The Pennsylvania State University

The Graduate School

College of Engineering

FUZZY LOGIC, NEURAL NETWORKS AND STATISTICAL CLASSIFIERS FOR THE DETECTION AND CLASSIFICATION OF CONTROL VALVE BLOCKAGES

A Thesis in

Mechanical Engineering

by

Michael R. Roemer

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2017

The thesis of Michael R. Roemer was review and approved* by the following:

Karl M. Reichard
Assistant Professor of Acoustics
Thesis Co-Adviser

Christopher D. Rahn
Professor of Mechanical Engineering
Thesis Co-Adviser

Asok Ray
Distinguished Professor of Mechanical Engineering & Mathematics
Thesis Reader

Mary Frecker
Associate Head for Graduate Programs, Mechanical and Nuclear Engineering

* Signatures are on file in the Graduate School.

Abstract

Given recent advances in data processing, information technology and machine learning techniques, condition-based maintenance (CBM) has increasingly been used on high-cost assets as a way to improve product reliability and decrease long-term maintenance costs. While long established, CBM is often applied in a primitive manner using simple threshold monitoring and trending techniques. As such, there are opportunities for expanding this work using more intelligent methods which are now viable in practical settings as sensor and IT solutions decrease in cost. Mechanical systems worth monitoring include the control valve, which is used in a myriad of industries, including both commercial and military applications, and is particularly valuable in many settings. As an example, control valves are critical components in submarine subsystems, such as carbon removal units, which can require precise valve movements. In this and related applications, ensuring that movements are fully realized is crucial to the control process in question.

This thesis explores a variety of techniques for classifying valve movements with the goal of detecting two specific types of blockages in control valves. Objects with different material properties can produce vastly different dynamic responses in a control valve when blocking the valve stroke. Therefore, it is the goal of this work to detect two types of these blockages in order to cover a broad range of potential obstructive sources. To do this, seeded-fault testing is performed to simulate these behaviors with the resulting dynamic responses captured by both external sensor instrumentation and inherent control system outputs. Both data sources are used separately for analysis.

Classification using these data is tested using fuzzy inference systems (FIS's), artificial neural networks (ANN) and a variety of geometric and instance-based statistical classification methods. Discriminating features are identified, extracted and subsequently input to these classifiers to test for overall accuracy. Furthermore, multiple approaches to processing these data are used, including real-time monitoring simulations and post-hoc movement analysis. In most cases, accurate blockage detection and classification is achieved from these analyses. As a result, it is hoped that this work can serve as the foundation for future research in the area of fault detection and identification on control valves which govern flow loops in important process control applications.

Table of Contents

List of Figures

List of Tables

List of Equations

Acknowledgements

This thesis was based upon work supported by the NAVSEA Contract Number N00024-12-D-6404, under Delivery Orders 0246 and 0437. The content of the information does not necessarily reflect the position or policy of NAVSEA, and no official endorsement should be inferred.

I would like to also acknowledge those at Emerson Process Management and Fisher who provided the valves used for this study, as well as additional expertise in a timely manner for the duration of the research.

To my mother and father

Chapter 1. Introduction

Motivation

In the realm of mechanical systems and engineering, there has always been a focus on developing processes and equipment in an efficient manner, with the goals of both reducing cost and increasing overall ability to a reasonable point. Specifically, engineers strive to strike a balance between cost and performance in the research and development of the work they produce. However, these goals are not always pursued in the most effective manner once the work itself is finished. Mechanical systems deal with power and force and are thus subject to fatigue, degradation and, eventually, failure. Maintaining their performance has arguably been, in recent years, a higher priority for organizations who depend vitally on their operation. There are many reasons for this, but the foremost is recent advances in information technology and related methods of data acquisition and analysis that allow engineers and scientists to view the performance of these systems in new and innovative ways [2].

Maintenance has always been a critical component in the successful operation of mechanical systems, and as such has continued to evolve in nature. In the traditional sense, maintenance takes place only once a system has clearly failed and is not working properly. Workers locate the fault manually, sometimes with the use of primitive data, and then perform maintenance actions accordingly. While simple in nature, this process relies almost entirely on human-based judgements and therefore can lead to critical errors in the repair mechanisms or even the diagnoses that lead to them. Furthermore, it can take a considerable amount of time to make the relevant diagnoses and repairs necessary to bring the system back online for operation. Other related “traditional” methods include constant-interval maintenance and age-based maintenance which, as their names suggest, involve the replacement or repair of certain components at a specific age or interval [3]. While this can help prevent potentially disastrous failures, it could still require significant, albeit scheduled, downtime. Ideally, the onset of a fault can be predicted before it actually takes place or detected immediately so that downtime can be reduced or eliminated and the associated costs can be recovered. This desirable scenario has led to the development of analysis techniques and other general processes with the goal of achieving it.

Shin et al. describes the distinct classification of three maintenance policies: breakdown maintenance, preventive maintenance and condition-based maintenance (CBM) [1]. Breakdown maintenance, as described above, can roughly be defined as taking corrective action after a failure has occurred in some mechanical asset, while preventive and condition-based maintenance policies are focused on the use of quantitative variables to describe the health state of some system. As noted by Kothamasu et al., these variables and monitoring algorithms will often be tailored to not only a specific system but specific faults as well. He defines CBM as a decision-making strategy where “the decision to perform maintenance is reached by observing the ‘condition’ of the system and/or its components” [3]. For the purposes of the work described in this thesis, this is a suitable definition.

CBM provides some very clear advantages over other types of maintenance that are immediately obvious. For one, it can provide warnings of critical failures before they take place if implemented correctly. Shin et al. note that the continuous advancement of available technological capabilities has enabled organizations to gather and use data in ways which make it possible to pinpoint the status of a product or asset more exactly [1]. Kothamasu similarly notes that CBM is helpful in diagnosing the failures of specific components through the monitored variables. Over time, as sensor and IT solutions decrease in cost while products become continuously more complex, CBM solutions can save money in maintenance budgets and have been reported to do so [4]. The primary disadvantage of CBM implementation is that it requires a period of relatively rigorous research and development to create algorithms and fit the solutions to specific systems, which comes at a large initial cost. A mass shift towards a CBM policy can also lead to a complete organizational overhaul of the maintenance structure and budget for an enterprise.

Nevertheless, the benefits are clear and explain why CBM continues to be pursued by those who employ high-cost assets. The industrial, manufacturing and military sectors alike use this policy to maintain a level of confidence in product reliability. While different in nature, both a complex piece of manufacturing machinery and a hard-to-reach submarine component can benefit from CBM technology. This versatility enabled by CBM in general is, in turn, the leading motivation for this work.

Background

Those in the manufacturing, industrial and oil and gas industries, along with nuclear and others, are generally familiar with control valves because of their pervasiveness in process plants. These plants, as noted by Fisher's *Control Valve Handbook*, consist of hundreds or thousands of control loops that are often interconnected in some manner to control pressure, flow or other related process variables. In order to compensate for load disturbance and, more importantly, maintain a set point for a given process variable, control valves are used in scores throughout industrial plants, pipelines and other applications where control of flow is necessary [5].

Given the widespread use of control valves and their importance as key components in process control, they are well suited for CBM research. The breakdown of a single control valve could jeopardize a much larger process, and there are certain parameters that can help indicate if something has gone wrong. For example, valves leave a well-defined signature during a movement when comparing flow and travel components as shown below in Figure 1.1 [5].

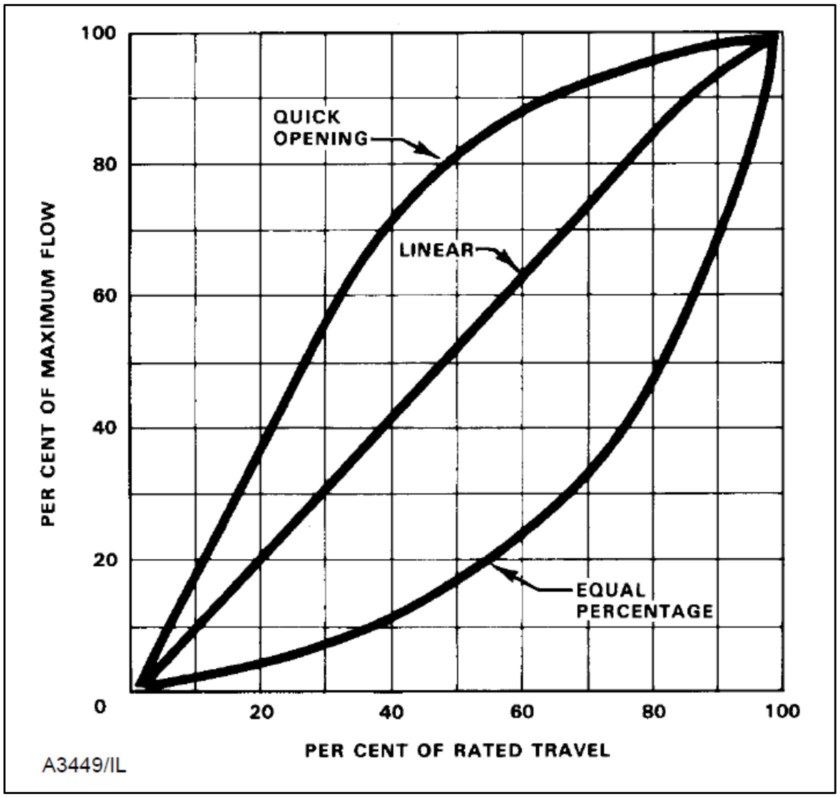


Figure 1.1: Visualization of the inherent characteristics of some control valves [5].

In most control valves, there is a range over which the input signal to the controller can be varied without any movement taking place. There can also be differences in the output value, or flow, of a valve from cycle to cycle. These two phenomena are referred to as dead band and hysteresis, respectively, and are shown below in Figure 1.2 [6].

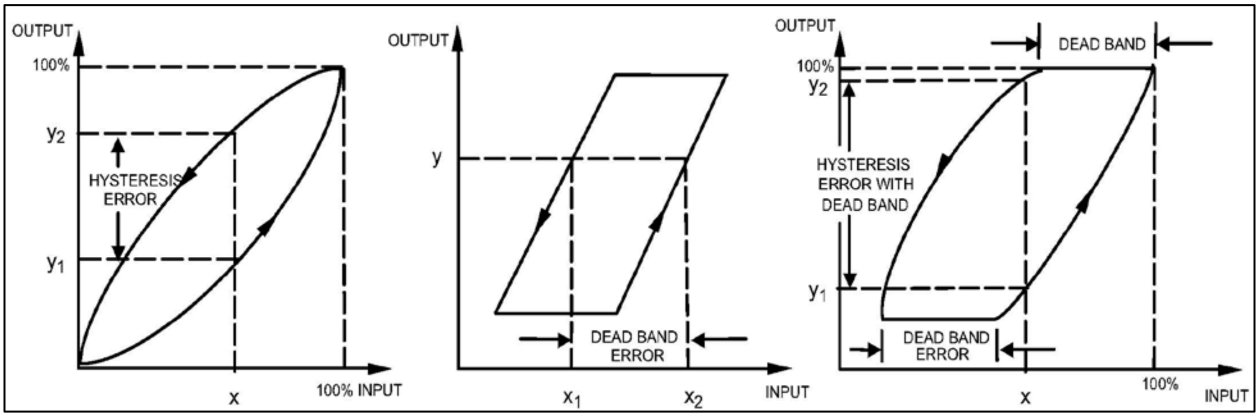


Figure 1.2: Hysteresis and dead band in control valves [6].

Ideally, these phenomena are minimized or kept within an acceptable range so that the process variable in question does not vary significantly from cycle to cycle. Organizations such as Emerson Process Management have marketed digital valve controllers (DVCs) with the capability of monitoring these variables and indicating to operators if the valve in question is falling outside of the acceptable range of performance. The software used can also monitor the values of certain parameters, such as supply pressure, and create notifications if there are discrepancies from an expected level, as well as trend the values over time. Other more sophisticated research has been performed to use this type of dynamic response data and its statistical properties for monitoring control valves [12].

This sort of condition monitoring is helpful and certainly more effective for maintenance purposes than breakdown maintenance policies. However, it fails to address the actual cause of the process variable differences or the severity of the issue, i.e. whether it is an impending catastrophic fault, a fully-realized minor one, or even an incorrect calibration. In addition, this approach leaves engineers vulnerable to a lack of understanding with regard to specific failure modes and what may cause them.

Indeed, research has been done into more advanced methods of fault diagnosis in control valves. Simani and others have referred to three basic model-based fault diagnosis methods using the following: output observers, parity equations and parameter estimation techniques [7]. Karpenko et al. note that the latter two methods, particularly state estimation techniques, require mathematical models of the systems in question. Therefore, they are generally restricted to linear systems and not as useful for dynamic systems such as control valves that exhibit nonlinear behavior [8]. As Simani notes, often only output signals are able to be measured through the use of sensors and analyzed through methods such as spectral analysis or mean and variance estimation. From there, classification techniques such as statistical methods, artificial neural networks (ANN) and fuzzy inference systems (FIS) or fuzzy clustering are used to perform the actual fault diagnosis using these data as inputs [7].

Much of the existing literature surrounding CBM on control valves involves the use of ANNs to take input data and classify it accordingly. Subbaraj et al. used a neural network-based approach to detect and classify an array of 19 different faults on a pneumatic valve in a cooler water spray system, four of which took place on or inside the valve itself and the others corresponding to external, motor or positioner faults [13]. Sundarmahesh et al. expanded on this work with a more detailed discussion of the algorithm used as well as the use of principal components analysis (PCA) to aid in dimension-reduction [14]. McGhee et al. also used an ANN approach to fault diagnosis and identification (FDI) for control valves and considers faults related to leakage due to corrosion and wear in seals and seats, blockage and sticking due to debris build-up and poor maintenance, and other failures related to the spring and diaphragm [15]. Karpenko uses an ANN with process variable data from the DVC to assess valve performance [8], allowing for the identification of faults related to the air supply, input air vents and leakage from the actuator diaphragm.

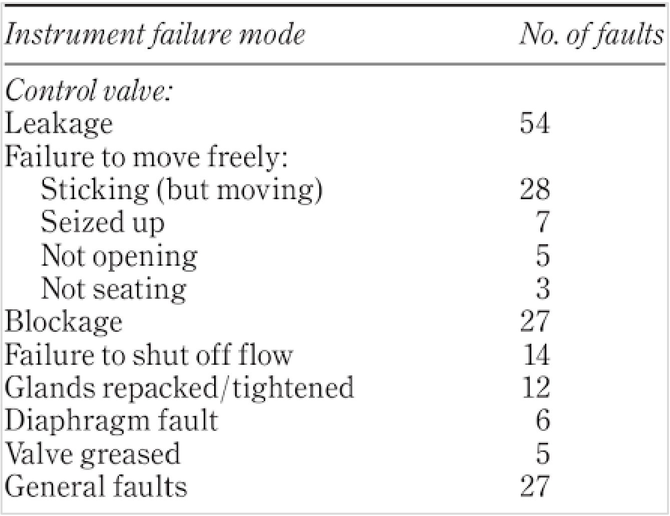
The use of fuzzy models and fuzzy logic is also prevalent in work related to fault diagnosis for process control valves. Fuzzy capabilities have been added to neural network models to create a "neuro-fuzzy" model with the goal of improving the interpretability of results [16]. Bocaniala et al. introduce a fuzzy classifier as a way of performing FDI on control valve actuators while Mendonca et al. used a similar analytic approach to fault detection but using parameter estimation techniques to predict outputs, which are then input to fuzzy models for each defined fault [17, 18]. Carneiro and Porto use arguably the most straightforward approach by taking basic DVC data and using a fuzzy model to assess valve stroke performance, similar to [8], enabling the diagnostics to improve beyond that of those provided by some major suppliers as noted above [19].

In addition to the work described above, CBM efforts on control valves and related systems have been proposed using statistical classification algorithms for the detection and classification of faults. Nair et al. describe a feature selection method and subsequent fault detection using the simple naïve Bayes classifier [11], and a similar approach was applied to an air-handling unit by House et al. [10]. Support vector machines (SVMs) have been shown to be effective in the detection and classification of faults on control valves in particular [20, 21]. Perhaps most relevant, at least in process, is the recent work performed by Adams et al., who rigged a hydraulic actuator with sensors and used the resulting data streams to record faults and test a variety of classifiers to determine the best performers in terms of accuracy and power usage [9]. This work in particular shows how data can be gathered and intuitive features extracted for input into relatively simple statistical classifiers for FDI purposes.

Research Question

All things considered, it is clear that process control valves are not only instinctively suited for CBM research, but have a demonstrated responsiveness to it with room for more investigation. This is particularly important to first uncover given previous suggestions that close to 30% of industrial equipment is not particularly receptive to CBM [1]. Results from the studies in the previous subsection combine to lay a solid foundation from which more testing and analysis in multiple different directions is possible. Some of these works, such as those in [9] and [20], rely on simulated data using software models, while most of the works focus on one type of analysis approach. Furthermore, the “faults” analyzed in each of the studies are scattered with little overlap. This leaves room for exploration into different types of faults entirely or different ways that these faults may present themselves. Table 1.1 below, taken from *Lee’s Loss Prevention in the Process Industries* [22], gives a broad survey of faults seen in control valves. Clearly, there are a wide variety of scenarios to be considered.

Table 1.1: Some failure modes for a control valve [22].



Instrument failure mode	*No. of faults*
Control valve:	
Leakage	54
Failure to move freely:	
Sticking (but moving)	28
Seized up	7
Not opening	5
Not seating	3
Blockage	27
Failure to shut off flow	14
Glands repacked/tightened	12
Diaphragm fault	6
Valve greased	5
General faults	27

One particular mode of interest, covered in some of the works above but as part of wider studies and without significant detail, is the *blockage* of a control valve. Table 1.1 shows that blockages accounted for a meaningful amount of the failures seen in the survey of control valves. However, thinking intuitively about the effects of a potential blockage, the issue could introduce many more of the modes given above. Leakage, for example, is arguably the most significant concern with respect to process control. Given that blockages will lead to leakage in a particular valve, combined with the no-flow conditions under which testing will be performed, blockages are thought of as a surrogate of sorts for detecting leakage in this case.

For the purposes of this work, we will define a valve *blockage* as some object or mechanism that prevents the valve from properly completing an expected movement. Considering a blockage under this definition, many of the faults included in Table 1.1 can be regrouped. A blockage that prevents a valve from fully closing, for example, would likely lead to the valve not seating correctly, leakage occurring and flow not being completely shut off, to name a few. Revisiting the table, over 70% of the failure modes are directly referred to or could directly result from a valve blockage. Also, while a slight “blockage” may not be particularly important to a large valve in, say, an oil and gas pipeline, it can be very critical in other applications such as carbon removal units and air purification components. In this case, even a slight leak caused by a small blockage could contaminate the process that, for example, allows submariners or astronauts to breath for extended periods of time.

As a result, the work presented in this thesis concerns the identification and subsequent classification of blockages in multiple types of control valves using a variety of approaches. By comparing the “black box” nature of the artificial neural network with the more easily interpretable fuzzy inference system and statistical classification methods, the three main sources of sensor-based fault diagnosis techniques will be explored and compared. Given the extensive work previously done with ANNs and the inherent limitations of a fuzzy logic structure that arise when certain relationships aren’t clear, statistical classification methods will end up as the primary focus.

This research intends to shed light on a few key questions regarding CBM for control valves. The following are some examples.

- Are there different types of blockages?
- Are blocked movements distinguishable from baseline movements?
- What external sensors can be used to detect them?
- What are the key features to focus on in the data?
- What kind of accuracy is achievable?
- Are the methods used transferable to valves of a different size or with a different controlling mechanism?

The objective is not only to demonstrate how this failure mode can be identified but to make the case that the methods used, particularly the statistical classifiers previously mentioned, can be effective in the fault detection of control valves and in CBM as a whole.

Chapter 2. Analysis Techniques

Introduction

The purpose of this chapter is to provide an overview of the methods used for the analysis of data collected for this work. As described in the previous chapter, the use of sensors to gather output signals from the mechanical system of interest is often the only feasible way to perform fault diagnosis in practical applications. This research details the use of sensors to gather data though baseline and seeded-fault testing and subsequent feature extraction methods for input into various classifiers. As a result, the classifiers themselves will be described in more detail to provide full context.

There are three primary classification techniques used for blockage detection and classification in the control valves used in this study. They are fuzzy logic, artificial neural networks and statistical classifiers, and all three are explored to some degree. A variety of statistical and other instance-based classifiers are available through MATLAB's *Classification Learner* application, part of the Statistics Toolbox, but only the handful that showed promising levels of classification accuracy are described. These are *discriminant analysis*, *k-nearest-neighbor, support vector machines*, and *ensemble* classifiers. The three techniques vary in terms of simplicity, performance, interpretability and accuracy. Therefore, they will be compared qualitatively in this manner to highlight key differences.

Fuzzy Logic Structures

To fully understand fuzzy inference systems and their applicability to CBM research, it is particularly important to first understand the concept of *crisp logic*. Crisp logic is something that most engineers and scientists have knowledge of, if not under that specific title. It is equivalent to the concept of Boolean logic, i.e. a rule-based logic structure which takes an input and classifies the associated rule as true or false.

Crisp logic is intuitive and easy to implement at a basic level with a small amount of rules. However, its complexity increases exponentially as inputs are added and rules become more complex to handle them. This makes it difficult to manually code and can quickly lead to classification errors if the entire space of rules is not deeply understood by the coder.

Furthermore, given the nature of this type of logic, discrete boundaries are required between classes. In other words, a set of observations belonging to a certain class must be clearly defined in the *n*-dimensional feature space in order to be suitable for crisp logic classification. As these clusters of observational data spread out, mix and become less distinct, classification error rates begin to rise and get quickly out of hand. A classic example is shown below in Figure 2.1, where two variable measurements on iris flower types explain three classes of flower.

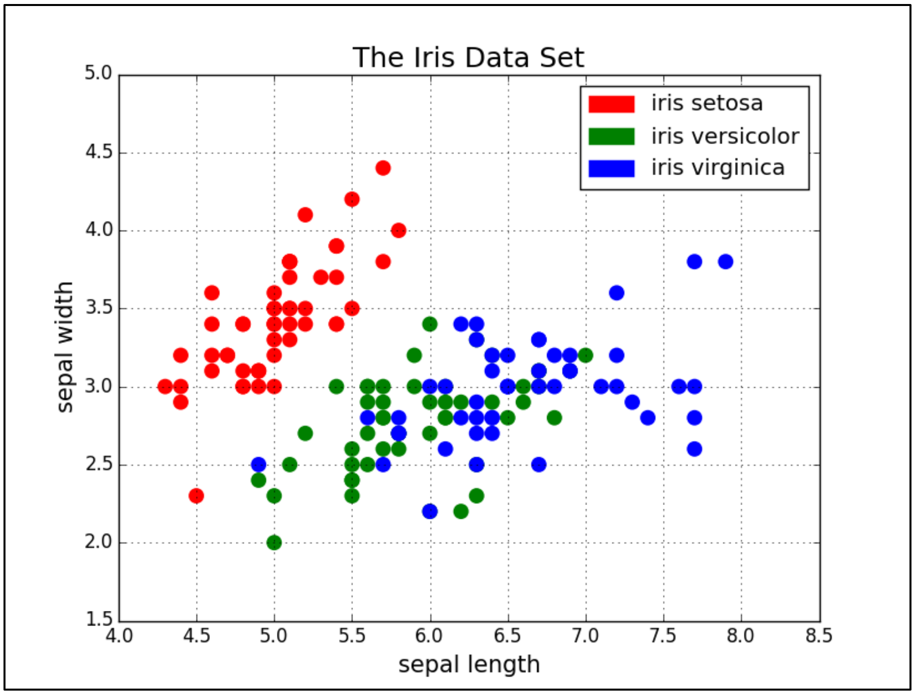


Figure 2.1: Data which may not be suited for crisp logic classification methods [27].

Given these two predictor variables and the scatterplot they produce, this data will probably not benefit from a crisp logic structure. In particular, the blending between the *versicolor* and *virginica* variables make it difficult to define a distinct set of rules for classifying these three flower types. It should be noted, however, that the addition of more predictor variables could help in creating a scenario that is more subject to classification using this method.

When working with datasets where variable values fall on a more continuous spectrum and classes are not as clearly defined, fuzzy logic emerges as a more suitable option for many reasons. The first is that fuzzy logic structures, by nature, are more flexible to imprecise data inputs and can deal with uncertainty more effectively than crisp logic systems. With respect to fault detection, there are often different levels of severity associated with specific faults. In these cases, fuzzy logic intuitively prevails because of its ability to define classifications along a continuum of class membership [28].

The primary concept tied to the use of fuzzy inference systems is that of the *membership function*. As described by Zadeh, membership functions associate a vector of inputs with a continuous output along a [0, 1] interval as opposed to crisp logic structures which output either 0 or 1 with no in-between [28]. This is useful because it allows the system to express more linguistic concepts such as “hot” versus “cold”, or “large” versus “small” by varying the degrees of membership mathematically [29].

Membership functions can vary in shape, with common types including triangular, trapezoidal, s-shape and Gaussian. MATLAB’s *Fuzzy Logic Toolbox*, which aided in the creation of the fuzzy inference systems used in this research, enables the use of these and others. Equation 2.1 below describes how a Gaussian membership function can be created by specifying statistical parameters (mean and standard deviation) to define its exact shape.

$$f(x;\ \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \qquad [2.1]$$

By specifying the parameters associated with the membership function(s) in question, one can create a relatively detailed structure covering a range of potential inputs for a particular variable. A generic example is shown in Figure 2.2 below.

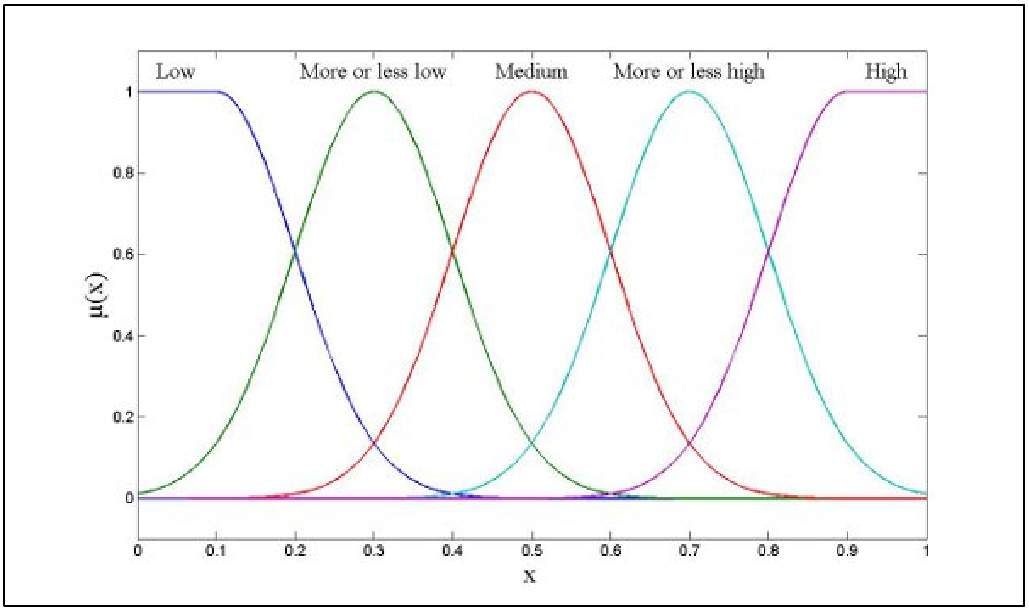


Figure 2.2: Use of Gaussian membership functions to provide a smooth surface for classifying a particular input [30].

The flexibility attained by defining this sort of *control surface*, as referred to by Hameed et al. [30], allows researchers to develop logic structures to deal with parameter uncertainty while simultaneously being able to define a clear set of rules for classification. Tools made available by developers such as MATLAB allow for the easy development of an FIS, both the control surfaces and corresponding rules, through interactive interfaces and simple functions.

Fuzzy sets such as the one shown in Figure 2.2 can be made for each variable of interest and combined to create an *n*-dimensional surface that represents the entire set of *n* features being used as inputs for classification. This allows a vector of crisp values to be input to the system and "fuzzified" according to the membership functions. These fuzzified values are then input into the fuzzy logic structure before being "de-fuzzified" into a crisp output. This configuration is summarized in Figure 2.3 below [31].

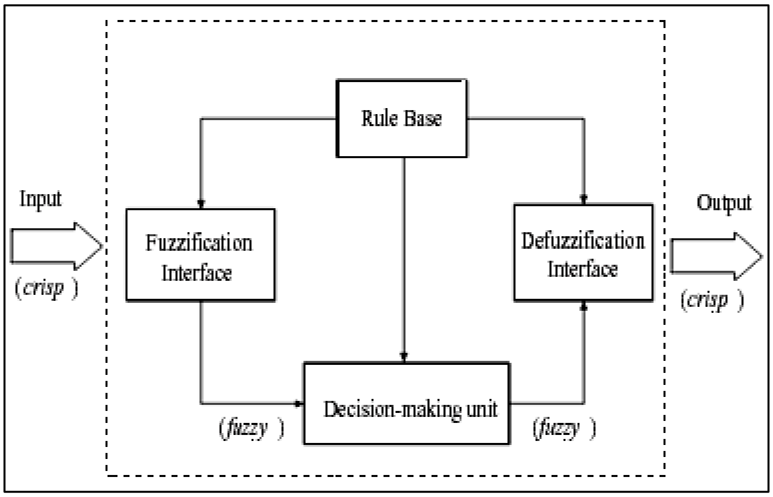


Figure 2.3: Fuzzy logic system configuration taken from de Abreu et al [31].

Overall fuzzy logic structures are not very dissimilar to crisp logic structures, but allow room for uncertainty in the inputs. While the advantages to this type of classification system have been described, mainly as compared to crisp logic, there are inherent disadvantages. For one, increasing the number of inputs quickly complicates the system and the number of rules involved. Furthermore, a single system is generally only helpful for describing the level of membership of one output or, for the purposes of this discussion, fault. Therefore separate systems are usually needed for describing different types of behavior. While this is expected from typical logic-based classification, the following classifiers discussed in this chapter provide additional flexibility in this domain.

Artificial Neural Networks

The artificial neural network (ANN) is a powerful tool that, while applicable in a myriad of fields, has been increasingly used in CBM research for fault diagnosis in past years. As described in the previous chapter, existing research includes work on control valves [13 – 16]. Their relevance to this work in particular concerns their ability to perform pattern recognition, not just for images and alpha-numeric characters but for classifying vectors of input data as well. The connectionist nature of the parallel-node structure of neural networks allows for enormous flexibility in application [32]. In other words, one is not bound by linearity, distribution types or other assumptions typically associated with probabilistic classifiers.

The most basic visual representation of an arbitrary feedforward ANN is shown below in Figure 2.4 [33]. At the highest level, inputs enter a *hidden layer* of nodes where weights are applied and bias is added. In supervised learning methods, common for CBM applications, a training set of data is supplied to the network where the output classifications are known. As the network trains, the weights within each node adjust. A simple example, provided by Shiffman [32], explains how the error from each prediction can subsequently be used to adjust the weight on a given node during training. Typically, the network will train until it converges on an error rate or a maximum number of training cycles (or *epochs*) has been reached.

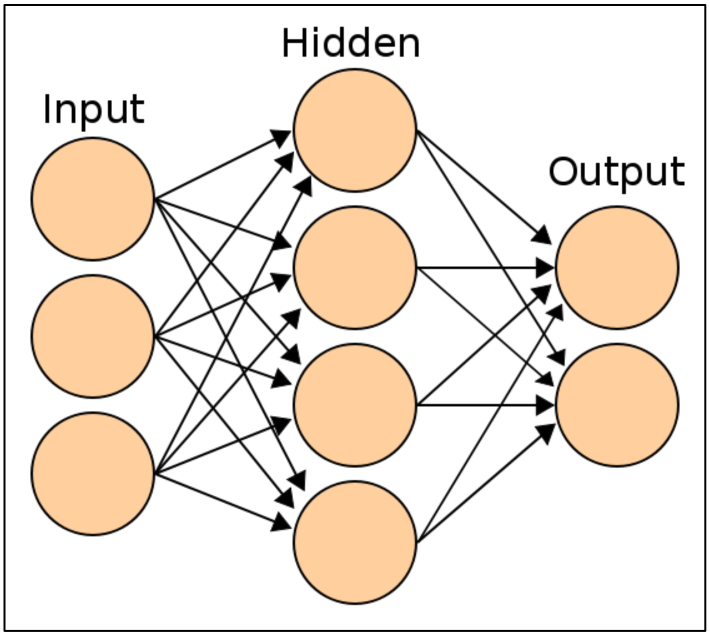


Figure 2.4: Neural network with a single hidden layer [33].

The hidden layer(s) of a neural network are essentially the backbone of the structure where most of the computation takes place. Each node in the hidden layer will take a linear combination of the inputs supplied and apply an activation function with the purpose of transforming it to some output. The activation function is usually one that is nonlinear, real-valued, continuously differentiable and relatively simple such as the sigmoid function shown in Equation 2.2 below [26, 36].

$$sig(x) = \frac{1}{1+e^{-x}} \qquad [2.2]$$

The adaptive nature of ANNs through the training process allows them to effectively "learn" from their mistakes, given a supervised learning strategy. Widespread research has been performed concerning the development of fast and efficient training algorithms [37], many utilizing the backpropagation algorithm to calculate cost function gradients. These training algorithms vary in the mechanics used to adjust weights and biases [38]. A common training method is the scaled conjugate gradient (SCG) algorithm, which removes the need for user-defined parameters that can affect convergence [34]. This is the algorithm used to train pattern recognition ANNs described in Chapter 5.

Many neural networks are designed to continuously adapt to new data points. However, for the purposes of this research and CBM in general, it makes more sense to train an ANN on a representative training set of data and use the resulting network with stored training patterns for future inputs. Neural networks of identical structure will train differently even with the same training set since initial conditions (i.e. weights and biases) will be reinitialized and changed. Thus it is often useful to train them many times until a suitable level of performance is achieved.

The foundational mathematics behind ANNs and training algorithms have been extensively described [34 – 37], and thus will not be explored in great detail, as the focus of this work is on their application to CBM research for control valves. In fact, given the complexity of these mathematics, combined with the inherent difficulty in understanding the way in which ANNs classify inputs, neural networks are often referred to as "black boxes" [24, 39]. So while ANNs have been shown to be effective in modeling complex nonlinear behavior and making decisions not easily made by standard computers, this gap of understanding can lead to misinterpretation or overtraining.

Statistical Classifiers

In the area of machine learning, classifiers can be grouped into different subsets that describe them more distinctly, i.e. probabilistic vs. non-probabilistic or parametric vs. non-parametric. Through the use of MATLAB's *Classification Learner Toolbox*, classifiers of different types are available and were used for this work. The broadest term that can describe these classifiers, less general than *machine learning*, is *statistical classifier*, and therefore will be used to reference these methods as a whole. The reason this is suitable is that a classifier can be non-Bayesian, non-probabilistic and non-parametric (SVMs, for example) and still use statistical inference or geometric distance calculations in its general procedure. There is literature that makes a distinction between statistical and geometric classifiers or instance-based learners [26, 53], but we will treat them as one and the same for simplicity.

As mentioned at the beginning of the chapter, the 4 general statistical classification methods that are considered in this work are *discriminant analysis*, *k-nearest-neighbor* (kNN), *support vector machines*, and *ensemble* classifiers. Discriminant analysis is a probabilistic classification method derived from Bayes' Theorem, while kNN and SVM methods use distance calculations to create separating hyperplanes or boundaries. Ensemble classifiers combine multiple classifier algorithms in an attempt to reach higher levels of classification accuracy, often at the cost of computational memory.

Equations 2.3 and 2.4 below provide some groundwork for these methods. Equation 2.3 is a modification of Bayes' Rule which states, in words, that the probability of membership to group *i* (designated by π_i), given the input **x**, is equal to the probability that an observation belongs to group *i* **and** has the value **x**, divided by the total likelihood that we observe **x**. This formula provides the basis for discriminant analysis.

$$p(\pi_i \mid \boldsymbol{x}) = \frac{p_i f(\boldsymbol{x} \mid \pi_i)}{\sum_{j=1}^{g} p_j f(\boldsymbol{x} \mid \pi_j)} \qquad [2.3]$$

Equation 2.4 is a generalized, or Mahalanobis, distance between two vectors which adjusts for variance through the addition of the variance-covariance matrix *S*. The Mahalanobis distance is a common measurement used in classifiers which utilize distance for classification purposes.

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \qquad [2.4]$$

Discriminant Analysis

The discriminant analysis procedure is a probabilistic classification method which involves the development of “score functions” derived from Bayesian probability equations. One of the key restrictions with this method comes from the assumption of a multivariate normal distribution among predictor variable data. In general, discriminant analysis for classification is robust to relatively small deviations from normality. However, there is no guarantee of normality when taking data and extracting certain features for classification purposes, so it is worth monitoring the distribution before continuing with classification using this strategy.

The general algorithm for discriminant analysis is as follows. Functions of the predictor variables, the score functions, are calculated for each class using a training data set to estimate certain parameters such as mean vectors and covariance matrices. Subsequently, new data can be input into these functions for classification. In general, the class with the highest score function value will be the one in which the new data point will be placed. The values of these scores can be further used to estimate misclassification probabilities.

The theoretical basis for discriminant analysis comes from Bayes’ Rule, a modification of which is shown in Equation 2.3, which poses conditional probabilities calculated using probability density functions (PDFs). Equation 2.5 below gives the PDF of a multivariate vector $\vec{\boldsymbol{x}}$ with a Gaussian distribution for population π_i.

$$f(\vec{\boldsymbol{x}}\,|\pi_i) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp[-\tfrac{1}{2}(\vec{\boldsymbol{x}} - \vec{\mu_i})'\Sigma_i{}^{-1}(\vec{\boldsymbol{x}} - \vec{\mu_i})] \qquad [2.5]$$

This formula identifies the population covariance matrix to be Σ with a population mean vector $\vec{\mu_i}$. There are p total classes to be classified into. Using Equations 2.3 and 2.5, we can simplify to the general form of a score function [44], summarized in Equation 2.6 for a single class i.

$$Q_i = (\vec{\boldsymbol{x}} - \vec{\mu_i})'\Sigma_i{}^{-1}(\vec{\boldsymbol{x}} - \vec{\mu_i}) + \log|\Sigma_i| - 2\log\pi_i \qquad [2.6]$$

As it stands, Equation 2.6 expands to a score function which is quadratic in $\vec{\boldsymbol{x}}$, hence the label Q for the equation. In this general form of the scores, the method is known as quadratic discriminant analysis. However, this form allows for varying covariance matrices for each class, giving the i designation for each Σ. A common additional assumption is that each class has approximately equivalent covariance matrices, or homoscedasticity among classes. With this in mind we can further simplify to Equation 2.7.

$$L_i = (\vec{\boldsymbol{x}}'\Sigma^{-1}\overrightarrow{\mu_i}) - \frac{1}{2}(\overrightarrow{\mu_i}'\Sigma^{-1}\overrightarrow{\mu_i}) + \log \pi_i \qquad [2.7]$$

This gives a linear score function in $\vec{\boldsymbol{x}}$, and is known as linear discriminant analysis. In general, while the quadratic analysis is able to create more sophisticated boundaries between classes which can improve accuracy, it requires more statistical parameters to be estimated as a result of the relaxed homoscedasticity assumption [44, 45].

After the score functions have been defined for each class, they can then be used for classification and estimating misclassification probabilities as explained previously. Graphically, the difference between the two methods, linear and quadratic, is demonstrated in Figure 2.5 below. Generally, an assumption of equal probability of membership to each class is made, though this assumption can be changed and adjusted for expected probabilities accordingly.

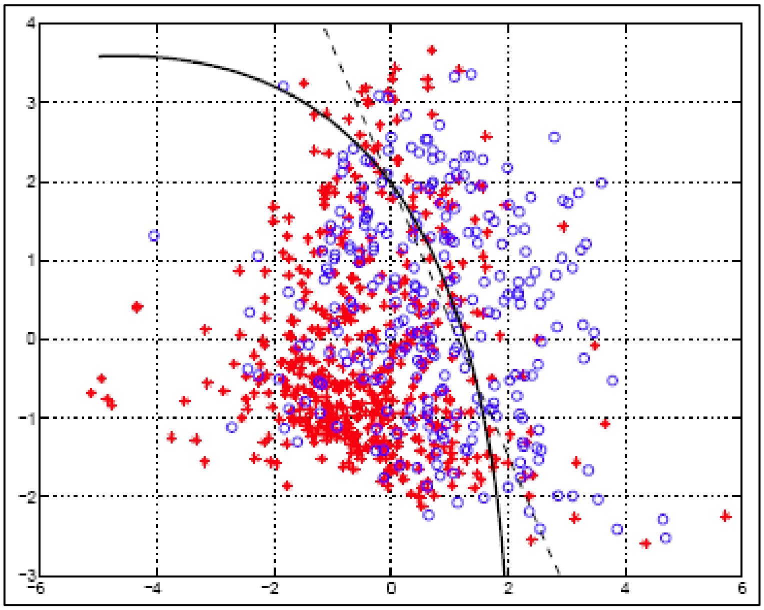


Figure 2.5: Comparing linear and quadratic discriminant analysis [46].

Given the advantages and disadvantages between the two methods, there are scenarios that where each one prevails over the other. As a result, both are tested in this work.

K – Nearest – Neighbor

The kNN classification problem is commonly known as one of the simplest machine learning applications and involves the classification of inputs based on the closest known value in the feature space [40, 41]. Therefore, as with the other classifiers discussed in this section, it is necessary to have access to a quality training data set that is as representative of the population of features as possible. For the purposes of CBM, the kNN algorithm is, at the very least, a very attractive starting point in fault classification because it requires no prior knowledge of the distribution of the data used [41]. This enables the process of data acquisition, feature extraction and subsequent classification to take place in a relatively streamlined manner, since little additional study or analysis is needed before training classifiers.

Given quality training data, kNN is quite simple and computationally efficient even for larger training sets. In general, the algorithm first calls for creating a distance matrix which stores the distance between all combinations of samples. As mentioned previously, there are a variety of distance measures that have been defined for statistical purposes. Peterson notes that the Euclidean distance, displayed in Equation 2.5 below, is often used for the calculation of this distance matrix [40].

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T(\vec{x}-\vec{y})} \qquad [2.8]$$

Note that this formula is identical to that of the Mahalanobis distance given in Equation 2.4, sans the inclusion of the variance-covariance matrix S. When information regarding the distribution of the data is not known, it is difficult to make assumptions about the use of certain hypothesis tests in statistical theory, such as Bartlett’s test for equal covariance matrices. As such, distance measures that involve these sorts of assumptions are avoided to maintain validity with respect to the chosen statistical methods. Euclidean distance is the default distance metric used in MATLAB-based kNN classifiers and is used in this work.

The feature data to be used in training and future testing of the classifiers may be on remarkably different scales, leaving room for potential bias. For example, if frequency magnitude data from an FFT is used and on the order of one one-thousandth along with time data on the order of 4-7 seconds, the data of higher magnitude (time data in this case) could have greater influence. The standardization of variables through calculation of the z-score is commonly used to alleviate this issue. This simple calculation is shown in Equation 2.9 below, where *i* is the sample in question and *j* is the feature it belongs to. In this case, *z* would replace *x* in Equation 2.8.

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \qquad [2.9]$$

Once the distance metric is defined and any necessary feature transformations have been performed, the distance matrix can be quickly calculated and used for classification. The “k” in k-nearest-neighbor stands for the number of neighbors that the test sample is to be compared to. In other words, once the distance matrix is calculated including the test sample in question, the *k* training points of closest proximity are collected. The test point is then classified into whichever class represents a majority of these points. A simple example is illustrated in Figure 2.6 below.

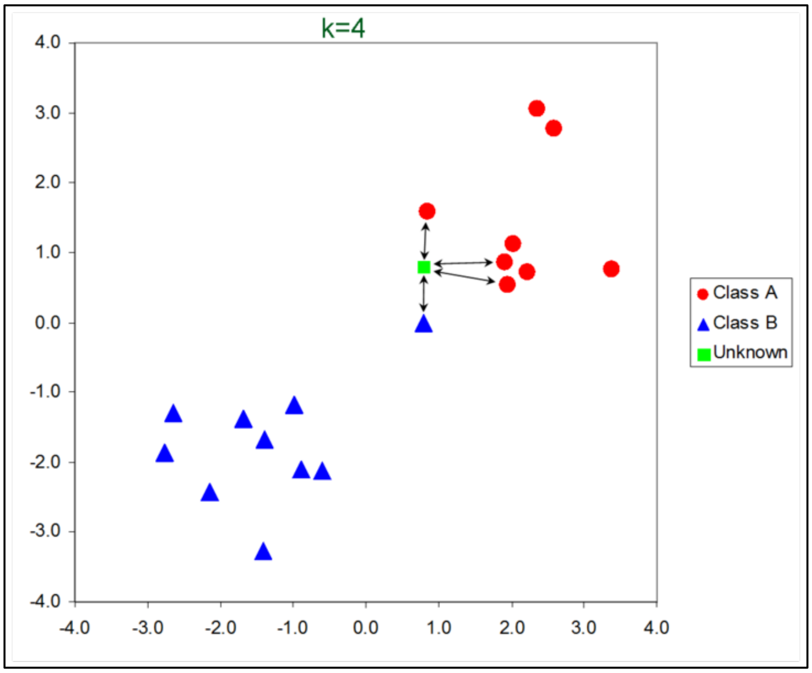


Figure 2.6: Visual example of kNN classification [40].

Since *k* is equal to 4 in this case, and Class A represents 3 neighbors compared to just 1 for Class B, the test point is classified into Class A. In an effort to avoid ties, an odd value is often chosen for *k*.

Most variations of the kNN classification strategy involve changing the distance metric used for calculations. Some have proposed the use of a kernel function to increase computational speeds [42]. Others have described the use of weighting functions to give more influence to closer data points, i.e. the distance-weighted kNN rule [43]. A variety of methods are tested in this research.

Support Vector Machines

The concept of support vector machines is a machine learning technique that is more mathematically complex than the kNN or discriminant-based classifiers discussed previously. The idea is simple enough; we aim to construct some hyperplane that separates the classes of training data as cleanly as possible before applying this boundary to new testing data. This is done by taking each class and maximizing the *margin* between them with this hyperplane and associated *support vectors* [49].

This process is relatively easy when the classes in question have distinct boundaries between them. When this is the case, linear SVM classifiers solve the problem of optimizing the separating boundary in order to minimize future classification error. An optimal hyperplane is found and the training observations that lie on the boundary that maximizes the margin between classes designate the location of the support vectors [47]. This basic case is illustrated in Figure 2.7 below.

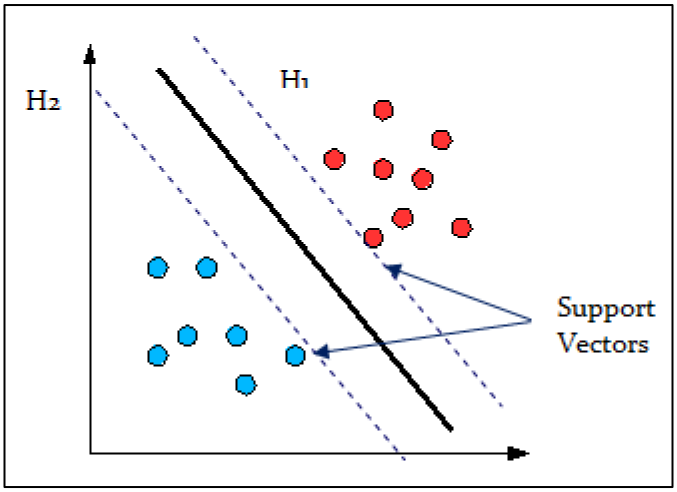


Figure 2.7: Linear SVM visualization [47].

However, many realistic scenarios yield data that is not easily separated by linear hyperplanes. Often times the data may be mixed, as in Figures 2.1 and 2.5, and it is impossible to define this plane. When this is the case, the SVM algorithm attempts to instead minimize the number of observations that lie on the incorrect side of the proposed hyperplane. As a result, the margin created by the support vectors in this scenario is commonly referred to as a *soft margin* because of the points that violate the boundary. In these types of mixed data, the violating points are often also used as support vectors [48].

While linear SVMs can be used in mixed data samples, there exist situations which do not lend themselves to linear separation boundaries. When this is the case, nonlinear transformations are used to develop boundaries that can classify more accurately than a simple linear hyperplane. To aid in this process, the kernel trick is used to expand the feature space to a larger dimension where it is possible to separate the data in a linear fashion [47, 48, and 50]. A variety of kernels exist for SVM classification, with quadratic, cubic and Gaussian variations available in MATLAB and used in this study.

It should also be noted that the methods developed for performing SVM classification are for binary data sets, i.e. sets with only 2 response variables. As is often the case, the data collected for this research corresponds to more than 2 distinct classes. Support vector strategies extended to more than 2 classes are referred to as multiclass SVMs, and there are multiple approaches to this problem. The two most common are one-against-all and one-against-one. In the former case, given *k* total classes, *k* binary classifiers are developed where each class is compared against all others and the best performing class is chosen. For the latter, all possible pairs of classes are combined and used to create distinct binary SVMs, with the class that wins the majority being chosen. In general, the one-against-one approach is the preferred method [47], but is also increasingly expensive from a computational standpoint as *k* increases in size. However, the number of predictors is purposely limited in this research and therefore the one-against-one strategy to multiclass SVM is used in all cases.

Overall, SVMs bring a nice balance with respect to classification accuracy, efficiency and computational performances. Furthermore, their approach to separating classes is flexible and is likely the primary reason for the promising results produced.

Ensemble Methods

Ensemble classifiers have been created with the goal of combining the results of multiple individual classifiers and weighting their results to achieve more accurate classification rates. Although often referred to as *weak learners* in the context of classifier ensembles, the individual learners used can be accurate alone. Therefore, when an ensemble is created from them, their averaged results can improve performance stability and reduce the possibility of overfitting the training data [54].

Generally, ensemble methods utilize the decision tree classifier as the baseline weak learner and combine many of them to make a final classification. Decision trees were utilized as classifiers at the genesis of the classification analysis performed for this work, but their predictive accuracy compared to other classifiers led them to be omitted from the final results. However, they were shown to be quite effective in certain configurations as part of *boosting* and bootstrap-aggregation (*bagging*) ensemble classifiers.

The concept of a decision tree is well-known because of their use for displaying a process or algorithm in a more visual manner. Decision tree learning comes from the idea of applying this concept to a training set of data in order to create a “tree” of rules for classifying new data. They are commonly constructed using top-down greedy splits, i.e. creating tree “branches” by recursively dividing a feature space until the nodal (leaf) values become constant and splitting becomes ineffective [56]. Using this general approach, there many algorithms which use the supplied training sets to systematically create a decision tree [55, 56]. A general example of a decision tree is shown in Figure 2.8 below.

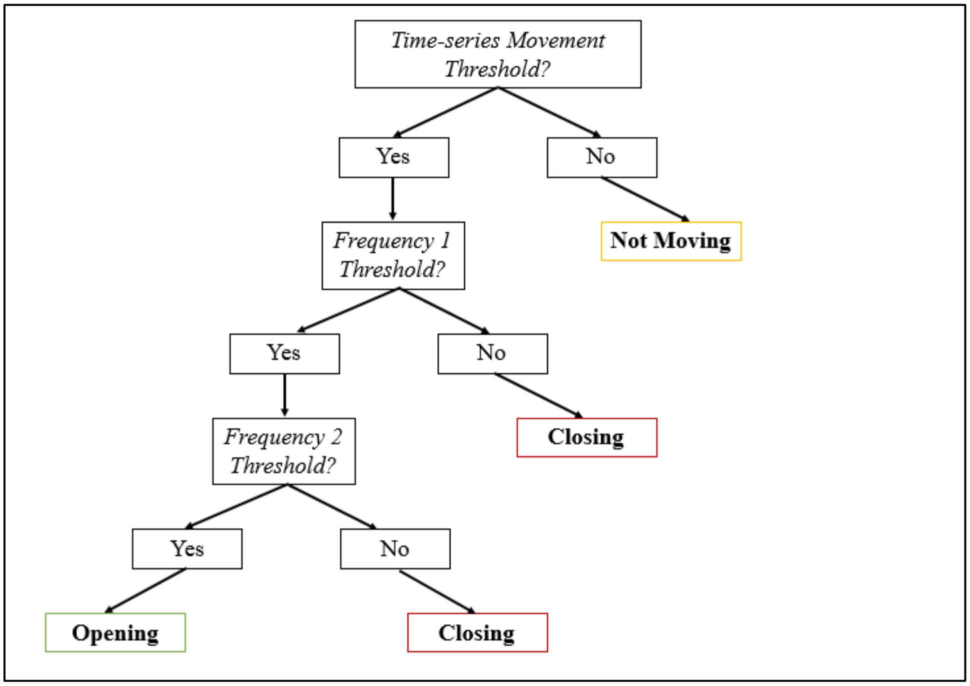


Figure 2.8: Decision tree for determining the state of a control valve.

The tree in this figure represents a generic logic structure used to determine the state of a control valve early in the research process for this thesis. While this logic structure was created manually and coded as such, the top-down decision tree induction process would ideally take the data associated with the three parameters used above (time-series movement, frequency 1 and frequency 2) and automatically determine the respective thresholds instead of defining them manually.

While simple in nature, intuitive and easy to interpret, decision trees are generally not as accurate as other machine learning classification methods and thus are commonly used in ensembles. There are multiple ensemble learning capabilities available in MATLAB which include the bagging and boosting methods referenced above along with random discriminant methods which, as the name suggests, combine discriminant classifiers.

Bagging trees refers to the process of taking the training data and resampling, with replacement, in order to create a set of subsampled decision trees [54]. All the decision trees in this set are subsequently used to classify new test data with the final classification made by majority rule. Boosting with decision trees, on the other hand, creates a sequence of trees and weights them based off of the error rates of the previous ones, in effect making it an intelligent averaging of all trees [57]. In this case, hundreds or thousands of trees are created, making the decision tree’s tendency to train quickly a particularly attractive feature for ensemble classifiers. Multiple algorithms have been developed for boosting purposes with the most common being the AdaBoost method, used for this work.

Given the scale on which individual classifiers, often decision trees, are produced for inclusion in classification ensembles, it makes sense that ensembles underperform in terms of computational efficiency compared to other machine learning methods. However, the way in which they are able to harness the typical accuracy of a single classifier with many others to create a more intelligent learner makes them worth exploring in the realm of CBM research and fault detection purposes. In particular, the way in which some are able to be adaptive in nature, as with AdaBoost boosted tree classifiers, makes them especially attractive if computing power is not a major concern.

Chapter 3. Hardware, Instrumentation & Test Setup

Introduction

This chapter stands to provide a detailed description of the valves used for testing and sensors used for data collection in this work. Additionally, the test plans for data acquisition on each valve will be given. For this research, three generic process control valves, graciously supplied by Emerson Process Management, were used for the simulation of baseline and seeded-fault testing. Two individual sensors, a high sample-rate accelerometer (ACCEL) and a more primitive, Arduino-powered inertial measurement unit (IMU), were used to collect response data along with basic control-system data (CS) supplied by the valve controllers.

By manually inducing blockages in the valve stroke, seeded-fault testing enabled response behavior to be recorded and used for subsequent feature extraction and classification purposes. By comparing this data to baseline valve movements, specific discrepancies between typical and atypical movements (i.e. those that result from a blockage) can be identified in the frequency and time domains. Further information regarding the evolution of the feature identification and extraction processes are provided in the following chapter.

Finally, specific descriptions of the actual faults are provided within this chapter. This work intends to discriminate specifically between a typical valve movement and two types of blockages, *hard* and *soft* that result in distinct behaviors. Relevant schematics are given to supplement the description of the seeded-fault methods.

Rotary Valves

Of the three valves used in this research, two were rotary ball valves. The valves are pneumatically actuated, indicating that air pressure is used as primary source of energy for controlling valve movement. Pneumatic valves are particularly common in the chemical, paper and petroleum industries [58] and are able to turn relatively little air pressure into a significant amount of output force for the valve stroke. The Fisher Vee-Ball V150 rotary control valve was used in two different sizes, five-inch and one-inch inlets, so that analysis methods could be compared in an attempt to demonstrate consistency as valve size is scaled.

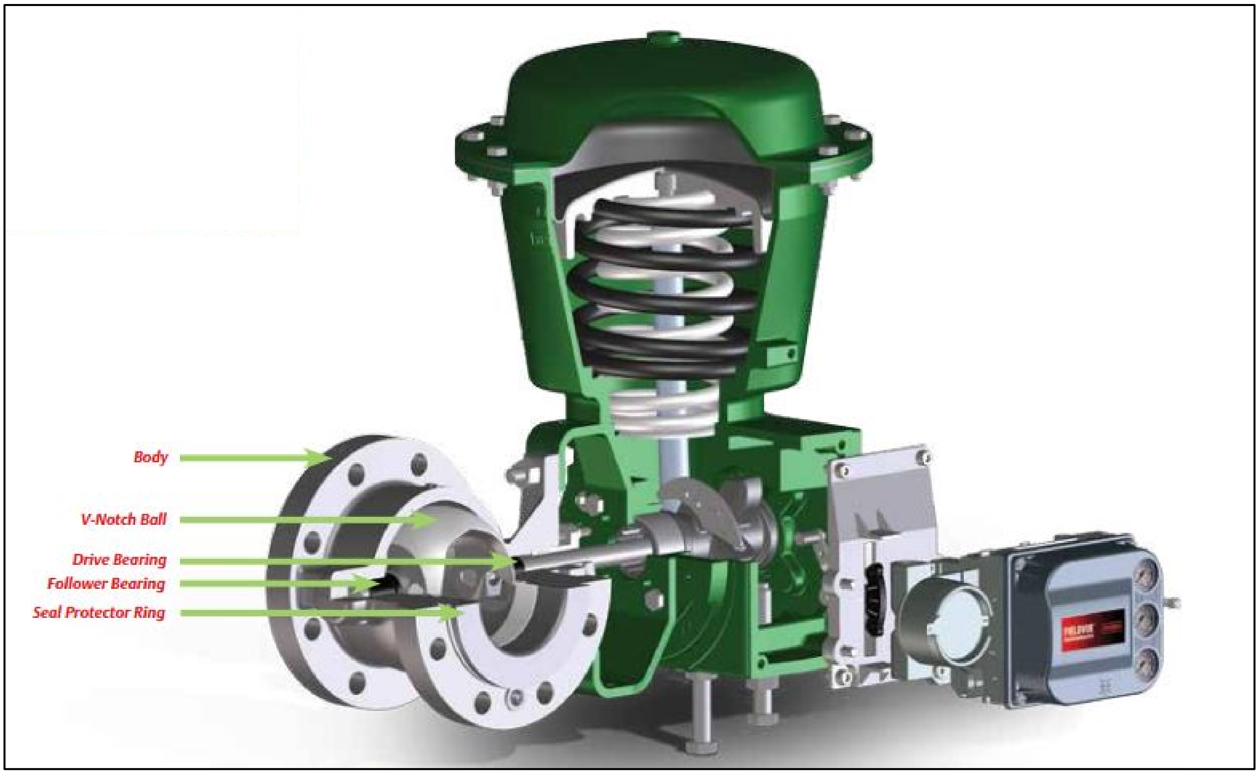


Figure 3.1: Computer rendering of the Fisher V150 rotary ball valve used [58].

Figure 3.1 above gives a schematic which displays some of the internal mechanics used by these types of valves. An air supply is connected to the diaphragm actuator (green) and is used to pressurize the diaphragm in order to provide a linear force to the spring-shaft within. This force is then translated to rotary motion through the use of a scotch-yoke mechanism. As a result, the valve can complete and full opening or closing movement with a quarter turn of the drive shaft. The spring-diaphragm actuator allows for an emergency response in the event of a loss of pressure or power to the control system. These events would cause the spring to decompress and close the valve automatically.

Figure 3.2 below provides actual images of the smaller of the two ball valves used. For scale, the visible inlet hole is one inch in diameter. While the inlet to the larger rotary valve is five inches in diameter, the entire body of the valve and diaphragm of the actuator is much more massive. It should also be noted that no flow loops were constructed using these valves. As a result, the testing performed took place under no-flow conditions with no pressure gradient across the valve inlets.

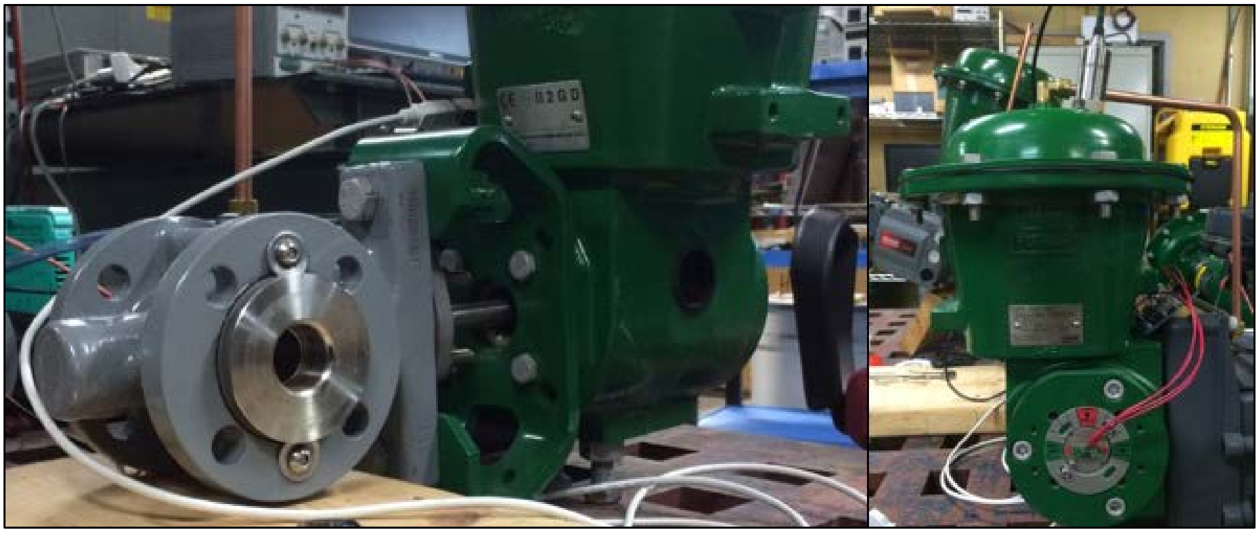


Figure 3.2: Images of the smaller notched rotary ball valve.

Sliding – Stem Valve

The third valve used for testing was a Fisher EZ Sliding-Stem valve, also provided by Emerson Process Management. Two views are provided below in Figure 3.3.

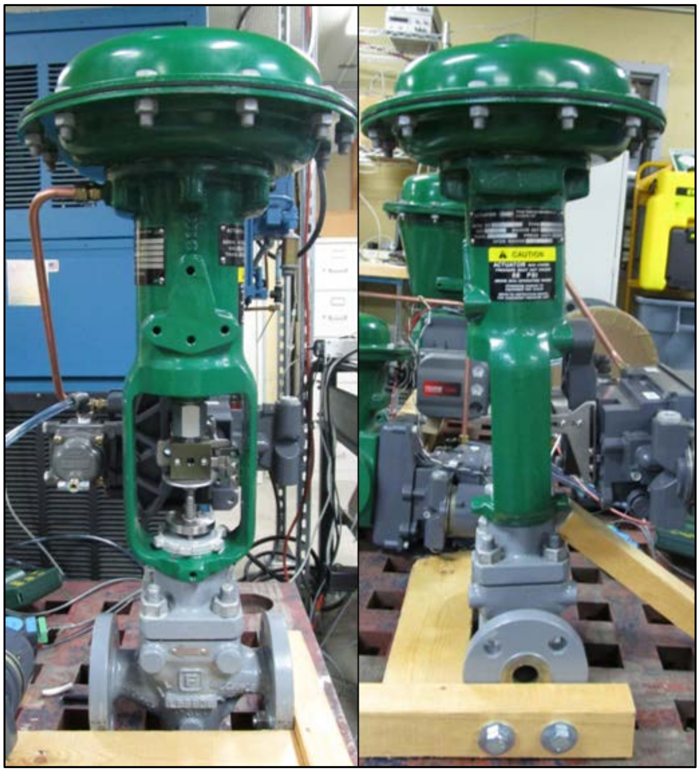


Figure 3.3: Images of the standing sliding-stem valve.

This valve was also equipped with a spring-diaphragm actuator supplied with air pressure to control movement. Similar to the mechanics of the rotary valves, the diaphragm is pressurized and the spring shaft compresses and releases to open and close the valve. However, instead of being translated into rotary motion, the linear force is applied directly to the drive shaft to plug the valve. Figure 3.4 below provides a general schematic for this type of actuator.

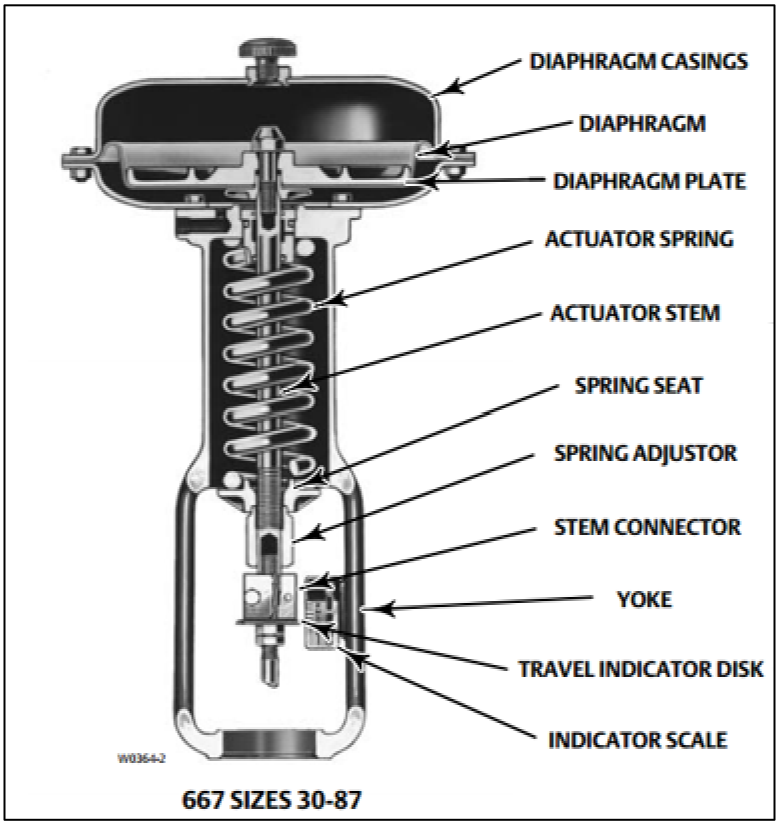


Figure 3.4: Internal schematic of the actuator for the sliding-stem valve used [59].

The purpose of performing testing on this valve, other than the fact that is was made available, was part of an effort to see if certain features carried over from valve to valve. From a more general standpoint, the goal was to show that certain analysis techniques discussed in the previous chapter could be applied to different types of valves. While the basic mechanical processes in each type are fundamentally different in certain ways, it would be important to discover that certain data streams are particularly effective in supplying features which discriminate between typical movements and blockage events.

Sensor Instrumentation

As mentioned in the introduction for this chapter, there were two sensors used for data acquisition during testing. The primary instrument which provided data that was critical to the continued development of this work was a USB-powered accelerometer from Digiducer, shown in Figure 3.5 below. The USB capabilities of this sensor allowed for easy data capture using MATLAB and effectively streamlined the process from acquisition to analysis.

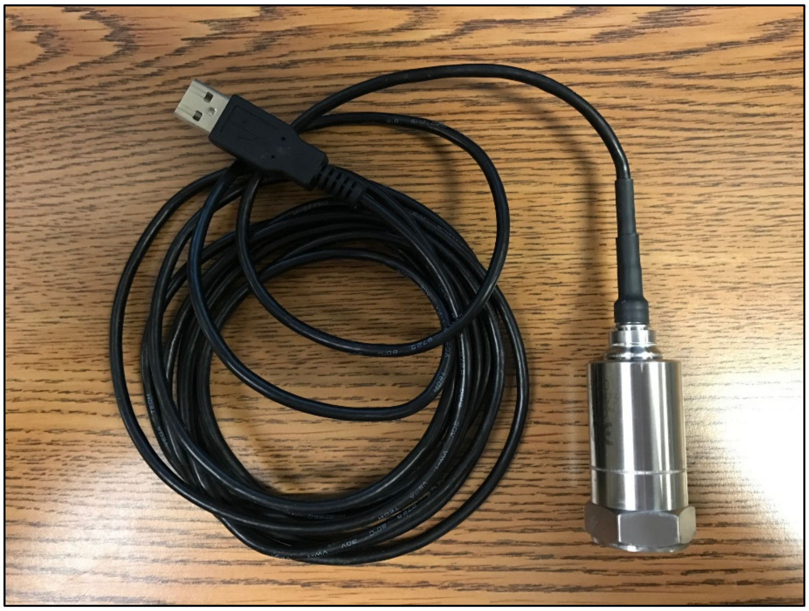


Figure 3.5: Digiducer USB-accelerometer used for collection of vibration data.

Given the historical importance of vibration data to CBM-related research, a quality accelerometer was sought at the beginning of this study. Specifically, the goal was to use sensors that are both easy to interface with software such as MATLAB and sample vibrations at relatively high frequencies so that a large range can be analyzed in the frequency domain. This accelerometer is capable of sampling at 48 kHz, a range deemed acceptable for this work. In many CBM applications, power usage and cost are often critical components to the selection of hardware components, so frequencies above 24 kHz were not of particular concern.

This Digiducer unit was identified as a good fit and was used to capture vibration on the valve actuators during baseline and seeded-fault testing with the goal of uncovering key discrepancies in the structural response of the valve and actuator to certain blockages. Figure 3.6 below shows the location of this sensor on one of the actuators. While not shown, it should be noted that approximately the same location was used on all three valve setups.

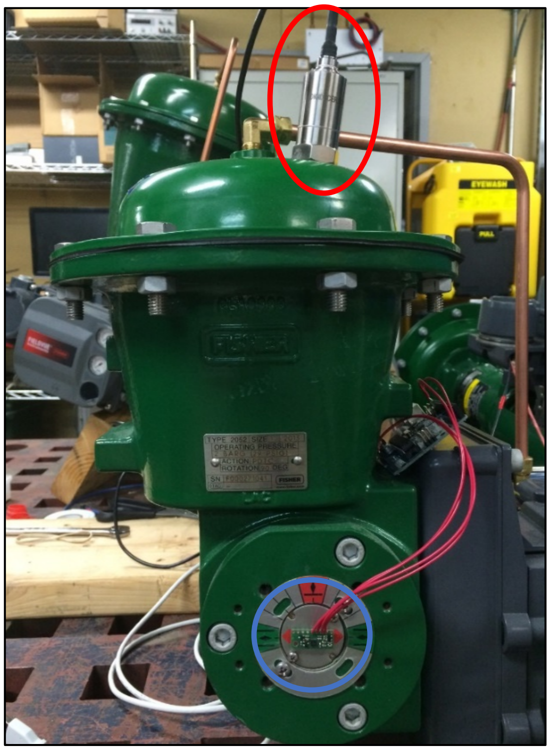


Figure 3.6: Location of Digiducer (red) and IMU (blue) on small rotary valve actuator.

The second sensor package used for data acquisition was a 6 degree-of-freedom inertial measurement unit purchased from Pololu (LSM6DS33). The IMU is capable of measuring acceleration in all three axial directions and can also measure angular velocity in these directions, giving it a total of 6 axes of measurement. The package itself is a breakout board which was wired to an Arduino Uno microcontroller for translation and calibration of output data for analysis. This connection is presented below in Figure 3.7.

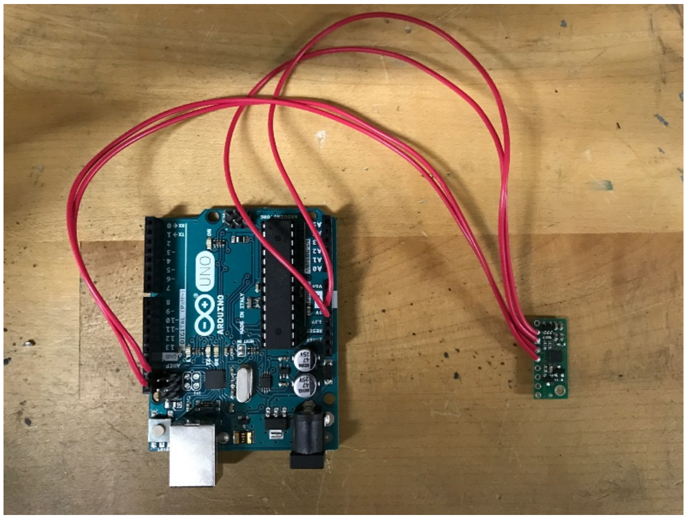


Figure 3.7: Arduino-IMU connective wiring.

The primary limitation to this sensor was the way in which data was collected from it. C code was provided for the IMU that allows the user to adjust the sample rate through simple commands. However, for both visualization and feature extraction purposes it was necessary for the data to come in some tabular or matrix format so that it could be saved in a spreadsheet and imported into MATLAB for analysis. This forced the use of Arduino's serial monitor to output data as it was read. As a result, although the baud rate for the serial monitor was maximized, the memory needed to print each value limited the sample rate to approximately 330 Hz. While this did not significantly affect the time domain analysis used on angular velocity measurements, for example, detailed analysis could not be performed in the frequency domain of this data. Low resolution in the data contributed further to this issue. As a result, potential underlying frequency-based characteristics were not explored using these data streams.

Given this limitation, the IMU was used primarily as a gyrometer to measure the angular velocity of the drive shaft in the rotary valves. It's placement on the turning indicator of these valve types is circled in blue in Figure 3.6. In an attempt to capture the translational velocity of the drive shaft on the sliding-stem valve, the IMU was positioned as indicated in Figure 3.8. The goal with this placement was to record lateral acceleration and integrate the signal to approximate velocity. As discussed in the following chapter, the lack of resolution posed problems to this approach.

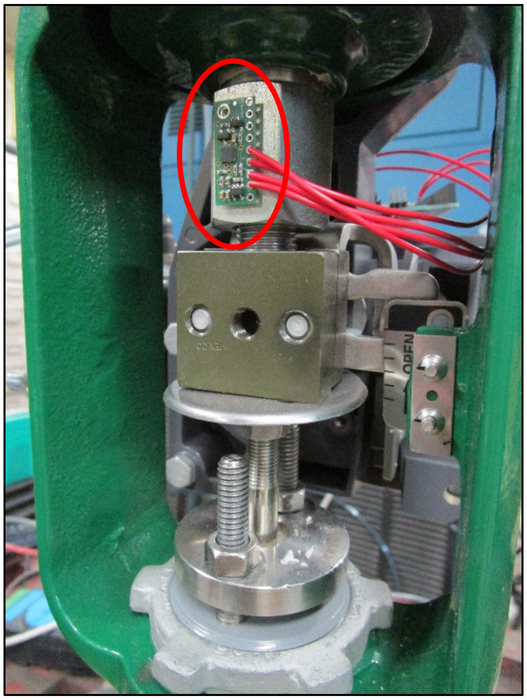


Figure 3.8: IMU location on sliding-stem valve.

Finally, it should be noted that for the majority of this testing, control system data was collected from the valves controllers which could provide it. In particular, the small rotary and sliding-stem valves were equipped with digital capabilities that allowed for both the calibration of the valve as well as control of specified movements. When these valves were controlled in this matter, three parameters, defined as Travel %, Drive % and Supply Pressure, are recorded at a low sample rate of just under 7 Hz. As a result, each movement results in 15-20 unique data points for each of these variables. While the low sample rate seems to imply that this is primitive data, it was shown to be remarkably consistent among movements of the same "type" as defined for this work. What makes this data particularly attractive is that it can be acquired without the use of external sensors. So from a practical perspective, it could be useful in applications that might be limited to no additional sensor instrumentation whatsoever. The following chapters discuss how it was used for classification purposes.

Description of Faults

The comparison of baseline data to those data associated with specific faults is critical to this work, and as such a brief discussion of the faults explored is warranted. As previously discussed, the scope of this work focuses on the idea of a valve blockage. To repeat the definition provided in the first chapter, a blockage is best described as some object or mechanism that prevents the valve from properly completing an expected movement. Given the potential consequences that can arise if a valve is not properly controlling flow, it might be valuable to identify when this abnormal behavior is occurring or worsening.

There is little doubt that, given the relatively broad definition of a blockage above, a myriad of events could occur that may be classified as a blockage. Considering foreign objects and internal mechanisms, we may see vastly different responses both between and among them, and it is likely possible to detect and classify these to high level of detail given the proper sensor instrumentations. However, taking into account both the resource limitations of this work and the impracticality of rigging control valves with many sensors in the real world, this research aims to take a more general approach which serves as a proof of concept as well as a foundation for future work. Therefore, it is proposed that two generic types of blockages, *hard* and *soft* blockages, be detected and classified against baseline movements, or *no* blockage.

The notation *hard* block is used to describe the object or internal mechanism failure which is so robust that it immediately stops the valve from continuing movement. Although the pneumatically controlled valves in question can exert a substantial amount of force, it is not out of the question that some object, such as one made from steel or rock, might find its way into the flow path. Trim and drain systems on submarines, for example, rely on the ability to pump water from the open ocean into their ballast tanks to control depth. Valves in situations such as this are more vulnerable to being compromised by some foreign object. Other valves are more intricate in their movements and might be stopped by smaller, weaker objects in a similar fashion. Smaller butterfly valves in submarine carbon removal units, for example, might be prevented from properly seating if even a very thin shim-type object slips in the wrong spot.

Furthermore, there are mechanisms within some valves which purposefully exist in order to affect movements. Valve stops exist in rotary valves, for example, to control where a valve might stop if desired. If these stops are improperly calibrated with the control system, the valve may not open and close as instructed. Figure 3.8 below provides a basic schematic.

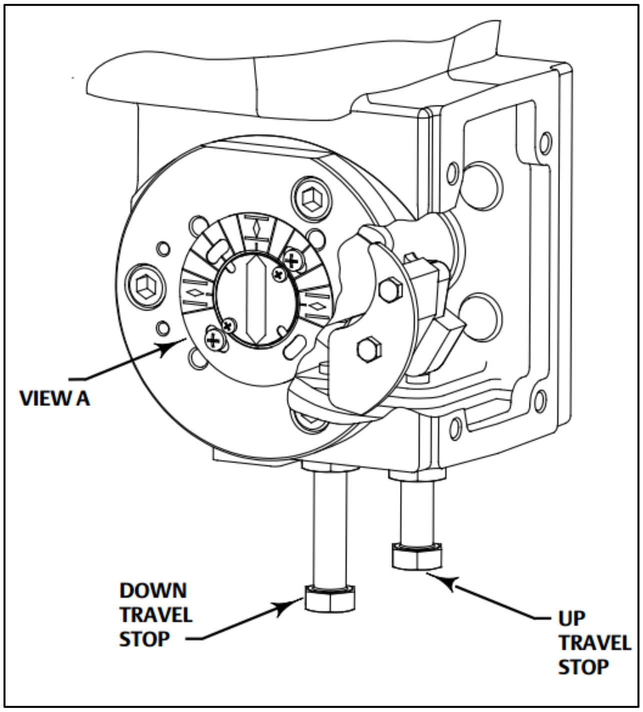


Figure 3.9: Rotary valve stops.

While a hard blockage may have many different causes, the commonality is that it always stops the valve stroke abruptly by striking an object, foreign or internal, before it has finished its movement. There are two key results from this happening. The first is that the impact will send a vibration throughout the valve body and actuator that could, ideally, be detectable. The second is that the movement will be shorter than expected. Using these two principles, the sensors described in the previous subsection could theoretically detect a hard blockage with relative ease.

When dealing with a hard blockage and the idea of simulating it in a controlled environment, the hardness of the objects used to block the valve must be considered. In order to avoid causing damage to the ball valve and its seat, the valve stops described above were used for introducing hard blockages in the two rotary valves used in this work. The continuous nature of these screw-based stops allows for many "levels" of blockage to be tested, and for open movements to be blocked as well as closings.

For the sliding stem valve, aluminum blocks of different sizes were cut and placed between the base of the valve actuator and a moving indicator connected to the stem as shown in Figure 3.10. While no objects were fit in the valve stroke because of the relatively complex flow path geometry, the same theorized general principles, induction of vibration and shorter movement time, still apply. It should be noted that only closing movements can be blocked using this approach. However, it is argued that these movements are more important because of the subsequent leakage that might be caused if they are blocked.

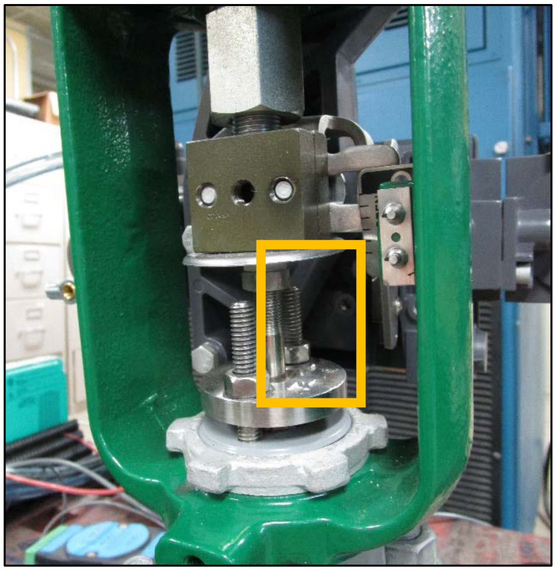


Figure 3.10: Location of hard blockage-inducing aluminum blocks and soft blockage-inducing die springs.

The concept of a *soft* blockage, as introduced by this work, is fundamentally different than a hard blockage and is thus worth distinguishing. The soft block is one in which a less robust object or mechanism interferes with a valve movement, causing the valve to eventually stop, possibly before the movement is complete, but not before working to push through the object in question. This results in a slower, crushing movement that will leave a much different travel signature than present in a hard block.

As with a hard blockage, there are two theoretical results from this type of movement. For one, the interfering object or mechanism being overcome will slow the movement down, increasing the overall time it takes to finish the movement, whether or not it ends up fully blocked. The second is that, as noted previously, the velocity profile of the soft-blocked movement will be drastically different from a typical movement in that it will be slowed and then potentially ramped back up if the object is fully overcome. A potential third result, as is the case with any fundamental change in a mechanical process, is that the frequency response from a soft blockage may excite or attenuate specific frequencies that can also be used for classification purposes. Again, as with the hard blockages described above, the simple two-sensor instrumentation used for this testing should theoretically be able to uncover key differences between this type of movement and a baseline run.

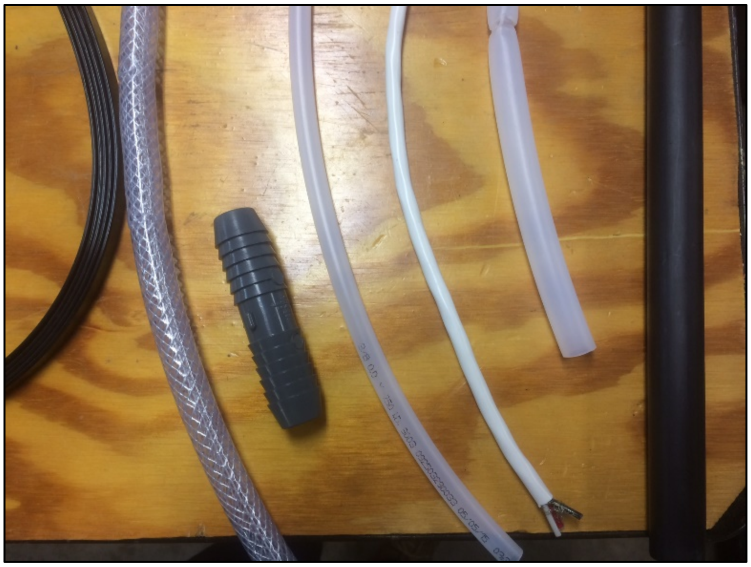


Figure 3.11: Objects used for soft blockage testing in small rotary valve. From left to right: zip ties (4), polyflex tubing, plastic coupling (1/2" ID), polyethelene tubing (3/8" OD), wire, polyethelene tubing (1/2" OD), PVC tubing (0.594" OD).

Given the nature of soft blocks and the weaker materials that may lead to them, they could be introduced to the valves used without the worry of damaging them. To test for the proposed behavior of the soft blocks, a variety of materials were inserted directly into the valve stroke path and closed on. Figure 3.11 above shows the different objects used in the initial testing on the small rotary valve, while Figure 3.12 below shows the objects used for soft blockages in the larger rotary valve.

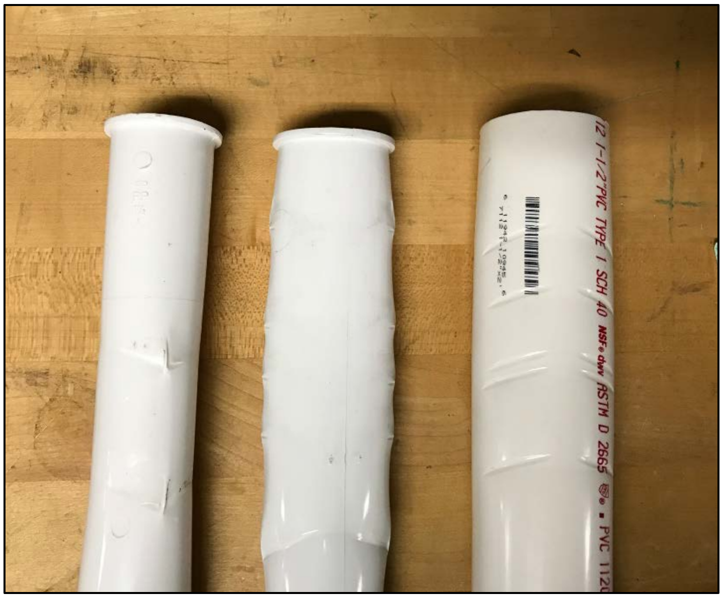


Figure 3.12: Objects used for soft blockage testing in large rotary valve. From left to right: 1.5" OD plastic sink tailpipe, 1.75" OD plastic sink tailpipe, 2" OD PVC pipe.

Given the smaller clearance in the sliding-stem valve and the curvature of the flow path within it, it was more difficult to select objects that would fit properly. Therefore a varied set of 5 heavy duty compression springs were placed in the position indicated in Figure 3.10 to induce this behavior. These springs are described in Figure 3.13 below. In much of the subsequent testing, the materials succeeded in stopping the valve movement before it was complete, particularly the more robust materials and the industrial die springs used with the sliding-stem valve. However, particularly in the case of the smaller rotary valve, some materials were weak enough to be eventually punched through. These cases are still of interest, however, because they still suggest some blockage occurring. It should also be noted that only closing movements could be impeded with these materials and thus the response of opening movements to soft blockage phenomena was not studied.

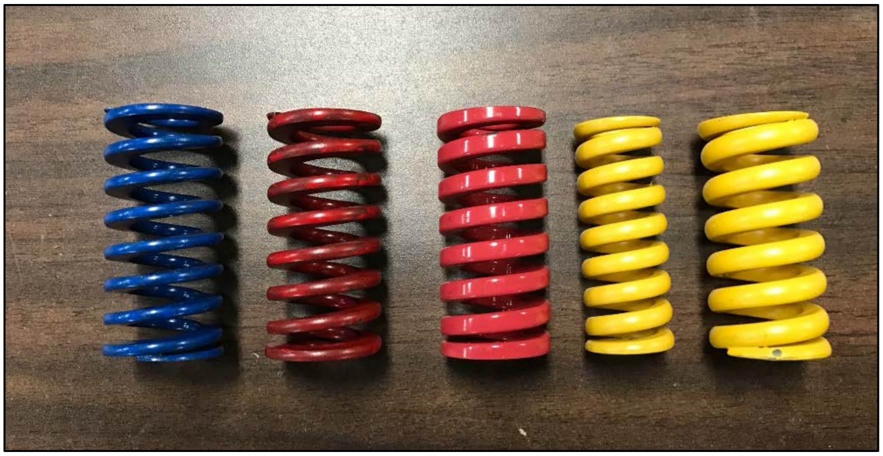


Figure 3.13: Compression springs used for soft blockage testing in the sliding-stem valve. From left to right: medium-duty (blue), medium-heavy-duty (dark red), heavy-duty (red), half-size extra-strong (yellow), full-size extra strong (yellow).

Description of Tests

The progression of the testing performed for this research is as follows. The initial bulk of tests were performed on the small rotary valve with the objective of comparing analysis techniques and experimenting with different feature extraction processes. Technical issues also made it difficult to control the other two valves in the beginning months of this work, lending focus to this valve in particular. All tests performed for the evaluation of the fuzzy logic and neural network classifiers were performed on this valve. The discovery that statistical classifiers and their corresponding feature extraction procedures produced the most promising results, led to their removal from the analysis of test data from the large rotary and sliding-stem valves. It is also noted that only the single USB accelerometer was used in data acquisition for these tests. The primary reason for this lies in specific methods used for extracting features (real-time vs. post hoc) and will be expanded upon in the following chapters.

Following the initial testing on the small rotary valve and evaluation of the fuzzy logic and neural network classification methods using this data, a posterior analysis evaluating the procedures was performed. Three primary decisions were made as a result. The first was that the IMU sensor would be added to the valve and used for data acquisition and subsequent analysis. The second was that all classifications would be made by analyzing movements in a post hoc manner as opposed to the attempted real-time simulations used for the fuzzy logic and neural network classifiers. Finally, it was decided that statistical classification methods would be the focus of further classification efforts for this work. These decisions enabled the ensuing tests to be performed in a more organized manner with a collective focus. After these decisions were made, a set of tests was performed on each valve simulating all three conditions (*hard* block, *soft* block, and *no*-block) and capturing the same data with the same sensors. Additionally, control system data output during these tests was pulled from the small rotary and sliding-stem valves. Because of compatibility issues, this data was unavailable for the large rotary valve.

Given the discrepancies and gaps in these testing procedures, an attempt at organizing them in a format which is easier to understand is given in Tables 3.1 and 3.2. These tables separate testing for the fuzzy logic and neural network classifiers from the test for the statistical classifiers. The numbers in Table 3.2 includes tests that were eventually used for training sets.

Table 3.1: Summary of test data collected for evaluation of FIS and ANN classifiers.

Fuzzy Logic & Neural Network Testing			
	Hard Blocks Analyzed	*Soft Blocks Analyzed*	*No-Blocks Analyzed*
Small Rotary	30	17	50
Large Rotary	N/A	N/A	N/A
Sliding-Stem	N/A	N/A	N/A

Table 3.2: Summary of test data collected for evaluation of statistical classifiers.

Statistical Classifier Testing						
	Hard Blocks Analyzed		*Soft Blocks Analyzed*		*No-Blocks Analyzed*	
Small Rotary	Two-Sensor	31	Two-Sensor	23	Two-Sensor	56
	Control System	12	Control System	25	Control System	57
Large Rotary	Two-Sensor	30	Two-Sensor	15	Two-Sensor	65
	Control System	N/A	Control System	N/A	Control System	N/A
Sliding-Stem	Two-Sensor	15	Two-Sensor	35	Two-Sensor	70
	Control System	15	Control System	35	Control System	70

Two final takeaways from these numbers will be noted. First, no-blocks were more populous in general because any movement that didn't directly result in a blockage was classified as a no-block, and every test that was performed included two movements. For example, a test which included a slow block would also include an opening movement after closing on the object in question. Such a test would therefore consist of one slow block and one no-block. Secondly it must be clarified that the number of classifications actually made for the fuzzy logic and neural network classifiers is not actually as low as it appears. As explained in the next chapters, these classifiers looped through data and made classifications continuously to simulate a more real-time collection and classification of movements. As such, one test could contain more than 100 individual classifications. A total of 44 tests (97 movements) were selected for this process.

Chapter 4. Feature Extraction Methodology

Introduction

Since a typical supervised learning classifier requires a set of inputs to both train and test its prediction performance, the selection of those inputs must be addressed. This chapter is provided to give an overview of the data supplied by the ACCEL, IMU and CS and the relevant extracted features for classification. Given the clear differences between the approach to classification with the ANN and FIS classifiers and the approach with statistical classifiers, the chapter will be divided accordingly to address them both.

First, a more detailed description of the specific differences in approach using these methods is warranted. It was mentioned in Chapter 3 that the ANN and FIS classifiers took a real-time simulation approach to diagnosis while the statistical classifiers took a more post hoc approach. The real-time simulations stemmed from an early effort to show how ACCEL data could be used to determine if a valve was opening or closing. While this effort will be expanded on in the following subsection, the general idea was to take data from ACCEL and input it into an algorithm which would then loop through, in intervals of 5000 data points or about 10 milliseconds, and perform quick analyses to flag specific behaviors. The resulting flags would be part of a logic structure to determine the direction of movement.

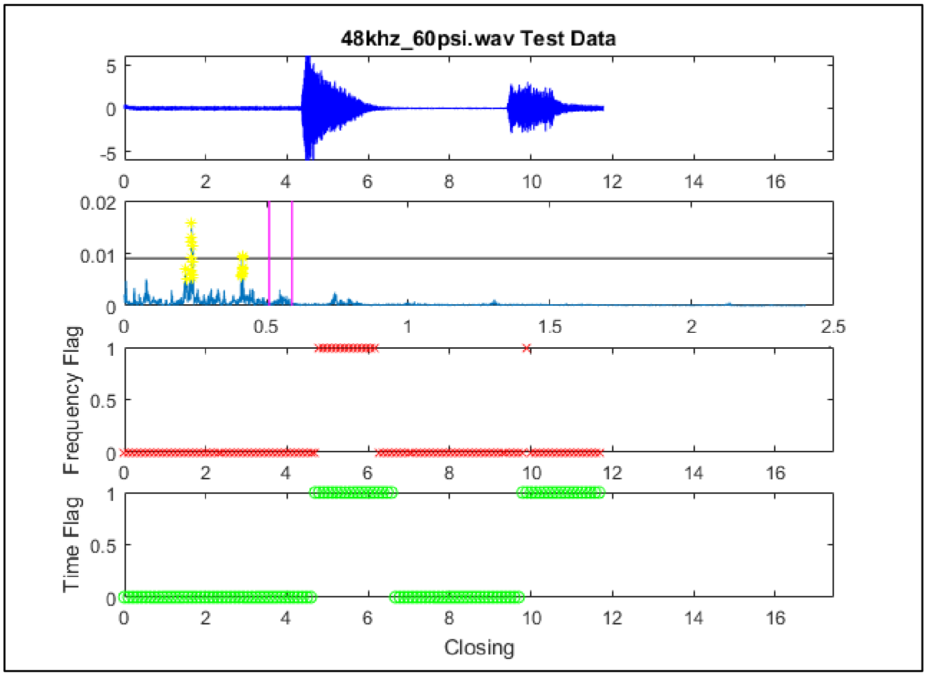


Figure 4.1: Example of a real-time simulation which determines whether the valve is opening or closing.

An example of this is shown in Figure 4.1 above. In this figure, the first plot shows the time-series vibration data while the second plot shows an updating FFT as the algorithm loops through the data. The final two plots show the binary values of two flags, one for time-domain analysis and the other for frequency-domain analysis, and combines their values in a crisp logic structure which outputs a movement direction given at the bottom. In this case, the valve was closing as indicated. The ability to monitor some mechanical system in real-time is advantageous because it removes the need for manually going back and checking the data after it was recorded. It was also particularly effective for detecting movement direction since it helped expose potential flaws in the simple classification logic structure used.

Once blockage testing began, it made sense to continue with this approach since it was shown to be effective up to that point. The data that was initially captured was then used to experiment with the neural network and fuzzy inference systems, and the results from this evaluation are shown in the next chapter. The focus then shifted towards statistical classifiers using the *Classification Learner* application in MATLAB. For this approach it was determined that it would likely be more effective to classify using a single vector of features from an entire movement as opposed to looping through and grabbing small chunks within each movement. This is the reason for the discrepancy in each approach.

ANN & FIS Features

As briefly mentioned in the previous chapter, data collected by the ACCEL sensor served as the only data stream which was used for feature extraction and classification using the ANN and FIS. The primary reason for this was that the IMU’s slow sample rate posed problems with respect to looping through the data sets of both sensors simultaneously. Furthermore, there was no available mechanism which provided the ability to record data from both sensors over identical intervals, making it nearly impossible to align the data in time. Therefore, it was attempted to detect and classify blockages with this single accelerometer placed on top of the control valve actuator. While limited in scope, this provided an opportunity to explore the frequency characteristics of these movements in particular detail. Given the multiple classes of movements that were looked at, a variety of discriminating features needed to be identified to assure a reasonable level of classification accuracy.

Conclusions drawn from the open-close detection study referenced above confirmed the fact that open and close movements in these valves leave different signatures in the frequency domain. As a result, distinguishing between them was important in the detection of blockages. Figure 4.2 below provides two plots comparing the frequency content of an open and close movement. The spike in magnitude around 5500 Hz, shown circled, was used as the focus of a frequency flag for the open-close decision logic.

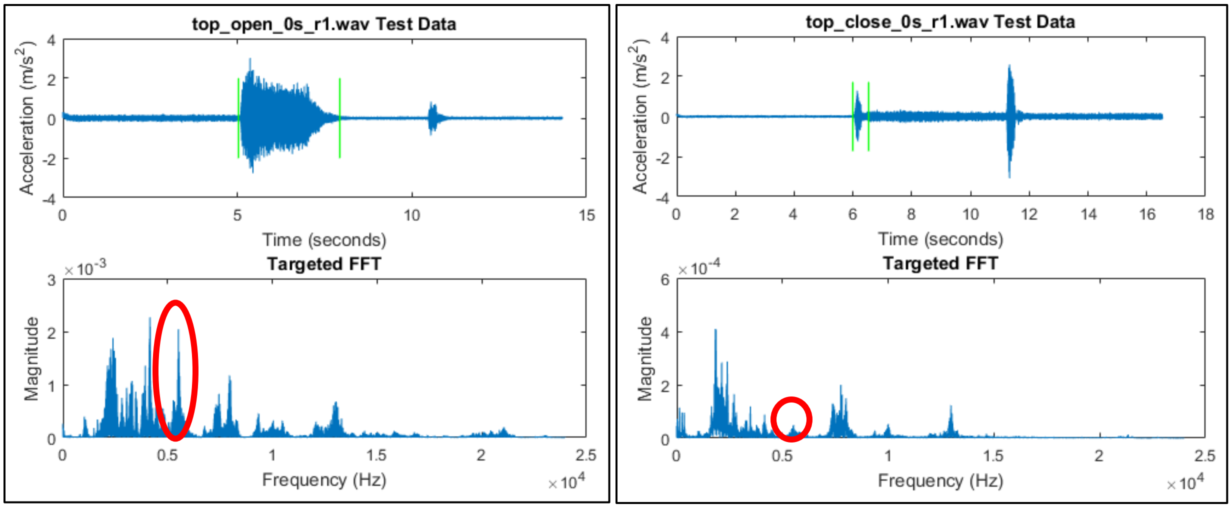


Figure 4.2: Comparison of frequency content in an open movement (left) and close movement (right). FFT magnitudes are normalized by FFT gain, and the spike at 5500 Hz was consistently seen in all comparisons.

Using this discovery, the resulting algorithms looked for the strength of frequency components in the 5100-5900 Hz frequency band. By taking a subset of the largest magnitudes within this range and averaging them, a single value describing this band could be obtained and used for subsequent classification.

While the frequency analysis helped in distinguishing between open and close, a simple time-domain analysis was performed to quantify whether or not the valve was moving in the first place. Recalling that the algorithm will loop through the data at 5000-point intervals, the standard deviation of those points was calculated and used to determine if the valve was moving or stationary. The reason for this approach is demonstrated in Figure 4.3 below. As the valve begins to move and vibrations increase in magnitude around 0, the overall standard deviation of each data subset increases dramatically as shown in the plot to the right. This gives a reliable gauge of movement in a single feature value.

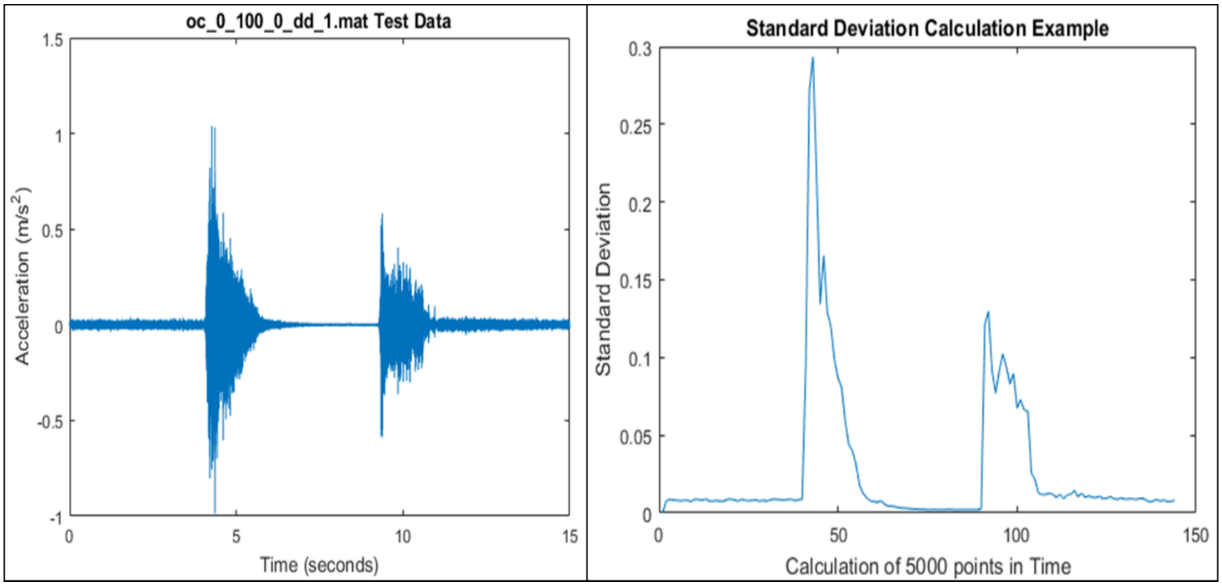


Figure 4.3: Simple open/close test (left) and standard deviation calculation over time (right).

Therefore, the results of this analysis provided two specific inputs, one describing frequency content and the other a simple time-domain calculation, which could be used to predict whether the valve was opening, closing or not moving at any point in time. These values could then be input to an ANN or FIS for quick classification.

The results of this open-close detection were then used in conjunction with additional frequency-based inputs to detect blockages. FFTs were once again used to study the frequency signatures of different movements in order to locate discrepancies. In particular, frequency spikes unique to movements resulting in either a soft block or hard block were searched for. The first obvious discovery was made by comparing a hard blockage to no-blocks and soft blocks. As theorized, the impact caused by a hard blockage appeared to be detectable in the 300-500 Hz range as demonstrated in the bottom plot of Figure 4.4 below. The spike visible in these hard-blocked movements and its absence in all other scenarios was consistently seen in weeks of collected data. The magnitude of frequencies in this band was quantified in a similar manner to the parameter used to distinguish between open and closing movements.

Given the smaller subsets of points that are analyzed in each loop using this real-time simulation approach, the detectable spikes in both hard blockages and soft blockages, as shown below, were much more prominent. As seen in the next subsection, performing analysis on an entire movement as one entity posed challenges to locating these discriminating characteristics.

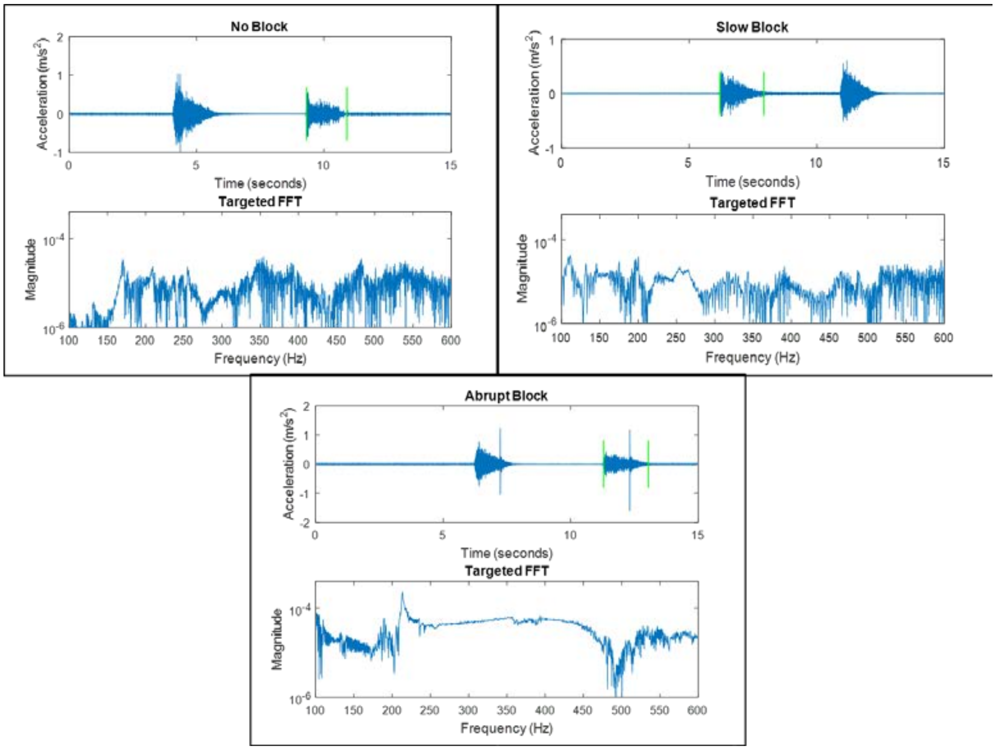


Figure 4.4: Comparison of a run with a hard blockage to runs with no blockage or a soft blockage over the 300-500 Hz band.

In an attempt to further divide the movements into distinct classes, an additional frequency spike unique to soft blockages was identified and similarly quantified for classification purposes. Seen in the bottom plot in Figure 4.5 below, the spike was consistently visible in the lower 10-30 Hz range as a slow blockage was occurring.

This additional analysis results in a total of four parameters to be input to both the ANN and FIS developed for this work. To summarize, the first two parameters described previously are used to first identify the movement status of the valve. The second two are frequency-based variables which aid in the subsequent classification of movement as a hard block, soft block or no-block. Within the algorithm used to process the data, extract features and make classifications, separate classifiers for open-close detection and blockage classification were created, with the result combined into a single final classification. One of the drawbacks of using a single accelerometer, therefore, is the need to identify movement direction because of how these identified characteristics change with it. This resulted in the need for the results of multiple fuzzy logic systems and neural networks to be combined into a final prediction. While only performed on one valve, it is sensible to assume that similar characteristics, albeit in different frequency ranges, could be found in other types or sizes of control valves.

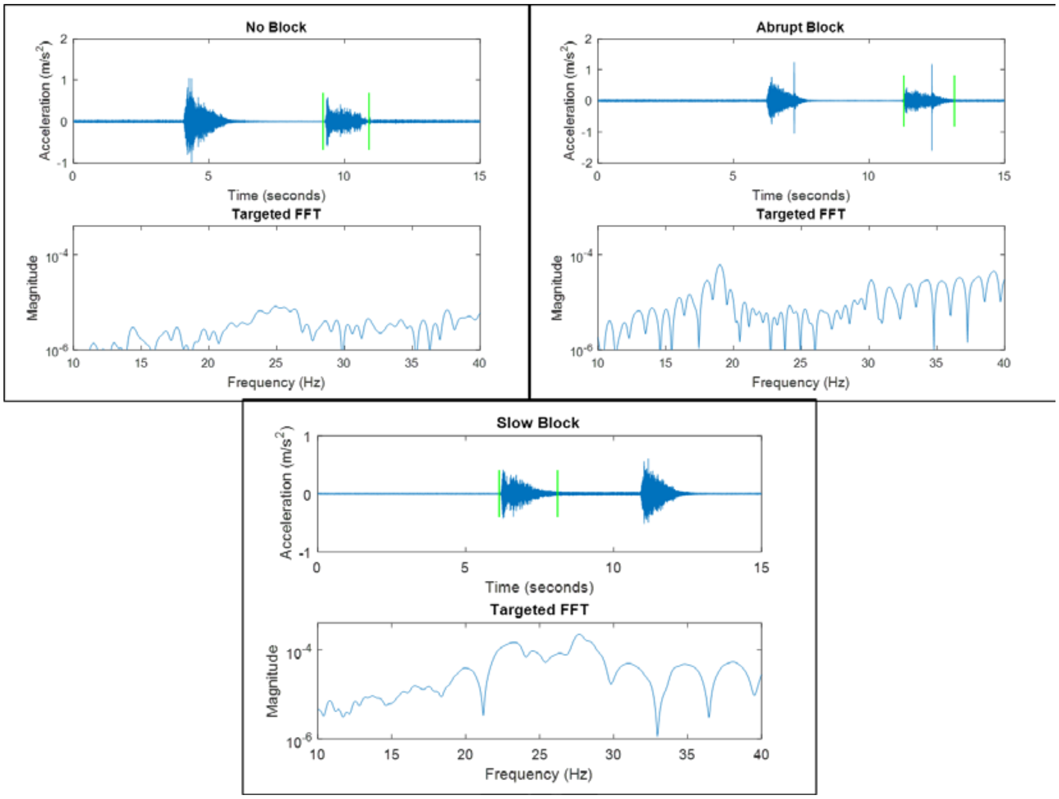


Figure 4.5: Comparison of a run with a soft blockage to runs with no blockage or a hard blockage over the 10-30 Hz frequency band.

Statistical Classifier Features

Following the training and evaluation of the ANN and FIS classifiers, the focus of this research turned to the use of a wider variety of classifiers, such as statistical, geometric and instance-based methods, given the work published by Adams et al. [9]. At this point, a more structured approach to testing and analysis took form. For each valve, the tests discussed in the previous chapter were performed and the resulting data was then analyzed individually. The overall objective was to find a small set of features that, in general, could be used in some form to accurately classify movements into types of blockages that may have occurred. For example, while it was not expected that each valve would show frequency spikes in identical bands for hard blocks, the fact that some discerning spike did exist was anticipated. Ideally, a feature template could be created which could be retrofit to any existing control valve with little additional research.

ACCEL Features

Features taken from the ACCEL data stream for this analysis were strictly frequency-based, unlike for the ANN and FIS classifiers that required additional time-domain features which were only relevant to open-close detection and not to actual blockage classification. This was the result of adding the IMU sensor package to this set of tests, as it was particularly useful for time-based feature selection. The analytical process for determining key features using this data was similar to that used for the ANN and FIS classification portion of this work. Specifically, frequency domain data taken from each movement was compared both to other movements of the same type as well as ones which would be classified differently with the goal of identifying specific frequency spikes that could be quantified in some manner. For blocked movements, frequency bands that were either excited or attenuated when compared to baseline testing were studied.

Testing was repeated for the small rotary valve and the distinguishing frequency-based features discovered were, as expected, consistent with the results from the previous set of tests. That is, a significant rise in magnitude over the 300-500 Hz frequency band was consistently seen during hard block movements. This can be seen again in Figure 4.6 below, which contains plots comparing the frequency content of a hard block (left) and a movement which took place directly after a hard block (right) with FFTs consisting of 8192 points. Given the time-based features described in the following subsection, it was particularly important to find a quantifiable frequency-based discriminator which was perfectly unique to hard blocks. Although not pictured, this excitation was also absent from soft block movements.

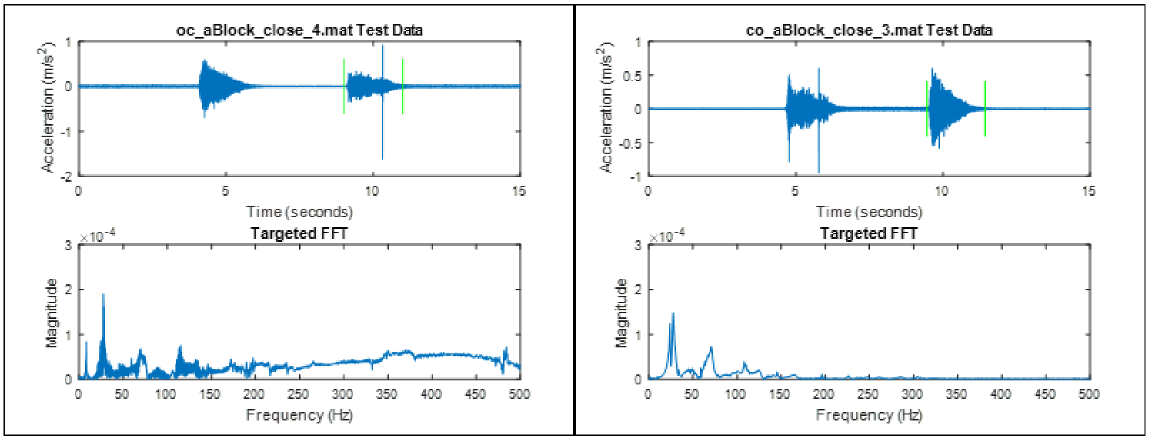


Figure 4.6: Comparison of a run with a hard blockage (left) to one with no blockage (right) for the small rotary valve.

A possible frequency feature was also discovered in first-glance looks at soft block data when compared to baseline and some hard blocks. Figure 4.7 below shows a soft block (left) with strong magnitudes around 50 Hz compared to a standard movement (right). As a result, this feature was also included in the extraction algorithm which aided in gathering training and test data. However, as discussed later, this feature was not shown to be completely unique to soft blocks from all tested materials and was ultimately not included in final classification testing.

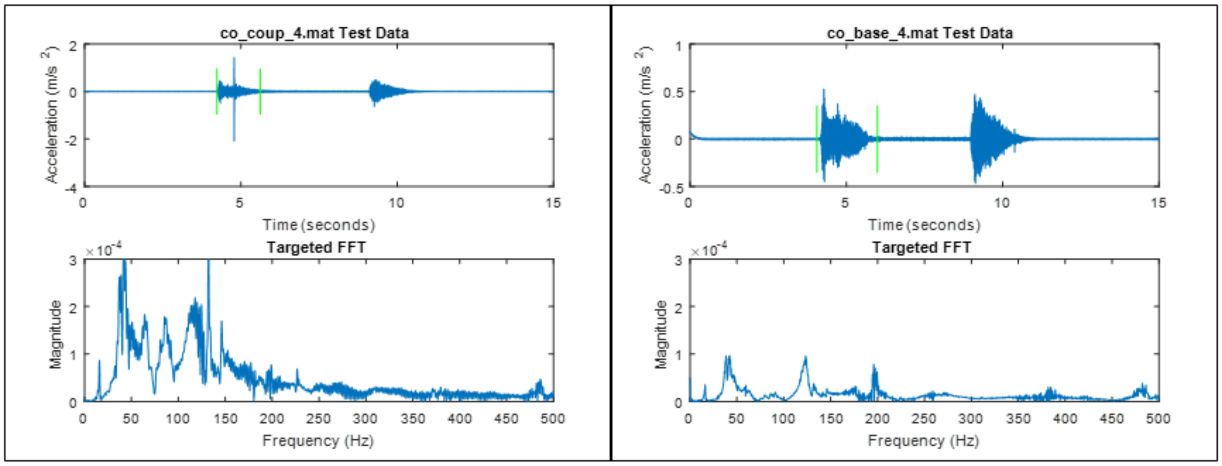


Figure 4.7: Comparison of a run with a soft blockage (left) to one with no blockage (right) for the small rotary valve.

Data from the large rotary valve yielded some different results, but ones which were useable nonetheless. While hard blockages revealed some frequency excitations, they were either not distinct enough to extract or not completely unique to a hard blockage. However, multiple bands were found to actually be attenuated during the hard blockage movements when compared to baseline. Specifically, spikes around 59, 340 and 2850 Hz were found in baseline movements and soft blocks that were not present to nearly the same extent in either hard blocks. Figures 4.8 and 4.9 below demonstrate the behavior of the 59 and 340 Hz spikes.

The discovery that discriminating features for the large rotary valve were only truly unique to baseline movements and some soft blocks is not ideal. In theory, this could lead to some ambiguity in classifying hard blockages correctly. However, the features taken from the IMU combined helped explain the differences between all movements when combined with these two features. Furthermore, an analysis of variance showed statistically that these features varied with class at a suitable significance level to be included in the final analysis. This process is discussed further in the final subsection of this chapter.

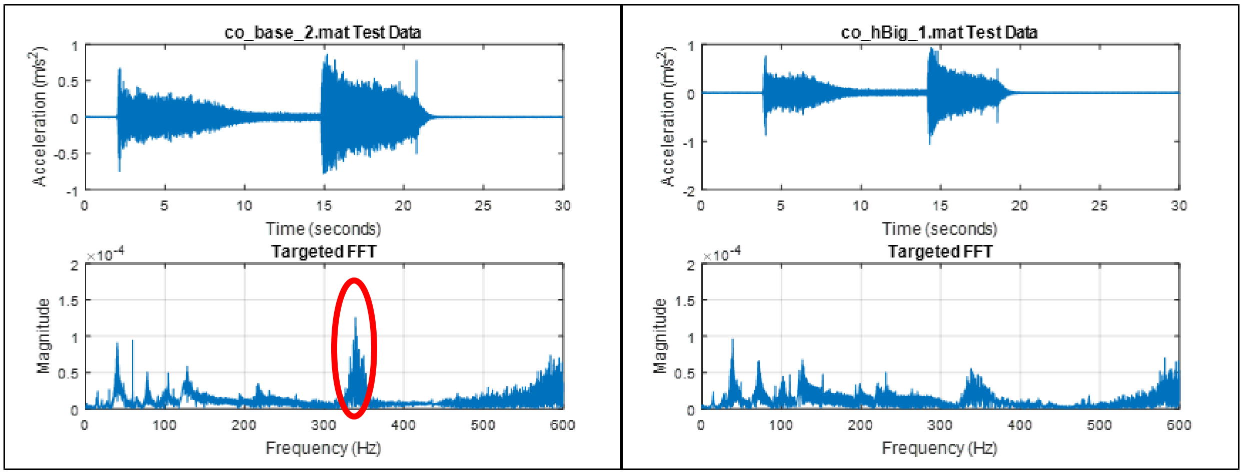


Figure 4.8: Comparison of a run with a baseline movement (left) to one with a hard blockage (right) for the large rotary valve. Note the spike around 350 Hz on the left plot.

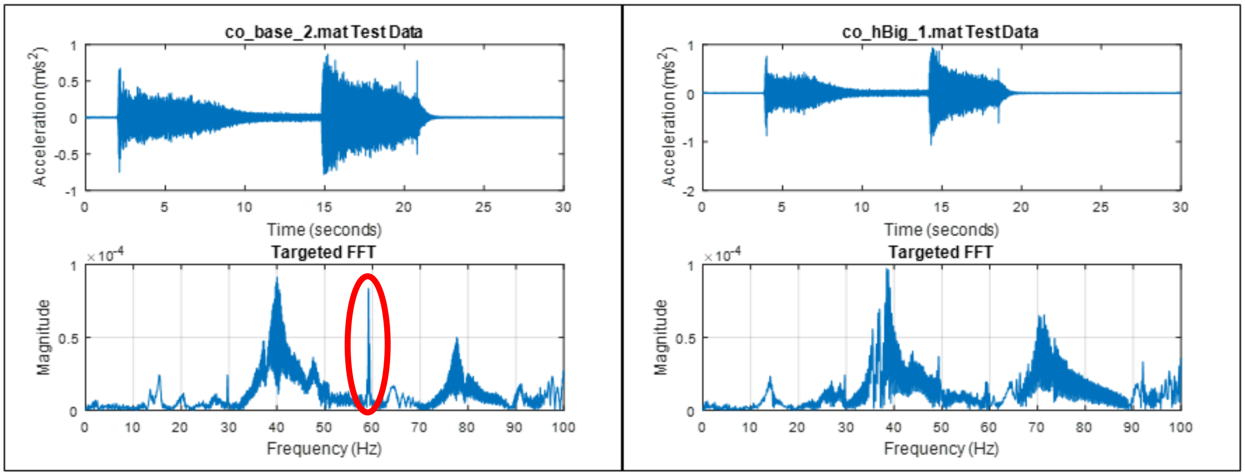


Figure 4.9: Comparison of a run with a baseline movement (left) to one with a hard blockage (right) for the large rotary valve. Note the spike at 59 Hz on the left plot.

Finally, data from the sliding-stem valve provided multiple features unique to both hard and soft blockages which could be leveraged for classification. Again, individual FFTs were analyzed then compared to movements of different class to identify quantifiable differences in the data. Two of these specific features are displayed in Figures 4.10 and 4.11 below.

During hard blockages, many spikes were seen in approximate 50 Hz intervals starting around 150 Hz and continuing to 400 Hz as shown in Figure 4.10. This behavior was consistent among all hard blockages and was not present to this extent in most other movements. Also discovered was a spike between 650 and 700 Hz which was often present in hard-blocked movements. This feature is not shown in the figures below but was collected during feature extraction for analysis. However, it was not as clear of a predictor as the magnitude spikes between 150 and 400 Hz.

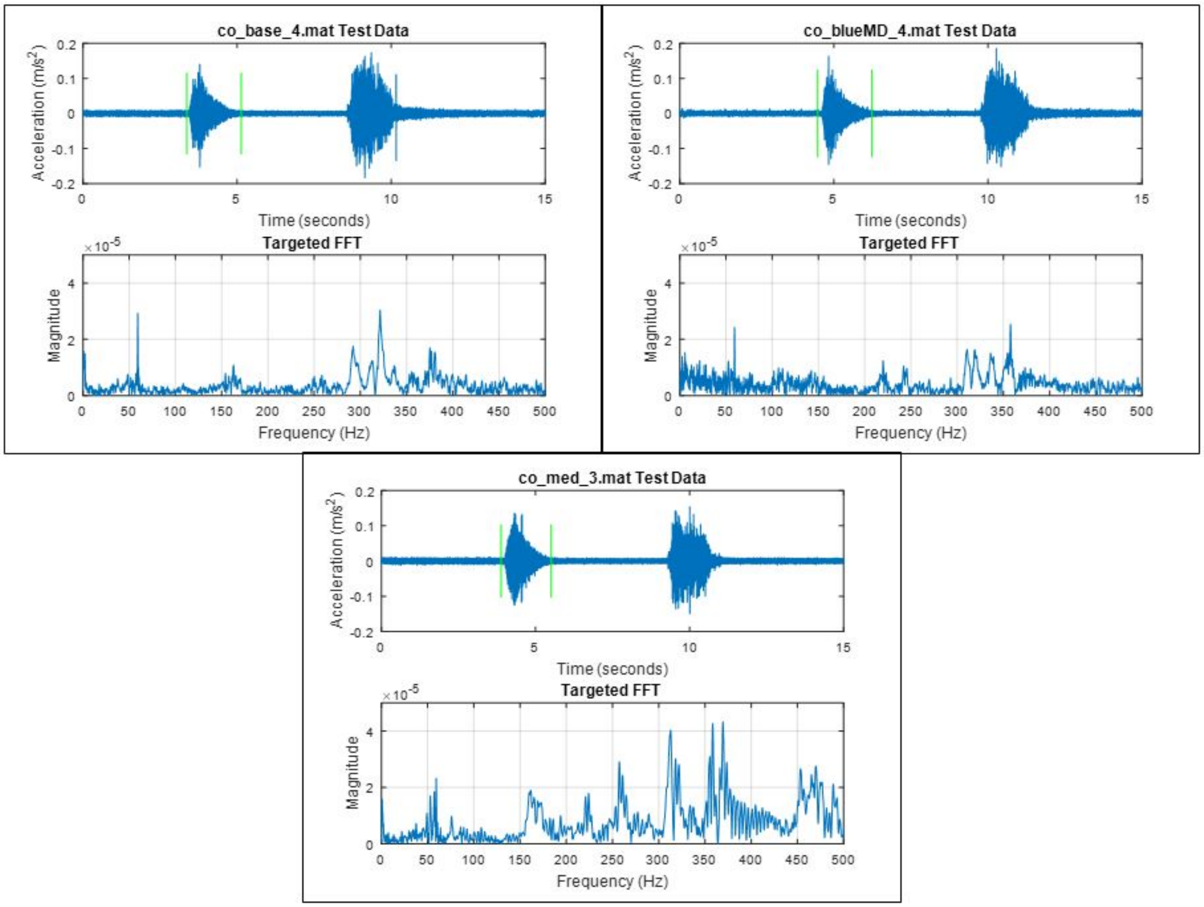


Figure 4.10: Comparison of a run with a hard blockage (bottom) to runs with no blockage (top left) or a soft blockage (top right) over the 150-400 Hz frequency band.

Soft blockages also produced unique behavior seen in Figure 4.11 below. There was, at the very least, a slight spike in the 1800-1900 Hz frequency band in all movements. However, it was significantly accentuated in soft blockage data as seen in example comparisons below. This feature was seen consistently across all soft blockage data sets, even in comparisons between the weakest soft blockages and strongest hard blockages. Overall, compared to both rotary valves, the sliding-stem valve seemed to provide the clearest distinguishing features for classification.

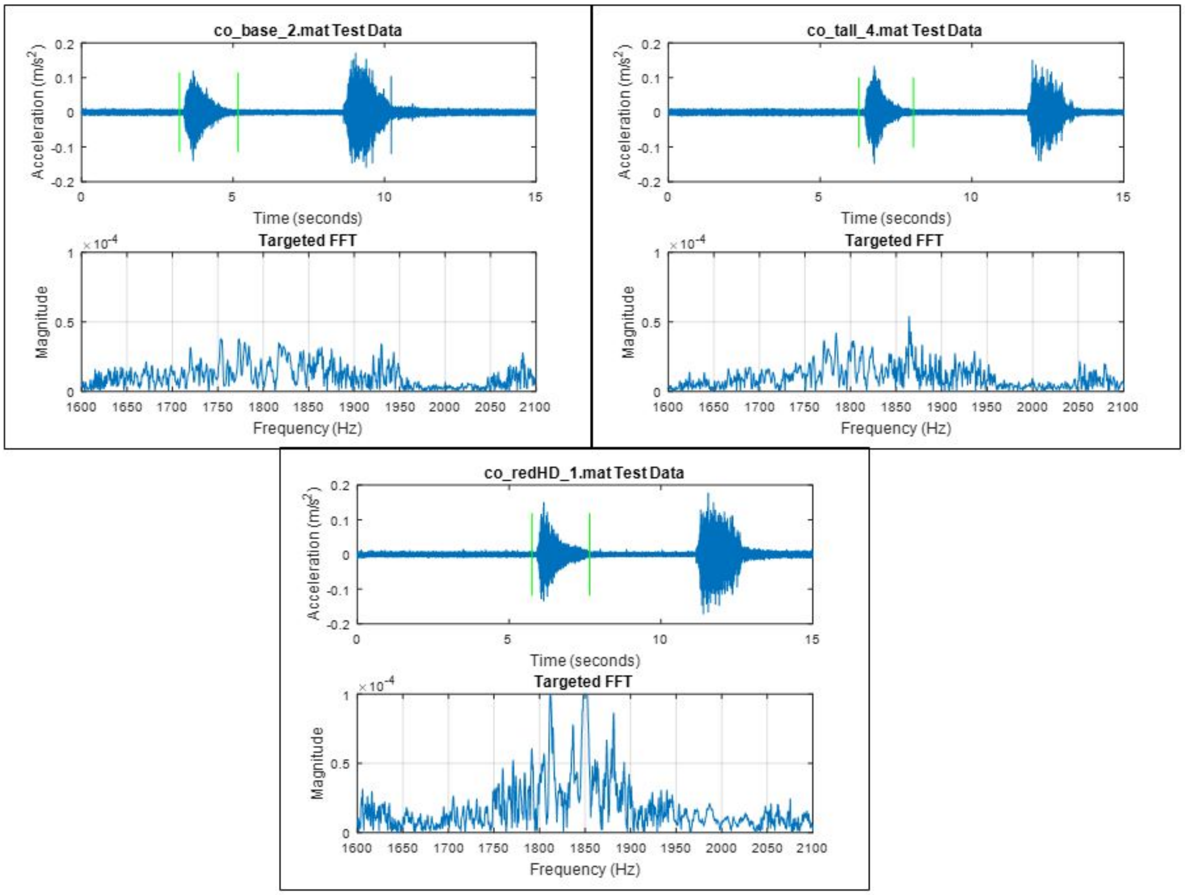


Figure 4.11: Comparison of a run with a soft blockage (bottom) to runs with no blockage (top left) or a hard blockage (top right) over the 1800-1900 Hz frequency band.

IMU Features

The IMU sensor package was initially set up to provide two outputs: angular velocity around the z-axis (rotational direction of ball valve) in degrees/second and lateral acceleration. The lateral acceleration parameter was chosen primarily as an attempt to verify the accuracy of the yaw rate measurement in determining the beginning and end of a movement. Since the acceleration parameters read *1* if the axis in question is stood vertically (corresponding to 1g of acceleration) and *0* if horizontal, the output in this case was expected to read *1* when the rotary valves were closed and *0* when they were open. Refer back to Figure 3.6, where the long edge of the IMU circled in blue indicates the x-axis.

Example data is provided in Figure 4.12 below, which shows both data streams plotted over a two-movement test. Note how the acceleration measurement goes from 1 to 0 as the valve opens and then back to 1 as it closes again. For the angular velocity measurements, a spike is seen at the onset of a movement which then gradually decreases as the valve comes to a stop. Clearly, as seen particularly at the onset of the first "open" movement in this plot, both parameters line up in their indication of the beginning and end of a movement.

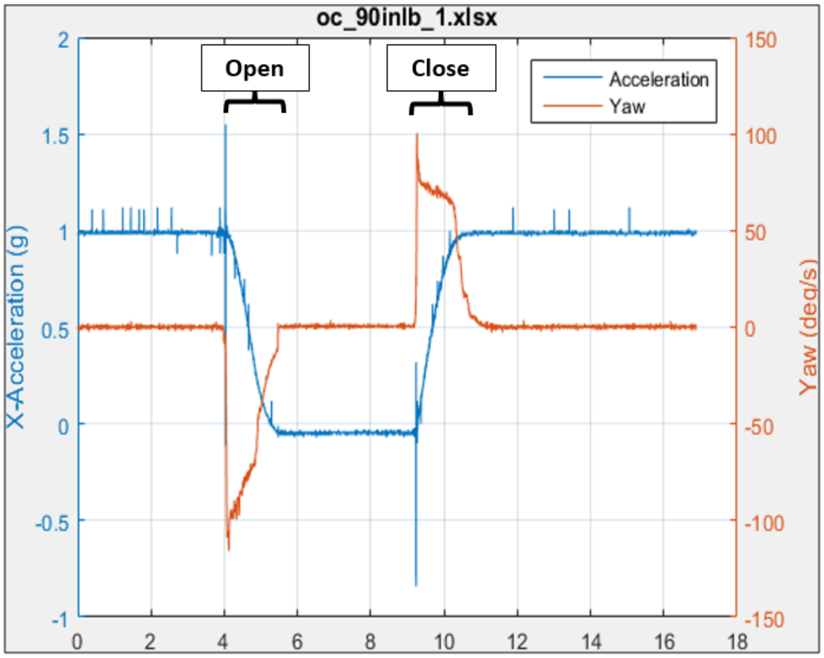


Figure 4.12: Yaw rate and X-Accel data gathered over two movements, an open followed by a close.

The next step was to define and extract simple features to be used for classification purposes. Figure 4.13 below provides a comparison of the yaw rate data for closing movements in each class considered for the small rotary valve. If we take where the movement lifts from 0 as the start point and where it reaches 0 again as the end point, we can make a rough calculation of movement time, or *MoveTime*. The approximations for these three movements are given in Figure 4.13. Clearly, there is a discrepancy between each movement. It makes sense intuitively for a hard-blocked movement to be shorter than a baseline movement since the valve stroke is abruptly stopped before expected. Likewise, the time taken by the valve as it attempts to push through material in a soft block would naturally extend the time of the movement. Although in the latter case there might be some crossover as the material in question varies in strength, overall it is reasonable to expect *MoveTime* to be a quality predictor and it was included as a result.

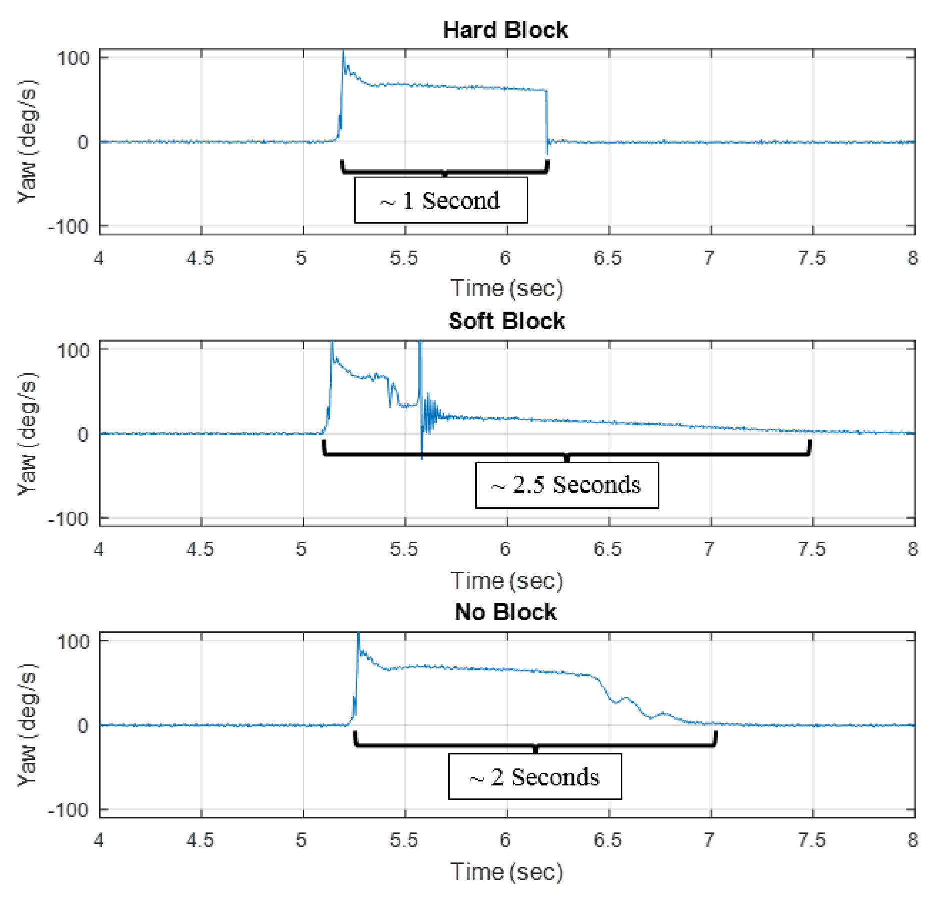


Figure 4.13: Comparison of the Yaw rate data stream for the 3 classes of close movements. Data taken from small rotary valve.

Looking at these three movements, it is also clear that the time-series signatures are significantly different. The angular velocity in the hard block just barely starts to even out and is immediately stopped before it has time to descend to 0. For the soft block, the behavior induced by the valve pushing through the blocking material causes a great deal of variation in the velocity. Finally, the no-block signature is similar to that of the hard block but with the addition of a gradual descent to 0 degrees/second as anticipated at baseline. Given that there is little variation in the hard block, a significant amount in the soft block and somewhere in the middle for the no-block, the standard deviation of this data for each movement was extracted for use as a predictor in the classifiers. The mean angular velocity was also calculated for convenience. These two parameters were also extracted for the x-acceleration data. Comparisons of the 3 movement classes for this data stream are shown in Figure 4.14 below.

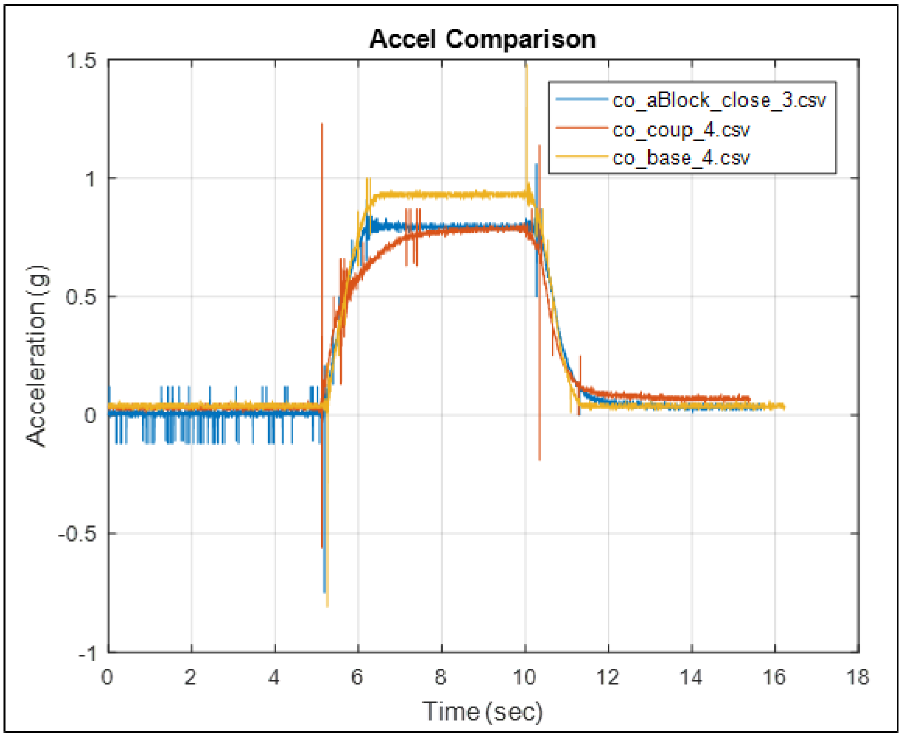


Figure 4.14: Comparison of the X-Accel data stream for the 3 classes of close movements. Data taken from small rotary valve.

Given this plot, it is clear that there are some differences between the 3 examples used. The maximum x-acceleration for the blocked movements, for example, never comes close to 1, which would be the value if the closing movement was fully realized in this case. There are also some slight differences in how each case rises to its maximum value. And furthermore, the decent back to zero (the second, open movement) is relatively consistent as expected since there were no blockages during open movements for any of these 3 examples. Overall, however, the differences between the 3 distinct cases are not very obvious, and become even less obvious when quantified in standard deviation and mean calculations. As a result, they ultimately were not found to be useful predictors and this acceleration data was not acquired from the rotary valve in an attempt to boost the sample rate from the Arduino serial monitor. The statistical justifications behind this choice are described in the last subsection of this chapter.

Although data from the IMU lent itself well to feature extraction for the rotary valves, it was found to be almost unusable for the sliding-stem valve. As described in the previous chapter, the IMU was attached to the stem of this valve with the goal of recording acceleration in the direction of stem travel and potentially integrating to obtain a velocity profile. Figure 4.15 below gives an example set of this data which was collected from the sliding-stem valve in this manner.

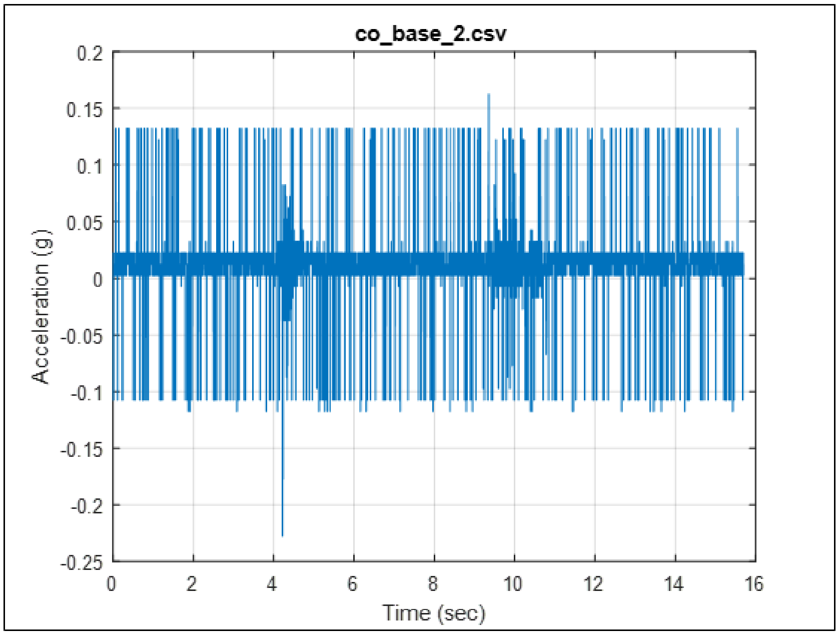


Figure 4.15: Lateral acceleration data obtained from IMU's location on sliding-stem valve shaft.

While there is some indication of each of the two movements in this dataset, this data overall is essentially worthless. The lack of resolution and noise makes integration an afterthought, and it is almost impossible in some other cases to obtain a reasonable estimate of the beginning and end indices for each movement. When physically observing this valve move, it was noted that the total range of movement is actually quite small given the small size of the valve. Furthermore, it does not appear to travel very fast during any given movement. As a result, the IMU barely registered the movements and produced the data above. Because of this, no IMU data was used in classifications for the sliding-stem valve, making it the lone valve subject to classification with only frequency-based features in this work.

Finally, it must be noted how exactly the feature extraction processes were performed in the algorithm developed to aid in the automation of this work. Given the relatively low resolution of the IMU data, it was difficult to program a procedure which could accurately locate the start and end points of each movement. The fact that each test contained two movements added to this difficulty. As a result, it was required that the user of the algorithm manually identify the start and end indices of each movement using the *ginput* function in MATLAB. These points were then organized accordingly so that the movement time, mean and standard deviation calculations could take place. This process was performed individually on the ACCEL and IMU data for each test set.

For the calculation of movement time, the yaw rate data was used as it made identifying these indices relatively easy as seen in Figure 4.13. Specifically, it is not particularly difficult to see where each movement begins and ends by observing this data. By approximating the start and end points of each movement, a simple subtraction can be performed to calculate total movement time. However, as particularly evident in the soft block data in Figure 4.13, and to a lesser extent in the no-block data, there were not always clear cutoffs for finding the end point of the movements. The potential for human error in this identification process could introduce significant variation to the parameters from test to test.

As a result, the process was performed multiple times on the training datasets to confirm that this variation did not significantly affect the training results. Furthermore, it was repeated 3 times for each test dataset to check for significant differences in test classification accuracy. Finally, the file names of each set of movements were not displayed during this process in an attempt to reduce potential bias in the index selections. While this procedure is not ideal, it provides an opportunity to test if the classifiers used could not only be accurate in training but robust to variation in feature data from future tests.

CS Features

While the ACCEL and IMU data were used together for feature extraction and blockage classification, control system data from the DVC was gathered and used in a separate classification study using the same general classifiers. As part of the primitive diagnostic capabilities built-in to the software provided with some of these valves, 3 parameters can be output during each test for monitoring. These variables are *Travel%*, *Drive%* and *Supply Pressure*. Travel% quantifies how open the valve is as a percentage, i.e. 100% indicates that the valve is completely open and 0% indicates that it is completely closed. Drive% is a measure of how hard the processor is pushing the current-to-pressure transducer in the DVC. Supply Pressure indicates just that – the air pressure being supplied to the actuator. This data is supplied at a low sample rate of 6.67 Hz. By only receiving data every fifteen hundredths of a second, the data for an entire blocked movement could consist of as few as 6-7 data points for each parameter.

For typical movements, these parameters appear to be remarkably consistent over time. Figure 4.16 below gives the plots of 3 separate 2-movement tests done at baseline (i.e. with no blockages). This data appears to line up almost exactly aside from a deviation in supply pressure in one of the tests.

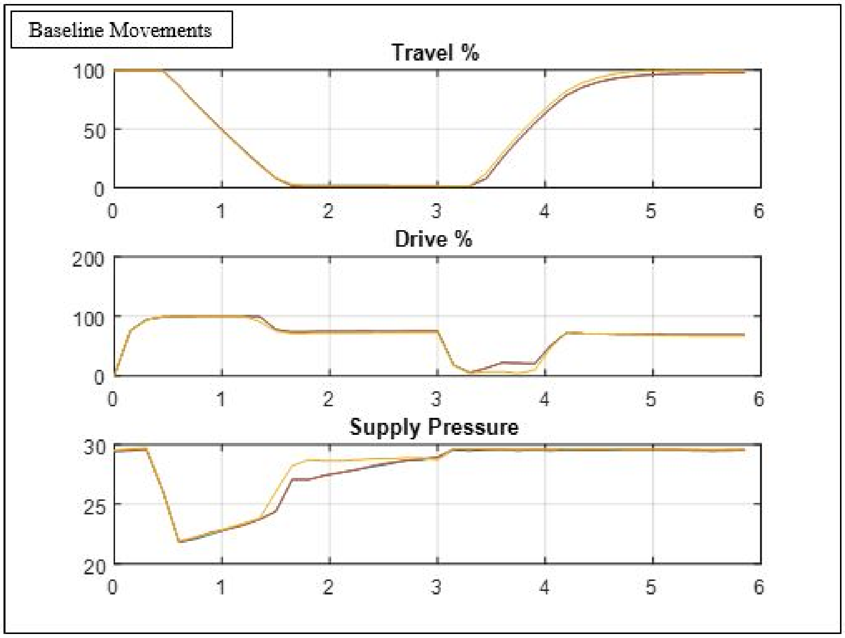


Figure 4.16: Three baseline tests plotted on top of each other for each CS parameters.

Figure 4.17 below provides the same plots as Figure 4.16 but for hard and soft blockages as indicated. Aside from some specific differences in the Drive% parameter, there do not appear to be many obvious discriminating features aside from stop points in the Travel % parameter.

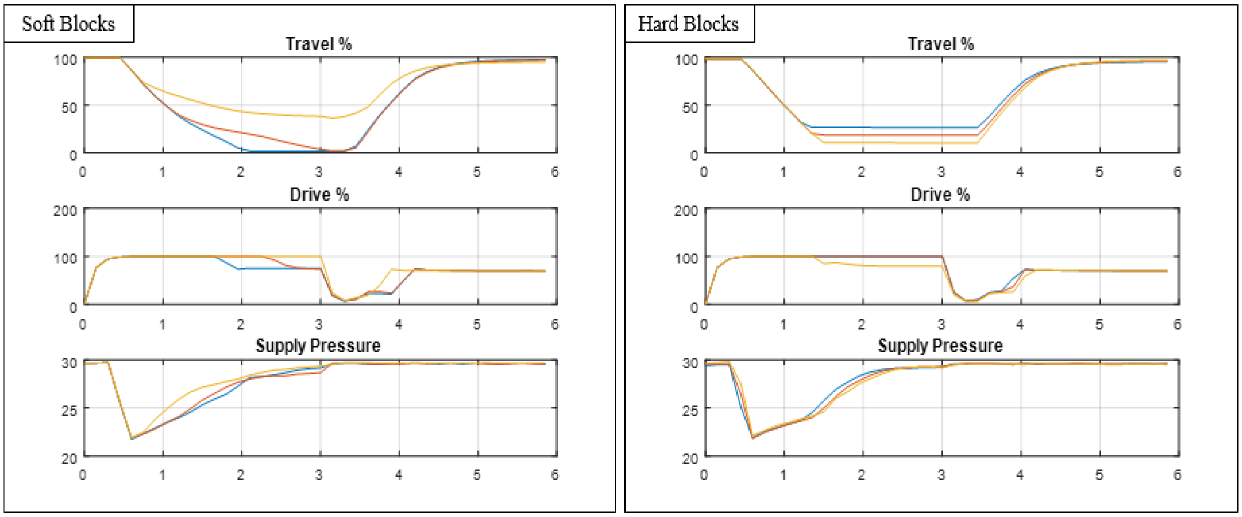


Figure 4.17: Three soft block (left) and hard block (right) tests plotted on top of each other for each CS parameters.

However, for consistency purposes, the mean and standard deviation of these three parameters were collected as classification predictors. To ensure that the calculations which took place only occurred over the interval during which the valve was actually moving, code was written to automate a process which could find the start and end points of each movement. Given the low sampling frequency of the data in question, this was able to be done with relative ease by taking the interval in the Travel% parameter which is not constant. An example showing the results of this analysis is given below in Figure 4.18. Along the x-axis, green points indicate the start of a movement while red points indicate the completion of one. Note that these intervals line up well with the apparent movement interval seen in the Travel% plot.

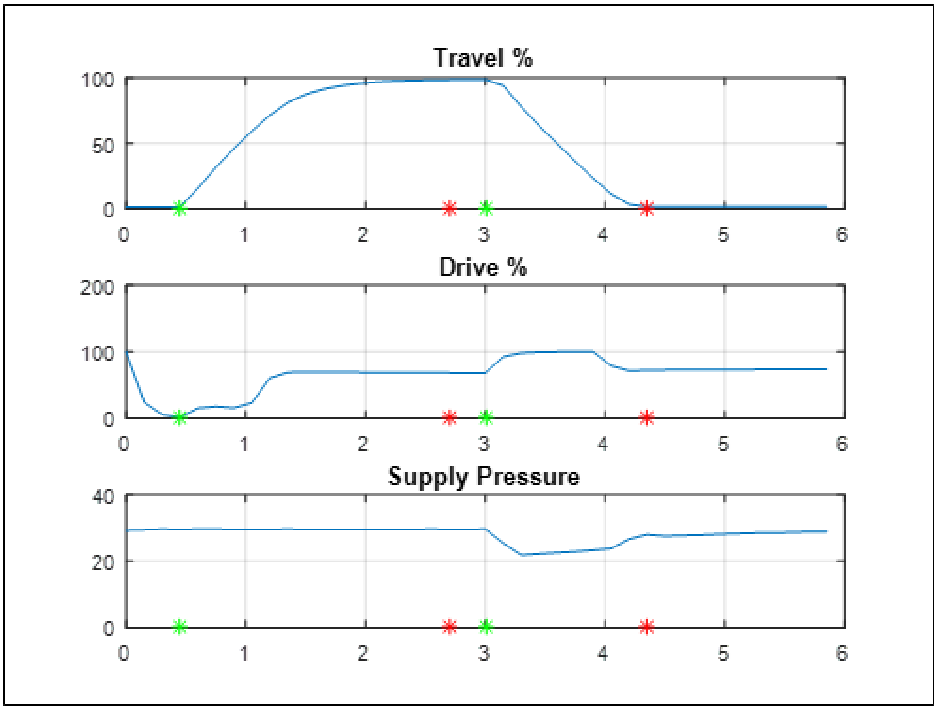


Figure 4.18: CS data from a baseline test with asterisks indicating the beginning (green) and end (red) of each movement as calculated by the algorithm.

Using these calculated intervals, a movement time calculation can easily be made and was added to the list of features to be predictors. As a result, a total of 7 features, the means and standard deviations of each CS parameter in addition to movement time, were collected for classification over the movement intervals found. Lastly, it is noted again that the CS data was only collected for the small-rotary and sliding-stem valves, as the diagnostic capabilities of the large rotary valve were inaccessible.

Final Selection for Statistical Classification

Once a complete set of potential predictors was defined and collected for each valve and each set of sensors, the logical next step would be to optimize the combinations of these parameters which yield the best results. While having more predicting variables can help further describe differences between classes, variables can be shown to have no effect on what class a dataset belongs in. When this is the case, the variable can actually add confusion to the classifier and reduce classification accuracy. As a result, it is in the best interest of researchers to remove these parameters which fail to explain data variance.

In order to do this, an analysis of variance (ANOVA) can be performed on each variable to determine whether or not it helps determine the correct class. More specifically, ANOVA can be used as a test for equal class means, or the hypothesis test that the mean of a single variable does not change regardless of what class each specific observation belongs to. Using software such as Statistical Analysis System (SAS), these hypothesis tests can be performed and the variables for which there is the most evidence for group mean discrepancies can be singled out for inclusion in the classifier models. An example ANOVA being performed on this data is shown below in Figure 4.19.

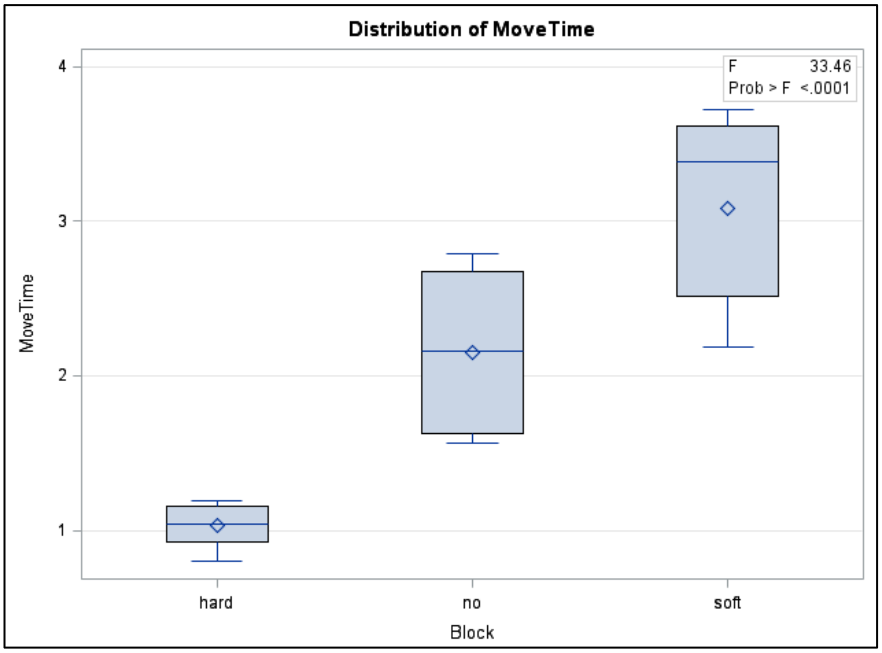


Figure 4.19: ANOVA boxplot results for movement time with hypothesis test results given also (top right).

Data taken from small rotary valve.

The boxplots shown correspond to the collection of MoveTime observations for each respective class. Given the discussion in the IMU features subsection regarding movement time, these results seem to make sense in that hard blocks are shorter movements than baseline and soft blocks are longer. This analysis takes a step further by applying the hypothesis test of equal group means, with the results given at the top right of Figure 4.19. The p-value resulting from this test is less than 0.0001, indicating that there is significant statistical evidence at the 99.999% confidence level that the null hypothesis of equal group means can be rejected. In other words, we are extremely confident that the MoveTime parameter varies significantly in at least one of the classes. By observation, this parameter appears to vary in all three classes. This doesn't need to be the case in order to yield a significant result, however, as shown in Figure 4.20 below.

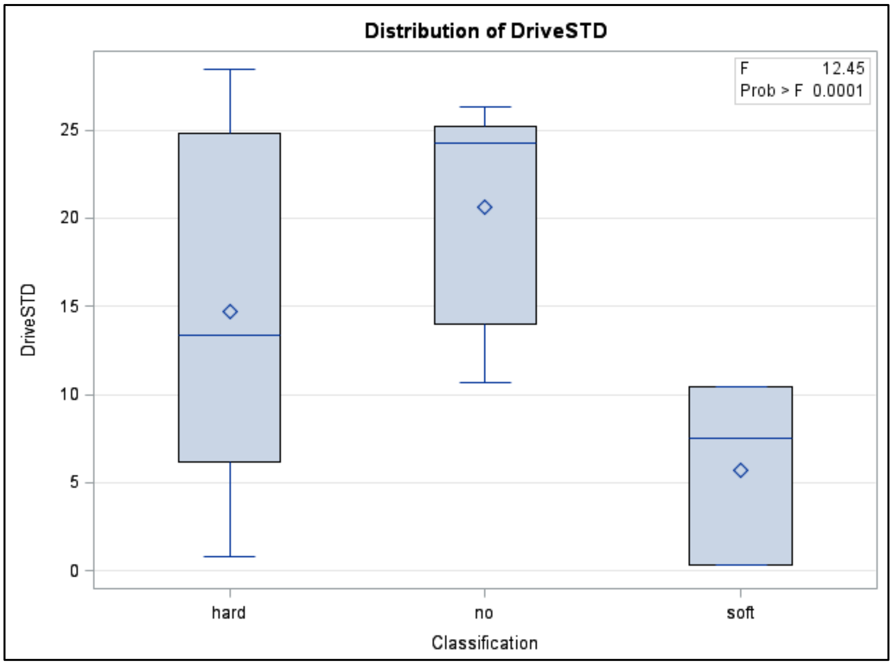


Figure 4.20: ANOVA boxplot results for standard deviation of Drive% with hypothesis test results given also (top right). Data taken from small rotary valve.

While the p-value resulting from this test is still extremely small, it is clear that the no-block versus soft block distinction is the source of this result. The conclusion is that only one class needs to vary from the others in order to yield a significant test result. Although it isn't ideal that all three classes do not vary from one another in these variables, they can still be effective predictors when combined with other parameters if there is only a single key distinguishing class.

With this concept in mind, two sets of ANOVA testing were performed on the data from each valve. One set was on the combined ACCEL/IMU feature data and the other on the CS data. The notable exceptions include the CS data for the large rotary valve, which was not available, and the lack of IMU data for the sliding-stem valve. Using the ANOVA results, the parameters which yielded the most significant test results were then chosen for classification. In some cases, many of these tests yielded significant results for a single set of parameters. Different combinations were subsequently used for training, with the best one for each set chosen for inclusion in this thesis. This process will not be expanded upon, but it further lends to the point that there is a level of intuitive, observation-based decision-making involved in the choosing of quality predictor variables for classification. A summary of the variables chosen for each test set and each valve is given in Tables 4.1 – 4.3.

Table 4.1: Summary of features used in final classifier training and testing exercises for the small rotary valve.

Small Rotary Valve	
ACCEL/IMU Data	*CS Data*
Frequency A (300-500 Hz Spike)	MoveTime
STDYaw	DriveSTD
MoveTime	PressSTD

Table 4.2: Summary of features used in final classifier training and testing exercises for the large rotary valve.

Large Rotary Valve	
ACCEL/IMU Data	*CS Data*
Frequency Base (~ 59 Hz Spike)	***N/A***
Frequency A (~ 340 Hz Spike)	
Frequency B (~ 2850 Hz Spike)	
STDYaw	
MoveTime	

Table 4.3: Summary of features used in final classifier training and testing exercises for the sliding-stem valve.

Sliding-Stem Valve	
ACCEL Data	*CS Data*
Frequency A (150-400 Hz Spikes)	MoveTime
Frequency B (650-700 Hz Spike)	DriveSTD
Frequency C (1800-1900 Hz Spike)	DriveMean
	PressSTD

Chapter 5. Classification Results & Discussion

Introduction

This chapter serves to describe the results of the 3 classification methods, FIS, ANN and statistical, for the respective valves they were performed on. To begin, the results of the FIS and ANN classifiers, which served as groundwork analyses for the later efforts, are given. More detailed explanations of the statistical classification results for each of the 3 valves are then provided. Given the classification accuracy and procedural parallels of the statistical classifier methods for each valve, these approaches are the focus of the results and discussion as indicated in the introductory chapter. This was intended once the analysis pivoted to considering features representing entire movements as opposed to smaller segments.

FIS Results

As described in previous chapters, classification using fuzzy inference systems was performed in a real-time simulation manner. ACCEL datasets which were pre-recorded at 48 kHz were taken and looped through in intervals of 5000 data points. At each interval, calculations were made and features were mined before being input into the FIS systems. This approach provided the opportunity to observe whether or not real-time data inputs were useful in small segments, given the potential advantages that could arise. For example, a CBM-based solution may not be properly equipped to determine the beginning and end of a random valve movement in time. However, if the data could be continuously analyzed without manual interventions, this problem potentially becomes irrelevant.

Unlike the ANN and statistical classification methods, no training sets of data are required when using an FIS for classification. Inputs and membership functions are simply defined as described in the chapter on analysis techniques. For this work, two distinct fuzzy logic structures, one for distinguishing between open and close movements and the other for classifying the movements, plus a separate mechanism for handling soft blocks, were needed. As a result, the first FIS determines movement status, and this status is further used as an input to the classifying FIS. The Gaussian membership functions described for the hard blockage portion of this structure are shown below in Figure 5.1.

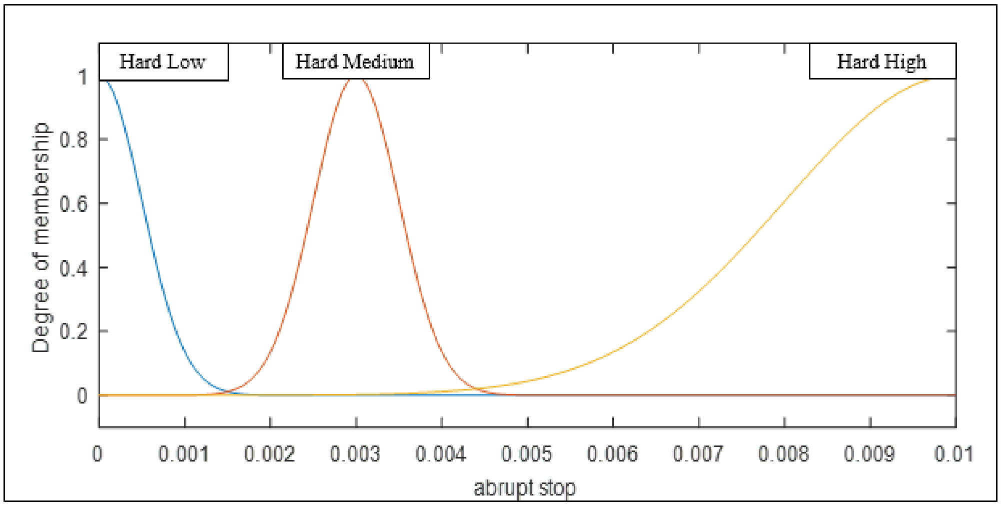


Figure 5.1: Input membership functions for the input to the hard blockage FIS for a closing movement.

As described in Table 3.1, a total of 97 movements from 44 data sets were analyzed for classification using both the FIS and ANN methods. These consisted of 30 hard blocks, 17 soft blocks and 50 no-blocks. Once the relevant fuzzy structures were defined, these data were input to an algorithm which looped through the data and made classifications after every loop. Two main flags were monitored for accuracy evaluation: movement direction and blockage classification. Since movement direction was a key input to the classification FIS, it was important to know if it was being determined correctly. An example of the monitoring for this testing is shown below in Figure 5.2.

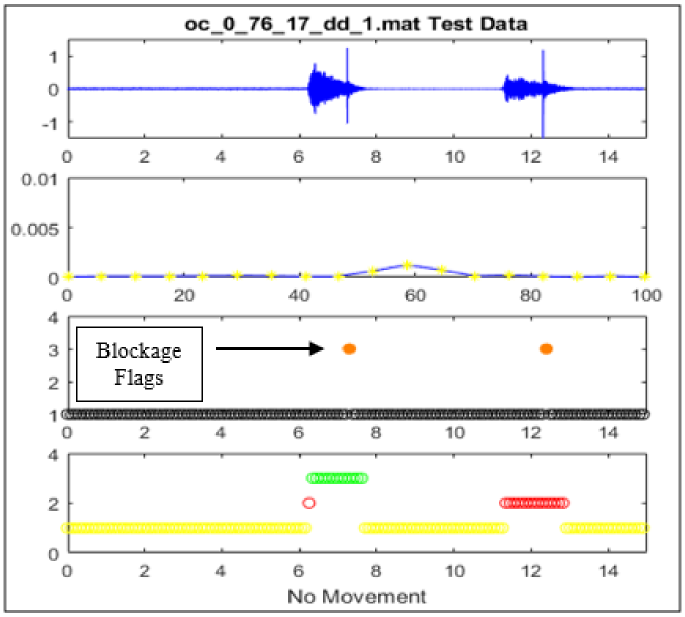


Figure 5.2: Example of the real-time monitoring setup algorithm for a test set of two hard blockages.

The configuration of the plots displayed in Figure 5.2 is as follows. The first plot contains the raw time-series vibration data of the movements associated with the data file in question, and the second plot contains an FFT of the current data subset being analyzed in the algorithm loop. In this case, the FFT was set in the 0-100 Hz range for observing some of the low frequency spikes seen in slow blockages as discussed previously. The third plot gives the flags associated with each type of block. In this case, a value of 1 indicates there is no blockage, 2 and 3 indicate various levels of hard blockage, and 4 indicates a soft blockage. For the data being analyzed in this plot, both movements recorded contained hard blockages which stopped the valve significantly before its intended stop point. As a result, the moments during which each hard blockage occurs are properly flagged by the fuzzy logic algorithm.

Finally, the fourth plot exists to monitor the movement direction of the valve. The yellow points with a value of 1 indicate no movement, the green points with a value of 3 indicate that the valve is opening, and the red points with a value of 2 indicate that it is closing. On the tail ends of many open movements, some points were seen to be classified as closing. This is likely the result of certain vibration-based classification parameters drop in magnitude as the valve just begins or ends a movement.

For the set of 97 movements, total classification accuracy was calculated for both movement direction and blockage classification since there were separate fuzzy logic structures for each. Movement direction accuracy was calculated by observing the points associated with actual movements from the fourth plot in Figure 5.2 and determining how many points were incorrect (i.e. a “close” flag occurring during an opening movement). This resulted in 1291 individual classifications, or approximately 13 for each movement. Classifications for “no movement” were not included since they are very simple for the classifier to determine and would inflate the reported accuracies.

For blockage flags, the movement as a whole was considered. In other words, if a hard blockage was correctly indicated for a movement in which one occurred, the movement would be considered to be classified correctly. If no blockage was expected but some blockage flag was raised, the movement would be considered to be classified incorrectly. Using these methods, Table 5.1 was constructed to summarize the results of the fuzzy logic testing for movement determination and blockage classification.

Table 5.1: High-level results from FIS testing.

	Classifications Made	**Incorrect Classifications**	**Percent Correct**
Open/Close	1291	110	91.5
	Movements Classified	**Incorrect Classifications**	**Percent Correct**
Blockages	97	14	85.6

Overall, we find the fuzzy logic structures to be effective in determining valve movement direction and relatively accurate for classifying blockages, though with room for improvement. There were a few stand-out statistics that help explain many of the inaccurate classifications seen in the above testing.

A handful of the movements used in this set of classifications were *partial* movements, meaning that the valve's intended stop point was not one which opened or closed the valve completely. For example, some commands had the valve open to 50%, then close back to 0%. These two movements would be considered partial. Partial movements were an interesting wrinkle since the typical time and frequency magnitudes which were used as predictors would not reach the same values associated with complete movements. As a result, they proved to be more difficult to handle, with 41% of the total incorrect open/close classifications corresponding to partial movements although only 27% of the total movements classified were partial. It should be noted that partial movements were not analyzed after FIS and ANN classification and were assumed to be atypical in normal control valve operation for simplicity.

As discussed briefly above and seen in Figure 5.2, it was common for misclassifications with respect to movement direction to occur at the beginning and end of open movements which resulted in lowered frequency magnitudes. This explains the majority of misclassified movement directions, with 95% of those attributed to this issue.

For the 14 misclassified blockages, 5 were misses and 9 were false positives. The misses are attributed to hard blockages for which the valve stops were set to a point very close to the intended end point of the movement. In cases such as these, the valve has already slowed movement significantly by the time it is blocked and certain frequency magnitudes are not excited to the extent generally associated with most other hard blockages. When the valve is being blocked within 5% of its intended stop point, leakage and related consequences would be minimal, and all hard blockages that would result in significant leakage were detected and classified correctly. The majority of the false positives seen were from baseline open movements directly following a closing soft blockage. In these cases, the valve had to still open out of the crushed objects, and these interactions are likely the cause of the false flags. In fact, one could make the argument that these objects are still, in effect, blocking the valve and these movements could be considered to be correctly classified.

ANN Results

The ANN was originally proposed in an effort to seek more accurate results once the FIS testing was completed. Although relatively accurate, the drawbacks of fuzzy logic structures make them unsuitable for classification with many inputs and classes. Soft blockages, for example, required a separate logic structure from hard blockages since the predictors used to classify them were different and thus could not be mixed with other classes. Additional faults would likely require additional logic structures to accommodate them. Neural networks, on the other hand, provide a more general approach and are relatively easy to implement for classification. While a separate ANN was made for identifying movement direction for consistency, both hard and soft blockages could be part of their own network as in most classifiers.

The neural networks developed for movement identification and blockage classification consisted of a single hidden layer of 20 neurons given the relative simplicity of the data sets being used. A sample of approximately 1000 data points were supplied to each for training, and resulted in quick and efficient training times (< 1000 epochs). The training data consisted of all frequency and time domain features needed to identify movement direction and detect blockages to a reasonable point of accuracy in the FIS’s. Subsequently, the same 44 data sets of 97 movements were supplied to the trained networks and evaluated in a manner identical to that which was described for the previous section, with outputs similar to those given in Figure 5.2 monitored as well. The results were gathered in Table 5.2 below.

Table 5.2: High-level results from ANN testing.

	Classifications Made	**Incorrect Classifications**	**Percent Correct**
Open/Close	1230	21	98.3
	Movements Classified	**Incorrect Classifications**	**Percent Correct**
Blockages	97	14	85.6

Immediately observed is the marked improvement in the classification of movement direction. The use of training data to help the networks learn to recognize data at the tail ends of opening movements, which proved to be an issue with the FIS testing, clearly improved overall classification in this regard.

The ANN did not provide additional accuracy in classifying movements into respective classes of blockages, as it yielded identical results to the FIS classifier. However, misclassifications did not correspond to the exact same set from FIS classification done before it. Overall, the amount of misses increased while the number of false positive decreased. Once again, all misses took place during hard blockages which happened very close to movement completion, and all soft blockages were correctly classified.

Given these results, as well as those from the FIS's, there is still room for improvement particularly in the domain of blockage classification. Furthermore, the real-time simulation approach used with these two sets of classifiers comes with some significant drawbacks. For one, the individual analysis of many small segments of data within a movement may mask key features which are more evident over an entire movement. Additionally, this approach is very computationally intensive because of the need to continuously monitor if transitioned to a more realistic setting. Many control valves may sit idle for a very long time, and recent efforts have shifted focus to more power-friendly approaches to CBM technology [9]. As a result, the approach of identifying the beginning and end of a movement and treating it as a single entity for feature extraction and FID is not only simpler but could prove to be more effective as well. With this in mind, this research transitioned to this approach using the statistical classifiers described in Chapter 2 for classification.

Statistical Classification Results

As summarized in the final pages of the previous chapter, there were 5 main sets of results which came from the analysis of test data using the geometric, instance-based and statistical classifiers previously introduced. These results correspond to the two-sensor instrumentation data from each of the 3 valves used, plus control system data from the small-rotary and sliding-stem valves (recall this data was not available for the large rotary valve). This section serves to describe these results, organized by valve, and discuss what they mean in some detail.

An overview of statistical classification process is as follows. Classification using data from each valve, both sensor-based and control system-based, was performed by first taking a representative sample of data for training. The training results were characterized by an *estimated-actual error rate* (EAER), found by performing a 5-fold cross-validation process. This procedure partitions the sample 5 times, with 4 being used as training data and the final being used as test validation data. This is performed 5 times, once for each subsample to obtain a final EAER. The cross-validation process is used to prevent model over-fitting common in simple re-substitution validation procedures, which often overestimate accuracy rates [60]. The classifiers which performed best in this training process were then exported for use.

The remaining test data was input to an algorithm which collected extracted features for input to these chosen classifiers. Given the manual element of this process described in the IMU features section in the previous chapter, this testing procedure was performed 3 times for the two-sensor data from each valve. The output classifications from the tests were then compared to the actual known classifications and an error rate for each of the 3 repetitions was calculated. From here, the final accuracy rate was found by averaging the rates from the 3 repetitions. The nature of the CS feature extraction process was more algorithmic in nature and thus was not subject to repeated testing to confirm consistency. This gives a set of accuracy rates (one for each classifier chosen) for each data type (sensor or CS) and each valve (small rotary, large rotary and sliding-stem). Since the large rotary valve did not provide CS data, a total of 5 sets of results, summarized by Tables 4.1 – 4.3, were subsequently created to summarize this data. This process of evaluating performance on training data followed by validation using test data is standard in machine learning applications.

Small Rotary Valve

Of the recorded movements described for the small rotary valve in Table 3.2, there were 8 hard blocks, 5 soft blocks and 15 no-blocks selected for a training set of sensor-based data. These data were selected in a manner which was representative of all sensor-based data collected for this valve. Figure 5.3 below gives a scatterplot comparing the *STDYaw* and *FrequencyA* parameters, with data colored by class to observe separation. Clearly, there is some visible separation between the classes when visualizing the data in the context of these two variables. Furthermore, this plot does not include the *MoveTime* parameter which was also used for classification of this data. Given the ANOVA results outlined in the previous chapter, it was hoped that this variable made the classes boundaries seen in Figure 5.3 even more distinct.

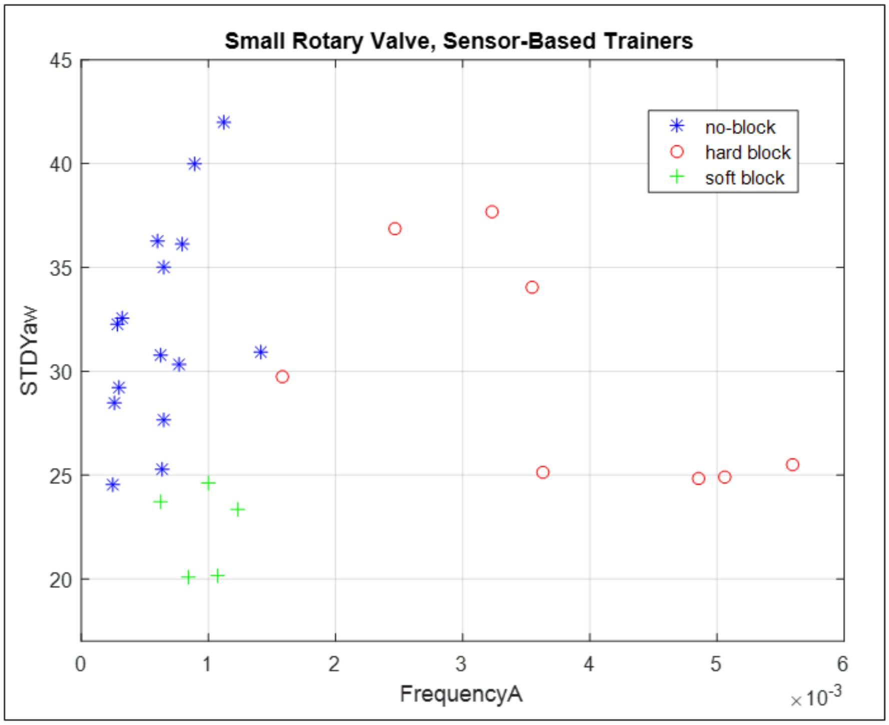


Figure 5.3: Scatter plot of sensor-based training data from 2 parameters for the small rotary valve.

Using the *Classification Learner* application in MATLAB, one is able to quickly use this data to train classifiers and view error rates as well as which data points in particular are being misclassified. After utilizing all available multi-class methods, the 4 classifiers which gave the lowest EAER were exported for classifying the rest of the data. For this data, the best classifiers were fine kNN (k = 1), quadratic discriminant, Gaussian SVM and bagged tree ensemble methods.

The rest of this data was subsequently input to an algorithm which took the ACCEL and IMU data from each test, extracted relevant features and input to these four classifiers for classification. As explained previously, this process was done 3 times because of the manual movement-selection component for identifying the beginning and end of movements. The final results for the sensor-based small rotary valve data are given below in Table 5.3. All classes are included in these results.

Table 5.3: Sensor-based small rotary valve accuracy rates.

Small Rotary Valve: Two-Sensor % Accuracy				
	Fine KNN	*Quadratic Discriminant*	*Gaussian SVM*	*Bagged Tree Ensemble*
Run 1	98.8	97.6	98.8	97.6
Run 2	100.0	96.3	96.3	97.6
Run 3	100.0	98.8	98.8	97.6
Averaged	**99.6**	**97.6**	**98.0**	**97.6**

Clearly, the chosen classifiers were very accurate in classifying the remaining 82 movements not utilized in training, with only about 2 misclassified movements on average. In all cases, the misclassifications corresponded to soft-blocked movements which were classified as no-blocks. These errors came from soft blocks induced with very light, flexible tubing which was the most easily crushed by the valve. As a result, the valve response recorded by the sensors was not very different from a baseline movement and could not, in these limited cases, be distinguished from one.

For the CS data, 5 hard blocks, 7 soft blocks and 20 no-blocks were compiled as a training set. Using these data, the same procedure was performed with respect to the training, selection and testing of final classifiers. Figure 5.4 below gives a scatter plot of this training data comparing the *DriveSTD* and *MoveTime* parameters (*PressSTD* was also used for classification). At first glance, there does appear to be some separation given by these two variables. Given the very low 6.67 Hz sampling frequency at which this data was collected, there was concern that the algorithm written to extract *MoveTime* might blur the line between no-blocks and soft blocks. This appears to be the case, although it was able to uncover a distinction between these movements and hard blocks, with the latter clearly giving a smaller total movement time.

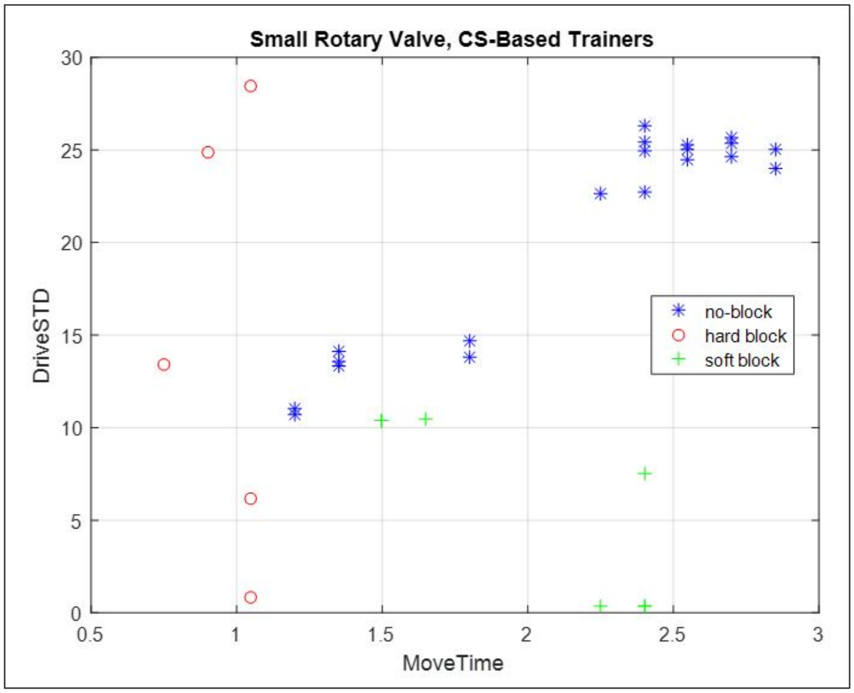


Figure 5.4: Scatter plot of CS-based training data from 2 parameters for the small rotary valve.

The classifiers which trained with the best EAER were the fine kNN, weighted kNN, Gaussian SVM and bagged tree ensemble methods. Final results are given in Table 5.4 below. KNN methods clearly proved superior in classifying the remaining 62 movements, although all final classifiers performed well, with the only misclassifications once again attributed to soft blockages with particularly weak material that was cut through entirely.

Given the low sampling rate of this data, which results in few data points to consider for each movement, the validity of these results are put into question as there did not appear to be many visible differences in the data as Figures 4.16 and 4.17 suggest. However, movements of the same class were seen to be remarkably consistent in nature, indicating that the control system supplies pressure and current in similar patterns given these different types of induced faults. This would be a good thing, but also means that these classifiers are at risk of being over-trained. It should be re-stated, however, that a variety of materials with different properties were used in the seeded-fault testing of soft blocks and different levels of hard blocks were also tested in an effort to provide variation to the data sets. Furthermore, the lack of observational variation in this data does not mean that there are not underlying characteristics uncovered by the standard deviation and movement time calculations which separate classes effectively.

As a result, there are two main interpretations that can be made. The first is that these seemingly over-accurate results suggest that soft blocks from a range of materials and hard blocks of different levels can all be classified accurately as these results suggest at first glance. The other is that there was not enough test data to provide sufficient variation in the data sets, which in turn provided over-fit models which could be vulnerable to misclassifying future data. Interpretations such as these which are unclear require future testing and analysis to be done under more realistic conditions in order to verify them. Regardless, the steps taken in this testing and analysis, as well as the consistency in the final results, provide a promising takeaway.

Table 5.4: CS-based small rotary valve accuracy rates.

Small Rotary Valve: CS % Accuracy				
	Fine KNN	*Weighted KNN*	*Gaussian SVM*	*Bagged Tree Ensemble*
Final %	100.0	100.0	95.2	96.8

Large Rotary Valve

Of the 110 total movements analyzed from the large rotary valve, 12 hard blocks, 6 soft blocks and 26 no-blocks were taken as a training set and trained in a manner identical to that which was described for the small rotary valve. Although there were discriminating features identified through manual observation of ACCEL and IMU data as summarized in Table 4.2, the class separation was found to be significantly more blurred when compared to features taken from the small rotary valve.

An example of this is shown in Figure 5.5 below. For the small rotary valve, it was relatively easy to find a scatter plot of two variables which uncovered clear class boundaries. By adding the additional features, accurate classification could be achieved. Figure 5.5 shows the clearest case of this for the large rotary valve data, with most other pairs of variables giving class boundaries which were much more mixed. While the complete 5D feature space cannot be visualized, it is difficult to determine whether or not the complete set of features selected will separate classes effectively, and the ANOVAs performed on this data only uncover univariate separations in the data. Therefore, high accuracy levels are hoped for but not necessarily expected.

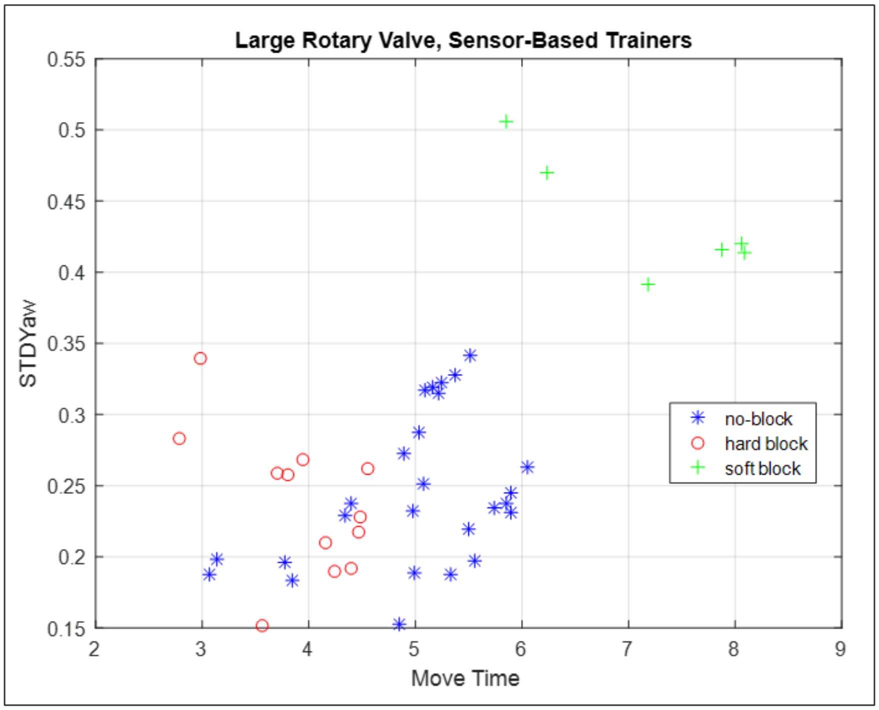


Figure 5.5: Scatter plot of sensor-based training data from 2 parameters for the large rotary valve.

The two most accurate classifiers found from the training data were kNN methods, fine and weighted. A third less accurate but sufficient cubic SVM classifier was also selected for testing. The final blockage classification results for the large rotary valve using these three classifiers are given in Table 5.5 below. Clearly, there is a significant drop in accuracy, with minimum error rates just getting under 15% as seen with the weighted kNN classifier.

Further investigation shows that these misclassifications primarily came from baseline movements which were incorrectly classified as hard blockages. There are multiple likely reasons for this. While *MoveTime* is an important parameter in determining whether or not a blockage occurred, Figure 5.5 shows how there is a mix between baseline movements and hard blocks with respect to movement time. This is because many of the recorded baseline movements occurred directly following a hard blockage. Thus, although no blockage occurred in these movements, the total time to move was smaller. This was the case in all valve testing and was one of the key reasons why additional parameters other than *MoveTime* were important in the classification process. In particular, the frequency-based parameters typically were chosen to further distinguish between these two classes of movements. Given that the classifiers were successful in doing this for the other valves, it can be reasoned that the chosen frequency-based parameters were simply not effective enough, possibly because of overlooked behavior which masked the differences between the two classes (no-block and hard block). In other words, the reduction in accuracy seen in the large rotary valve data is likely the fault of the feature selections rather than classifiers themselves.

The obvious solution to this issue would be to introduce signal processing techniques such as filtering and enveloping in an attempt to uncover features which may have been masked by noise in each movement. The size of the large rotary valve, for example, results in more force in the valve stroke and more air being used for pressure. As a result, movements were generally much louder. This could have affected the vibration data collected during each movement. However, it should be noted that band-pass filters were applied to a variety of important frequency ranges, and no potential features of note were uncovered. More advanced techniques could aid in this process in future work if similar issues are encountered.

In the end, the feature described in Figure 4.9, which shows how a frequency spike in baseline movements is not present in hard blocked movements, was in retrospect found to also be absent in baseline open movements following a hard blockage. As a result, these movements were misclassified as hard blockages, causing the dive in accuracy rate. However, true hard blockages and soft blockages, as well as data sets consisting only of baseline movements (i.e. ones which did not include *any* hard blockages) were classified correctly. This stresses the need for complete verification when defining features which are claimed to be *unique* to a certain class.

Table 5.5: Sensor-based large rotary valve accuracy rates.

Large Rotary Valve: Two-Sensor % Accuracy			
	Fine KNN	*Weighted KNN*	*Cubic SVM*
Run 1	78.8	84.8	72.7
Run 2	80.3	86.4	72.7
Run 3	78.8	78.8	66.7
Averaged	**79.3**	**83.3**	**70.7**

Sliding-Stem Valve

The sliding-stem valve, which was the final valve used in this analysis, provided only frequency-based predictors because of the lack of IMU data quality as previously described. Data acquisition from this valve yielded both sensor-based and control system data for separate use in classification. A scatter plot of two of the three frequency parameters used for training of the sensor-based classifiers is given below is Figure 5.6.

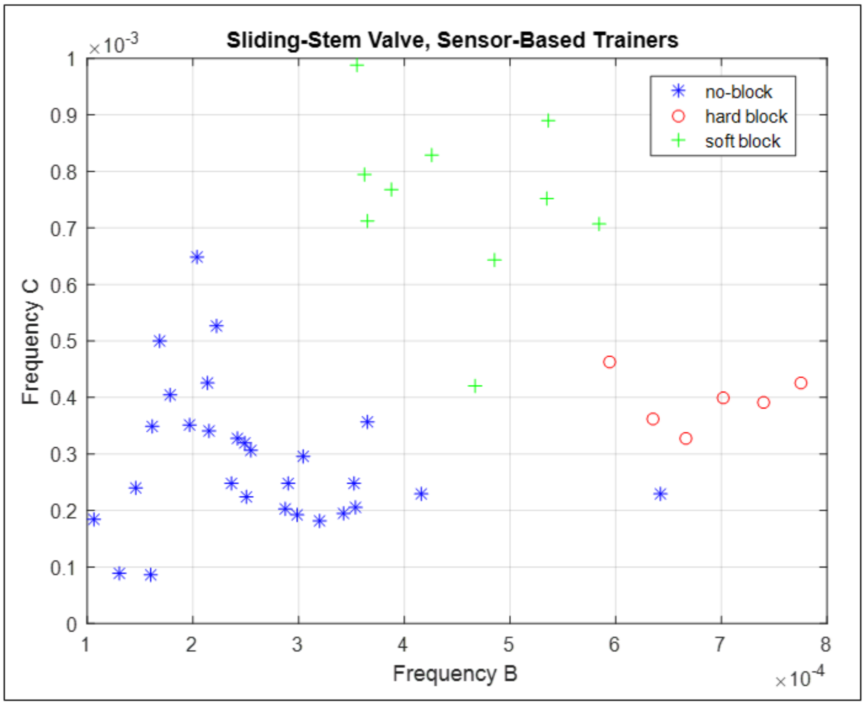


Figure 5.6: Scatter plot of sensor-based training data from 2 parameters for the sliding-stem valve.

Once again, promising separation boundaries are revealed and give some confidence that these two parameters, along with the third which was also shown to vary with blockage type with ANOVA, can be used for accurate classification. For the sensor-based (vibration only) data collected during testing, the most accurate trained classifiers chosen for testing were fine kNN, cubic kNN (cubic distance metric), Gaussian SVM and quadratic discriminant analysis. After training, the remaining 41 data sets (82 movements) were input for classification. The final results from this analysis is given in Table 5.6 below.

Table 5.6: Sensor-based sliding-stem valve accuracy rates.

Sliding-Stem Valve: Two-Sensor % Accuracy				
	Fine KNN	*Cubic KNN*	*Gaussian SVM*	*Quadratic Discriminant*
Run 1	96.1	90.2	93.1	94.1
Run 2	96.1	90.2	93.1	94.1
Run 3	96.1	90.2	93.1	94.1
Averaged	**96.1**	**90.2**	**93.1**	**94.1**

At first glance, a return to error rates under 10% is seen, suggesting that the chosen classifiers were effective in processing new data. Once again, the fine kNN classifier provides the highest accuracy out of the test classification methods. For all 4 methods, the majority of misclassified test data corresponded to baseline movements which were classified as hard blockages. Although this is similar to the problem seen in the large rotary valve, it was not an issue to nearly the same extent (3 or 4 misclassifications on average). The most likely reason for some overlap here is the fact that no movement time parameter was extracted for classification. Since only vibration data was analyzed, clear beginning and end points of each movement could not be deciphered accurately. Assuming that none of the frequency-based predictors could create a distinct boundary between these two classes, a level of misclassification should not be particularly surprising. However, this does indicate that improvement in either the frequency bands selected as features or the calculation algorithm which quantifies them into predictors may be warranted.

For the control system data, the same three parameters were used for classification as in the small rotary valve, with the addition of a fourth *DriveMean* predictor. It was hoped that the same three predictors alone could be used for both valves to demonstrate consistency between two valves with different internal mechanisms, but this additional parameter was shown to help stabilize prediction accuracy at a high level. An example of this is shown in a scatter plot of the training data in Figure 5.7 below, with *DriveMean* on the horizontal axis. While there is some overlap between no-blocks and soft blocks seen around 65%, there is clearly a reasonable amount of separation between classes. This is therefore a visual supplementation of the ANOVA results which assert that *DriveMean* changes with class and could therefore be an effective predictor for future test data.

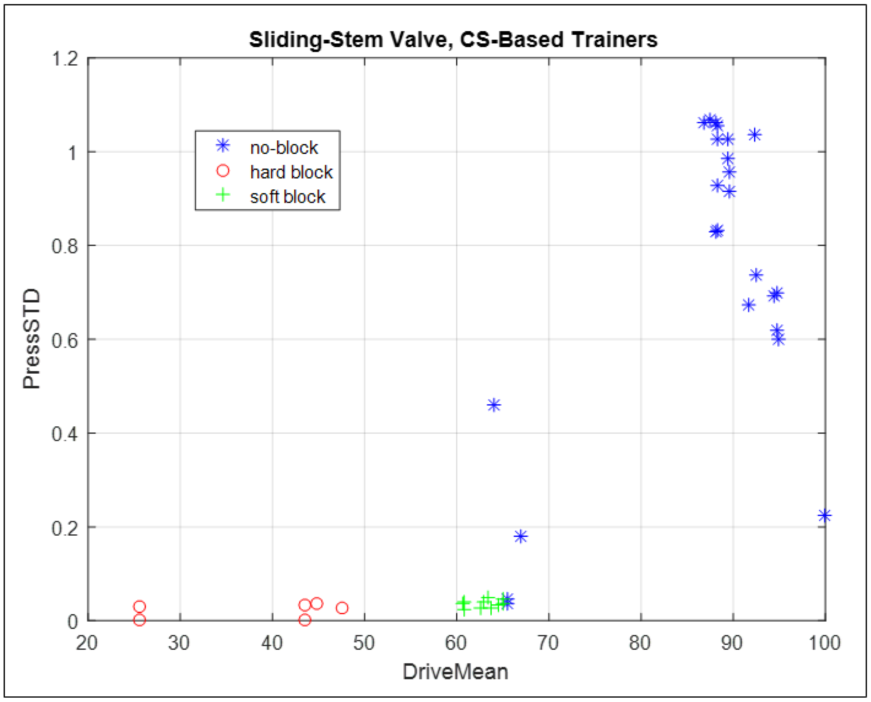


Figure 5.7: Scatter plot of CS-based training data from 2 parameters for the sliding-stem valve.

The final classifiers selected after the training process for test data were fine kNN, weighted kNN, quadratic discriminant analysis and a bagged tree ensemble. The results from the final classification of the remaining 40 data sets (80 total movements) are given below in Table 5.7.

Table 5.7: CS-based sliding-stem valve accuracy rates.

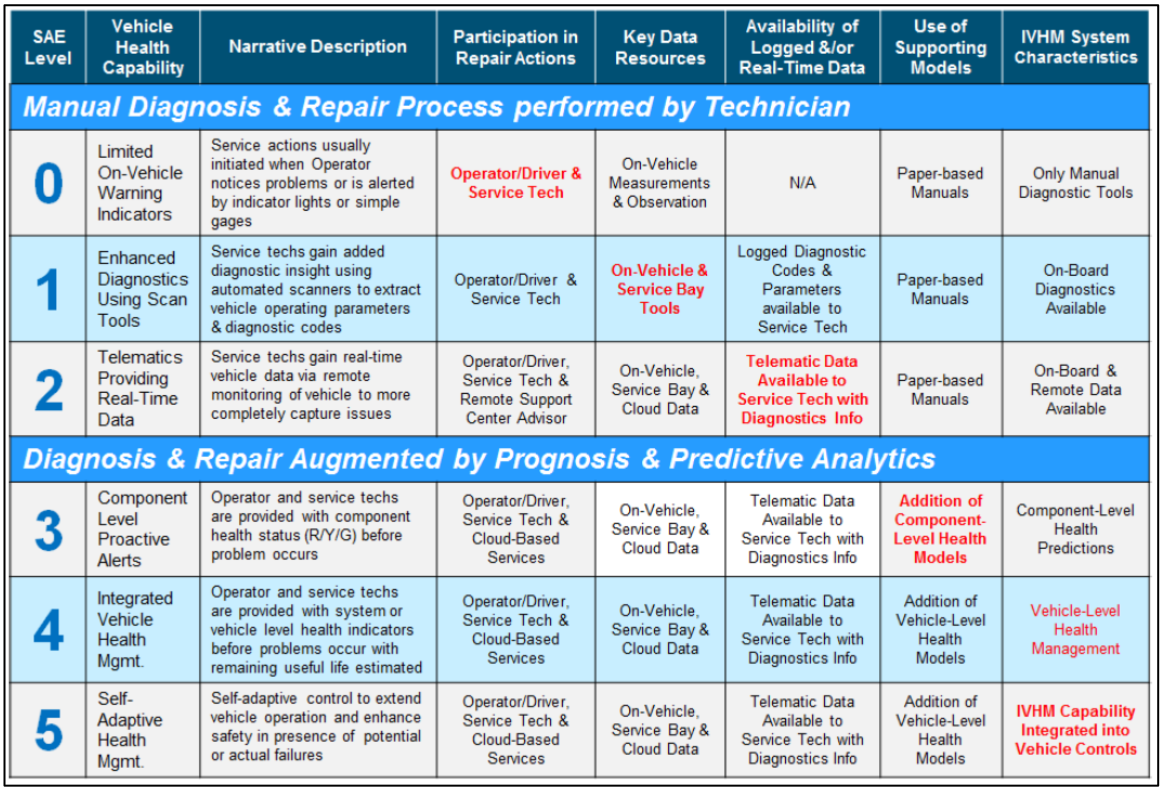
Sliding-Stem Valve: CS % Accuracy				
	Fine KNN	*Weighted KNN*	*Bagged Tree Ensemble*	*Quadratic Discriminant*
Final %	98.8	97.5	100.0	97.5

High accuracy rates are seen similar to the analysis performed on the small rotary valve. Given that different materials were used to simulate similar behavior in each of the two valves, these results further validate those provided previously using CS data. This suggests that, although primitive in nature, the CS data analyzed in this work is useful in detecting valve stroke blockages in a variety of control valve setups.

Chapter 6. Summary & Conclusions

Procedural Overview & Discussion

Since this research has explored a variety of methods to quantify predictors for control valve movement classification, a number of important takeaways can be observed. Health management systems continue to take on a more standard, structured identity as seen in, for example, integrated vehicle health management (IVHM) systems. Figure 6.1 summarizes the SAE standard for these types of systems, clearly defining distinct levels of system intelligence. Although prognostics and self-adaptive health management techniques can be the ideal solution to maintenance issues, they are often not achievable given the significant research investment which can involve complex test rigs and large amounts of time to fully understand system behaviors and implement solutions accordingly. However, enhanced manual diagnostic solutions, such as those given by levels 0-2 in Figure 6.1, are much more manageable and encompass the scope of this work. Although some of the terminology in the figure may not be directly applicable, the goal of this research was to explore solutions which fall in the range of levels 1 and 2, i.e. post-hoc analysis and real-time data monitoring.



SAE Level	Vehicle Health Capability	Narrative Description	Participation in Repair Actions	Key Data Resources	Availability of Logged &/or Real-Time Data	Use of Supporting Models	IVHM System Characteristics
Manual Diagnosis & Repair Process performed by Technician							
0	Limited On-Vehicle Warning Indicators	Service actions usually initiated when Operator notices problems or is alerted by indicator lights or simple gages	**Operator/Driver & Service Tech**	On-Vehicle Measurements & Observation	N/A	Paper-based Manuals	Only Manual Diagnostic Tools
1	Enhanced Diagnostics Using Scan Tools	Service techs gain added diagnostic insight using automated scanners to extract vehicle operating parameters & diagnostic codes	Operator/Driver & Service Tech	**On-Vehicle & Service Bay Tools**	Logged Diagnostic Codes & Parameters available to Service Tech	Paper-based Manuals	On-Board Diagnostics Available
2	Telematics Providing Real-Time Data	Service techs gain real-time vehicle data via remote monitoring of vehicle to more completely capture issues	Operator/Driver, Service Tech & Remote Support Center Advisor	On-Vehicle, Service Bay & Cloud Data	**Telematic Data Available to Service Tech with Diagnostics Info**	Paper-based Manuals	On-Board & Remote Data Available
Diagnosis & Repair Augmented by Prognosis & Predictive Analytics							
3	Component Level Proactive Alerts	Operator and service techs are provided with component health status (R/Y/G) before problem occurs	Operator/Driver, Service Tech & Cloud-Based Services	On-Vehicle, Service Bay & Cloud Data	Telematic Data Available to Service Tech with Diagnostics Info	**Addition of Component-Level Health Models**	Component-Level Health Predictions
4	Integrated Vehicle Health Mgmt.	Operator and service techs are provided with system or vehicle level health indicators before problems occur with remaining useful life estimated	Operator/Driver, Service Tech & Cloud-Based Services	On-Vehicle, Service Bay & Cloud Data	Telematic Data Available to Service Tech with Diagnostics Info	Addition of Vehicle-Level Health Models	Vehicle-Level Health Management
5	Self-Adaptive Health Mgmt.	Self-adaptive control to extend vehicle operation and enhance safety in presence of potential or actual failures	Operator/Driver, Service Tech & Cloud-Based Services	On-Vehicle, Service Bay & Cloud Data	Telematic Data Available to Service Tech with Diagnostics Info	Addition of Vehicle-Level Health Models	**IVHM Capability Integrated into Vehicle Controls**

Figure 6.1: IVHM capability levels [61].

The very first steps in this work, taken even before the analysis included in this thesis, aimed to simulate the real-time gathering of vibration data with a goal of detecting valve blockages via construction of simple crisp logic structures. As new inputs and predictors to these structures were proposed, the difficulty of implementing changes to accommodate them increased dramatically. Furthermore, the tendency for these predictors to not necessarily follow distinct class boundaries made it difficult to develop these crisp logic structures effectively.

These issues led to the development of the fuzzy inference systems (FIS) described in the previous chapters. Fuzzy logic systems were seen to be easily implementable and gave promising test results. However, they still failed to alleviate some of the issues which were present in the crisp logic structures. For one, an FIS is still, at its core, a simple structure which hopes to improve results by deciphering the "fuzziness" of continuous-scale inputs. Because of this, there are no sophisticated mathematical mechanisms at play which could be utilized for classification purposes. This requires the classification structures be built completely from the ground up, since there are no model templates to be leveraged. In other words, while it could be easier to implement given tools such as MATLAB's *Fuzzy Logic Toolbox*, the results may not show dramatic increases in accuracy overall, and they certainly did not in this research. The issues of handling new inputs and fault types still exist and further complicate structure development.

With these points in mind, artificial neural networks (ANN) were explored as a possible improvement to the FIS's. While more mathematically advanced, ANN's can in theory provide a more streamlined approach to classification in that they can simply collect inputs and classify accordingly. No internal structures need to be developed on a case-by-case basis, and multiple networks are not required to handle different faults or types of the same fault. A properly trained ANN with a relatively simple set of predictors should be expected to classify accurately. However, for this application there was no tangible improvement in blockage classification accuracy. Movement direction was seen to be detected more accurately, but this isn't particularly useful in practical applications since it can be known by the control system at all times. Given the results, it became clear that some part of the entire classification process up to that point was holding back the error rates that could be achieved.

The hypothesized reason for hitting an accuracy plateau using both the FIS and ANN was the actual approach used to classification. As described previously, there are many advantages to real-time data monitoring, the foremost being that it can simply be more intelligent by nature. Having a constant data stream to analyze could intuitively reduce the harmful effects of any unexpected event, whether or not the system in question is actually dynamic at the time. Therefore, this approach was used in conjunction with the first two classification methods, with the idea being that small segments of data could be analyzed as they are brought into the monitoring software. However, by focusing in on a small subset of data without taking into account the data before or even after it, features that require more data points to uncover may be hidden. These features, in turn, may be particularly important or effective discriminators for classification.

This train of thought led to the procedural change of leveraging the data associated with movements as a whole, rather than treating it in small segments, and this approach was used with the variety of statistical classifiers described in Chapter 2. While different, it is not intended to suggest that it could not be used as part of a “real-time” monitoring process in practical applications. Rather, the aim is to separate the two approaches. Additionally, the manual element of this procedure described in Chapter 4, where the beginning and end of each movement must be manually identified, suggests a process in which the data is being reviewed post-hoc by some technician or engineer.

By analyzing each movement as a single entity and, in the case of the rotary valves, adding an additional sensor, a clear improvement in classification accuracy was achieved. Furthermore, many types of classifiers were able to predict accurately, specifically k-nearest-neighbor (kNN) based methods as well as support-vector machines and bagged tree ensembles. In particular, the fine kNN classifiers checking only the single nearest neighbor seemed to consistently give some of the highest accuracy rates regardless of the predictors used. Given the relatively small test data sets (< 100 tests) and the distinct but nonlinear class boundaries seen in Figures 5.3-5.7, this is of no surprise. Support-vector machines (SVM) were effective but, for the most part, inferior to kNN. Bagged tree ensembles, while very accurate in a few test sets, are very computationally intensive. Overall, the simplicity and accuracy of kNN methods seem to suggest that it is the best resulting classifier for this application. However, kNN methods have their own issues with computational power given the number of calculations involved for large training sets.

These findings reinforce the importance of considering the specific application of the CBM solution when performing necessary research. In unmanned systems, for example, computational power might be a concern, so the constant gathering of data and use of classifiers with larger processing requirements are methods to be avoided. Computing power is of particular concern if the health monitoring is to be embedded with the mechanical system of interest. This makes the results of classification using only high-level control system data, given in Chapter 5, to be very attractive. Utilizing data outputs which are baked into the system of interest adds a layer of efficiency to the process as a whole. Given the results presented in the previous chapter, this is certainly an area to be explored when a more bare-bones approach is warranted.

Future Work

While the testing and analysis presented in this thesis is relatively “clean” and easy to interpret, it should be recognized that there have been a handful of assumptions and purposefully omitted conditions which should provide a basis for future work in this area. Testing in the most realistic possible context is important specifically to capturing vibration data, as environmental influences could affect the data being collected by these sensors. Furthermore, there should be attempts to verify and stay adherent to test settings such as supplied air pressure and current which can affect the behavior of the dynamic system in question. For example, a constant air supply at 60 PSI was supplied to all valves used in this study (not to be confused with the *Supply Pressure* CS variable which measured the air pressure actually supplied to the actuator). It was shown early in this study that the speed with which a valve movement was made began to slow as this pressure was adjusted significantly below this value. A supply pressure of 40 PSI would therefore completely change frequency magnitudes and potentially the excited frequency bands themselves. These operational conditions are therefore of great interest not only in the context of this research but in CBM work in general.

The most immediate logical next step for this work would be to expand to a more realistic environment. Specifically, future test rigs would ideally have a fully operational flow loop so that fluid could flow through the valves during testing. This would allow researchers to observe how fluid flow affects the acoustic characteristics associated with specific blockages or faults in general. Pressure gradients within the flow loop itself may also affect this data. Additionally, the presence of flow through the valves brings new sensor instrumentation opportunities. For example, sound recordings or pressure sensors could be used to detect leaks in the valves which may be caused by a blockage or related issues.

Once a level of analysis has been performed on data corresponding to more realistic operating environments, efforts would shift to integrating the hardware and software needed to perform the analysis to some concentrated package for use in relevant industries. There are a variety of approaches to this type of work and would depend entirely on the sensors involved, software required and other spatial, computational and logistical requirements.

Further expansion upon this work should also be pursued in the area of prognostics. It stands to assume that many blockages in control valves may not be an instantaneous event and that there could be build-up over time which may eventually present itself as a blocking force. As a result, detailed study of the progression of faults may provide crucial information into how they form and how to effectively respond to or prevent them. From an analytical standpoint, density estimation techniques may be explored to better understand the frequency and circumstances under which these types of blockages may form.

Final Summary

At the highest level, the analysis and results presented in this thesis outline how a variety of data inputs and analysis techniques can be utilized for the detection and subsequent classification of blockages in process control valves. Given the wide range of control valve applications and the previous CBM research performed on them, it is argued that the motivation for this work is well founded and provides a foundation for future research for FDI not only in control valves but for other mechanical systems as well. Specifically, the completed testing and analysis suggests that vibration data is particularly helpful in identifying the types of valve blockages proposed, and the use of motion-detecting sensors such as gyrometers may also be useful supplementary data sources. It was shown how control system data can be leveraged for classification purposes as well. While classification accuracy rates were shown to have a legitimate dependence on the predictors being used as inputs, it was also shown how a variety of classification methods could accurately perform fault detection given the quality of these predictors, regardless of what type of sensor data they represent. As such, control valves have been shown to be quality candidates for this type of health monitoring from the data presented here from simple test environments.

References

[1] Shin, Jong-Ho, Hong-Bae, Jun. "On Condition-Based Maintenance Policy." *Journal of Computational Design and Engineering* 2.2 (2015): 119-27.

[2] Prajapati, Ashok, Bechtel, James, Ganesan, Subramaniam. “Condition based maintenance: a survey”. *Journal of Quality in Maintenance Engineering* 18.4 (2012): 384-400.

[3] Kothamasu, R., Huang, S.H. & VerDuin, W.H. “System Health Monitoring and Prognostics – A Review of Current Paradigms and Practices”. *International Journal of Advanced Manufacturing Technologies* 28.9 (2006): 1012-1024.

[4] Bengtsson M. *Condition Based Maintenance Systems an Investigation of Technical Constituents and Organizational Aspects*. Licentiate Thesis. Mlardalen University. (2004).

[5] Emerson Process Management. (2005). *Control Valve Handbook 4th Edition*. Retrieved from http://www.documentation.emersonprocess.com/groups/public/documents.

[6] Processor. "DO (dissolve Oxygen) Control System to Support BNR Processes." *Control Systems Engineering*. WordPress, 26 Sept. 2011. Web. 18 Jan. 2017.

[7] Simani S. *Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Doctoral Dissertation. University of Modena and Reggio Emilia. (2004).

[8] Karpenko, M., Sepehri, N. “A Neural Network Based Fault Detection and Identification Scheme for Pneumatic Process Control Valves”. *2001 IEEE International Conference on Systems, Man and Cybernetics, 7-10 October 2001*. IEEE, 06 August 2002.

[9] Adams, S., Beling, P., Farinholt, K. “Condition Based Monitoring for a Hydraulic Actuator”. *Annual Conference of the Prognostics and Health Management Society October 2016*. PHM Society, 07 September 2006.

[10] House, J., Lee, Won Y., Shin, D. “Classification Techniques for Fault Detection and Diagnosis for an Air-Handling Unit”. *Ashrae Journal* 105.1 (1999): 1088-1097.

[11] B.B. Nair, Preetam, M. T. Vamsi, Panicker, V. R., Kumar, G., and Tharanya, A. “A Novel Feature Selection method for Fault Detection and Diagnosis of Control Valves”. *International Journal of Computer Science Issues* 8.3 (2011): 415-421.

[12] Sharif, M. A., Grosvenor, R. I., “Sensor-Based Performance Monitoring of a Control Valve Unit”. *Journal of Process Mechanical Engineering* 213.2 (1999): 71-84.

[13] Subbaraj, P., Kannapiran, B. “Artificial Neural Network Approach for Fault Detection in Pneumatic Valve in Cooler Water Spray System”. *International Journal of Computer Applications* 9.7 (2010): 43-52.

[14] Sundarmahesh, R., Kannapiran. B. “Fault Diagnosis of Pneumatic Valve with DAMADICS Simulator using ANN based Classifier Approach”. *IJCA Proceedings on*

International Conference on Innovations in Intelligent Instrumentation, Optimization and Electrical Sciences (2013) ICIIIOES (9):18-17.

[15] McGhee, J., Henderson, J., & Baird, A. "Neural Networks Applied for the Identification and Fault Diagnosis of Process Valves and Actuators". *Measurement* 20.4 (1997): 267-275.

[16] Uppal, F. J., Patton, R.J. "Fault Diagnosis of an Electro-Pneumatic Valve Actuator using Neural Networks with Fuzzy Capabilities". *2002 European Symposium on Artificial Neural Networks*. ESANN, January 2002.

[17] Bocaniala, C. D., da Costa, J. S., & Louro, R. "A Fuzzy Classification Solution for Fault Diagnosis of Valve Actuators". *Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference, KES 2003, Oxford, UK, September 2003. Proceedings, Part I.* (pp. 741–747). inbook, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[18] Mendonca, L. F., Sa da Costa, J. M. G., Sousa, J. M. "Fault Detection and Diagnosis using Fuzzy Models". *2003 European Control Conference*. Technical University of Lisbon, 2003.

[19] Carneiro, A. L. G., Porto Jr., A. C. S. "An Integrated Approach for Process Control Valves Diagnosis Using Fuzzy Logic". *World Journal of Nuclear Science and Technology* 4.3 (2014): 148-157.

[20] L. Hu, K. Cao, H. Xu and B. Li. "Fault Diagnosis of Hydraulic Actuator based on Least Squares Support Vector Machines". *Proceedings of the IEEE International Conference on Automation and Logistics, 2007*. pp. 985-989.

[21] J. Gao, W. Shi, J. Tan and F. Zhong, **"**Support vector machines based approach for fault diagnosis of valves in reciprocating pumps". *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, 2002*. pp. 1622-1627.

[22] Mannan, M. Sam. *Lee's Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control*. Vol. 1. Amsterdam: Elsevier Butterworth-Heinemann, 2005. Print.

[23] Jameel, R., Singhal, A., Bansal, A. "A Comparison of Performance of Crisp Logic and Probabilistic Neural Network for Facial Expression Recognition". *1st International Conference on Next Generation Computing Technologies, September 2015, Dehradun, India*. IEEE, 11 January 2016.

[24] Xhemali, D., Hinde, C., Stone, R. "Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages". *International Journal of Computer Science Issues (IJCSI)* 4.1 (2009): 16-23.

[25] Correa, M., Bielza, C., Pamies-Teixeira, J. "Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process". *Expert Systems with Applications* (2008), doi: 10.1016/j.eswa.2008.09.024.

[26] Leonhardt, S., Ayoubi, M. “Methods of Fault Diagnosis”. *Control Engineering Practice* 5.5 (1997): 683-692.

[27] Tonyromarock. "My First Time Using Matplotlib." Blog post. *Pybloggers.com*. PyBloggers, 21 Sept. 2015. Web. <http://www.pybloggers.com/my-first-time-using-matplotlib/>.

[28] Zadeh, L. A. "Fuzzy Sets." *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*. Vol. 6. River Edge, NJ: World Scientific, 1996. 338-53. *World Scientific*. Web.

[29] Klir, George, and Bo Yuan. *Fuzzy sets and fuzzy logic*. Vol. 4. New Jersey: Prentice hall, 1995.

[30] Hameed, Ibrahim A., and Claus G. Sorensen. *Gaussian Membership Functions*. Digital image. *Intech*. N.p., 1 Feb. 2010. Web. 23 Jan. 2017. <http://www.intechopen.com/books/fuzzy-systems/fuzzy-systems-in-education-a-more-reliable-system-for-student-evaluation>.

[31] Luiz C. M. de Abreu, G., Ribeiro, J., *Basic configuration of a fuzzy logic system*. Digital Image. *SciELO*. N.p., Dec. 2003. Web. 23 Jan. 2017. http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592003000400005.

[32] Shiffman, Daniel. *The Nature of Code*. S.l.: Selbstverl., 2012. Print.

[33] Cburnett. *Artificial Neural Network*. Digital image. *Wikipedia*. N.p., 27 Dec. 2006. Web. 24 Jan. 2017. <https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg>.

[34] Moller, M. *Neural Networks*, Vol. 6, 1993, pp. 525–533.

[35] Haykin, Simon S. *Neural Networks: A Comprehensive Foundation*. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999. Print.

[36] Kocak, Taskin. *Sigmoid Functions and Their Usage in Artificial Neural Networks*. N.p.: UCF School of Electrical Engineering and Computer Science, Spring 2007. PPT.

[37] Rojas, R. *Neural Networks: A Systematic Introduction*. Chapter 8. Springer-Verlag, Berlin, 1996.

[38] Nielsen, Michael A. *Neural Networks and Deep Learning*. Determination Press, 2015.

[39] Segaran, T. *Programming Collective Intelligence*. U.S.A: O’Reilly Media Inc. 2007.

[40] Peterson, Leif E. *K-nearest neighbor*. Scholarpedia, 4.2 (2009):1883.

[41] Bailey, T., Jain, A. “A note on distance-weighted k-nearest neighbor rules”. *IEEE Trans. Systems, Man, Cybernetics* 8 (1978):311-313.

[42] Yu, K., Ji, L. Zhang, X. “Kernel Nearest-Neighbor Algorithm”. *Neural Processing Letters* 15 (2002): 147-156.

[43] Dudani, S. A. “The Distance-Weighted k-Nearest-Neighbor Rule”. *IEEE Trans. Systems, Man, Cybernetics* 6.4 (1976): 325-327.

[44] Venables, W. N., and Brian D. Ripley. "Classification." *Modern Applied Statistics with S*. 4th ed. New York: Springer, 2002. 331-51. Print.

[45] Johnson, Richard A., and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. 6th ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2007. Print.

[46] LDA vs QDA decision boundaries. Digital image. Penn State Eberly College of Science. Penn State, n.d. Web. <https://onlinecourses.science.psu.edu/stat505/node/96>.

[47] "Lesson 10: Support Vector Machines." Lesson 10: Support Vector Machines. The Pennsylvania State University, n.d. Web. 26 Jan. 2017. <https://onlinecourses.science.psu.edu/stat857/node/211>.

[48] Hastie, Trevor J., Tibshirani, Robert J., Friedman, Jerome H. *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. 2nd ed. New York, NY: Springer, 2009. Print.

[49] Meyer, David, and FH Technikum Wien. "Support vector machines." *The Interface to libsvm in package e1071* (2015).

[50] Sayad, Saed. "Support Vector Machine." Support Vector Machines. N.p., n.d. Web. 26 Jan. 2017. <http://www.saedsayad.com/support_vector_machine.htm>.

[51] Hsu, Chih-Wei. Lin, Chih-Jen. “A Comparison of Methods for Multiclass Support Vector Machines”. *IEEE Transactions on Neural Networks* 13.2 (2002): 415-425.

[52] Dietterich, Thomas G. "Ensemble methods in machine learning." *International workshop on multiple classifier systems*. Springer Berlin Heidelberg, 2000.

[53] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." *Emerging Artificial Intelligence Applications in Computer Engineering*. (2007): 3-24.

[54] Breiman, Leo. "Bagging Predictors." *Machine learning* 24.2 (1996): 123-140.

[55] Murthy, Sreerama K. "Automatic construction of decision trees from data: A multi-disciplinary survey." *Data mining and knowledge discovery* 2.4 (1998): 345-389.

[56] Quinlan, J. Ross. "Induction of decision trees." *Machine learning* 1.1 (1986): 81-106.

[57] Hastie, T. *Boosting*. N.p.: Stanford University. N.d. PPT.

[58] Emerson Process Management. *Fisher Vee-Ball Rotary Control Valves*. Marshalltown, IA: Emerson Process Management, 2015. Web. <http://www.emerson.com/resource/blob/134994/e4b0115ee118832851a174d776ece2d5/d350004x012-data.pdf>.

[59] Emerson Process Management. *Fisher 657 and 667 Diaphragm Actuators*. Marshalltown, IA: Emerson Process Management, 2016. Web. <http://www.emerson.com/resource/blob/122352/24074c231331f3c1f770660ab9e8e06e/d100087x012-data.pdf>.

[60] Braga-Neto, Ulisses, et al. "Is cross-validation better than resubstitution for ranking genes?." *Bioinformatics* 20.2 (2004): 253-258.

[61] SAE JA6268 Design & Online Information Exchange Standard for Health-Ready Components.