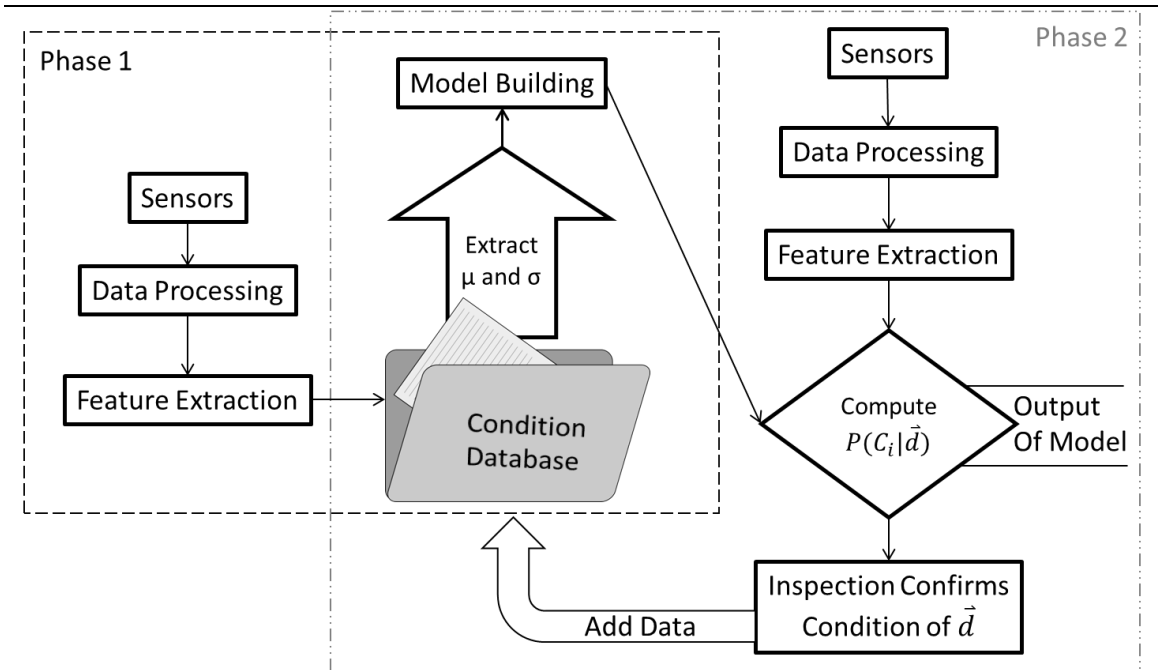


Figure 10: The phase one and the phase two of proposed method

3.3.1 PHASE 1

In phase 1, sensors are selected and placed on the equipment, data processing methods are selected and implemented, and features correlated to the conditions are discovered and extracted. In a ‘Condition Database’ as indicated in the center of Figure 10, feature values from the experimental data collected in phase 1 are stored. For example, if there are two possible conditions, C_1 and C_2 , and two features extracted, F_1 and F_2 , and only 5 sets of data for each condition have been collected, then the condition database could be shown follows:

Condition 1		Condition 2	
F1	F2	F1	F2
0.776	0.067	0.474	0.559
0.263	0.019	0.546	0.351
0.662	0.481	0.640	0.517
0.249	0.970	0.401	1.394
0.207	0.019	0.977	0.386
$\mu(F1 \in C1)$	$\mu(F2 \in C1)$	$\mu(F1 \in C2)$	$\mu(F2 \in C2)$
$\sigma(F1 \in C1)$	$\sigma(F2 \in C2)$	$\sigma(F1 \in C2)$	$\sigma(F2 \in C2)$

Figure 11: Illustration of a condition database with 5 sets of data in each condition

Using the method to generate likelihood probabilities for Normally distributed data explained in section 3.2.1.2, distribution characteristics, μ and σ , are extracted from the history of values of each feature in each condition in the condition database. For example, from the conditional database shown in Figure 11, four means, $\{\mu(F_1 \in C_1), \mu(F_2 \in C_1), \mu(F_1 \in C_2), \mu(F_2 \in C_2)\}$ and four standard deviations, $\{\sigma(F_1 \in C_1), \sigma(F_2 \in C_1), \sigma(F_1 \in C_2), \sigma(F_2 \in C_2)\}$

C_2), $\sigma(F_2 \in C_2)$ } are extracted to be used for model building. Next, model building occurs. Again, as described in section 3.2.1.2, if the features are Normally distributed, the likelihood probabilities are generated by equation (1)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \text{ for } -\infty < x < \infty$$

where x = the variable for the current value of F_j , $\mu = \mu(F_j \in C_i)$ from the condition database, and $\sigma = \sigma(F_j \in C_i)$ from the condition database.

3.3.2 PHASE 2

In phase 2, the sensors, data processing, and feature extraction is performed using the same hardware and methods established in the offline state. The resultant of processing the current sensor data through data processing and feature extraction is the vector of current features \vec{d} . As was seen in the section 3.2.2, \vec{d} is a vector of features $\{F_1, \dots, F_m\}$ from a particular point in time. Indicated by the diamond shaped block in Figure 10, the vector of features are input to the Bayesian model defined by (50), to compute the probabilities of the machine tool being in each condition $\{P(C_1|\vec{d}), \dots, P(C_n|\vec{d})\}$. These conditional probabilities are known as posterior probabilities and are the nondeterministic output of the algorithm. Again, the posterior probabilities represent the probability the machine tool is in condition C_i given the data set \vec{d} .

3.3.3 MODEL UPDATING

While in the functional state of phase 2, the algorithm is designed to easily update the generated models as new data is collected. Updating the models is performed as follows. First, a new data set is collected and input into the algorithm when the state of the

machine is unknown. Using the outputted posterior probabilities from the algorithm, the operator must make the decision to stop the machine and perform an inspection or to wait until a more convenient time to perform inspection. When the inspection is performed and the condition associated with the data set is confirmed, the new set of feature values are added to the condition database. Now that there are more data in the condition database, the distribution characteristics for each feature μ and σ , are recalculated. This updating process is illustrated in Figure 10 by the thick curved arrow (Add Data) from the ‘inspection confirms condition’ block to the ‘Conditional Database’ block. This process of model updating can occur each time data is collected and its condition confirmed.

3.4 METRICS FOR RESULTS

The first metric for the method results are error. Error is calculated as:

$$\varepsilon = Y_e - Y_a \quad (57)$$

where ε is error, Y_e is the expected output, and Y_a is the actual output of the model. Two facts are needed to understand the expected output of the model. First, in all testing, the test data is sourced from the same population as the conditional database data. Second, in the experimental and simulated tests, there are only two conditions. To explain, it is known that the test data either belong to condition 1, ($\vec{d} \in C_1$), or belong to condition 2, ($\vec{d} \in C_2$); therefore, the theoretical posterior probabilities should be 1 and 0 or 0 and 1. For example, if the test data $\vec{d} \in C_1$ then theoretically $P(C_1|\vec{d}) \approx 1$ and $P(C_2|\vec{d}) \approx 0$. Similarly, if the test data $\vec{d} \in C_2$ then theoretically $P(C_1|\vec{d}) \approx 0$ and $P(C_2|\vec{d}) \approx 1$.

The second metric for the method results is accuracy. Accuracy is defined as the ratio of solutions where the test “if $\vec{d} \in C_1$ then $P(C_1|\vec{d}) > P(C_2|\vec{d})$ and if $\vec{d} \in C_2$ then $P(C_1|\vec{d}) < P(C_2|\vec{d})$ ” is not violated to the total number of trials.

3.5 EXPERIMENTAL IMPLEMENTATION: TOOL DIAGNOSTICS FOR A THREE INSERT TOOL

3.5.1 ASSUMPTIONS

1. Inspection is assumed to be perfect. Therefore, assume data used to build models do not contain incorrect values.
2. Observations from multiple sensors are random and conditionally dependent on the machine state.
3. Conditions, also known as the model classes, are mutually exclusive and collectively exhaustive.
4. Features are conditionally independent of each other

These assumptions are explained in section 3.1 and are the standard assumptions for the method presented in this thesis.

3.5.2 PURPOSE

The purpose of the experimental implementation is to demonstrate a simple real world implementation of the general process and show the results. In this methods section, the purpose is to present the process used to implement the Bayesian models. This includes describing feature selection, data processing, feature extraction, model building and model updating. The section is laid out as follows. First, the equipment, materials and

experimental procedure for the experimentation are provided. Then the methodology used to implement phase 1 and phase 2 are provided following the structure of the flow chart from Figure 10.

The goal of the experimental case is to detect the condition of an actively cutting three insert end milling tool using sensor fusion. As shown in Figure 12, the two specific conditions of interest are “Healthy” where all three tool inserts are unchipped and not worn, and “Damaged” where one of the three inserts is chipped and the other two are unchipped and not worn.



Figure 12: Milling tool and possible insert conditions

3.5.3 EQUIPMENT

Machine tool: Okuma Millac 44V

Tool: HERTEL Indexable Copy End Mill, 3 Insert, 1” Shank Diameter and Maximum Cutting Diameter, Inserts RP.10. Made of Steel

In addition to the manufacturing equipment, there were three types of sensors used: two accelerometers, one dynamometer, and one power meter.

Accelerometers:

PCB Piezoelectric Accelerometer, Model 353B15, High Frequency Quartz Shear ICP

Analog Devices, MEMS Accelerometer, Model ADXL203EB

Dynamometer: Kistler Multicomponent Dynamometer, Type 9257B SN 456863

Power meter: Load Controls Incorporated Universal Power Cell, Model UPC-FR

Data Acquisition:

Kistler Dual Model Amplifier, Type 5010 (X3)

Kistler Power Supply/Coupler Type 5134

National Instruments, CompacDAQ, type cDAQ-9178 with sensor based I/O modules type NI 9232

LabVIEW 2014 Student Edition Software

3.5.4 MATERIALS

Alloy A-286 Stainless, Hardened HRC 40.2, 6” Round Bar stock

3.5.5 EXPERIMENTAL PROCEDURE

The experiment had two independent variables, radial depth of cut and the condition of the tool. In all, there were three radial depths of cut and two conditions tested; thus six runs of experimentation were performed (Table 3).

Table 3: Layout of Experiments

Run #	Radial Depth of Cut	Tool Condition
1	3 mm	Healthy
2	3 mm	Damaged
3	4 mm	Healthy
4	4 mm	Damaged
5	5 mm	Healthy
6	5 mm	Damaged

For all experimentation, the axial depth of cut was 1mm with a speed of 3000 RPM. Milling was performed with flood coolant in a down milling direction. An illustration defining radial depth of cut and axial depth of cut is shown in Figure 13.

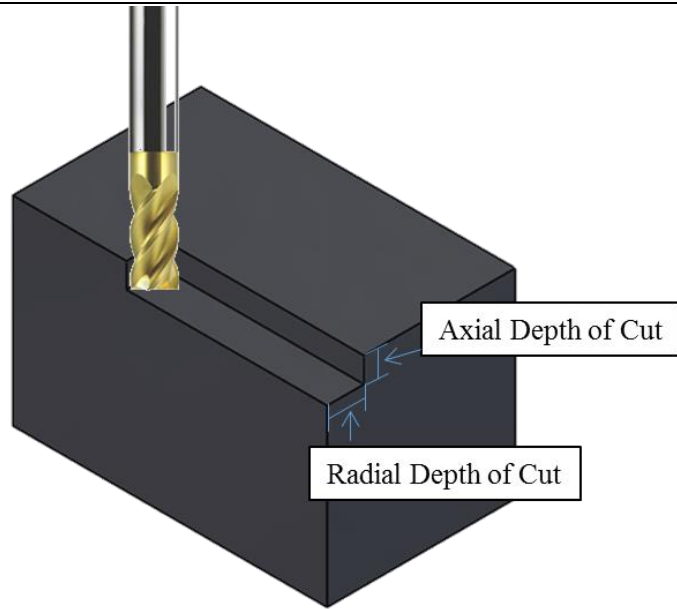


Figure 13: Illustration of Radial Depth of Cut vs. Axial Depth of Cut in End Milling

The test equipment was set up as shown in Figure 14. The MEMS and piezoelectric accelerometer were placed on the spindle housing away from the spindle motor. Both accelerometers were secured magnetically and with accelerometer wax. For accurate force measurements, the stock cutting material was bolted directly to the dynamometer. The dynamometer was secured to a plate that was gripped in the machine vice. The power meter was installed into the back panel of the machine as directed by manufacturer specifications. All sensors were sampled at a rate of 50 kHz with the National Instruments CompacDAQ DAQ system.

Experimental Test Equipment Setup

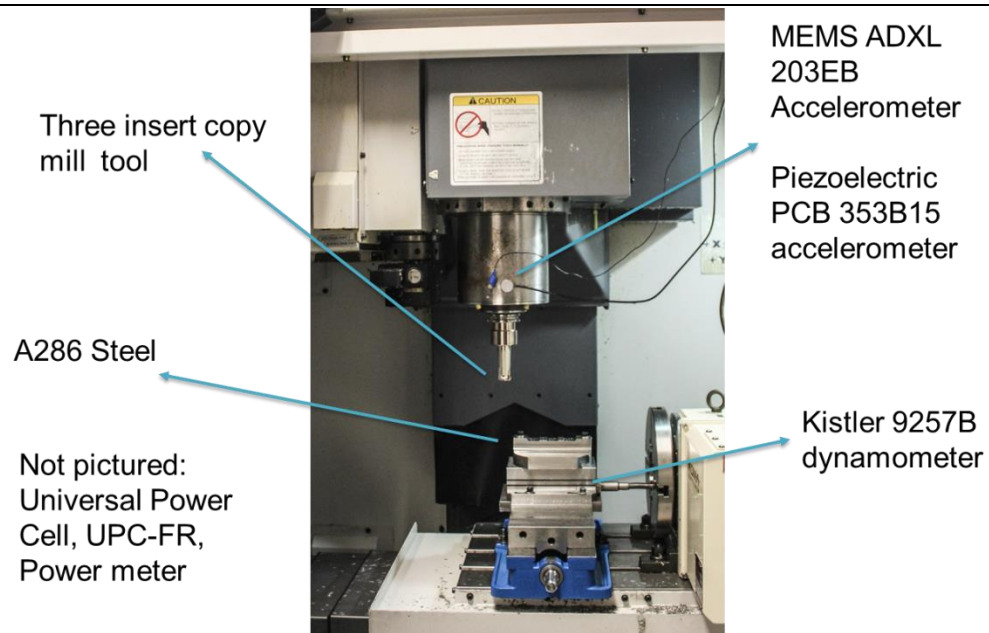


Figure 14: Experimental Setup

3.5.6 PHASE 1

The discussion in the following section ‘Sensors and Data Processing’ explains logically how the data features were found and then discusses how they were used to choose data processing techniques. The, the next section, ‘Feature Extraction’ explains the methods used for automated feature extraction for the features chosen in ‘Sensors and Data Processing.’

3.5.6.1 Sensors and Data Processing

Initial data were taken at all six runs (see Table 3) and studied for data features that correlated to the two conditions of interest, healthy tool and damaged tool. The following sections discuss the findings from each type of sensor.

3.5.6.1.1 MEMS & Piezoelectric Accelerometer

No data characteristics were found in time domain accelerometer data that correlated strongly with the two conditions of interest, healthy and damaged. On the other hand, when data was converted to the frequency domain by a fast Fourier transform (FFT), graphs of the FFT consistently unveiled two distinct types of scenarios; one correlated with the healthy condition and the other with the damaged condition. To illustrate, observe the FFT of accelerometer data for a healthy and damaged tool in Figure 15 and Figure 16. The first figure is for the MEMS accelerometer and the second figure is for the piezoelectric accelerometer. The plot at the top of each figure is from a 3mm radial depth of cut pass with a healthy tool. The plot at the bottom of the figure is from a 3 mm radial depth of cut pass with a damaged tool. The dependent axis is acceleration (mg) and the independent axis is the frequency in Hertz.

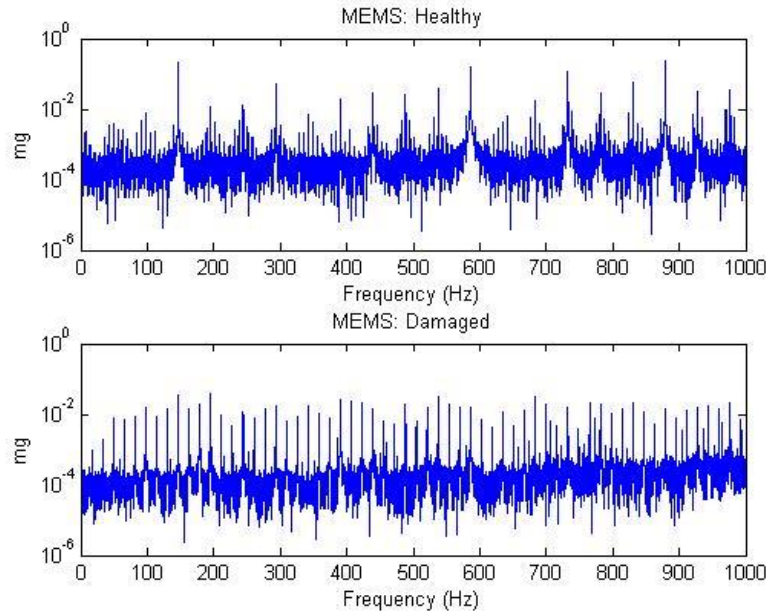


Figure 15: FFT of MEMS Accelerometer Data for a Healthy and Damaged Tool

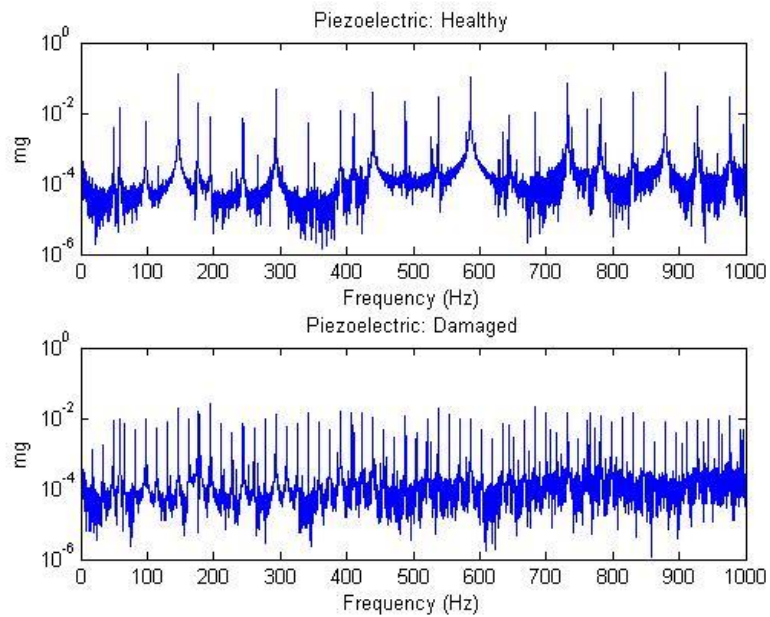


Figure 16: FFT of Piezoelectric Accelerometer Data for a Healthy and Damaged Tool

For the MEMS and piezoelectric accelerometer, the FFT of the healthy tool is observably different than the FFT of a damaged tool. For example, between 100-200 Hz and 500-600 Hz there is a noticeably taller and wider FFT peak in the healthy case than in the damaged case. This is true for both the MEMS and piezoelectric accelerometer. Also, in the damaged case there are noticeably more peaks above 10^{-2} mg acceleration than in the healthy case for both the MEMS and piezoelectric accelerometer.

Although promising and a good first step, visually observing unique patterns is not enough to be able to implement a statistical Bayesian model. Specific features that can be isolated and numerically quantified must be discovered. An educated approach to discover isolatable features with only small amounts of data is to exploit the physics of the machining operation. For example, since a rotating three insert milling tool was used in this experiment, it is expected that an FFT peak at the rotation frequency of the tool and an FFT peak at three times the rotation frequency (the insert frequency) should be observed from the vibration data. Using this knowledge the plotting methods for the FFT were altered to better reveal the expected features of interest, the rotation frequency and tool insert frequency. To demonstrate, data from Figure 15 and Figure 16 were replotted to create Figure 17 and Figure 18 respectively. Although from the same data, the independent axis was normalized by the rotation frequency of the cutting tool. Then the independent axis was limited to 10 times the frequency of the rotation of the cutting tool. The rotation of the cutting tool is also known as the shaft frequency. By normalizing the frequency response by the shaft frequency, the independent axis displays multiples of the shaft frequency. Because the expected features, rotation frequency and tool insert

frequency, are at the shaft frequency and three times the shaft frequency, they are observed at the first and third marker of the horizontal axis.

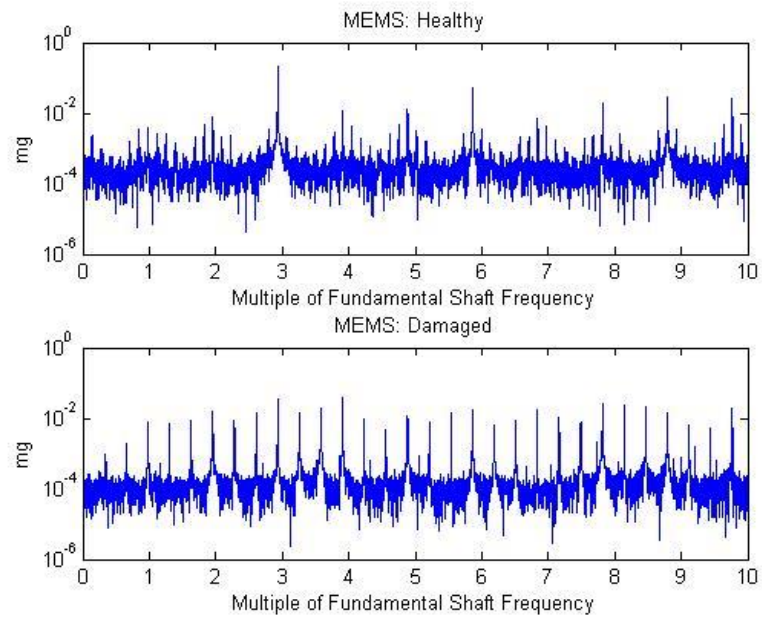


Figure 17: Normalized FFT of MEMS Accelerometer Data for a Healthy and Damaged Tool

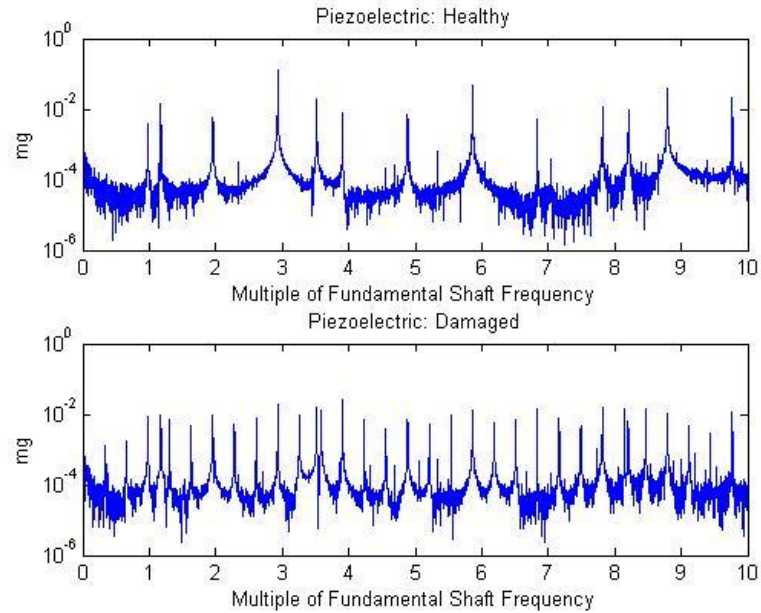


Figure 18: Normalized FFT of Piezoelectric Accelerometer Data for a Healthy and Damaged Tool

Observing these figures, it is seen that each of the multiples of the shaft frequency noticeably increases or decreases in magnitude between the healthy and damaged case. For example, when comparing the healthy case to the damaged case the first and second multiple clearly increase and the third multiple decreases. The causation for this is due to the physics of the frequency of impact of the tool with the workpiece. Because the causation for this response is understood and the magnitude and location of these three FFT peaks are easily quantifiable, these three features are chosen as extractable features from the accelerometer data for the Bayesian models.

Having found features from the accelerometer data to use in the models, the next step is to choose optimal data processing techniques for the raw accelerometer data. By the

nature of the features selected, choosing to perform the FFT and data processing methods to provide high quality FFT results were selected. Inspired from bearing health diagnostic techniques that use FFT data to identify bearing faults, time domain data was chosen to be windowed by 50% and the FFTs averaged before feature extraction was to be performed (Betta, G., et al., 2001)(McFadden et. al., 2000). This thesis does not go into the effects of windowing or FFT averaging, however, the methods used are explained. In this thesis, windowing is the process of subdividing data into overlapping sets prior to performing an FFT on each set. A window of 50% means that first 50% of the data in the current set is the last 50% of the data from the previous set. For example, looking at Figure 19, if data is subdivided into D width sections for FFT computation without windowing, there are three subsets of data, D_I , D_{II} , and D_{III} . However, if windowing of 50% is used, there are five subsets of data, $D_1 \dots D_5$.

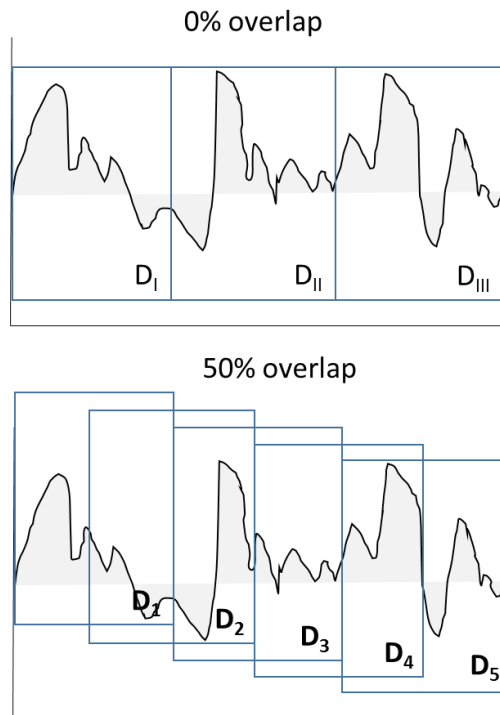


Figure 19: Illustration of Windowing by 50%

In this experimental case, data were separated into subsets of one second for FFT computation. An average of the FFT peaks was taken from every 10 subsets of data and there was 10.5 seconds worth of consistent cutting data used for analysis per pass.

3.5.6.1.2 Dynamometer

From understanding the accelerometer data, the initial dynamometer data was inspected in the frequency domain and similar characteristics found. To illustrate, observe Figure 20 and Figure 21 to view FFT plots of dynamometer data for a healthy and damaged tool respectively. In both figures, the topmost plot is the FFT of the X-axis dynamometer data, the middle plot is the FFT of the Y-axis dynamometer data, and the bottommost plot is the FFT of the Z-axis dynamometer data. The X-axis is along the feed direction, the Y-

axis is perpendicular to the feed direction, and the Z-axis in the direction of the tool. Again the FFTs are normalized by dividing by the shaft frequency. Note, the vertical axis of the plots is not uniform across all 6 plots.

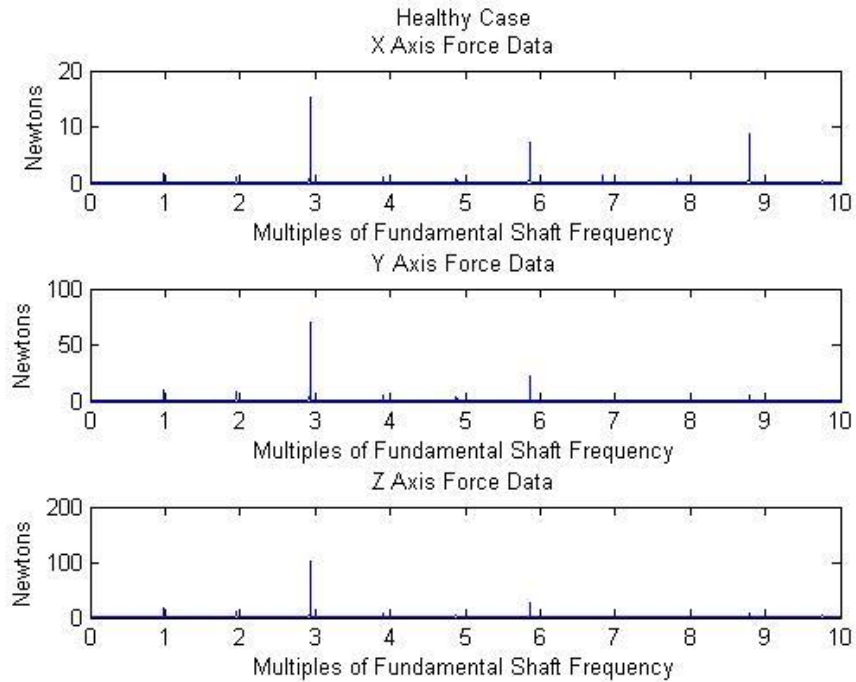


Figure 20: FFT of X, Y, & Z axis of 'Healthy' Dynamometer Data

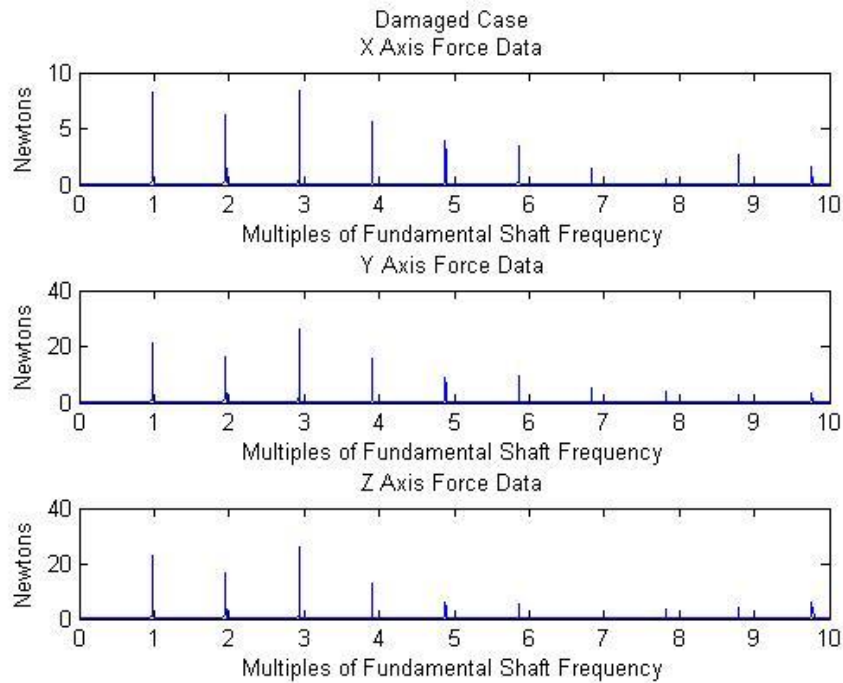


Figure 21: FFT of X, Y, & Z axis of ‘Damaged’ Dynamometer Data

As was perceived in the accelerometer case, it is seen that each of the multiples of the shaft frequency noticeably increases or decreases in magnitude between the healthy and damaged case. As was chosen in the Accelerometer case, the first three multiples of the shaft frequency for each axis, X Y & Z, were chosen as features.

Based on the similar choice of features, data processing methods selected for the accelerometer data was also used as data processing methods for the dynamometer data.

3.5.6.1.3 Power meter

After observing the initial data response from the time domain power meter data it was found that the magnitude of the power meter data was affected by the condition of the

tool and the radial depth of cut. The initial results are plotted in Figure 22 below. The average power meter data from when the tool was in a healthy condition is shown in black and the average power meter data from when the tool was in a damaged condition is shown in light grey. Every 5k of raw data was average to arrive at these plots. The error bars represent $\pm 1\sigma$. The top plot is from the 3mm radial depth of cut condition. The middle plot is from the 4mm radial depth of cut condition. Finally, the bottom plot is from the 5mm radial depth of cut condition.

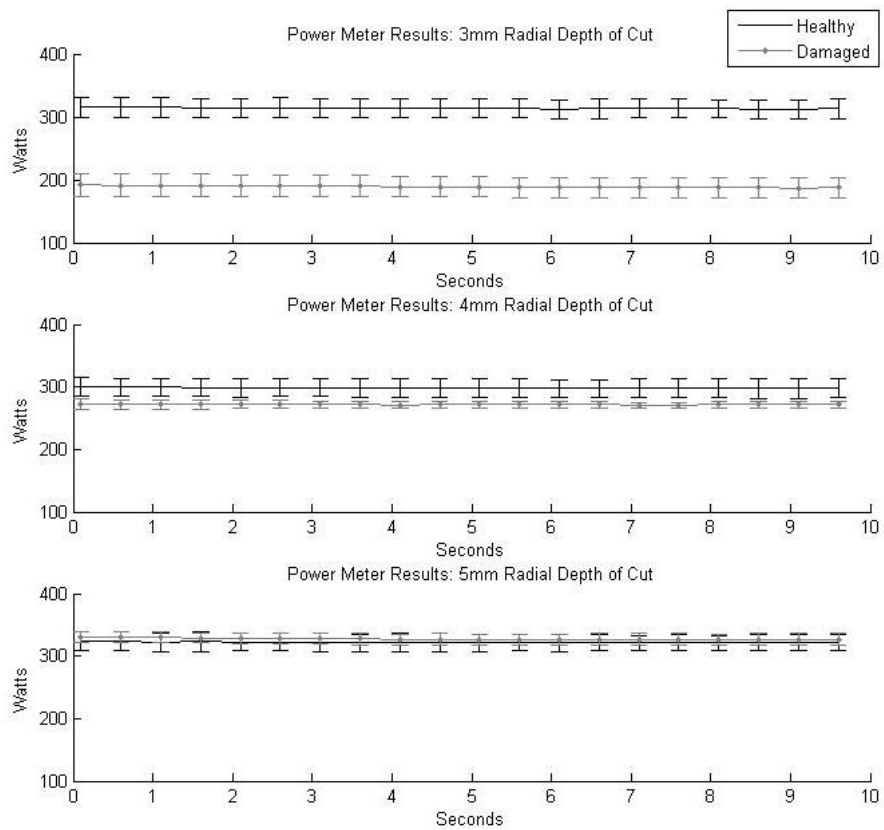


Figure 22: Initial Results of Power Meter Data for 3, 4, & 5 mm Radial Depth of Cut

In the 3mm radial depth of cut case, it is clear when the tool is healthy and when it is damaged as each group is widely separated from each other. In the 4mm radial depth of cut case, it begins to become unclear around 280-300 Watts. In the 5mm radial depth of cut case, the healthy case and damaged case are nearly indistinguishable. Since mean of the power data can be useful in determining the state of the tool, it is selected as a feature.

To statistically compare the distributions of the healthy and damaged power meter data, Figure 23 is helpful.

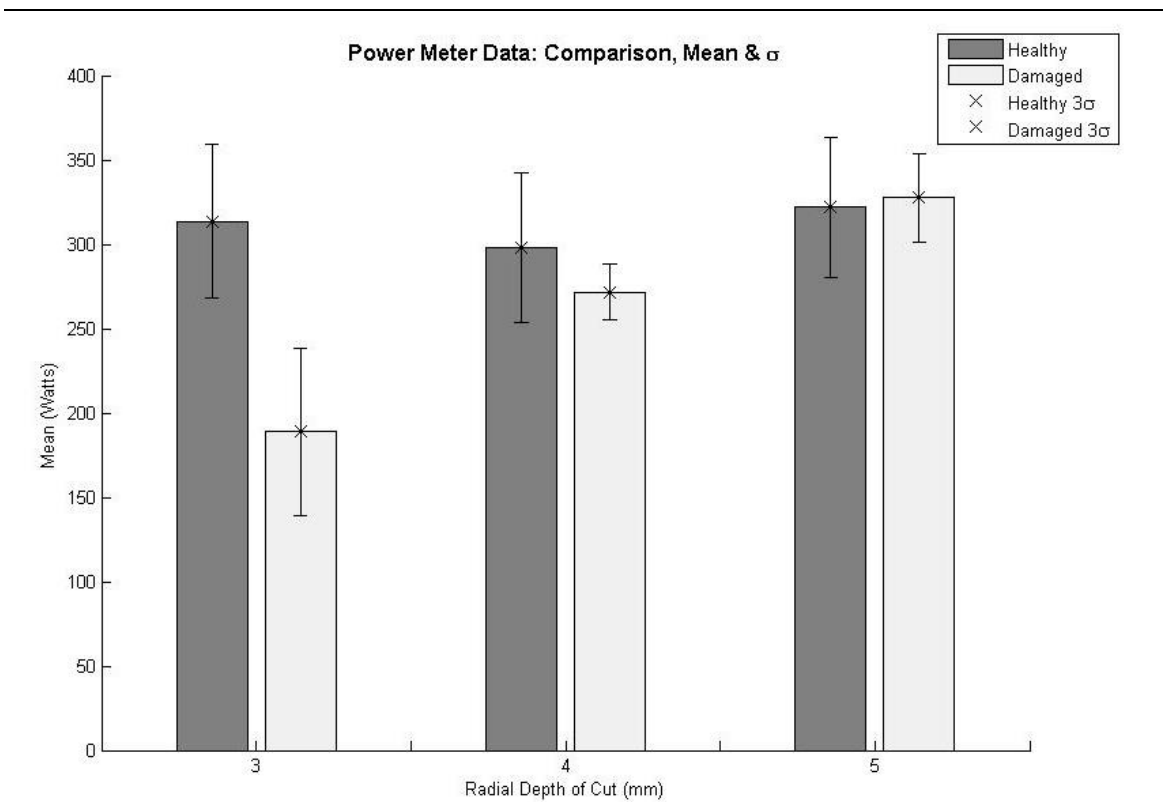


Figure 23: Distributions of Power Data from the Healthy and Damaged Condition

The dark grey bar represents the mean of the healthy power meter data. The light grey bar represents the mean of the damaged power meter data. Lastly, the error bars are the $\pm 3\sigma$ limits for each bar respectively. Each set of bars on the bar chart represent the 3mm, 4mm, and 5mm radial depth of cut case respectively. Again, it is seen that the distributions for the 3mm radial depth of cut case are quite separate and that the 4mm and 5mm radial depth of cut distributions have significant overlap. On the other hand, this figure makes it clear that the magnitude of the standard deviation itself is a unique identifier for the 4mm and 5mm radial depth of cut cases. With further analysis, it was found that the spread of the standard deviation for each set of power data was quite small. As a result, the standard deviation of the power meter data has a high probability of being a reliable feature. Thus, the final features are chosen, the standard deviation of the power data and the mean of the power data.

Given the simple nature of the features chosen from the power meter data, the data processing technique was to take the mean and standard deviation of the raw data over 10 seconds of continuously engaged cutting.

All data processing was written into MATLAB functions in order to be able to automate the process in phase 2.

3.5.6.2 Feature Extraction

While choosing sensors and selecting data processing methods in phase 1, individual data features that change in response to the two conditions of interest, healthy tool and damaged tool, were found. A vector of these features becomes the structure of the input, \vec{d} , for the Bayesian model in phase 2 (see Figure 10, block “Compute $P(C|\vec{d})$ ”).

The vector is organized in Table 4. The left column is the final vector of extracted features. The right column is the description of the feature. MSF stands for “multiple of shaft frequency.”

Table 4: Final Vector of Extracted Features for Experimental Implementation \vec{d}

Feature	Feature Description
A_{m1}	Magnitude of 1 st MSF from normalized FFT of MEMS accelerometer
A_{m2}	Magnitude of 2 nd MSF from normalized FFT of MEMS accelerometer
A_{m3}	Magnitude of 3 rd MSF from normalized FFT of MEMS accelerometer
A_{p1}	Magnitude of 1 st MSF from normalized FFT of piezo accelerometer
A_{p2}	Magnitude of 2 nd MSF from normalized FFT of piezo accelerometer
A_{p3}	Magnitude of 3 rd MSF from normalized FFT of piezo accelerometer
D_{nx1}	Magnitude of 1 st MSF from normalized FFT of x-axis of dynamometer
D_{nx2}	Magnitude of 2 nd MSF from normalized FFT of x-axis of dynamometer
D_{nx3}	Magnitude of 3 rd MSF from normalized FFT of x-axis of dynamometer
D_{ny1}	Magnitude of 1 st MSF from normalized FFT of y-axis of dynamometer
D_{ny2}	Magnitude of 2 nd MSF from normalized FFT of y-axis of dynamometer
D_{ny3}	Magnitude of 3 rd MSF from normalized FFT of y-axis of dynamometer
D_{nz1}	Magnitude of 1 st MSF from normalized FFT of z-axis of dynamometer
D_{nz2}	Magnitude of 2 nd MSF from normalized FFT of z-axis of dynamometer
D_{nz3}	Magnitude of 3 rd MSF from normalized FFT of z-axis of dynamometer
P_{om}	Mean of 10 seconds of power data
P_{os}	Standard deviation of 10 seconds of power data

To ensure consistency and allow for automation of the Bayesian model formulation, methods for extracting the features from the processed data must be defined. In this experiment, there are three main types of features. The first 15 features of \vec{d} are magnitudes of frequency responses from normalized FFTs of specific sensor data. The second to last feature of \vec{d} is a mean of specific sensor data and the last feature is the standard deviation of that specific sensor data. As such, there are three methods for feature extraction defined: one for FFT peak extraction, one for data mean extraction, and one for data standard deviation extraction.

The methods for FFT peak extraction utilized the MATLAB function 'PeakFinder.m' and is as follows. The inputs are the spindle RPM, the FFT resolution and the averaged normalized FFT. The outputs are the magnitudes of the three peaks, 1 MSF 2 MSF and 3 MSF, from the input FFT. To extract the peak magnitudes, the code utilizes MATLABs built in 'max.m' function to find the maximum peak in a small region around the frequency bin of interest. For the 1 MSF peak, the search region is covered by $1 \text{ MSF} \pm 2 \text{ Hz}$. For the 2 MSF and 3 MSF, the search region is covered by $2 \text{ MSF} \pm 5 \text{ Hz}$ and $3 \text{ MSF} \pm 5 \text{ Hz}$ respectively. The MATLAB code for the function 'PeakFinder.m' is included in Appendix B.

The method for mean extraction from sensor data was simply using MATLAB's built in 'mean.m' function over 10 seconds of cutting data. This simply computed the mean of the data.

The method for standard deviation extraction from sensor data was simply using MATLAB's built-in function 'std.m' over 10 seconds of cutting data. This simply computed the population standard deviation of the data.

3.5.6.3 Building the Condition Database

The condition data base can be built as a series of matrices in MATLAB or as multiple files in Excel. Both types of condition databases were tested and used. In MATLAB, there was one matrix per run (see Table 3: Layout of Experiments). Each column of the matrix represented a specific feature. The rows recorded the history of feature values.

In Excel, there was one file per case and one page per condition, as shown by the "Damaged Data" and "Healthy Data" pages in the RDOC_3mm Excel file shown in Figure 24.

	A	B	C	D
1	Data Names	<i>Accel_mems3_1</i>	<i>Accel_mems3_2</i>	<i>Accel_mems3_3</i>
2	Mean	228.285	378.251	1521.338
3	Standard Deviation	80.569	82.001	277.777
7	Data Values 1	430.699	560.829	1114.149
8	2	399.334	572.365	995.455
9	3	148.537	279.527	1328.910
10	4	149.912	342.919	1170.945
11	5	202.294	291.248	1589.606
12	6	186.551	322.780	1376.140

Figure 24: Condition Database in Excel

As in the MATLAB matrix, the columns of the Excel page represent the extracted features, and the rows store the recorded history of feature values. In the condition database in Excel, the mean and standard deviation are computed at the top of the condition page.

3.5.6.4 Model building

For later comparison two models are built: one with all 17 features, and one with only the five features that come from the MEMS accelerometer data and the power meter data. This is repeated for each case, 3mm 4mm and 5mm radial depth of cut. From the mathematical representation of a general Bayesian model for n conditions and m features equation (50) the condensed Bayesian model is:

$$P(H|\vec{d}) = \frac{P(\vec{d}|H)P(H)}{P(\vec{d}|H)P(H) + P(\vec{d}|D)P(D)} \quad (58)$$

$$P(D|\vec{d}) = \frac{P(\vec{d}|D)P(D)}{P(\vec{d}|H)P(H) + P(\vec{d}|D)P(D)} \quad (59)$$

where the multivariate likelihood probabilities $P(\vec{d}|H)$ and $P(\vec{d}|D)$ for the 17 feature model are

$$\begin{aligned}
& P(\vec{d}|H) \\
&= P(A_{m1}|H)P(A_{m2}|H)P(A_{m3}|H)P(A_{p1}|H)P(A_{p2}|H)P(A_{p3}|H)P(D_{nx1}|H)P(D_{nx2}|H)P(D_{nx3}|H) \\
&\quad P(D_{ny1}|H)P(D_{ny2}|H)P(D_{ny3}|H)P(D_{nz1}|H)P(D_{nz2}|H)P(D_{nz3}|H)P(P_{om}|H)P(P_{os}|H)
\end{aligned} \tag{60}$$

$$\begin{aligned}
& P(\vec{d}|D) \\
&= P(A_{m1}|D)P(A_{m2}|D)P(A_{m3}|H)P(A_{p1}|D)P(A_{p2}|D)P(A_{p3}|D)P(D_{nx1}|D)P(D_{nx2}|D)P(D_{nx3}|D) \\
&\quad P(D_{ny1}|D)P(D_{ny2}|D)P(D_{ny3}|D)P(D_{nz1}|D)P(D_{nz2}|D)P(D_{nz3}|D)P(P_{om}|D)P(P_{os}|D)
\end{aligned} \tag{61}$$

and the multivariate likelihood probabilities $P(\vec{d}|H)$ and $P(\vec{d}|D)$ for the five feature model are

$$P(\vec{d}|H) = P(A_{m1}|H)P(A_{m2}|H)P(A_{m3}|H)P(P_{om}|H)P(P_{os}|H) \tag{62}$$

$$P(\vec{d}|D) = P(A_{m1}|D)P(A_{m2}|D)P(A_{m3}|H)P(P_{om}|D)P(P_{os}|D) \tag{63}$$

Prior to using the models, the assumption of condition independence should be inspected. A useful tool to understand dependence and independence is to draw the Bayesian network. Graphically illustrated, the Bayesian network for the 17 variable models is shown in Figure 25:

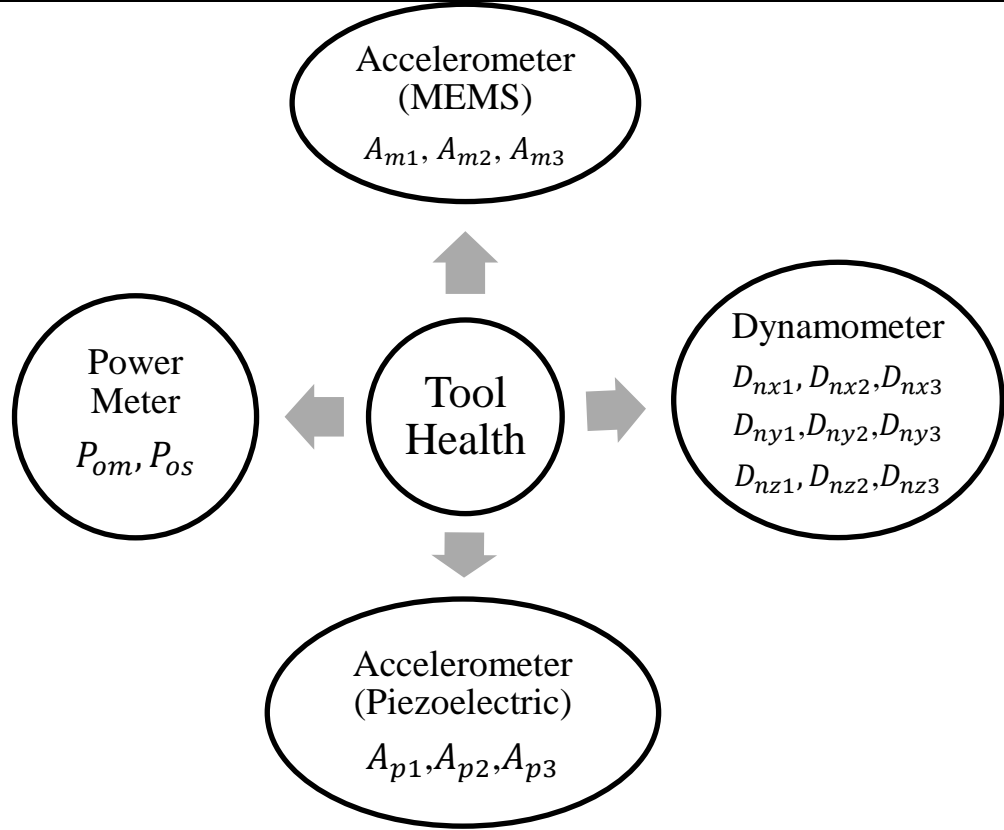


Figure 25: Naïve Bayesian Network for Tool Health Model

In the figure, the sensor data are all dependent on the state of the tool and independent from each other. The presence or absence of data from one sensor does not affect the results of the others. In other words, although the data features are all correlated, causation is due to tool health and not influenced by the results of the other sensors. Mathematically this is expressed as:

$$P(A_m | D, A_p, D_n, P_o) = P(A_m | D) \quad (64)$$

and so forth for each sensor. Although it can be argued whether or not the features extracted from the sensor data are entirely conditionally independent of each other, the

use of naïve Bayesian models reasonably holds for these features (see section 2.6.1 for further explanation).

The models are generated using a MATLAB function, called BuildBayes, laid out below.

In this experimental case, the inputs are:

Input	Meaning
CDatabase_H	A matrix from the conditional data base structure representing all of the values in the conditional database for condition healthy. The rows represent individual data points and the columns represent individual features. Ex. rdoc5.accelP.h.peaks
CDatabase_D	The same set up as CDatabase_C1 except for condition damaged. Ex. rdoc5.accelP.d.peaks
testData	A vector of data representing the input \vec{d} . Each column is a feature F_i .
prior	A vector representing the prior probabilities for the conditions.

The output of the model is a vector of posterior probabilities called PosteriorP. Column 1 of the output is $P(H|\vec{d})$ and column 2 is $P(D|\vec{d})$. In this implementation, the input testData is a vector \vec{d} . This code is specifically set up for only two conditions but technically can be used for an unlimited number of features. MATLAB code for the experimental BuildBayes function is shown in Figure 70 in Appendix B.

This function, although specific to this experimental case, is revised into a more general code for Bayesian implementation of sensor fusion in section 3.6.

3.5.7 PHASE 2

3.5.7.1 *Sensors, Data Processing, & Feature Extraction*

In phase 1, data processing and feature extraction methods were defined. In this experimental case, code to perform the data processing and feature extraction was also created in phase 1. As such, packets of raw sensor data are able to be inserted into said functions to arrive at a vector of features, \vec{d} . This vector is then inserted into the Bayesian model and the posterior probabilities are generated. In this case, this is done by inputting the current \vec{d} into the code BuildBayes.m which was also generated in phase 1.

3.5.7.2 *Model updating*

When a new set \vec{d} is ready to be added to the condition database, the vector is input as an additional row to the appropriate MATLAB condition database matrix or Excel condition database page. When a dataset is added to a condition's database in Excel, the mean and standard deviation for each feature auto updates thus updating the likelihood probabilities. When a dataset is added to a condition's database in MATLAB, the means and standard deviations are updated by the BuildBayes.m function prior to calculating the likelihood probabilities while running the model.

3.6 SIMULATED IMPLEMENTATION

3.6.1 PURPOSE

A simulated approach is taken to illustrate the classical limitations during model building and model use. In the simulated testing, the effects of three aspects of the models are explored: the quantity of data used to build the model, the amount of distribution overlap

for the different conditions, and the quantity of variables (features) used to build the model. The purpose of the simulated implementation methods section is to explain how the testing was performed and how the components of the tests were created. The organization is as follows. First the inputs for the Bayesian model and the simulated model set up are discussed. Next, the computational tools used to generate the components of the models and the models themselves are presented. Then, the procedure for generating and testing the effect of overlapping distributions on multivariate Bayesian models is explained. Finally, the procedure for testing the effect of model updating is discussed.

3.6.2 INPUTS & SIMULATED MODEL SETUP

The Bayesian Model for the simulated experimentation has two conditions, C_1 and C_2 , and three features F_1 , F_2 , and F_3 . Using the mathematical representation for a general Bayesian model for n conditions and m features (50), the Bayesian Models are

$$P(C_1|\vec{d}) = \frac{P(C_1)P(F_1|C_1)P(F_2|C_1)P(F_3|C_1)}{P(C_1)P(F_1|C_1)P(F_2|C_1)P(F_3|C_1) + P(C_2)P(F_1|C_2)P(F_2|C_2)P(F_3|C_2)} \quad (65)$$

$$P(C_2|\vec{d}) = \frac{P(C_2)P(F_1|C_2)P(F_2|C_2)P(F_3|C_2)}{P(C_1)P(F_1|C_1)P(F_2|C_1)P(F_3|C_1) + P(C_2)P(F_1|C_2)P(F_2|C_2)P(F_3|C_2)} \quad (66)$$

The models are generated using a MATLAB function, called BuildBayes, which is now described. This is a generalized version of the BuildBayes code shown on page 117. The inputs are:

Input	Meaning
CDatabase_C1	A matrix representing all of the values in the conditional database for condition C1. The rows represent individual data points and the columns represent individual features.
CDatabase_C2	The same set up as CDatabase_C1 except for condition C2
testData	testData is a matrix and each row represents an test vector \vec{d} . Each column in vector \vec{d} is a feature F_i .
prior	A vector representing the prior probabilities for the conditions.

The output of the model is a vector of posterior probabilities called PosteriorP. Each column of PosteriorP represents the posterior probability for a specific condition. For example, since there are two conditions, column 1 is $P(C_1|\vec{d})$ and column 2 is $P(C_2|\vec{d})$. In this implementation, the input testData is a matrix of input data vectors where each row represents an input vector \vec{d} . For each row of testData there is an associated output row of posterior probabilities in PosteriorP. This code is specifically set up for only two conditions but unlimited number of features. The MATLAB code is located in Figure 72 in Appendix B.

3.6.3 COMPUTATIONAL TOOLS

The simulated experimentation was done completely in MATLAB. To test the models, Normally distributed datasets were randomly generated using MATLAB's "normrnd.m" function. For each distribution, three individual sets of 1020 points were generated and stored as a matrix. Each of the three sets represented an individual feature. As shown in

Figure 26, Quantile Quantile regression plots were used to guarantee that simulated data were Normally distributed.

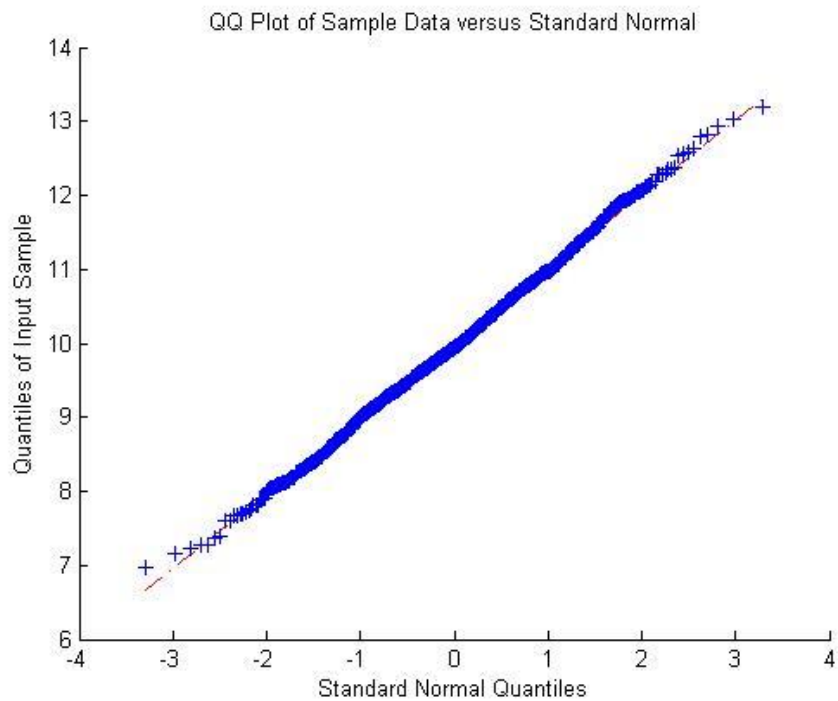


Figure 26: Quantile-Quantile regression plot of simulated data

MATLAB is also used to perform kfold cross validation during testing. Cross validation is a way to measure the predictive performance of a statistical model when an explicit validation set does not exist. In other words, when the data used to test a model are sourced from the same population as the training data used to build the model, cross validation is used to ensure that results reflect the performance of the model independent of the subset of data chosen for testing. Kfold cross validation is a cross validation

technique that allows every point in a population to be tested and used for training. For kfold cross validation, the original data set is partitioned into k equal sized subsets. From these subsets, one is chosen to be used for testing and the remaining subsets are used as training data. This is repeated k times until each subset has been used as testing data. In this thesis, the original data set is the data in the conditional database. As noted above Figure 26, each condition has 1020 data instances per feature in the conditional database. With this, a k value of 51 was chosen for folds of 20 points each.

Kfold cross validation was performed while also testing for the effect of the number of points in the conditional database. This was then repeated for each set of distribution overlaps. The code script used is in Figure 71 in appendix B.

3.6.4 SIMULATED PROCEDURE: OVERLAPPING DISTRIBUTIONS

Ten distribution sets are used to test for the effect of overlapping distributions for multivariate Bayesian models. The distribution overlap is characterized by the distribution range overlap and the distribution area overlap. The range overlap is calculated by the following equation:

$$Range\ Overlap = \frac{(\mu_2 - 3\sigma_2) - (\mu_1 + 3\sigma_1)}{(6\sigma_1)} \times 10 \quad \mu_1 < \mu_2, \quad \sigma_1 = \sigma_2 \quad (67)$$

where μ_1 and σ_1 represent the mean and standard deviation of condition 1 and μ_2 and σ_2 represent the mean and standard deviation of condition 2. The distribution area overlap is also known as the overlapping coefficient (OVL). In cases where $\sigma_1 = \sigma_2$, distribution area overlap is calculated using

$$OVL = 2\Phi\left(-\frac{|\delta|}{2}\right) \quad (68)$$

where Φ is the cumulative distribution function of the standard Normal distribution and δ is the population Cohen's d (69).

$$\delta = \frac{\mu_1 - \mu_2}{\sigma} \quad (69)$$

This calculation only holds if there are homogenous population variances ($\sigma_1 = \sigma_2 = \sigma$)

Table 5 below provides the distribution characteristics for each set of distributions and Figure 27, Figure 28, Figure 29, and Figure 30, illustrate four of the distribution overlaps. For illustrations of the distribution overlaps not illustrated in this section, please go to Appendix A.

Table 5: Parameters for test of overlap effect

Condition 1		Condition 2		Distribution Range Overlap	Distribution Area Overlap
μ_1	σ_1	μ_2	σ_2		
10	1	16	1	0.00%	0.00%
10	1	15.5	1	8.33%	0.60%
10	1	15	1	16.67%	1.24%
10	1	14.5	1	25.00%	2.44%
10	1	14	1	33.33%	4.55%
10	1	13.5	1	41.67%	8.01%
10	1	13	1	50.00%	13.36%
10	1	12.5	1	58.33%	21.13%
10	1	12	1	66.67%	31.73%
10	1	11.5	1	75.00%	45.33%

In Table 5, the mean μ and standard deviation σ are listed for each distribution. Condition 1 is always represented by a Normal distribution with a mean of ten and standard deviation of one. In the following figures, condition 1 is always on the left side of the figure and condition 2 is always on the right side of the figure. Also, condition 2 is always outlined with asterisks. The overlapping region of the two distributions is noted by the dark triangular shaped region between the two distributions in the figures.

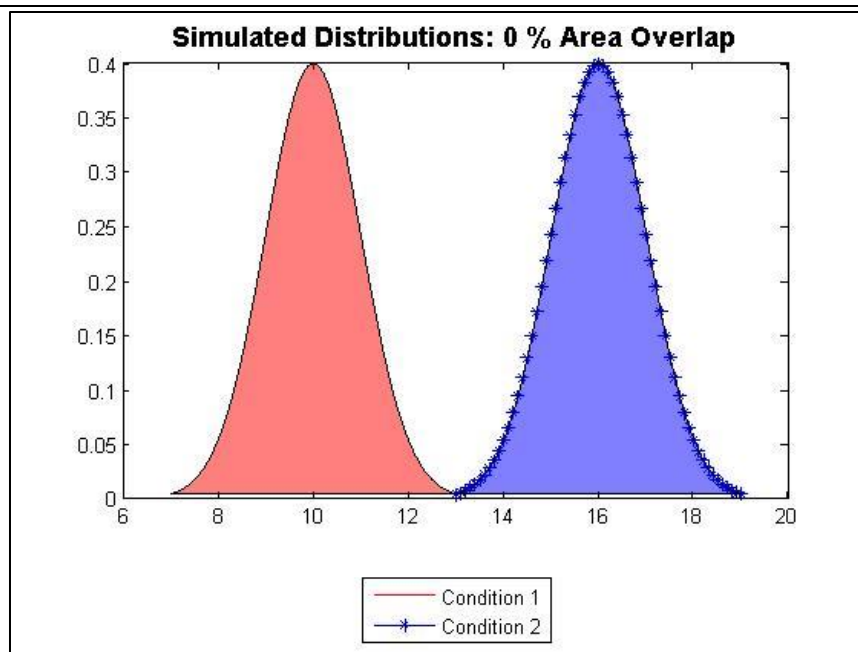


Figure 27: Simulated Distributions, 0% Area Overlap

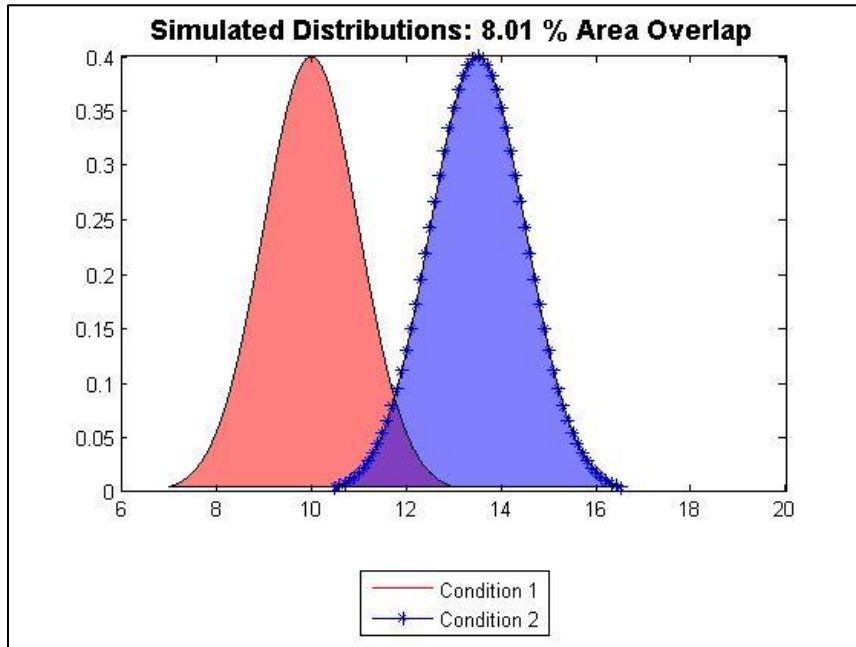


Figure 28: Simulated Distributions, 8.01% Area Overlap

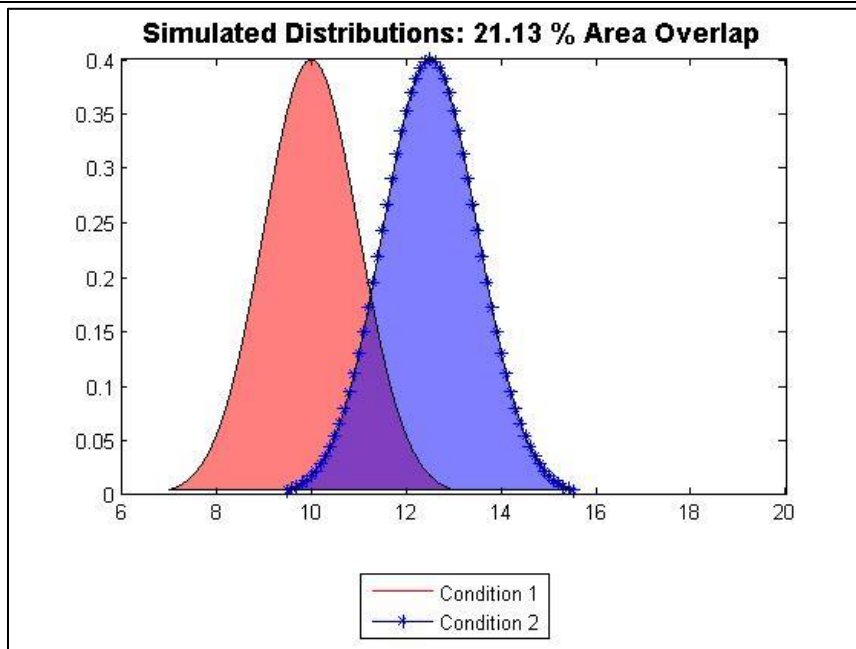


Figure 29: Simulated Distributions, 21.13% Area Overlap

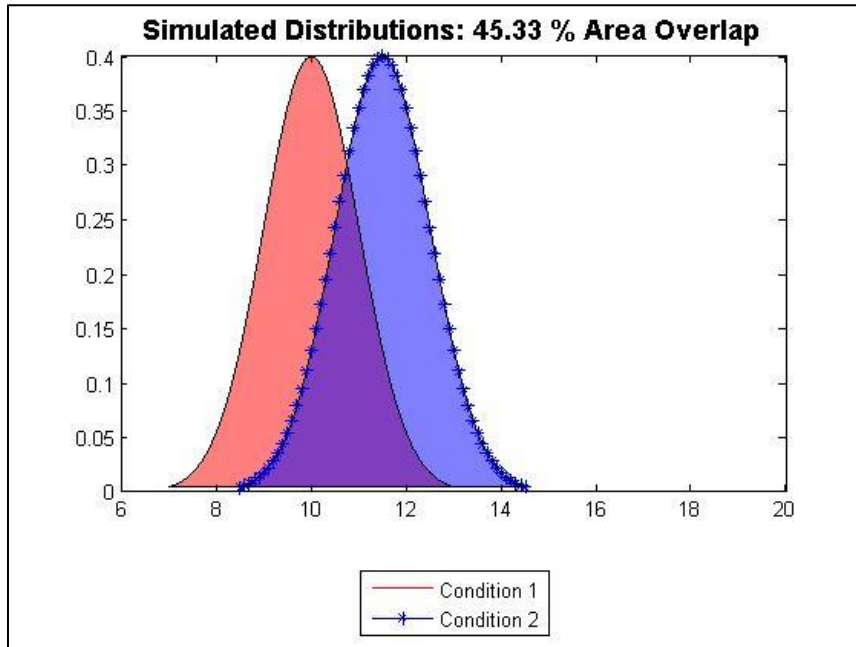


Figure 30: Simulated Distributions, 45.33% Area Overlap

For testing the effect of overlapping distributions, the conditional database had 1000 sets of data for each condition and 20 sets of data used for testing. The code used for these tests are shown in Figure 72 and Figure 71 in Appendix B. This was repeated for each pair of distributions.

3.6.5 SIMULATED PROCEDURE: MODEL UPDATING

The ten distribution sets from the “Simulated Procedure: Overlapping Distributions” section are used again in this model updating section. For each distribution set, the Bayesian model is tested with various amounts of data in the conditional database. Using code shown in Figure 71 for cross validation, the Bayesian model is tested from 10 points to 1000 points (incremented by 10) for each feature in each condition of the condition database.

4 RESULTS AND DISCUSSION

4.1 EXPERIMENTAL IMPLEMENTATION

As mentioned in the Methods section, two models were tested in the experimental trial case: a 17 variable model which contained all variables noted in Table 4, and a 5 variable model with only features from the MEMS accelerometer and the Power Meter. This was repeated for three cases; once for 3mm radial depth of cut passes, once for 4mm radial depth of cut passes, and once for 5mm radial depth of cut passes. The summary of the results are presented in Table 6. Each of the columns of the table represents the average posterior probability that was output from the algorithm. The expectations are if $\vec{d} \in \text{Healthy}$ then the expected posterior probabilities are $P(\text{Healthy}|\vec{d}) \approx 1$ and $P(\text{Damaged}|\vec{d}) \approx 0$ else if $\vec{d} \in \text{Damaged}$ then the expected posterior probabilities are $P(\text{Healthy}|\vec{d}) \approx 0$ and $P(\text{Damaged}|\vec{d}) \approx 1$.

Table 6: Results for 3mm, 4mm, & 5mm Radial Depth of Cut Case with 10 instances of Data in Conditional Database for each condition

Condition of Tool	17 Feature Model		5 Feature Model	
	Average $P(\text{Healthy} \vec{d})$	Average $P(\text{Damaged} \vec{d})$	Average $P(\text{Healthy} \vec{d})$	Average $P(\text{Damaged} \vec{d})$
$\vec{d} \in \text{Healthy}$	1	0	1	0
$\vec{d} \in \text{Damaged}$	0	1	0	1

The results strongly match the expectations for the 17 feature model and for the 5 feature model. Looking directly at the error of the individual results, Table 7 illustrates how small the error is for both experimental models. If the value is less than 4.94×10^{-324} then MATLAB rounds to zero.

Table 7: Results for 3mm Radial Depth of Cut Case with 10 instances of Data in Conditional Database for each condition

Calculated Error						
Tool Condition	17 Feature Model			5 Feature Model		
	3 mm RDOC	4 mm RDOC	5 mm RDOC	3 mm RDOC	4 mm RDOC	5 mm RDOC
$\bar{d} \in \text{Healthy}$	0	0	4.47E-167	0	9.00E-34	3.02E-30
	0	0	7.81E-161	0	5.42E-32	6.66E-30
	0	3.18E-205	2.17E-112	3.43E-136	2.41E-17	7.46E-21
	0	8.37E-179	5.10E-112	9.95E-132	8.76E-18	6.39E-20
	0	9.38E-171	2.21E-99	1.06E-137	9.96E-19	9.04E-17
	0	2.82E-141	1.41E-100	1.01E-132	4.75E-13	5.04E-16
	0	5.30E-183	5.45E-96	5.11E-146	2.16E-20	8.82E-15
	0	1.01E-142	9.36E-90	5.51E-142	1.19E-17	6.65E-13
	0	4.22E-195	2.61E-103	1.10E-173	4.20E-27	7.27E-15
	0	1.10E-151	1.43E-96	1.56E-170	4.82E-20	2.32E-12
	0	2.13E-240	1.93E-126	9.61E-214	4.53E-36	1.08E-18
	0	1.99E-170	1.96E-118	3.45E-212	1.66E-27	8.02E-16
$\bar{d} \in \text{Damaged}$	0	0	0	0	2.88E-26	1.49E-93
	0	0	0	0	6.59E-53	6.72E-119
	0	0	0	3.36E-250	1.89E-62	2.51E-54
	0	0	0	5.95E-280	6.20E-51	1.33E-68
	0	0	0	6.45E-242	5.89E-90	1.39E-66
	0	0	0	1.45E-264	9.66E-84	5.38E-82
	0	0	0	1.23E-261	1.31E-81	9.91E-76
	0	0	0	2.03E-288	2.32E-68	2.28E-95
	0	0	0	2.10E-251	1.32E-104	2.09E-122
	0	0	0	1.10E-274	4.22E-108	1.87E-147
	0	0	0	3.64E-281	3.41E-96	5.88E-192
	0	0	0	2.16E-312	1.28E-89	2.30E-226

In Table 7, the rows represent individual input samples of data. The condition of the tool, is indicated on the left. From both Table 6 and Table 7, it is seen that the expectations match the results in 100% of the experimental trials that were performed. In other words, if $\vec{d} \in \text{Healthy}$ then the posterior probabilities are $P(\text{Healthy}|\vec{d}) \approx 1$ and $P(\text{Damaged}|\vec{d}) \approx 0$. Likewise, if $\vec{d} \in \text{Damaged}$ then the posterior probabilities are $P(\text{Healthy}|\vec{d}) \approx 0$ and $P(\text{Damaged}|\vec{d}) \approx 1$. Given that there were features in both models with very separate distributions, these results make sense. This aspect is further discussed in the next section.

4.1.1 EFFECT OF OVERLAPPING DISTRIBUTIONS IN EXPERIMENTAL CASE

Three variables are selected to illustrate the effect of overlapping distributions in the experimental trial case. The variables are D_{ny1} , D_{ny2} , and D_{ny3} which are the magnitude of three peaks from the FFT of the Y axis force data from the dynamometer (see Table 4). Figure 31 illustrates the distributions for each peak and for each condition. The dark grey bars represent the healthy distributions, and the light grey bars represent the damaged distributions. The magnitude of the bars in the bar graph represents the mean value of the distribution in Newtons. The error bars represent the $\pm 3 \sigma$ range for the distributions. On the independent axis, 1, 2, and 3 represent the 1st, 2nd, and 3rd multiples of the shaft frequency (which are D_{ny1} , D_{ny2} , and D_{ny3}).

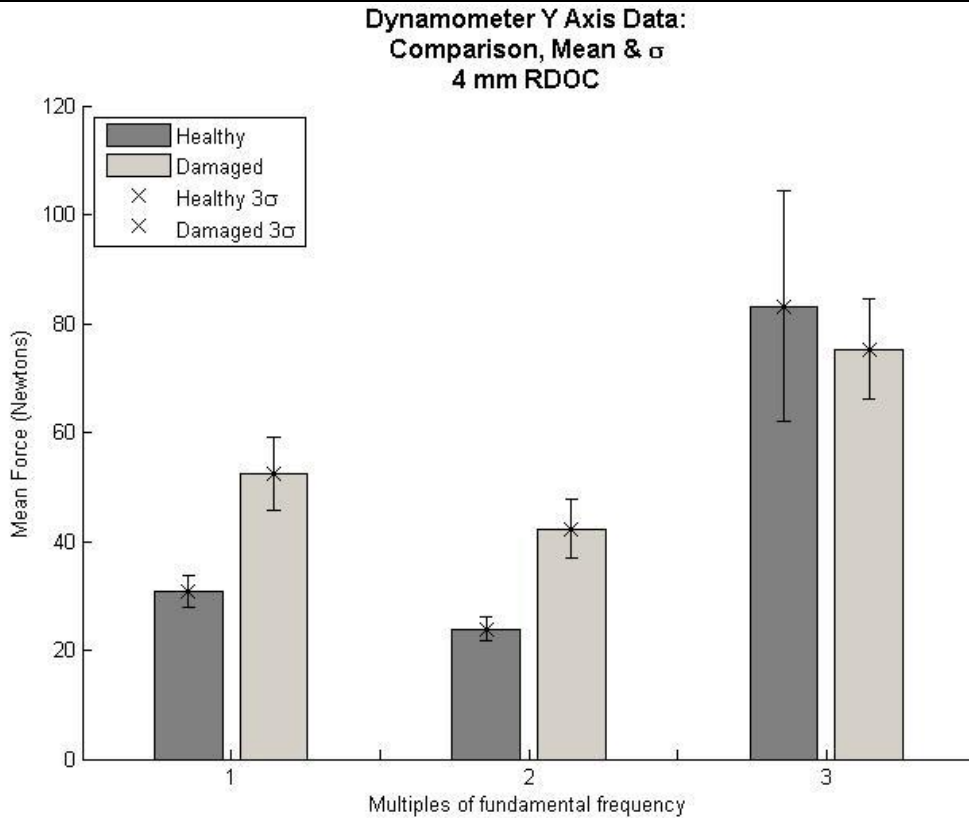


Figure 31: Distributions for Dynamometer in the Y axis

From Figure 31, it can be seen that the 3rd multiple of the fundamental frequency has a large overlap in distributions for the healthy and damaged case. More clearly illustrated, Figure 32 shows how the $\pm 3\sigma$ range of the damaged distribution is entirely within the $\pm 3\sigma$ range of the healthy distribution.

**Experimental Distributions:
4 mm RDOC
Dynamometer Y Axis 3rd Multiple**

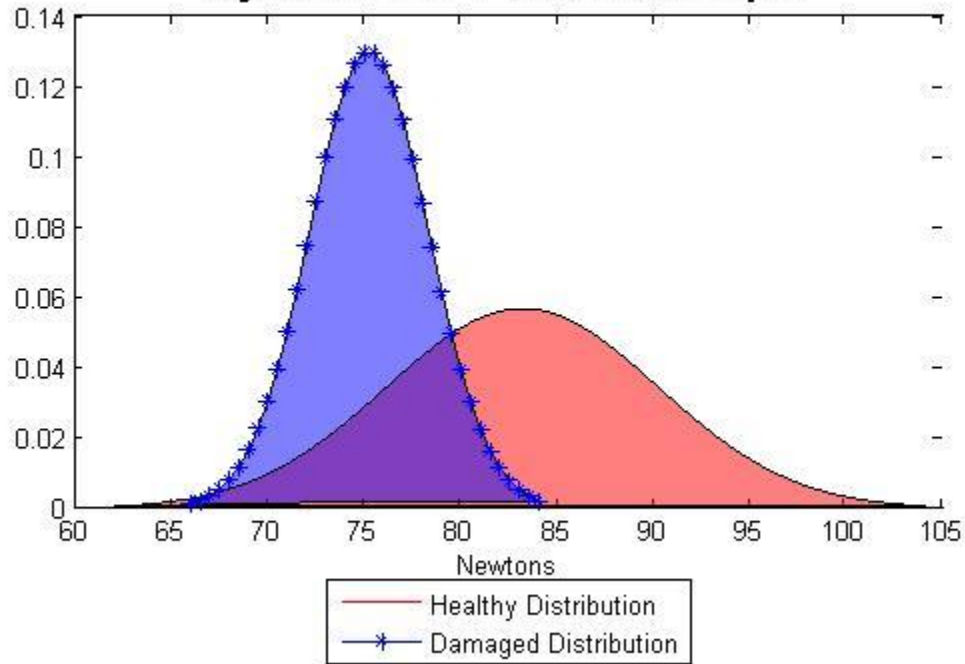
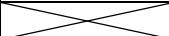


Figure 32: Distribution for healthy and damaged case, Dynamometer 3rd peak Y axis

If only using this variable, the output of the proposed model has a higher amount of error. However, if the distributions are clearly separate, as with D_{ny1} and D_{ny2} , even the univariate models produce clear results with small amounts of error. The most powerful implementations of Bayesian models use multiple features where at least one feature has clearly separate distributions. To illustrate this effect, Table 8 displays a result from one of the damaged test points. The first row is the test point represented in Newtons. In the second row, each column represents the error in the Bayesian univariate posterior probability for D_{ny1} , D_{ny2} , and D_{ny3} respectively. The next row shows the error in the

multivariate posterior probability for a model with only D_{ny2} , and D_{ny3} as features. The final row represents the error of the three feature model.

Table 8: Three variable multivariate model, Dynamometer Y axis

Test			
	D_{ny1}	D_{ny2}	D_{ny3}
Damaged Test Point (Newtons)	55.39	44.52	78.33
Univariate Error	3.11e-133	1.27e-177	0.3607
Multivariate Error (2 feature)			8.845e-178
Multivariate Error (3 feature)	2.753e-310		

As was expected, the error from the univariate case is miniscule for variable D_{ny1} and D_{ny2} while it is significant for D_{ny3} . When moving to a multivariate case with both D_{ny2} , and D_{ny3} , the results of 8.845×10^{-178} are a small improvement over the univariate results of 1.27×10^{-177} . This demonstrates that features with overlapping distributions are still a valuable asset to the Bayesian models to reduce error. Finally, with all three features in the model, an error of 2.753×10^{-310} is significantly lower than any of the univariate cases. By having multiple features with clearly separate distributions, the confidence in results is increased. This provides insight into the strong correlation between expected and actual results for the 17 and 5 feature models. Because there were multiple features with clearly separate distributions for each condition, the model was able to be accurate 100% of the time.

To observe the effect of the increase in number of samples in the conditional database, error as a function of number of samples for the five feature model has been plotted in

Figure 33 below. The error value and range are plotted. For comparison, the results for the 3mm, 4mm, and 5mm radial depth of cut cases are all shown.

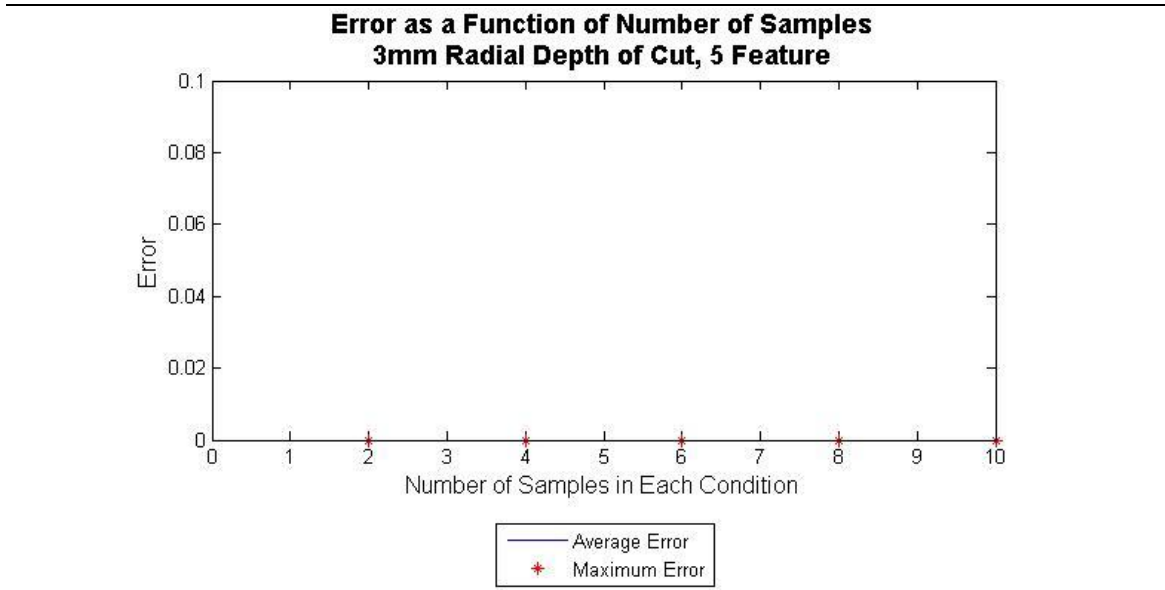


Figure 33: Error as a function of number of samples for 3mm Radial Depth of Cut

From the figure, it is seen that the error and spread is negligible regardless of the number of samples between 2-10 in each condition.

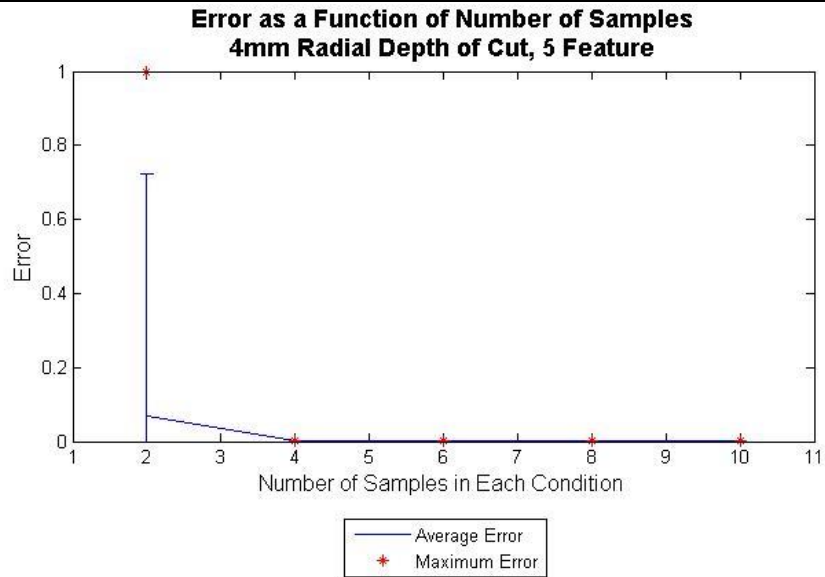


Figure 34: Error as a Function of Number of Samples for 4mm Radial Depth of Cut

From the figure above, it is seen that for 4mm radial depth of cut, the error and spread decrease rapidly as the number of samples increase. When there are only two samples in each condition, the error reached the maximum of one even though the average error is only 8%. By four samples of data in each condition, both the average and range of the error are negligible.

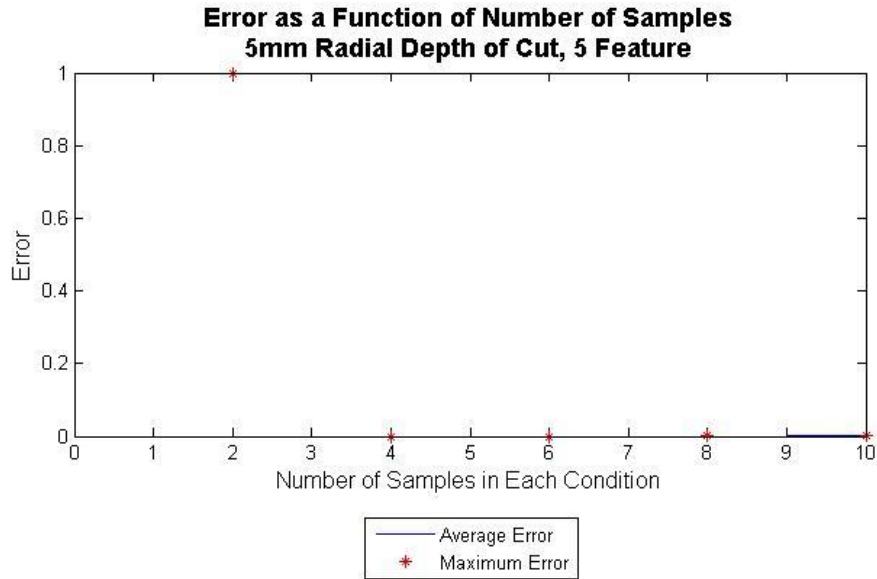


Figure 35: Error as a Function of Number of Samples for 5mm Radial Depth of Cut

Similar to Figure 33 Figure 35 shows that the error and spread are negligible for the number of samples of 2-10 in each condition of the condition database. From these three figures, it is suggested that the updating feature of the algorithm would produce accurate models after collection of 4-6 samples of data per condition when using the specified five variables for this experimental case. To better demonstrate the effects of model updating, a simulated approach was taken in the section “Effect of Model Updating.”

4.2 SIMULATED IMPLEMENTATION

The results of the simulated implementation are organized as follows. First the effect of overlapping distributions is discussed. Next the effect of model updating is presented. Finally, additional results are presented.

4.2.1 EFFECT OF OVERLAPPING DISTRIBUTIONS

The results from testing the effect of overlapping distributions on multivariate Bayesian models are organized in Table 9 below. The results are also illustrated in Figure 36. In the first two columns of the table, the amount of distribution overlap is characterized by the distribution range overlap and the distribution area overlap. In the next three columns are the average value of the error, the standard deviation of the error, and the range of the error for each set of overlapping distributions. The final column is the % accuracy of the Bayesian model. Error and accuracy for these models are defined in section 3.4.

Table 9: Effects of distribution Overlap for 3 Variable Bayesian Model

Simulation: Effect of Distribution Overlap					
Distribution Range Overlap	Distribution Area Overlap	Average Error	Standard Deviation	Range	Accuracy
0.00%	0.00%	0.0000	0.0000	0.0000	1.00
8.33%	0.60%	0.0000	0.0000	0.0000	1.00
16.67%	1.24%	0.0000	0.0000	0.0001	1.00
25.00%	2.44%	0.0001	0.0051	0.2312	1.00
33.33%	4.55%	0.0006	0.0216	0.9462	.9995
41.67%	8.01%	0.0021	0.0295	0.8956	.9990
50.00%	13.36%	0.0091	0.0724	0.9993	.9936
58.33%	21.13%	0.0366	0.1358	0.9867	.9721
66.67%	31.73%	0.0816	0.1963	0.9998	.9387
75.00%	45.33%	0.2029	0.2936	0.9991	.8260

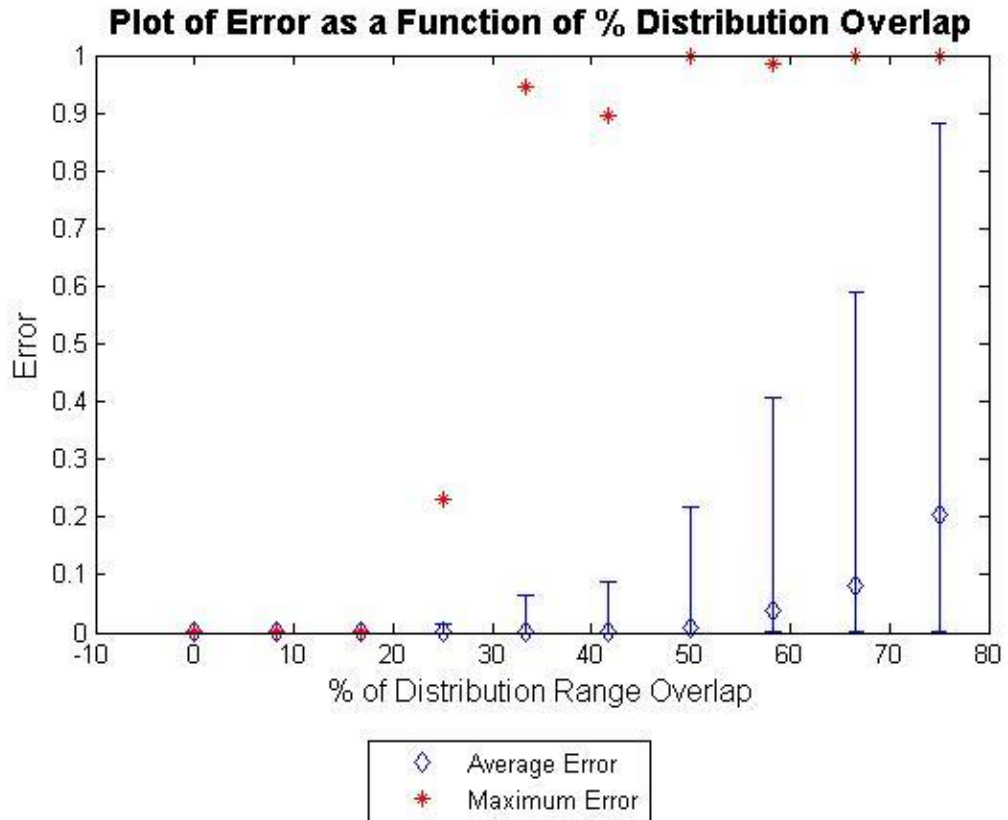


Figure 36: Error vs. Percent of Distribution Range Overlap

In Figure 36, the error bars represent the three sigma limits. As was shown in Figure 2, 99.7% of data fall within the three sigma limits for Normally distributed data. The diamond represents the average error and the asterisk represents the maximum error observed from the test data sets.

From Figure 36, it is seen that the maximum possible error begins to be of significance around 20-30% distribution range overlap (2-5% distribution area overlap). Also from the table and the figure, it is seen that the accuracy begins to quickly diminish as the overlaps

pass 50% distribution range overlap (13.36% distribution area overlap). This discussion continues in the next section while discussing the effects of model updating.

4.2.2 EFFECT OF MODEL UPDATING

The results from testing the effect of model updating on multivariate Bayesian models are summarized in Figure 37, Figure 38, Figure 39, and Figure 40. Each figure shows the error as a function of number of samples for a different % distribution area overlap. To view the figures for all ten distribution sets, please see Appendix A.

As seen previously, the diamond represents the average error and the asterisk represents the maximum observed error. The error bars represent the 3σ limits of the results.

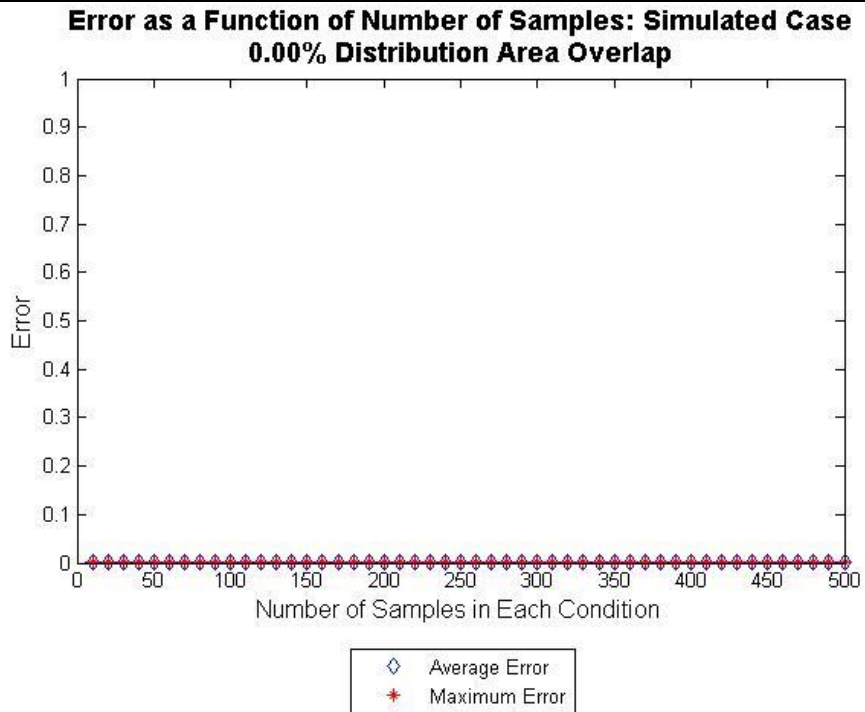


Figure 37: Error as a Function of Number of Samples: 0.00% Distribution Area Overlap

From Figure 37, it is seen that with 0% overlap, the error is minimal. The maximum error observed is less than 10^{-6} and in the majority of instances the error is less than 10^{-12} . Although the error decreases as the number of samples increase, the error is never significant. Given what the average error is, the maximum observed error is also negligibly small. From a statistical perspective, these results are expected.

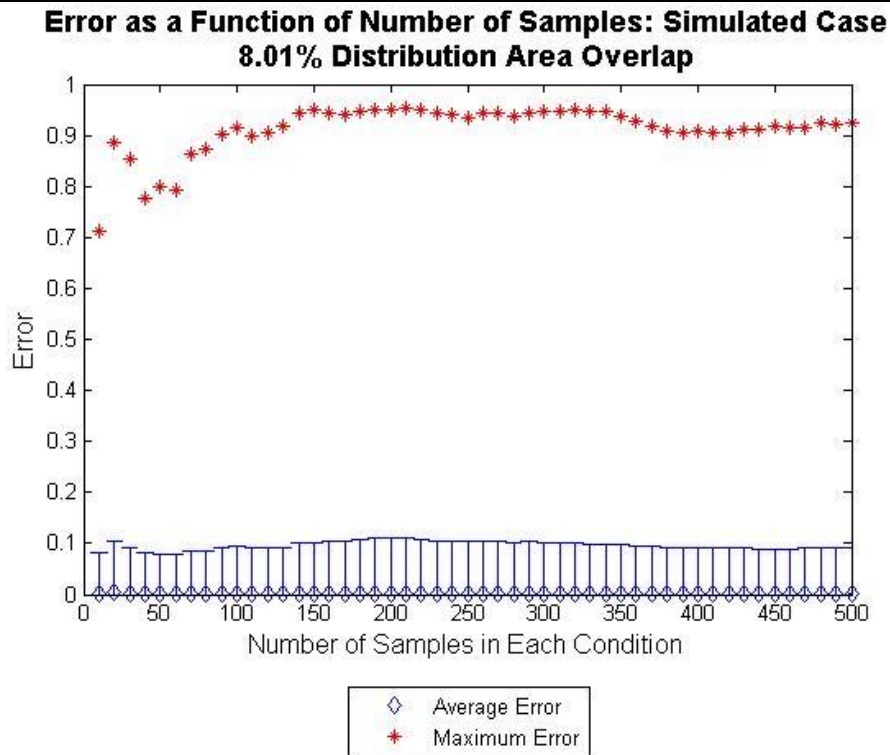


Figure 38: Error as a Function of Number of Samples: 8.01% Distribution Area Overlap

As the overlap increases, the spread of error increases rapidly. Although the average error in the 8.01% case is still low (with a maximum average of 0.003) the range is drastically increased to between 0.70-0.94. This indicates that as overlap increases, uncertainty increases. As the number of samples in the conditional database increase, the standard deviation does not noticeably decrease and the range does not significantly decrease. Instead, as the number of samples increase, the mean error, standard deviation, and range begin to stabilize.

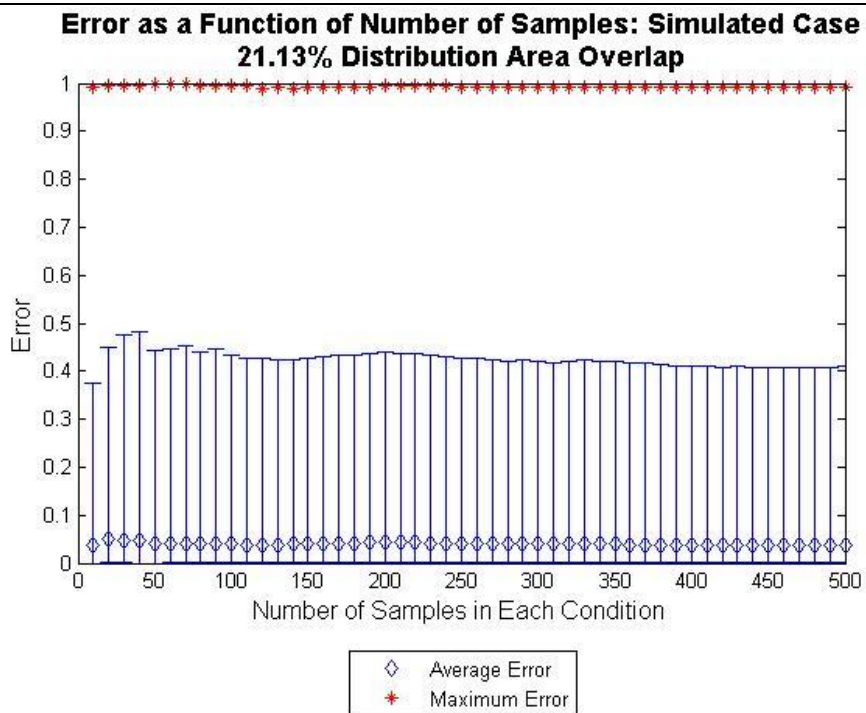


Figure 39: Error as a Function of Number of Samples: 21.13% Distribution Area Overlap

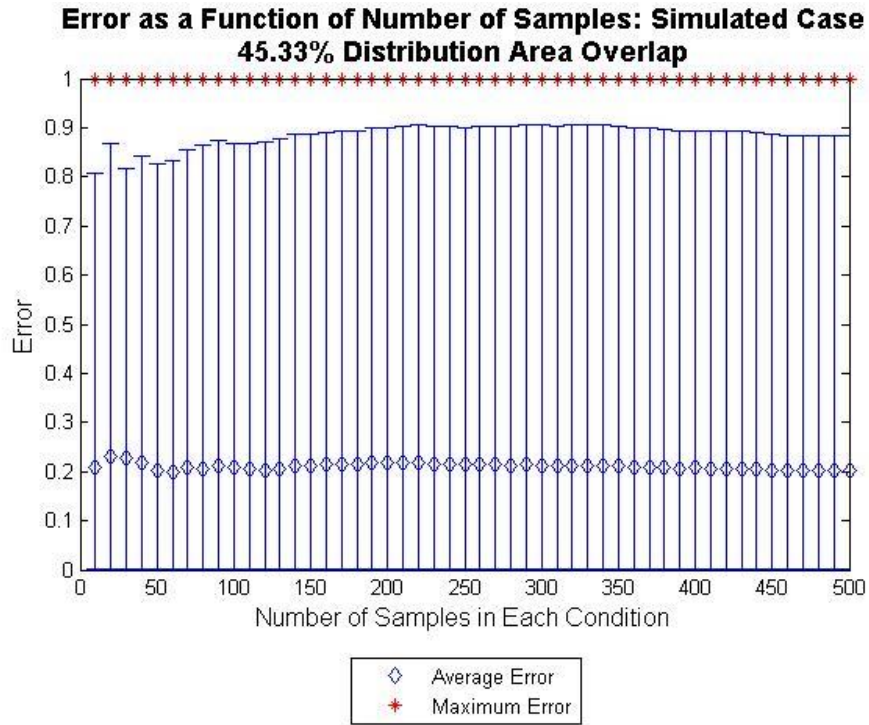


Figure 40: Error as a Function of Number of Samples: 45.33% Distribution Area Overlap

In Figure 39 and Figure 40, the trends identified in Figure 38 continue. For example, as the overlap increases, the spread continues to increase rapidly while the average error increases less rapidly. Also, as the number of samples increase, the error and range of error begin to stabilize.

From the previous four figures it was seen that the models perform relatively consistently after having about 100 samples for each condition in the condition database. Putting all of the distributions together in a histogram better illustrates the effect of model updating on the multivariate Bayesian model regardless of the amount of distribution overlap. In the

histogram, the x axis is the number of samples in each condition from 0 to 200, the y axis is the percent distribution range overlap and the z axis is the average error.

**Error as a Function of Number of Samples and Percent Distribution Range Overlap:
Simulated Case**

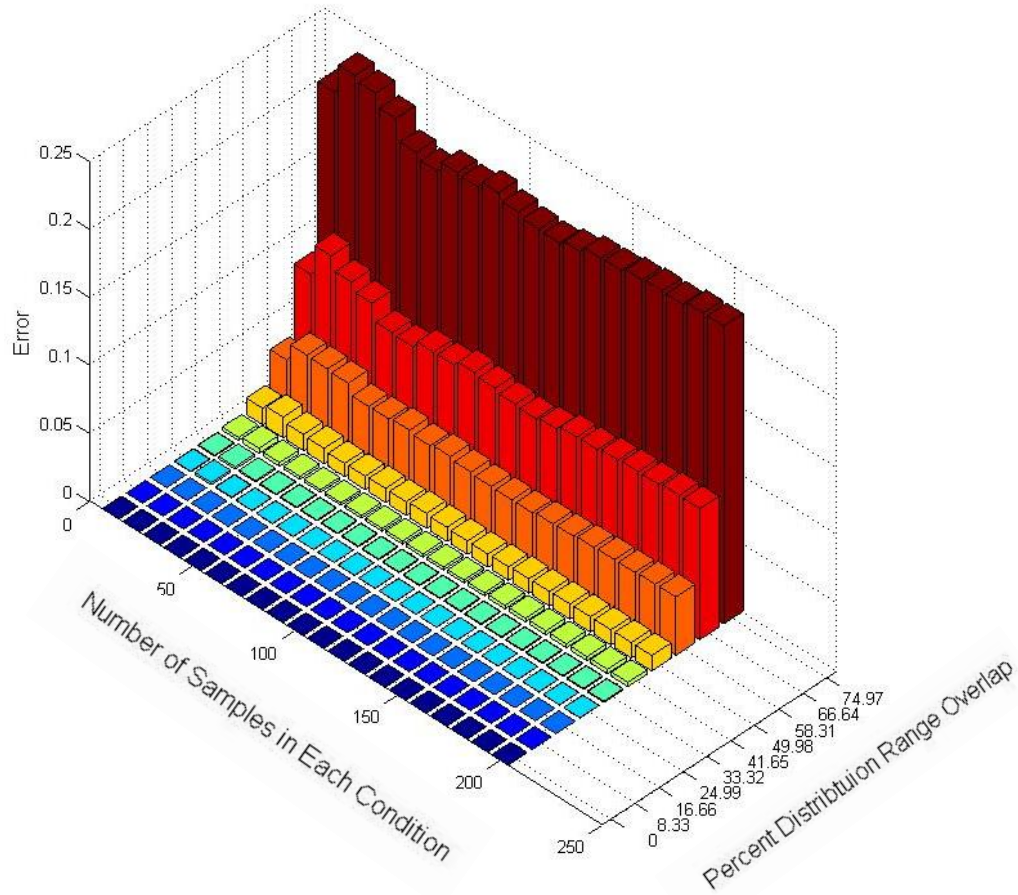


Figure 41: Histogram of Error as a Function of Number of Samples and Percent Distribution Range Overlap

Figure 41 highlights that the variation in the average error observed when building up the model from 10 to 50 points, is more substantial as distribution overlap increases.

By changing the figure above from histogram to a mesh plot, and changing the x axis from a maximum of 200 samples in each condition, to 1000, it is easier to observe how the error stabilizes with increasing numbers of samples. From Figure 42 it is also clear that the amount of samples required for the error to stabilize is related to the amount of distribution overlap. The more overlap, the more samples needed for error to stabilize. Note, although the x,y, and z axis values are the same, the positive direction of the x axis has been reversed for clarity.

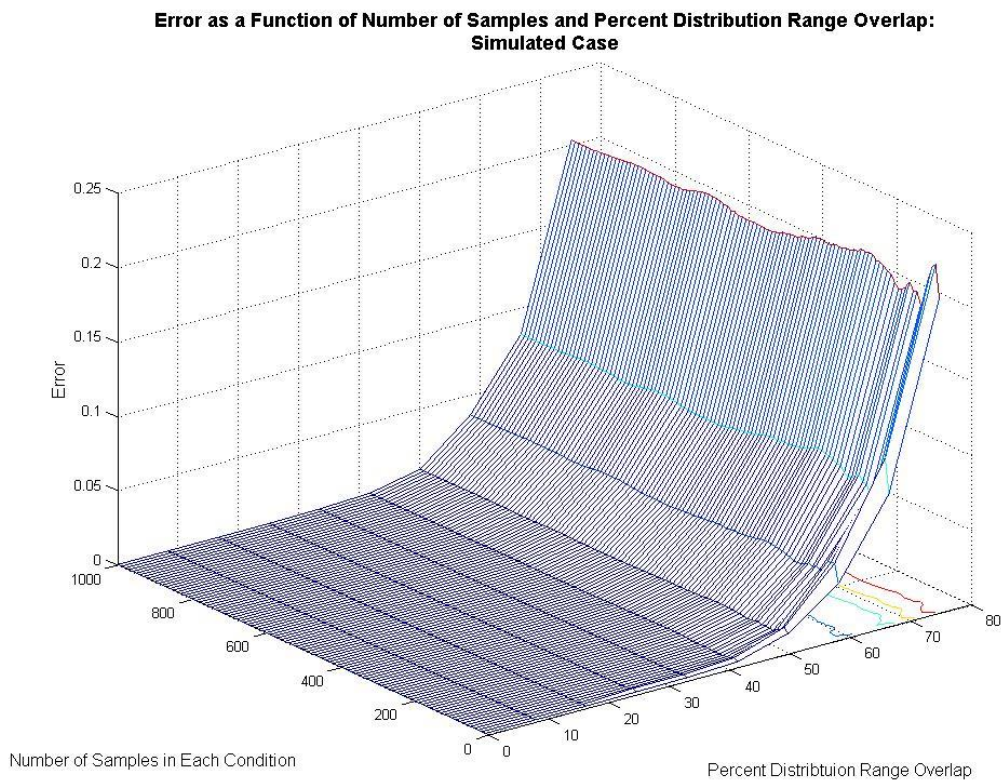


Figure 42: Mesh Plot of Error as a Function of Number of Samples and Percent Distribution Range Overlap

4.2.3 ADDITIONAL RESULTS

From the experimental data, it was seen that a single feature with clearly separate distributions severely reduces a multivariate Bayesian model's error (Table 8). To better view this effect, the testing for the effect of number of samples and percent distribution overlap on error (Figure 41) was repeated with each model having an additional feature that had clearly separate distributions. The results are shown in Figure 43.

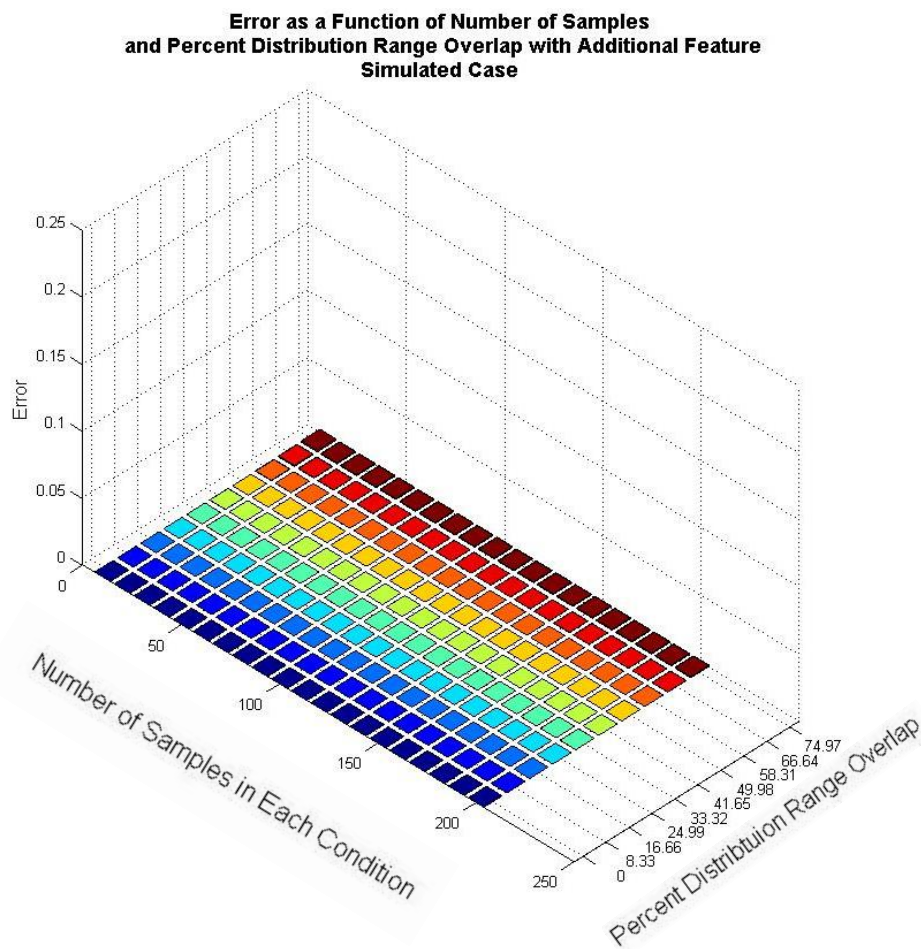


Figure 43: Histogram of Error as a Function of Number of Samples and Percent Distribution Range Overlap When Feature with Zero Overlap is Added to Model

Figure 43 illustrates how a single feature with clearly separate distributions significantly reduces the error regardless of the amount of distribution overlap of the majority of the features in the model. Figure 43 is plotted with the same axis limits as Figure 41 (case without additional zero overlap feature) to enable comparison. Figure 44 has the z axis limits such that one can distinguish variations between the data sets.

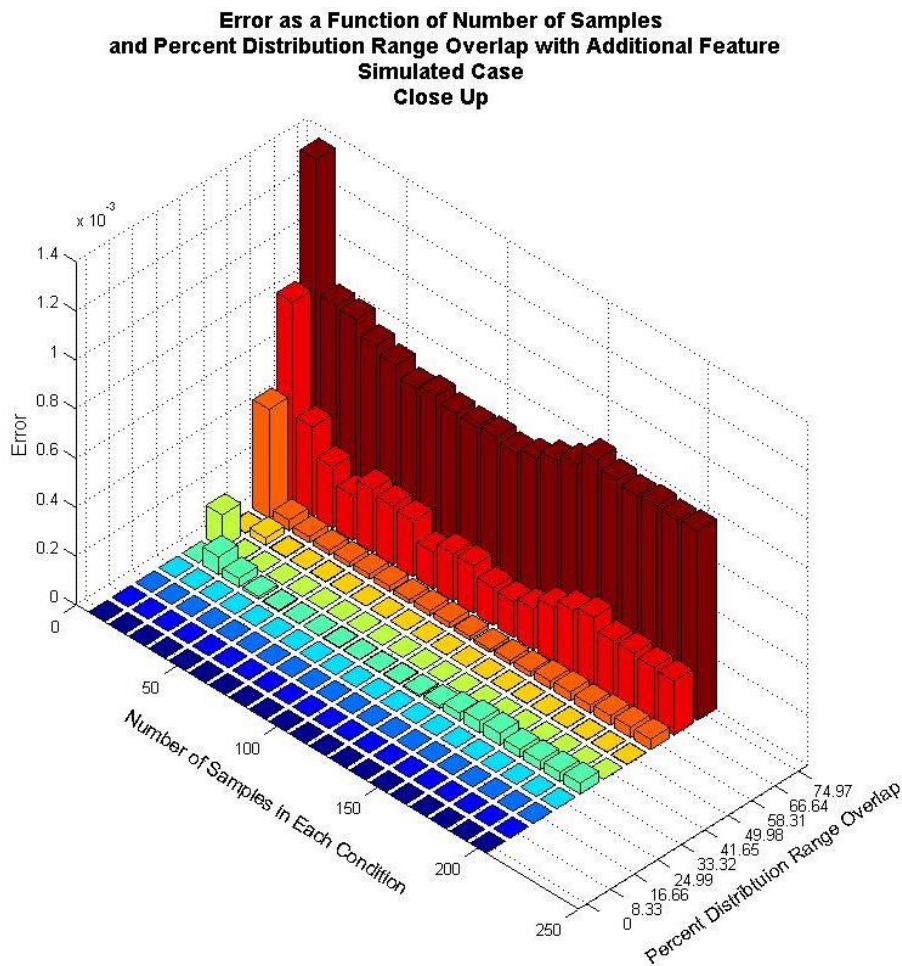


Figure 44: Close up of Histogram of Error as a Function of Number of Samples and Percent Distribution Range Overlap When Feature with Zero Overlap is Added to Model

From Figure 44 it is seen that the larger the amount of distribution overlap of the majority of the features in the model, the higher the error is.

When making measurements, it is common practice to take multiple measurements of the same feature and use the average of the result in order to reduce uncertainty. For example, when extracting the magnitude of FFT peaks from accelerometer or Dynamometer data, it was chosen to take the average of 10 FFT peaks instead of using just one. Instead of averaging the result, one can also use each measurement as an individual feature. The effect of using multiple instances of the same feature is tested in the same way to testing the effect of using multiple features because each repeated test of a feature was treated as an independent feature in the Bayesian model (clearly violating the assumption of conditional independence). The test was performed from 1-9 features with the same range of distribution overlaps as defined in the methods section. The results are illustrated in Figure 45. Percent distribution range overlap is plotted on the x axis, number of features is plotted on the y axis, and average error is plotted on the z axis.

**Error as a Function of Number of Features
and Percent Distribution Range Overlap:
Simulated Case**

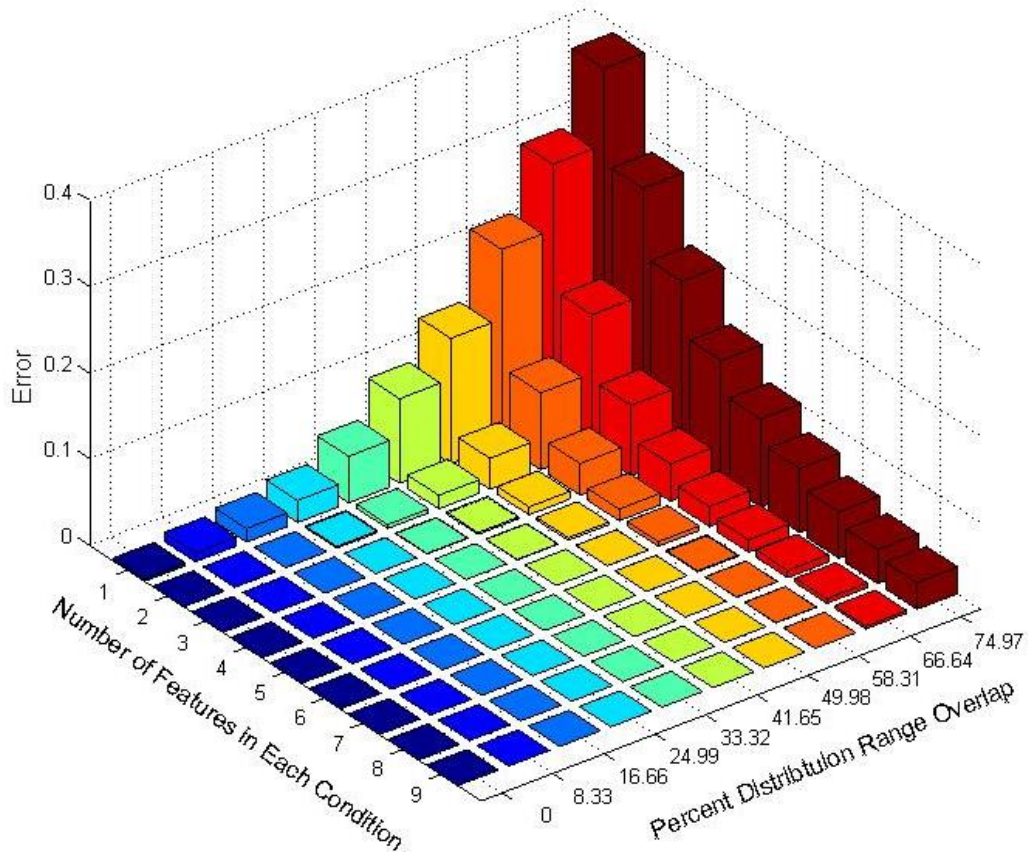


Figure 45: Effect of Number of Features and Distribution Overlap on Error

From Figure 45 it is seen that the number of data instances or features needed to cut through the uncertainty resulting from the distribution overlap, is related to the amount of distribution overlap present. For example, if there is a feature with about 41% distribution range overlap, the figure suggests that gathering 3 data points are enough to provide a clear result from the Bayesian model. If on the other hand, one has a feature with 0% distribution range overlap, a single measurement of that feature is enough to provide clear results.

5 CONCLUSIONS AND RECOMMENDATIONS

This thesis demonstrates a general Bayesian approach for manufacturing equipment diagnostics using sensor fusion. To accomplish this, a method to build and implement Bayesian models is formulated and then demonstrated via an experimental case. Classical limitations during model building and model use are explored by discovering the effects of model updating, the effects of the quantity of signal features used to build the models, and the effects of distribution overlap.

In this thesis, models are built from conditional and non-conditional probabilities that are related to features and conditions of a machine tool. Conditions represent the state of the machine tool, such as broken tool or chipped insert. Features are indicators extracted from sensor data that are used to determine the most probable condition of the machine tool. A feature has a unique distribution for each condition of the tool. For example, if a feature is the average value of power consumption during a cutting operation, P_m , and there are two conditions: damaged tool and healthy tool, then there is a probability density function (PDF) representing the distribution of P_m when the machine tool is damaged (damaged PDF) and a PDF representing the distribution of P_m when the machine tool is healthy (healthy PDF). The PDFs are built from distribution characteristics that are extracted from historical data stored in a condition database. Model updating is the process of adding additional data to the condition database and re-extracting the distribution characteristics. Studying the effect of model updating is studying the effect of the quantity of historical data stored in the condition database. Distribution overlap is defined as the overlap in the PDF distributions that represent the conditions for a single feature (i.e. the area or range overlap of the healthy PDF with the

damaged PDF for the feature P_m). A large overlap of these distributions makes it difficult to probabilistically distinguish between machine conditions using the specific feature. Studying the effect of distribution overlap is studying the effects of distribution overlap on Bayesian model error and investigating methods to reduce this error when large distribution overlap is present. Finally, the quantity of signal features used to build the models is the number of extracted features from sensor data that is used as indicators of machine state in the model. Studying the effects of the quantity of signal features used to build the model is studying the effect of having more or less signal features in the Bayesian model.

5.1 CONCLUSIONS

To implement a Bayesian model, a two phase approach is employed to separate the model development and the system development of phase 1 from the functional phase 2. The main tasks of phase 1 are: to discover signal features that correlate to the machine state, to standardize data acquisition, data processing, and feature extraction methods to be used in phase 2, to build a condition database, and to build the Bayesian models. For phase 1, the equation used to build the Bayesian model is represented in equation (50).

$$P(C_i|\vec{d}) = \frac{P(C_i) \prod_{j=1}^m P(F_j|C_i)}{\sum_{i=1}^n P(C_i) \prod_{j=1}^m P(F_j|C_i)}$$

where C_i represents condition i , \vec{d} represents a vector of features, F_j represents a feature j , n represents the total number of conditions, and m represents the total number of features. The output of the model, $P(C_i|\vec{d})$, represents the probability of the machine tool being in state C_i given the extracted information \vec{d} .

A method to find signal features that correlate to the state of the machine tool for use in the Bayesian models is to exploit the physics of the machining operation. The main tasks of the functional phase 2 are to employ the models to aid decision making on the manufacturing floor and to update the models when new data are classified. To update the models while in phase 2, the additional data are concatenated to the appropriate location in the conditional database. The feature distribution characteristics for the Bayesian model are then re-extracted. This two phase process is represented by the flow chart illustrated in Figure 10: The phase one and the phase two of proposed method.

The purpose of the experimental case is to demonstrate the steps taken to implement both phases and perform model building. Two models, a 17 feature and a 5 feature model, are built and tested. The 17 feature model is resultant from four distinct sensors: a MEMS accelerometer, a piezoelectric accelerometer, a dynamometer, and a power meter. The 5 feature model is resultant from two sensors: a MEMS Accelerometer and Power Meter. Both the 17 feature and the 5 feature model are accurate for 100% of the trials. These results are verified with k-fold cross validation. Due to the nature of the experimental case, taking a Bayesian approach does not add significant information to the model. As such, a simulated approach is exploited to better illustrate the classical limitations of building and applying the Bayesian models.

The effects of model updating are tested by determining the effect of quantity of data in the condition database. The amount of data in the condition database does not affect the mean or spread of model error, however, it does stabilize the magnitude of error observed. In the simulated test cases, it takes about 50-100 instances of data in the condition database for most results to converge. Nevertheless, the amount of data needed

in the condition database to stabilize the error is dependent on the amount of distribution overlap present. Generally, the less distribution overlap present, the less data is needed in the condition database. For example, for a distribution range overlap of about 30% for all features in a three feature model, only 30 instances of data are needed for consistent results. On the hand, for a distribution range overlap of about 60% for the same model, 140 instances of data are needed for consistent results. If there is at least one feature with clearly separate distributions in the multivariate model, simulated results demonstrate that the results become consistent within 10 samples per condition in the condition database. In the experimental case, there were a number of features with clearly separate distributions, thus results converged in 4-6 samples per condition.

From the simulated testing the general effects of distribution overlap on Bayesian model performance are analyzed. Model error increases as distribution overlap increases. To illustrate, when all features of a three feature model have distribution area overlap of 8.01%, the average error is 0.002. If all features of a three feature model have a distribution area overlap of 31.73%, the average error is 0.08. Increasing the number of features in the Bayesian models or increasing the number of samples input into the model is a method to decrease the error in the presence of distribution overlap. This is illustrated in Figure 45: Effect of Number of Features and Distribution Overlap on Error.

If there is a single feature with no distribution overlap in the multivariate model, the error is diminished to negligible amounts. For instance, when a single feature with 0.0% distribution overlap is added to the multivariate model whose features have distribution area overlaps of 31.73%, the average error decreases from 0.08 to less than 2×10^{-4} . This effect is clearly observed when comparing Figure 41: Histogram of Error as a Function of

Number of Samples and Percent Distribution Range Overlap and Figure 43: Histogram of Error as a Function of Number of Samples and Percent Distribution Range Overlap When Feature with Zero Overlap is Added to Model.

5.2 RECOMMENDATIONS

As the training and trial data for model testing were both sourced from the same population, further research should explore the consequences of model updating if the sample distributions are different than the population distribution, if the population distribution is shifting, or if the population distribution is unknown. Outside of simulated testing, further research should explore if model updating in phase 2 allows the models to better represent the stochastic nature of some manufacturing operations on the manufacturing floor. Another opportunity for experimental exploration is using the updating method to completely build the Bayesian models while the machines are operating. Although the updating feature was developed to update the models, this could be used to capture data that occurs at low frequencies such as tool failure or machine crash. For example, presently there are no data driven Bayesian models of CNC machine crash. However, when machine crashes occur in the field, a model could be built using the sparse amount of data available. If and when another crash occurs, model updating can be used to further develop the model. Finally, a logical next step is to investigate a general approach to develop and implement Bayesian models that can relate the features over adjacent time steps. In other words, develop general dynamic Bayesian models instead of Naïve Bayesian models.

APPENDIX A

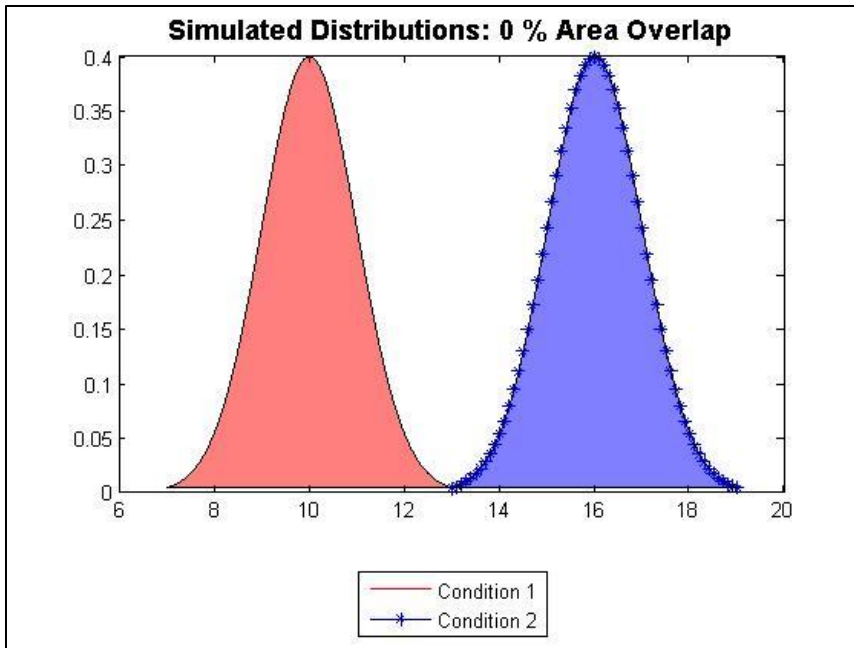


Figure 46: Simulated Distributions, 0% Area Overlap

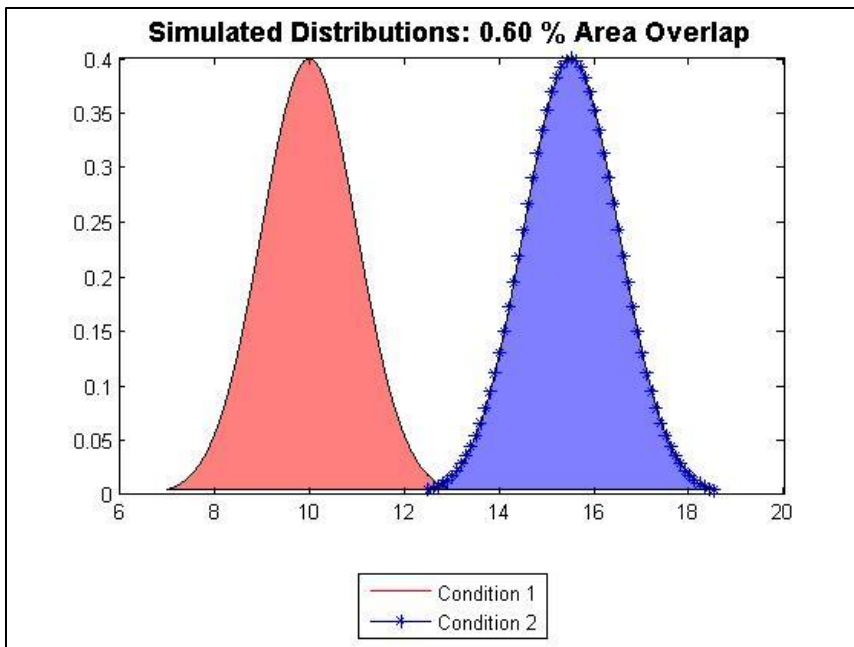


Figure 47: Simulated Distributions, 0.60% Area Overlap

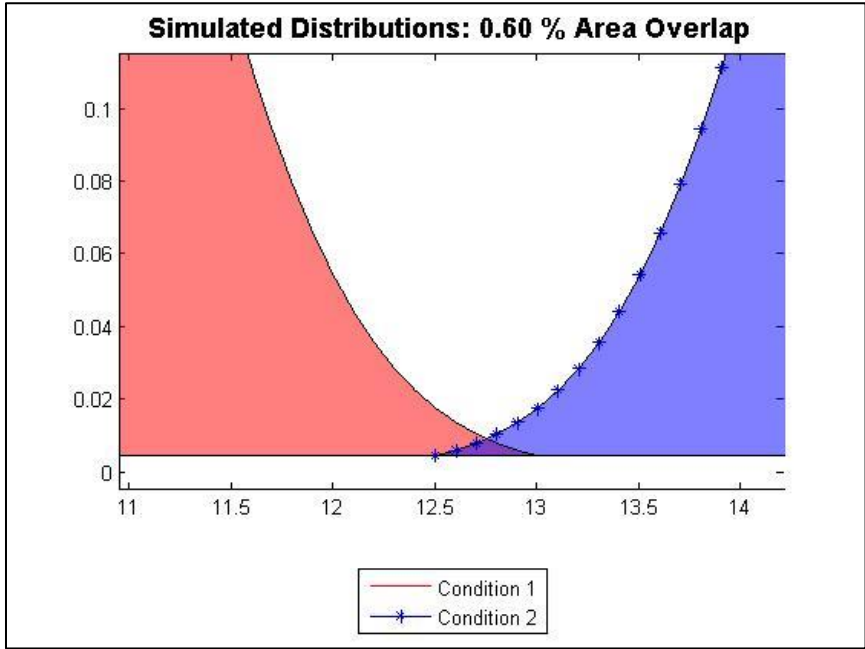


Figure 48: Close Up Simulated Distributions, 0.60% Area Overlap

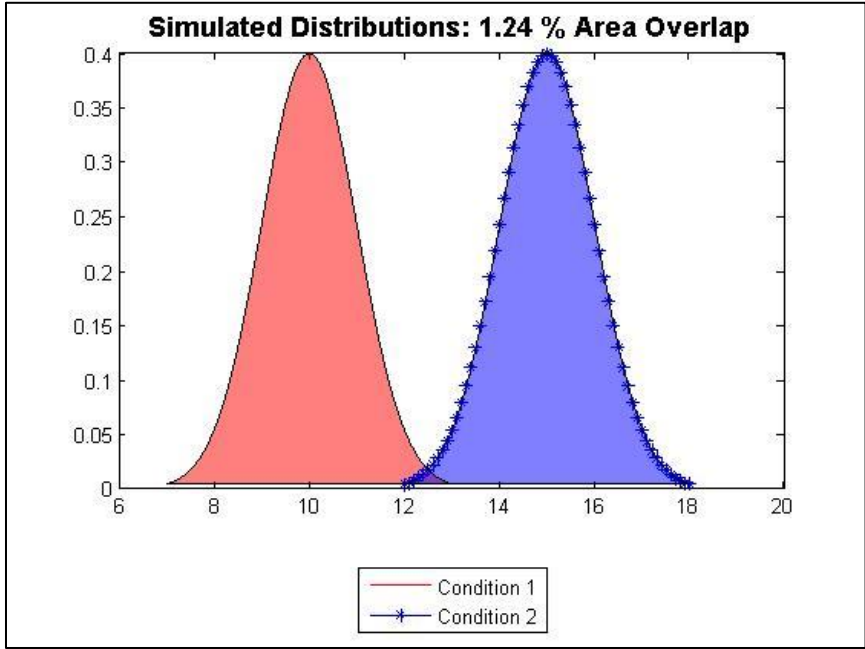


Figure 49: Simulated Distributions, 1.24% Area Overlap

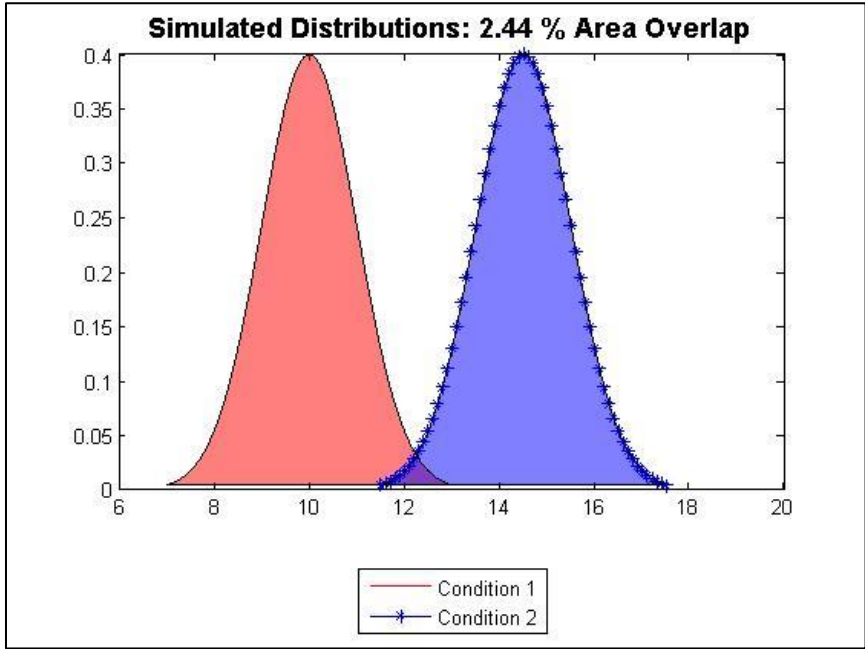


Figure 50: Simulated Distributions, 2.44% Area Overlap

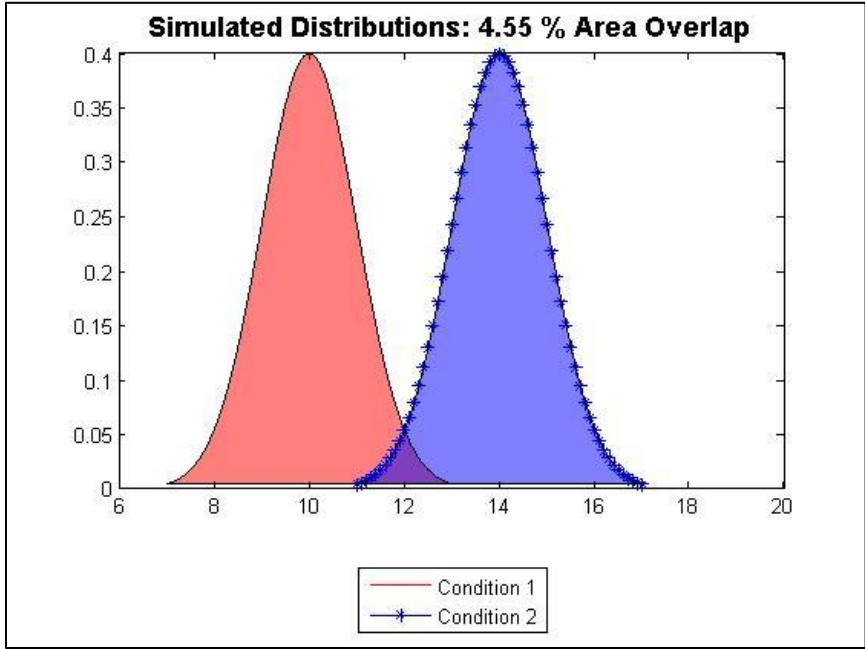


Figure 51: Simulated Distributions, 4.55% Area Overlap

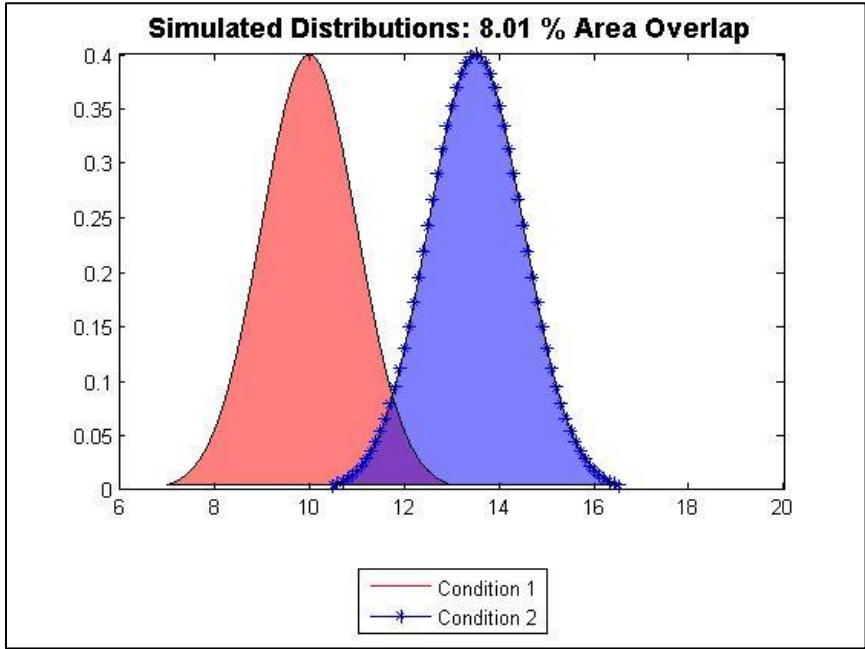


Figure 52: Simulated Distributions, 8.01% Area Overlap

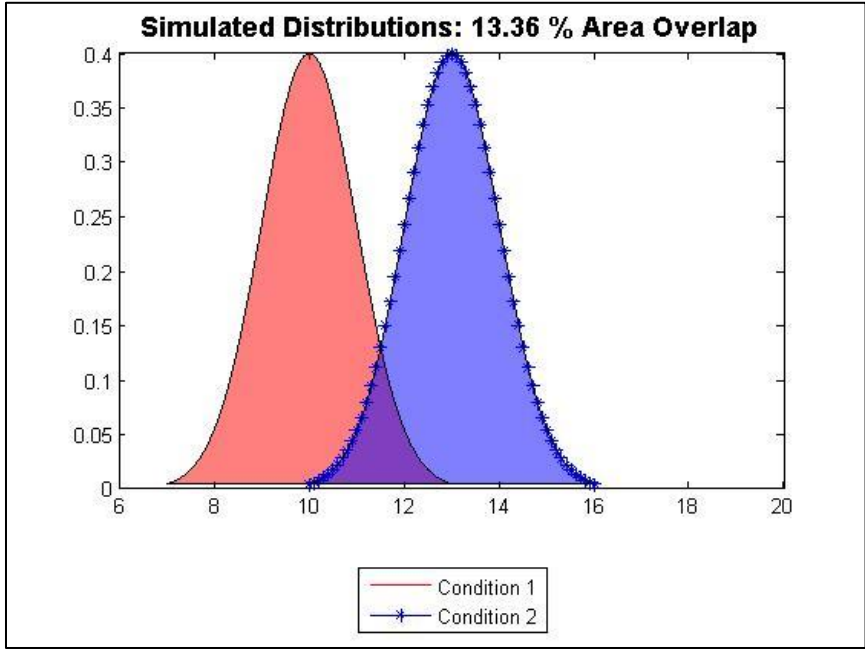


Figure 53: Simulated Distributions, 13.36% Area Overlap

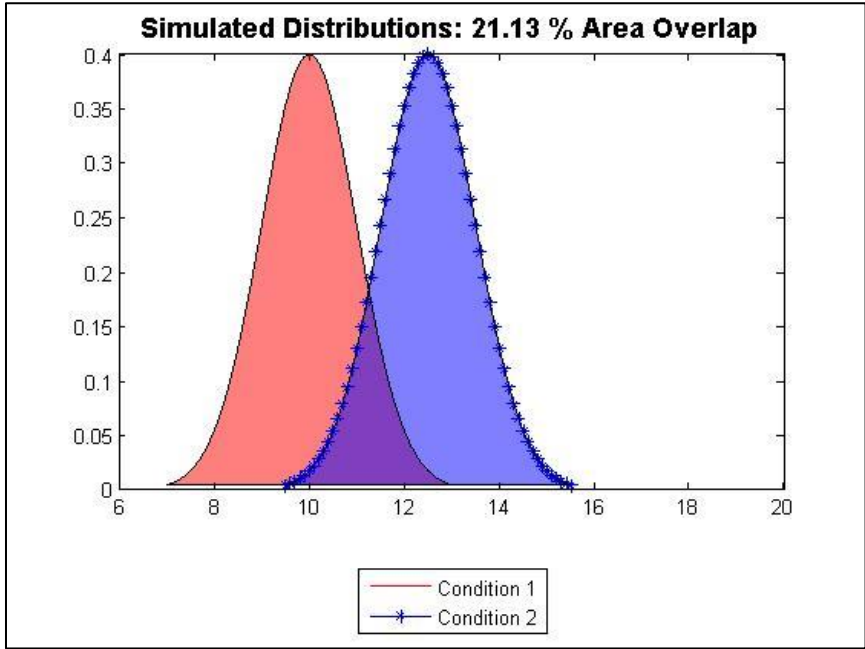


Figure 54: Simulated Distributions, 21.13% Area Overlap

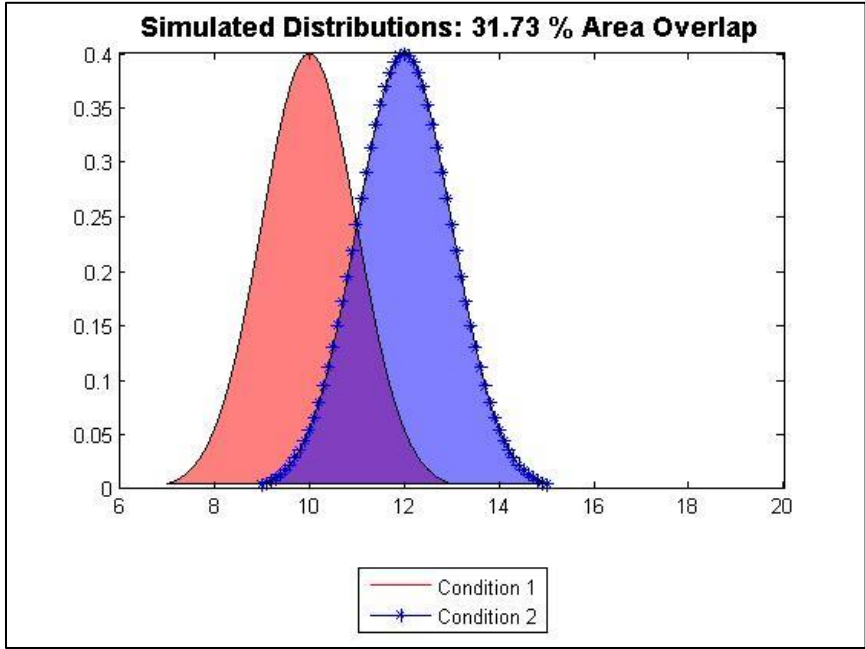


Figure 55: Simulated Distributions, 31.73% Area Overlap

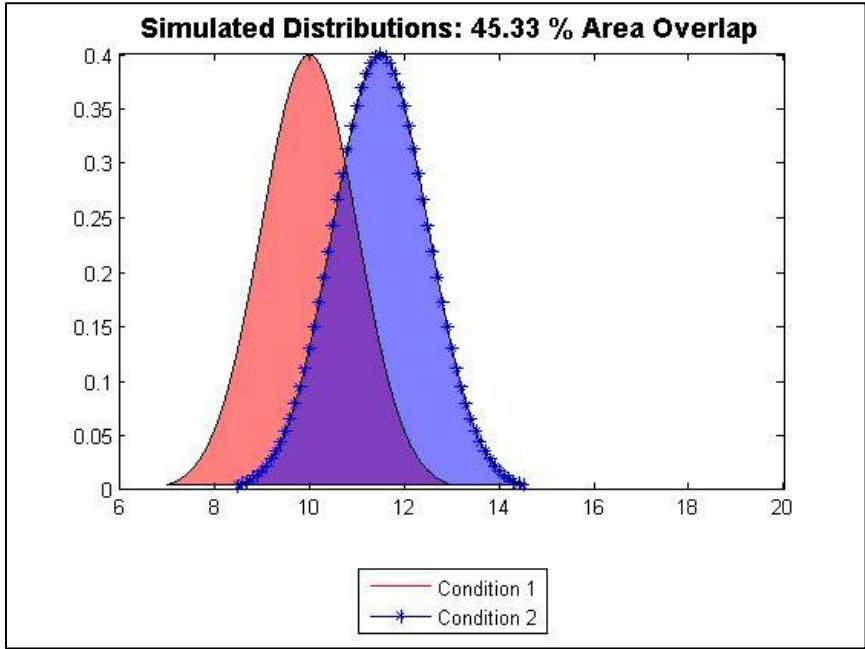


Figure 56: Simulated Distributions, 45.33% Area Overlap

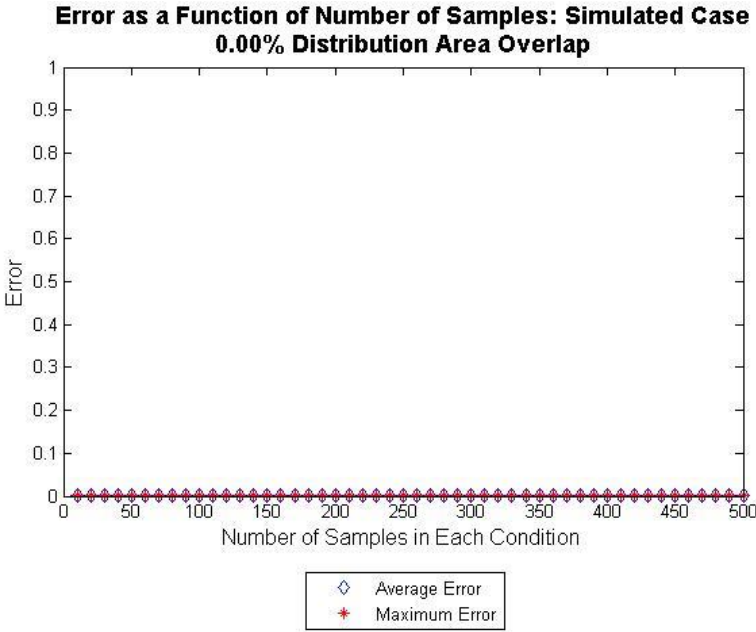


Figure 57: Error as a Function of Number of Samples: 0.0% Distribution Area Overlap

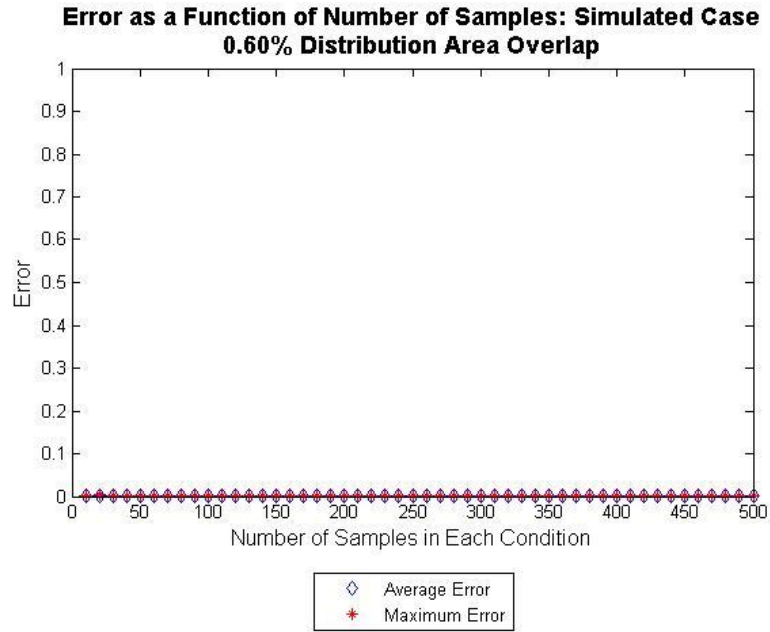


Figure 58: Error as a Function of Number of Samples: 0.60% Distribution Area Overlap

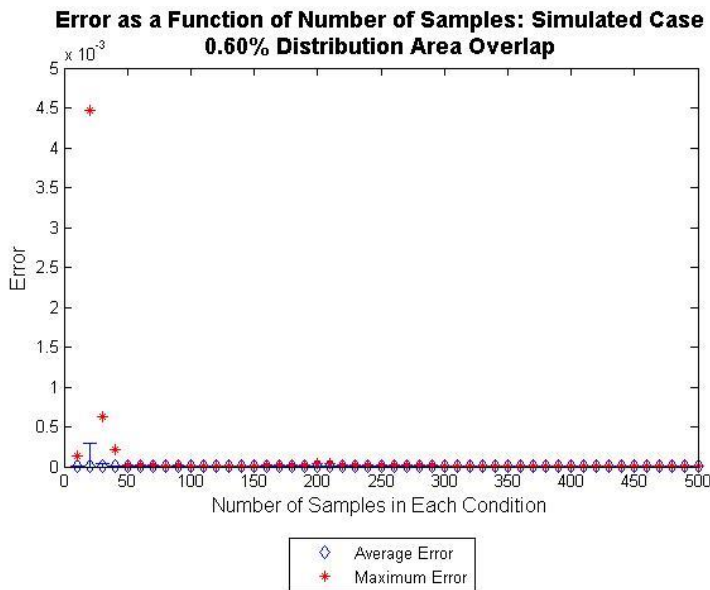


Figure 59: Close Up Error as a Function of Number of Samples: 0.60% Distribution Area Overlap

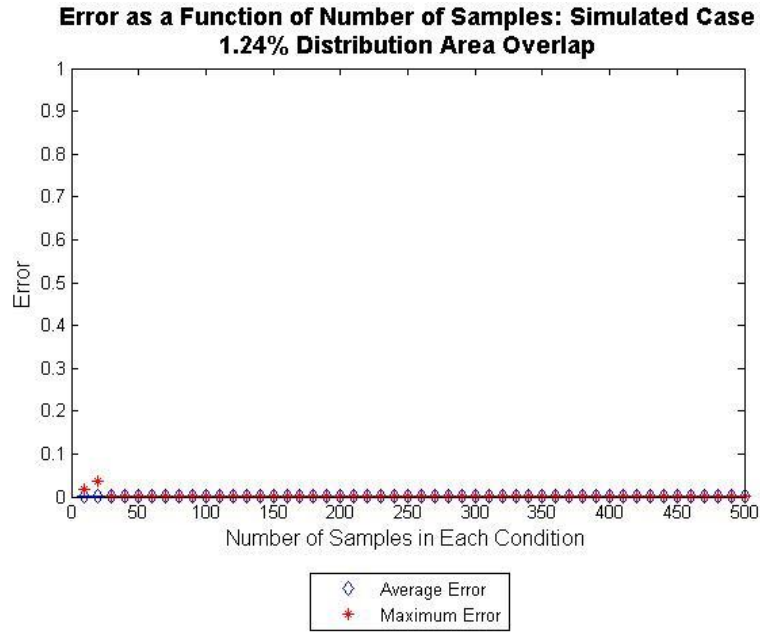


Figure 60: Error as a Function of Number of Samples: 1.24% Distribution Area Overlap

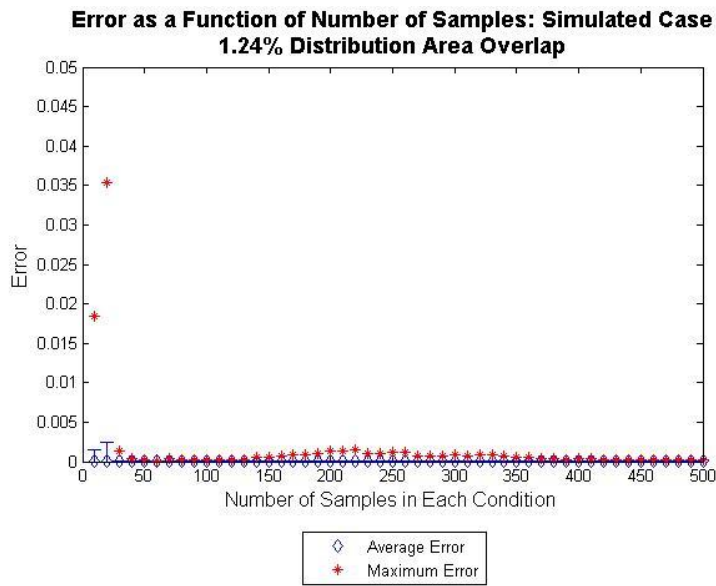


Figure 61: Close Up Error as a Function of Number of Samples: 1.24% Distribution Area Overlap

**Error as a Function of Number of Samples: Simulated Case
2.44% Distribution Area Overlap**

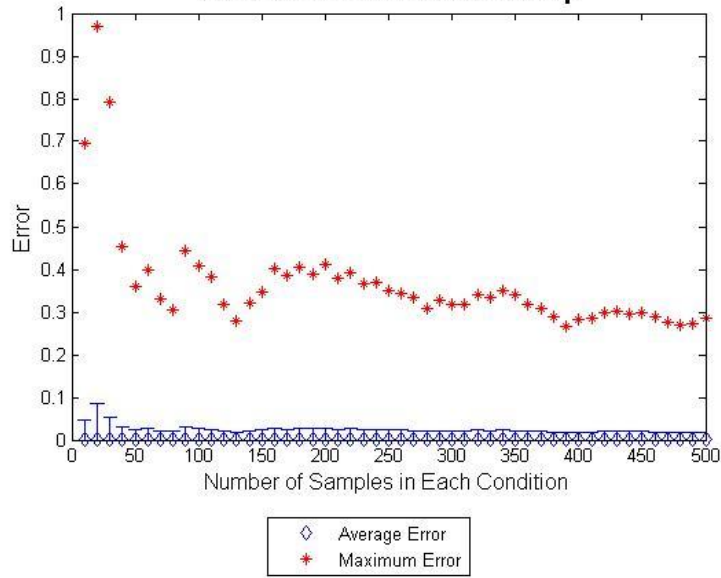


Figure 62: Error as a Function of Number of Samples: 2.44% Distribution Area Overlap

**Error as a Function of Number of Samples: Simulated Case
4.55% Distribution Area Overlap**

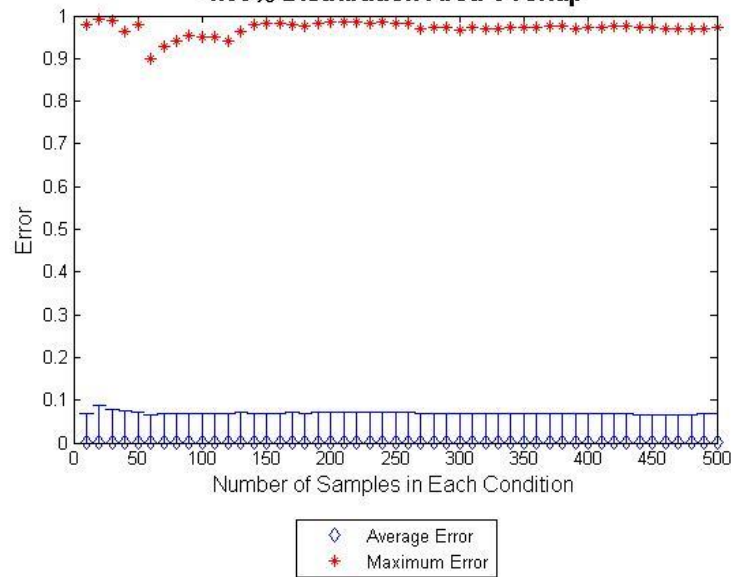


Figure 63: Error as a Function of Number of Samples: 4.55% Distribution Area Overlap

**Error as a Function of Number of Samples: Simulated Case
8.01% Distribution Area Overlap**

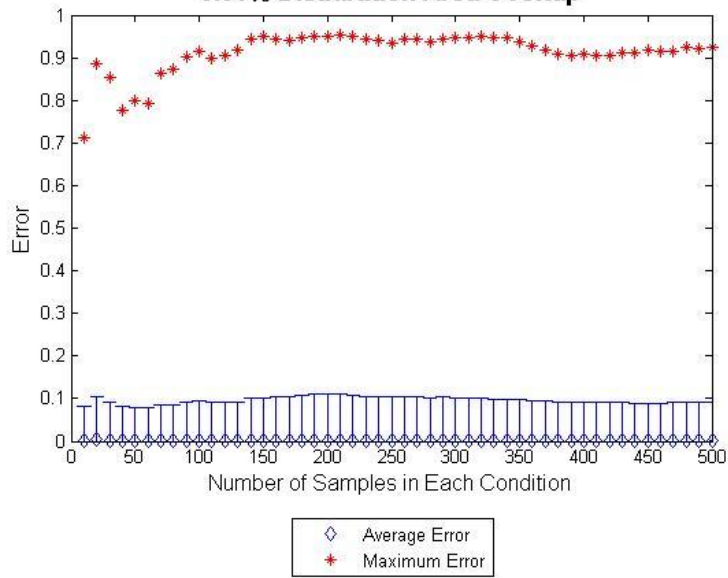


Figure 64: Error as a Function of Number of Samples: 8.01% Distribution Area Overlap

**Error as a Function of Number of Samples: Simulated Case
13.36% Distribution Area Overlap**

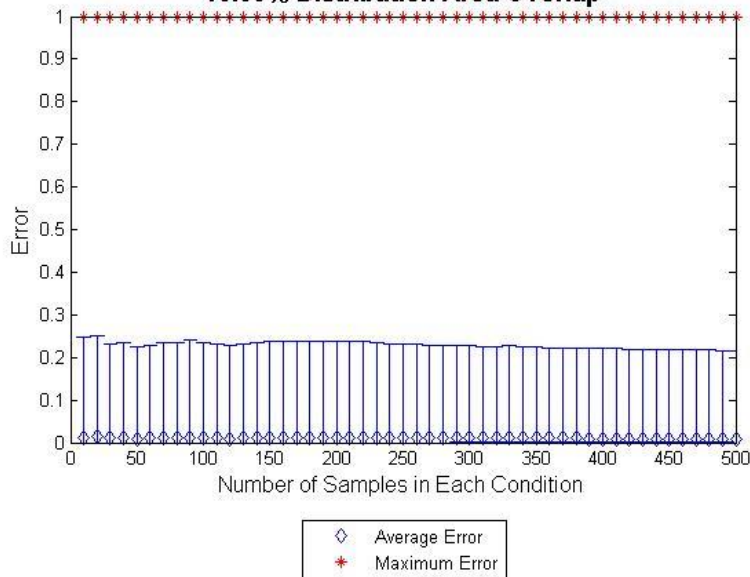


Figure 65: Error as a Function of Number of Samples: 13.36% Distribution Area Overlap

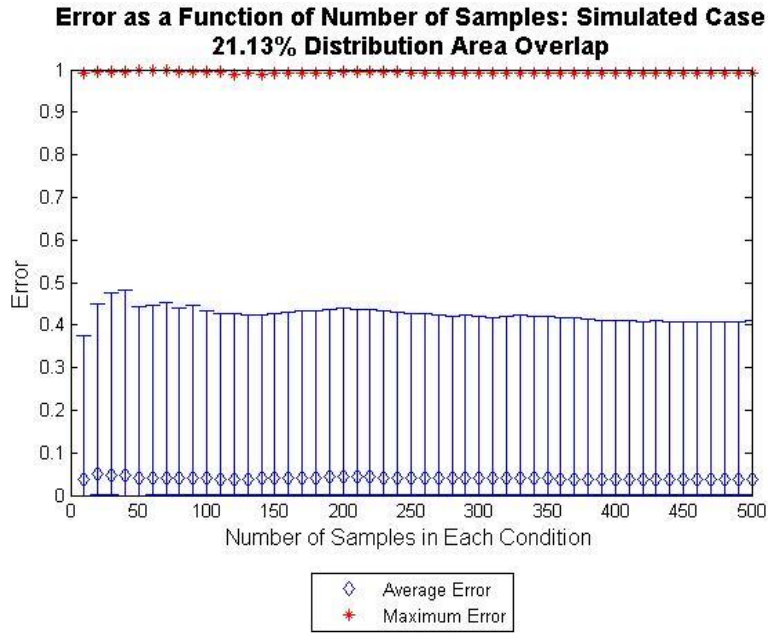


Figure 66: Error as a Function of Number of Samples: 21.13% Distribution Area Overlap

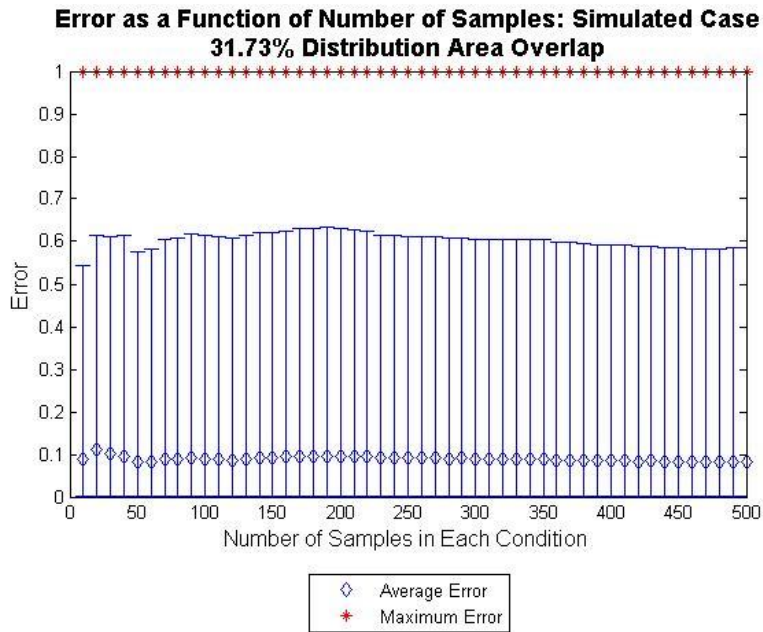


Figure 67: Error as a Function of Number of Samples: 31.37% Distribution Area Overlap

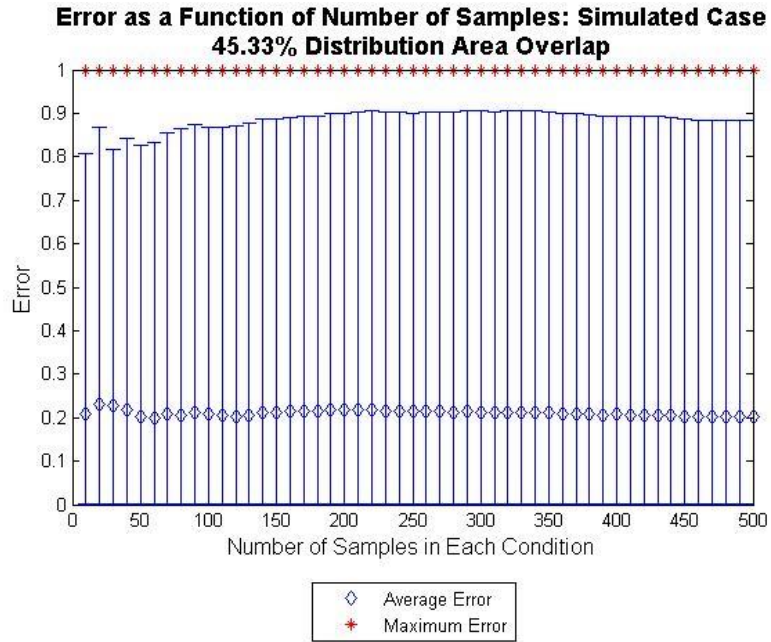


Figure 68: Error as a Function of Number of Samples: 45.33% Distribution Area Overlap

APPENDIX B

```
function [peak1,peak2,peak3]=PeakFinder(fft,RPM,fftRes)
%inputs: fft = NORMALIZED FFT, RPM= rpm of spindle, fftRes= the FFT
resolution

ff=RPM/60; %fundamental shaft frequency in Hz
%% Calculate value of fundametal peak (aka 1st multiple)
l1=(ff-2)/fftRes;%lower limit bin to look in
u1=(ff+2)/fftRes;%upper limit bin to look in
ff_max1=max(abs(fft(l1:u1))); %max peak found in area where we expect
to see fundamental peak
%% Calculate value of the 1st harmonic (aka 2nd multiple) peak
ff2=ff*2;
l12=(ff2-5)/fftRes;%lower limit bin to look in
u12=(ff2+5)/fftRes;%upper limit bin to look in
ff_max2=max(abs(fft(l12:u12))); %max peak found in area where we expect
to see @ 2*fundamental peak
%% Calculate the value of the 2nd harmonic (aka 3rd multiple) peak
ff3=ff*3;
l13=(ff3-5)/fftRes;%lower limit bin to look in
u13=(ff3+5)/fftRes;%upper limit bin to look in
ff_max3=max(abs(fft(l13:u13))); %max peak found in area where we expect
to see @ 3*fundamental peak

peak1=ff_max1;
peak2=ff_max2;
peak3=ff_max3;
end
```

Figure 69: MATLAB Code for FFT Peak Extraction

```
function PosteriorP=BuildBayes(CDatabase_H,CDatabase_D,testData,prior)

[numSets,feats]=size(testData); %feats= the # of features
PosteriorP=zeros(numSets,2); %numSets = the # of test data sets, each
set = 1 d vector

%extract databse features
means_H=mean(CDatabase_H);
means_D=mean(CDatabase_D);

std_H=std(CDatabase_H);
std_D=std(CDatabase_D);

likelihoodH=zeros(numSets,feats);%set up variables to store likelihood
probabilities
likelihoodD=zeros(numSets,feats);
```



```

for i=1:numSets
    %get the likelihoods of healthy condition
    likelihoodH(i,:)=normpdf(testData(i,:),means_H,std_H);

    %get likelihoods of damaged condition
    likelihoodD(i,:)=normpdf(testData(i,:),means_D,std_D);

    %compute the posterior probabilities
    PosteriorP(i,1)=(prior(1)*prod(likelihoodH(i,:)))/...
(prior(1)*prod(likelihoodH(i,:))+prior(2)*prod(likelihoodD(i,:)));
    PosteriorP(i,2)=(prior(2)*prod(likelihoodD(i,:)))/...
(prior(1)*prod(likelihoodH(i,:))+prior(2)*prod(likelihoodD(i,:)));
end
format long e
end

```

Figure 70: MATLAB Function for Bayesian Model for Experimental Case

```

clear
load('simulationData.mat')
prior=[.90,.10]; %the prior probabilities for condition 1 and condition
2
c1=m10; %condition one
c2=m13; %change for each distribution being tested, this is condition 2
[numSam,numEl]=size(c1); %numSam= number of samples, numEl= number of
elements
xInc=10; %x value increment, aka number of points
fold=20; %number of samples per fold
numXpoints=(numSam-fold)/xInc; %how many x points there are
Kfold=numSam/fold; %Kfold is number of folds for testing in cross
validation

Ep_c1=zeros(numSam-fold,numXpoints);%for error values with test data
from C1
Ep_c2=zeros(numSam-fold,numXpoints);%for error values with test data
from C2

h=0; %counter for number of points in conditional database
for j=1:numXpoints
    c=1; %counter for folds
    for i=1:Kfold
        c1hold=c1; %save conditions
        c2hold=c2;
        c1hold(c:c+fold-1,:)=[]; %take out test data
        c2hold(c:c+fold-1,:)=[];
        c1hold=c1hold(1:numSam-fold-h,:); %reduce num of points for graph
        c2hold=c2hold(1:numSam-fold-h,:);
        Post=BuildBayes(c1hold,c2hold,c1(c:c+fold-1,:),prior); %note: here
we test c1 data
        Ep_c1(c:c+fold-1,j)=Post(:,2);
    end
end

```

```

        c=c+fold;
    end
    h=h+xInc; %decrease number of points in conditional database by xInc
end

h=0;
for j=1:numXpoints
    c=1;
    for i=1:Kfold
        c1hold=c1; %save conditions
        c2hold=c2;
        c1hold(c:c+fold-1,:)=[]; %take out test data
        c2hold(c:c+fold-1,:)=[];
        c1hold=c1hold(1:numSam-fold-h,:); %reduce num of points for graph
        c2hold=c2hold(1:numSam-fold-h,:);
        Post=BuildBayes(c1hold,c2hold,c2(c:c+fold-1,:),prior); %note: here
we test C2 data
        Ep_c2(c:c+fold-1,j)=Post(:,1);
        c=c+fold;
    end
end
h=h+xInc;
end

x=(xInc:xInc:numSam-fold);
x=fliplr(x);
full=[Ep_c1; Ep_c2];
means=mean(full);
upper=3*std(full)-means;
lower=means-min(full);
errorbar(x(:,50:100),means(:,50:100),lower(:,50:100),upper(:,50:100),'d
')
hold on
plot(x(:,50:100),max(full(:,50:100)),'r*')
hold off
title({'Error as a Function of Number of Samples: Simulated Case';...
'13.36% Distribution Area
Overlap'}, 'fontweight', 'bold', 'fontsize', 14)
xlabel({'Number of Samples in Each Condition'; ''}, 'fontsize', 12)
ylabel('Error', 'fontsize', 12)
ylim([0,1])
xlim([0,500])
legend('Average Error', 'Maximum Error', 'Location', 'SouthOutside')

```

Figure 71: Code for Kfold Cross Validation and Effect of Number of Points

```

function
PosteriorP=BuildBayes(CDatabase_C1,CDatabase_C2,testData,prior)

[numSets,feats]=size(testData); %feats= the # of features
PosteriorP=zeros(numSets,2); %numSets = the # of test data sets, each
set = 1 d vector

```

```

%extract databse features
means_c1=mean(CDatabase_C1);
means_c2=mean(CDatabase_C2);

std_c1=std(CDatabase_C1);
std_c2=std(CDatabase_C2);

likelihoodC1=zeros(numSets,feats);%set up variables to store likelihood
probabilities
likelihoodC2=zeros(numSets,feats);

for i=1:numSets
    %get the likelihoods of C1
    likelihoodC1(i,:)=normpdf(testData(i,:),means_c1,std_c1);

    %get likelihoods of C2
    likelihoodC2(i,:)=normpdf(testData(i,:),means_c2,std_c2);

    %compute the posterior probabilities
    PosteriorP(i,1)=(prior(1)*prod(likelihoodC1(i,:)))/...
(prior(1)*prod(likelihoodC1(i,:))+prior(2)*prod(likelihoodC2(i,:)));
    PosteriorP(i,2)=(prior(2)*prod(likelihoodC2(i,:)))/...
(prior(1)*prod(likelihoodC1(i,:))+prior(2)*prod(likelihoodC2(i,:)));
end
format long e
end

```

Figure 72: General Code for Bayesian Model

```

%%build structure 3mm case
rdoc3=struct('accelP',[],'accelM',[],'dyno',[],'power',[],'MTC',[]);

% set up accelP structure
rdoc3.accelP=struct('h',[],'d',[]);
rdoc3.accelP.h=struct('peaks',[],'mean',[],'std',[]);
rdoc3.accelP.d=struct('peaks',[],'mean',[],'std',[]);

% set up accelM structure
rdoc3.accelM=struct('h',[],'d',[]);
rdoc3.accelM.h=struct('peaks',[],'mean',[],'std',[]);
rdoc3.accelM.d=struct('peaks',[],'mean',[],'std',[]);
% set up dyno structure
rdoc3.dyno=struct('h',[],'d',[]);
rdoc3.dyno.h=struct('peaks',[],'mean',[],'std',[]);
rdoc3.dyno.d=struct('peaks',[],'mean',[],'std',[]);
% set up power structure
rdoc3.power=struct('h',[],'d',[]);

```

```

rdoc3.power.h=struct('avgsSpread', [], 'mean', [], 'std', []);
rdoc3.power.d=struct('avgsSpread', [], 'mean', [], 'std', []);
% set up MTConnect structure
%currently leaving empty since data is not conclusive

% end 3mm case

%% 4mm case

rdoc4=struct('accelP', [], 'accelM', [], 'dyno', [], 'power', [], 'MTC', []);

% set up accelP structure
rdoc4.accelP=struct('h', [], 'd', []);
rdoc4.accelP.h=struct('peaks', [], 'mean', [], 'std', []);
rdoc4.accelP.d=struct('peaks', [], 'mean', [], 'std', []);

% set up accelM structure
rdoc4.accelM=struct('h', [], 'd', []);
rdoc4.accelM.h=struct('peaks', [], 'mean', [], 'std', []);
rdoc4.accelM.d=struct('peaks', [], 'mean', [], 'std', []);
% set up dyno structure
rdoc4.dyno=struct('h', [], 'd', []);
rdoc4.dyno.h=struct('peaks', [], 'mean', [], 'std', []);
rdoc4.dyno.d=struct('peaks', [], 'mean', [], 'std', []);
% set up power structure
rdoc4.power=struct('h', [], 'd', []);
rdoc4.power.h=struct('avgsSpread', [], 'mean', [], 'std', []);
rdoc4.power.d=struct('avgsSpread', [], 'mean', [], 'std', []);
% set up MTConnect structure
%currently leaving empty since data is not conclusive

% end 4 mm case

%% 5mm case

rdoc5=struct('accelP', [], 'accelM', [], 'dyno', [], 'power', [], 'MTC', []);

% set up accelP structure
rdoc5.accelP=struct('h', [], 'd', []);
rdoc5.accelP.h=struct('peaks', [], 'mean', [], 'std', []);
rdoc5.accelP.d=struct('peaks', [], 'mean', [], 'std', []);

% set up accelM structure
rdoc5.accelM=struct('h', [], 'd', []);
rdoc5.accelM.h=struct('peaks', [], 'mean', [], 'std', []);
rdoc5.accelM.d=struct('peaks', [], 'mean', [], 'std', []);
% set up dyno structure
rdoc5.dyno=struct('h', [], 'd', []);
rdoc5.dyno.h=struct('peaks', [], 'mean', [], 'std', []);
rdoc5.dyno.d=struct('peaks', [], 'mean', [], 'std', []);
% set up power structure
rdoc5.power=struct('h', [], 'd', []);
rdoc5.power.h=struct('avgsSpread', [], 'mean', [], 'std', []);
rdoc5.power.d=struct('avgsSpread', [], 'mean', [], 'std', []);

```

```
% set up MTConnect structure
%currently leaving empty since data is not conclusive
% end 5mm case

save('E2_analysis_structs')
```

Figure 73: Code Used to Create Condition Database Structures in MATLAB for Experimental Case

REFERENCES

- A. Albarbar, A. B. (2009, June). Performance evaluation of MEMS accelerometers. *Measurement*(5), 790-795.
- AlexanderBillington, S. (1997). Sensor and machine condition effects in roller bearing diagnostics.
- Andreas Vogl, D. T. (2009, August 3). Design process and characterisation of a high-performance vibration sensor for wireless condition monitoring. *Sensors and Actuators A: Physical*, 153(2), 155-161.
- B.A. Paya, I. E. (1997, September). Artificial Neural Network Based Fault Diagnostics Of Rotating Machinery Using Wavelet Transforms As A Preprocessor. *Mechanical Systems and Signal Processing*, 11(5), 751-765. Retrieved from <http://dx.doi.org/10.1006/mssp.1997.0090>
- Betta, G., et al. (2001). A DSP-based FFT-analyzer for the fault diagnosis of rotating machine based on vibration analysis. *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE.*
- C. Sbarufatti, M. C. (2016, February). Sequential Monte-Carlo sampling based on a committee of artificial neural networks for likelihood state estimation and residual lifetime prediction. *International Journal of Fatigue*, 83, 10-23. Retrieved from <http://dx.doi.org/10.1016/j.ijfatigue.2015.05.017>
- Chaochao Chen, G. V. (2012, April). Machine remaining useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach. *Mechanical Systems and Signal Processing*, 28, 597-607. Retrieved from <http://dx.doi.org/10.1016/j.ymsp.2011.10.009>
- Cousineau D., & C. (2010). Outliers detection and treatment: A review. *International Journal of Psychological Research*, 58-67.
- Domingos, P. &. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 103-130.
- Drake, A. W. (1967). *Fundamentals of Applied Probability Theory*. McGraw-Hill, Inc.
- Fafa Chen, B. T. (2013, January). A novel fault diagnosis model for gearbox based on wavelet support vector machine with immune genetic algorithm. *Measurement*, 46(1), 220-232. Retrieved from <http://dx.doi.org/10.1016/j.measurement.2012.06.009>

- Farzaneh Ahmadzadeh, J. L. (2013, November). Remaining useful life prediction of grinding mill liners using an artificial neural network. *Minerals Engineering*, 53, 1-8.
- Francesco Di Maio, S. S.-L. (2011). Naïve Bayesian Classifier for On-line Remaining Useful Life Prediction of Degrading Bearings. *MMR2011*, 1-14.
- Friedman, N. G. (1997). Bayesian network classifiers. *Machine learning*, 131-163.
- Ganesan, A. P. (2012). Condition based maintenance: a survey. *Journal of Quality in Maintenance Engineering*, 18(4), 384 - 400.
- Gebraeel, N. Z. (2005). Residual-life distributions from component degradation signals: A Bayesian approach. *IIE Transactions* , 543-557.
- Hand, D. J., & Yu, K. (2001). Idiot's Bayes—not so stupid after all?. *International statistical review*, 69(3), 385-398.
- Hilden, J. (1984). Statistical diagnosis based on conditional independence does not require it. *Computers in biology and medicine*, 429-435.
- John, G. H. (1995). Estimating continuous distributions in Bayesian classifiers. *Eleventh conference on Uncertainty in artificial intelligence*, 338-345.
- Jaouher Ben Ali, N. F.-M. (2015, March). Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, 89, 16-27. Retrieved from <http://dx.doi.org/10.1016/j.apacoust.2014.08.016>
- Jun Ma, J. C. (1995, January). Detection of localised defects in rolling element bearings via composite hypothesis test. *Mechanical Systems and Signal Processing*, 9(1), 63-75. Retrieved from <http://dx.doi.org/10.1006/mssp.1995.0005>
- K.C. Gryllias, I. A. (2012, March). A Support Vector Machine approach based on physical model training for rolling element bearing fault detection in industrial environments. *Engineering Applications of Artificial Intelligence*, 25(2), 326-344. Retrieved from <http://dx.doi.org/10.1016/j.engappai.2011.09.010>
- Keogh, E. J. (1999, January). Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. *AISTats*.
- Khmais Bacha, S. S. (2012). Power transformer fault diagnosis based on dissolved gas analysis by support vector machine. *Electric Power Systems Research*, 83(1), 73-79. Retrieved from <http://dx.doi.org/10.1016/j.epr.2011.09.012>

- Kruse, R. (2008, October 24). Scholarpedia.org. Retrieved from Fuzzy Neural Network: http://www.scholarpedia.org/article/Fuzzy_neural_network
- Kunpeng Zhu, Y. S. (2009, June). Wavelet analysis of sensor signals for tool condition monitoring: A review and some new results. *International Journal of Machine Tools and Manufacture*, 49(7-8), 537-553.
- Langley, P. I. (1992). An analysis of Bayesian classifiers. *Aaai*, 90, 223-228.
- McFadden, P. D. and M. M. Toozhy (2000). "Application Of Synchronous Averaging To Vibration Monitoring Of Rolling Element Bearings." *Mechanical Systems and Signal Processing* 14(6): 891-906.
- Mehta, P. A. (2015). Condition based maintenance-systems integration and intelligence using Bayesian classification and sensor fusion. *Journal of Intelligent Manufacturing*, 331-346.
- M. Elangovan, K. R. (2010, March 15). Studies on Bayes classifier for condition monitoring of single point carbide tipped tool based on statistical and histogram features. *Expert Systems with Applications*, 2059-2069.
- Michael L. Fugate, H. S. (2001, July). Vibration-Based Damaged Detection Using Statistical Process Control. *Mechanical Systems and Signal Processing*, 15(4), 707-721. doi:<http://dx.doi.org/10.1006/mssp.2000.1323>
- Montgomery, D. C. (2009). *Engineering statistics*. John Wiley & Sons.
- Mosallam, A. K. (2014). Data-driven prognostic method based on Bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 1-12.
- N. Baydar, Q. C. (2001, March). Detection Of Incipient Tooth Defect In Helical Gears Using Multivariate Statistics. *Mechanical Systems And Signal Processing*, 15(2), 303-321. Retrieved from <http://dx.doi.org/10.1006/mssp.2000.1315>
- Nagi Z. Gebraeel, M. A. (2005). Residual-life distributions from component degradation signals: A Bayesian approach. *IIE Transactions*, 543-557. doi:10.1080/07408170590929018
- Psomas, N. (2003). *The Normal Distribution*. Retrieved from www.marin.edu
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann.

- Ramasso, E. a. (2014). Remaining useful life estimation by classification of predictions based on a neuro-fuzzy system and theory of belief functions. *Reliability, IEEE Transactions*, 555-566.
- Rish, I. (2001). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 3(22), 41-46.
- Rish, I., Hellerstein, J., & Thathachar, J. (2001). An analysis of data characteristics that affect naive Bayes performance. *IBM TJ Watson Research Center*, 30.
- Saha, B. a. (2008). Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques. *IEEE Aerospace Conference*. IEEE.
- S. Kavitha, R. J. (2016, January). High performance MEMS accelerometers for concrete SHM applications and comparison with COTS accelerometers. *Mechanical Systems and Signal Processing*, 410-424.
- Soualhi, A. e. (2014). Prognosis of bearing failures using hidden Markov models and the adaptive neuro-fuzzy inference system. *Industrial Electronics, IEEE Transactions*, 2864-2874.
- Thanagasundram, S. a. (2006). Comparison of integrated micro-electrical-mechanical system and piezoelectric accelerometers for machine condition monitoring. *Journal of Mechanical Engineering Science(Part C)*, 1135-1146.
- The Basic Statistical Foundation. (2002, April 17). Retrieved from <http://nitro.biosci.arizona.edu>
- Tobon-Mejia, D. A. (2012). CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. *Mechanical Systems and Signal Processing*, 167-182.
- Wang, W. (2012, May). A simulation-based multivariate Bayesian control chart for real time condition-based maintenance of complex systems. *European Journal of Operational Research*, 218(3), 726-734. Retrieved from <http://dx.doi.org/10.1016/j.ejor.2011.12.010>
- Wang, W. (2002). A model to predict the residual life of rolling element bearings given monitored condition information to date. *IMA Journal of Management Mathematics*, 13(1), 3-16.
- Zhang, H. (2004). The optimality of naïve Bayes. *American Association for Artificial Intelligence*, Retrieved from www.aaai.org