A FRAMEWORK FOR REAL-TIME FAULT DETECTION AND RESPONSE IN MULTI-AGENT TEAMS

MATTHEW B MIDDLETON

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2010

ACKNOWLEDGEMENTS

This thesis would not have been possible without the assistance and support of a great number of people. I owe a great debt of gratitude to Suzanne for standing by me during every step of the way. I would also like to thank my parents for always believing in me and instilling the idea within me that I can achieve whatever I set my mind to. My education also would not have been possible without the support of my grandmothers Shirley and Robbie who constantly encouraged me throughout this journey. My broad set of interests and technical curiosity that eventually led me on this course is the result of growing up with three incredibly capable brothers, and for that fact I'm grateful.

I would specifically like to thank Dr. Frank Lewis for taking a chance on me and hiring me despite my non-traditional path into the engineering field. The technical content of this thesis is only possible with the education and support given by Dr. Lewis. It has been a tremendous honor to work for and with someone who so truly appreciates his students. This thesis also would not have come together without the assistance of Dr. Farhad Kamangar who helped me refine my ideas and research into this document. Special thanks to Dr. Vassilis Athitsos for serving on my committee as well as proving a sound background in Artificial Intelligence.

Finally, I would like to thank my colleagues that I have worked with at the Automation and Robotics Research Institute. The people with whom I have shared the ACS lab with for the past several years have not only made quantifiable contributions to this thesis, but have been terrific friends as well.

November 22, 2010

ABSTRACT

A FRAMEWORK FOR REAL-TIME FAULT
DETECTION AND RESPONSE IN
MULTIAGENT TEAMS

Matthew Benjamin Middleton, M.S.

The University of Texas at Arlington, 2010

Supervising Professor: Frank Lewis and Farhad Kamangar

This thesis details the creation of a fault detection and response framework that can be applied to mobile multi-agent teams. This framework unites existing concepts of conditioned based maintenance and real-time machine diagnostics with a high-level matrix based discrete event command and control system that can respond to detected mechanical faults in agents in real-time. This allows for automatic corrective actions to be taken when an agent experiences failure if redundant resources are available to replace the failed unit or for the mission planner to be notified that the mission is no longer achievable given the operating conditions of the resources. In addition to positing a theoretical framework, this thesis details the design and implementation of a device consisting of hardware, firmware, and software that allows for condition based maintenance methods to be used for diagnosis of mechanical faults in the context of a team of mobile, networked agents.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 Background

The use of highly complex robotic drones for mission-critical tasks is becoming increasingly commonplace, especially for tasks that are incredibly difficult, dangerous, or even impossible for humans to accomplish. After the explosion of the Deepwater Horizon drilling rig in the Gulf of Mexico in the Summer of 2010, underwater drones were used extensively not only for reconnaissance, but also for physically modifying and repairing failed oil production equipment located nearly a mile below the ocean's surface. With few alternative solutions available, the world held its breath while these machines performed delicate operations under extremely hostile conditions.

Such complex machinery operating in especially harsh conditions runs a comparatively high risk of experiencing some failure of its systems. Had one of the underwater drones catastrophically failed in the middle of a critical cutting operation while attempting to repair the oil leak, stopping the flow of oil could have been made even more difficult. It is apparent that responsible use of these systems in critical situations requires great confidence in a drone's operating condition, including the health of its electrical and mechanical systems. As these machines begin to take on even more difficult and critical tasks, especially when working around humans, it becomes imperative to have a means of predicting faults before they happen and having some method of responding to such faults.

Since the mid to late 1990s, many militaries around the world already field tele-operated drones that have been weaponized. The United States Air Force has operated the armed Predator aerial drone has flown combat missions in Afghanistan, Pakistan, Iraq, Yemen, and the former Yugoslav countries. In an attempt to develop these systems in to true force multipliers, the current trend is to increasingly automate these systems such that one operator can direct a networked team of drones that can scout larger areas and conduct increasingly complex missions. In these use-cases, system failures could have especially catastrophic consequences, especially when these systems operate in more highly populated places. The use of these autonomous systems presents a paradox, then. The reason behind their existence is to relieve the workload of human beings and to accomplish critical tasks outside the realm of human capability, but the maintenance and monitoring required for confidence in the operation of these systems becomes enormous.

In order for these systems to actualize their intended purpose while operating safely, they must be able to predict and detect operating faults reliably and quickly. This requirement becomes even more important when teams of autonomous agents are deployed to accomplish a given mission. When many of these agents are used together, the probability of encountering a failure is increased. If fault prediction can be made to work reliably, mission commanders can make better choices about whether or not to conduct a mission given the health condition of the available resources. Similarly, timely fault detection can allow the mission commander to expediently and dynamically reorganize resources in light of a failure during a mission, or make an informed choice to abort the mission if it is determined that the resources necessary for successful completion no longer exist. A system for fault prediction, fault detection, and dynamic recovery is critical for the proper and efficient operation of any complex team of robotic systems.

1.2 Thesis Objective

The research presented in this work, conducted at the Automation and Robotics Research Institute, puts forth a method for real-time wireless detection of mechanical and electrical faults in a team of networked agents and the means for analyzing and responding to the effects of that fault on a larger mission. While fault monitoring in static industrial environments has been researched for some time, the application of this research to a large team of mobile, wireless agents presents some novel challenges and opportunities. This thesis will detail the development of an embedded system comprised of hardware, firmware, and software intended for this purpose. This research also focuses on the design of a high level mission controller that can dynamically respond to detected faults. Specifically, the thesis will detail the construction of a Discrete Event Controller, an expert system used for dynamic, mission-level dispatching of resources with the presence of faults. A visualization of this framework can be seen in Figure 1.1 below.

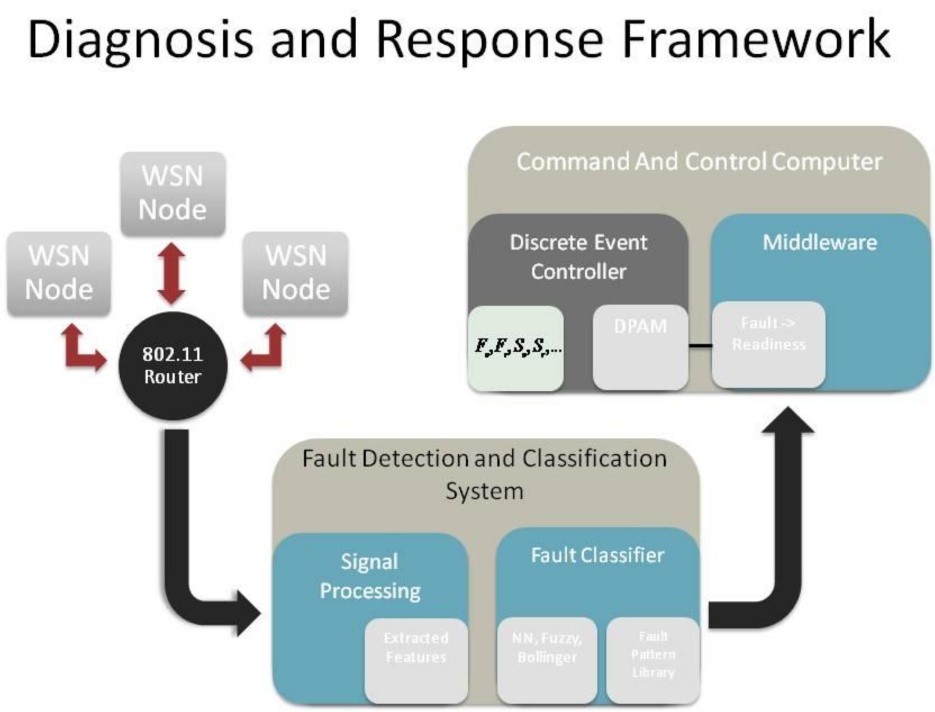

Figure 1.1 High-Level View of Diagnosis and Response Framework

1.3 Thesis Organization

This thesis contains 4 remaining chapters. The next chapter will detail the theoretical background of the research contained in this thesis and examine related work. Chapter 3 will detail the specific theoretical and mathematical framework of the thesis including signal analysis for fault detection and fault handling through the use of a discrete event system. Chapter 4 will detail the simulation and implementation of the thesis work including the design of a physical device that allows for real-time diagnostics in a multi-agent framework.. Finally, Chapter 5 will discuss the conclusions of this research and present information about interesting future work that can be done in this area of research.

CHAPTER 2

ASSOCIATED RESEARCH

2.1 Introduction

The framework described in this thesis draws from and unites two existing research areas that have traditionally been relevant to the domain of factory automation. The first component of the framework is a method for detecting faults in a vehicle in real-time that prevent it from performing its mission. The methods for this sort of fault detection come from a research area called condition-based maintenance. The second part of the framework is a method for responding to this diagnostic data. This particular framework involves the extension of another research topic in industrial automation known as discrete event control. This thesis utilizes these two areas in conjunction to provide a unified platform for detecting faults in a mobile network in order to provide a proper response to the failure of agents.

2.2 Condition-Based Maintenance

Fault prediction and detection has historically been used primarily in industrial and factory settings in the form of Condition Based Maintenance. Sensors that measure vibration, current consumption, and other metrics are attached to complex machinery and the measurements from these sensors are sent to a computer for processing. Signal processing and/or model-based algorithms then sift through the data looking for abnormalities in the sensor readings that indicate a current or potential fault. Machines used for fabrication, for example, often exhibit vibrations with a signature frequency content when their cutting blade or bit becomes dull [9]. By measuring the vibrations given off by a machine and analyzing the frequency content of those measurements, the condition of the cutting tool can be analyzed in real-time and the tool can be replaced if necessary. Before this means of analysis existed, it was often necessary to take the machine out of operation on a regular schedule and physically examine or test the tool in question. Such a method is grossly in efficient and expensive. Machines are often taken down for maintenance before it is actually necessary or alternatively cutting tools that dulled earlier than expected end up damaging or destroying work pieces. Real-time analysis, by contrast, allows for the optimal maintenance schedule based on the current condition of the machine.

In a factory setting, it is often possible to use the optimal sensor and data acquisition (DAQ) equipment without many constraints. Most factories have plenty of power and room for the necessary equipment and there generally aren't vast numbers of machines that need to be outfitted with that equipment. It is also usually possible to hardwire the connections between the sensors, DAQ and processing hardware and to count on reliable connections and centralized resources.

On the other hand, there are myriad challenges involved in developing a fault detection system for mobile platforms, especially teams of mobile platforms. Such platforms often have a very limited power supply, limited space to install the equipment, and must operate and no hardwired means of communication. Moreover, because the costs add up quickly when each agent in a large team is outfitted with this equipment, it is important to minimize the per unit cost. The development of a system that meets these constraints and works reliably is a challenging engineering task.

The implementation of a sensing system on a networked team of agents implies the creation of a type of Wireless Sensor Network (WSN). A WSN is comprised of small, inexpensive, power-efficient nodes connected in a network structure each capable of measuring some physical condition using a sensor and then locally or jointly processing that data [19]. Wireless sensor networks have many potential applications such a monitoring an area or an environment. WSNs themselves are the subject of much academic research on topics such as sensor choice and placement, selection of the network topology. The WSN research relevant to the topic of this thesis is primarily that concerned with ensuring reliable communications between agents as well as distributed processing and analysis for fault detection. In [17], the authors detail specifically using these methods for fault detection in vehicles which is typical of the mobile agents that this framework is useful for.

Each node in the Wireless sensor network is comprised of a sensor or set of sensors connected to a device capable of acquiring the signal from the sensor, possibly processing or pre-processing the data, and transmitting the data to another node on the network. A sensor is a device, usually a transducer, which is capable of converting some physical phenomenon into an analog voltage. The first task of a wireless sensor node is to acquire that signal using an Analog to Digital Converter (ADC) which will convert the analog signal to a digital numeric representation that can be processed by the computer. In order acquire a usable signal with a high Signal-to-Noise Ratio, Linearity and High Dynamic Range however, most applications require signal conditioning and filtering and digital signal processing.

Once the analog signal has been properly converted to a usable digital representation, it must then be processed in order for the system to detect and classify the presence of a fault condition. In the subject area of fault diagnosis and classification, there are a wide variety of methods; however this work will only focus on two classes. The first class of methods that is the subject of this thesis is Signal Based fault detection. These methods involve searching for changes in the output signal from the sensors that occur during fault conditions. Signals can be analyzed in the time-domain or the frequency domain. The time domain refers to the set of samples, or measurements from the sensor at certain time intervals. A motor with a short in its wiring might draw a surge of current which would be captured by an attached current sensor and represented on the output analog signal by samples with a high amplitude during the current spike. This thesis will primarily focus on frequency domain analysis. In this case, the sensor time series is converted to the frequency domain using a Fast Fourier Transform algorithm (FFT). In this domain, the frequency content of the data can be analyzed for abnormalities. For example, A 60Hz motor attached to a 4-blade fan would be expected to produce a signal with significant frequency components at 60Hz and 240Hz and certain multiples. If a blade became damaged however, these frequency components would shift and a fault could be deduced.

A second means of fault detection focused on in this thesis is model-based fault detection. Given a working model of a dynamic system, its output behavior can be reasonably predicted given its input. If the system also has sensors to measure various aspects of its output, those measurements can be compared to the expected output. When there is significant divergence between the measured output and the expected output, there must be some fault in the system. This technique seems to have a lot of potential, especially in large networked teams.

2.3 Discrete Event Control

Once the diagnostic data has been received and faults have been detected, some method is required to respond to that data. For large networked teams, it is impossible for a human operator to respond in real-time to diagnostic data coming in from many agents. The framework presented in this thesis relies on a matrix-based discrete event controller that provides for real-time dispatching of resources. This framework shows the extension of this controller to allow for dispatching based on the real-time operating condition of the vehicles. The particular matrix-based discrete event controller used in this research was developed at and patented by the Automation and Robotics Research Institute (ARRI) at the University of Texas at Arlington where the research for this thesis was carried out. The mathematical framework for the discrete event controller is detailed originally in [18]. Originally it was used to specify, analyze, and control a factory workflow. In [1,6,14,7], researchers at ARRI demonstrate the use of the discrete event controller as a supervisory controller for mobile wireless sensor networks. Based on research presented in [1,7,6,10,4], this controller was updated to respond to real-time diagnostic data detailing the current operation condition of the resources and to provide automatic redeployment of resources if possible or an alert to the mission commander that the available resources do not permit mission completion.

CHAPTER 3

RESEARCH

The objective behind this thesis is to provide a framework for real-time fault detection and response in the context of multi-agent teams conducting programmed missions. The basic flow of operation behind this framework is that relevant data will be collected from sensors attached to each agent that might indicate the presence of a fault. Using concepts and techniques from the field of condition-based maintenance, this data will be analyzed to determine if a fault truly exists and to classify that fault. The primary classification of faults in this context is a binary decision as to whether or not the presence of that fault prevents functionality of that agent as a resource in the mission. In the case that the fault precludes its functionality, the mission controller is updated and replacement resources are appropriately dispatched if possible.

3.1 Fault Analysis and Diagnosis

3.1.1 Introduction

The fundamental task of fault diagnosis is the identification of certain features of the captured signal that indicate the presence of a fault. [20] details that faults can induce changes in the energy of the system, the entropy of a system and the power spectrum of the observed signal as well as other changes. Furthermore, ideal features are those are computationally inexpensive, have a physical or mathematical interpretation, and clearly delineate between different classes of faults, or the presence or absence of a fault. In Industrial environments, it is often very important to distinguish between different classes of faults and much research has gone into studying the distinguishability of features and the identifiability of different faults. In the context of swarms of disposable agents, it is often more important to concentrate on the detectability and the severity of a fault. This is because in most scenarios, the goal is not to pinpoint a specific fault for repair, but rather to determine if a fault exists that prevents an agent from accomplishing its function within the mission framework. In many cases relevant to this research, it is impossible or simply not worth it to retrieve the failed agent and repair it.

In traditional conditioned based maintenance, techniques for fault detection and classification are typically categorized into one of two approaches. The first approach is referred to as a data-driven approach that relies on collecting statistical information from sensors. This approach does not require a dynamic model of the system, but has the drawback that it can only detect anticipated faults [20]. An alternative method is to use a model-based framework that looks for a discrepancy between the outputs of the model and the actual system to determine a fault. This method allows for the discovery of unanticipated faults, but requires the creation of an accurate dynamic model. Sections 3.1.2 and 3.1.3 will detail the statistics that underlie the mathematical framework for the data-driven approach. In Chapter 4, experiments using these methods to detect bearing faults will be presented. [3] provides specific details of the use of the following methods to that domain.

3.1.2 Time Domain Analysis

3.1.2.1 Probability Density Functions

When analyzing data in the time-domain, it is helpful to build a probabilistic model of the operation of the machinery when it is functioning normally. This can be done by constructing the probability density function (PDF) from observed data. The PDF is constructed by calculating the number of samples that fall into each of a number of “bins” that represent the range of the data. Changes in the operation of the machinery are often visually apparent when looking at how the samples are distributed.

3.1.2.2 Moments about Zero

While the density function provides a visual understanding of the changes in a system with the introduction of a fault, there are many calculations that allow for quantification of those visible changes. Those calculations are some of the primary features used in fault detection and include moments about zero and the moments about the mean. [13] details the specific use of moments in vibration analysis used to detect bearing failure which is the focus of the experiments detailed in Chapter 4. The moments about zero are defined by the equation below (where p represents the *p*th moment):

$$\frac{1}{N}\sum_{k=1}^{N} x_k^p \tag{3.1}$$

The first moment about defines the sample mean:

$$\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k \tag{3.2}$$

The second moment is defined as the moment of inertia of the signal. Other important calculations include the energy of the time series which is defined by:

$$\sum_{k=1}^{N} x_k^2 \tag{3.3}$$

Finally, some features use the root-mean-square value of the signal which is defined as:

$$\sqrt{\frac{1}{N}\sum_{k=1}^{N} x_k^2} \tag{3.4}$$

3.1.2.3 Moments about the Mean

Once the mean has been calculated, several important moments about the mean can be computed. Generally, the *p*th moment about the mean is defined by:

$$\frac{1}{N}\sum_{k=1}^{N}(x_k - \bar{x})^p \tag{3.5}$$

The second moment about the mean is referred to as the variance σ^2 with σ defining the standard deviation. The higher moments about the mean begin to reveal greater information about the operation of the system. The third moment about the mean is referred to as the skewness of the data and measures the symmetry of the distribution of the data. The skewness is defined as:

$$\frac{1}{N\sigma^3}\sum_{k=1}^{N}(x_k - \bar{x})^3 \tag{3.6}$$

The fourth moment about the mean is called kurtosis. It represents the peakedness of the distribution of the data set. A distribution with higher kurtosis means that the variation in the data is attributed to infrequently occurring samples with a higher difference from the mean. A distribution with lower kurtosis, on the other hand, has samples that more frequently deviate from the mean, but by a smaller difference.

3.1.2.4 Windowed Moments

For many systems that have time varying output, it may be necessary to calculate the moving moments of the system using a time window. When computing moving moments, conceptually a window of a determined number of samples is slid over the data and the moment is calculated for each move of the window. The moving average is a type of Finite Impulse Response Filter. It is defined mathematically as (where *N* is the window size and *n* is the current time):

$$\bar{x}(n) = \sum_{k=n-(N-1)}^{n} x_k \tag{3.7}$$

3.1.2.5 Analysis in the Time Domain

The nature of the analysis of a specific system is highly dependent on that system. Few features provide useful information outside of the context of the specific system and a specific operating envelope. The features mentioned above provide a general framework for analysis, but they may be individually more or less useful depending on what is being analyzed. For the vibration tests on the mechanical testbed used in this research, when a kernel smoothed probability density is calculated, the samples are symmetrically distributed around and between the peaks and troughs of the fundamental signal waveform. In the case of the motor experiment, most of the energy of the signal is in the 60Hz waveform. As the severity of the bearing fault increases, the standard deviation of the signal increases dramatically and the fundamental shape of the density function can change dramatically. Figures 3.1 – 3.3 show the probability densities of three different fault cases over the period of several days. While there are a number of factors that can affect the operation of the machinery over this short period of time including fluctuations in the electrical source, the PDF remains very stable for each type of fault condition. Figure 3.4 shows the difference in PDFs between fault conditions and demonstrates that the distribution of sensor values in the sample varies quite dramatically between fault cases.

Quantitatively, for the data used in Figures 3.1-3.4, the mean of the data for each set of samples is virtually identical. For this sort of vibration analysis to locate bearing faults, the mean then is not a useful feature. On the other hand, the PDFs show that there is a large difference in the deviation of the sensor readings from the mean between the fault conditions. The standard deviation of the no fault case is significantly lower than the standard deviation of the fault cases. In the case of a bearing fault, the sensor values empirically tend to more uniformly take on values across the full range of the measured vibration rather than converging around a central mean. While in this case the skewness of the distribution does not obviously provide insight into the operating condition of the machine, one can see that the conditions are well delineated by their kurtosis. While the no-fault case appears to be leptokurtic, it should be noted that there is a definitive lack of a tail on this distribution. In fact, the probability density function for the no-fault case has the lowest kurtosis value of the three distributions. This means that during normal operation, the sensor values fall within a comparatively narrow deviation from the mean and there are few extreme outliers. As the severity of the bearing fault increases however, the kurtosis of the probability density function increases due to the increasing presence of outliers that substantially deviate from the mean. Since kurtosis measurements give comparatively more weight to isolated samples with a large deviation, it tends to be a useful metric for identifying critical failure versus the gradual onset of problems.

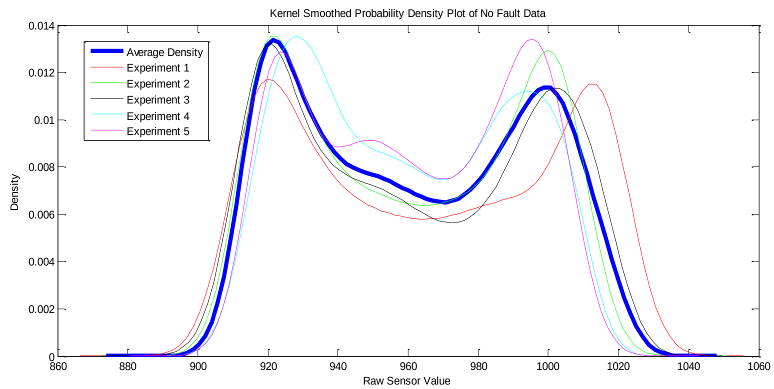

Figure 3.1 Example PDF of a Simulated No Fault Condition (0g Offset)

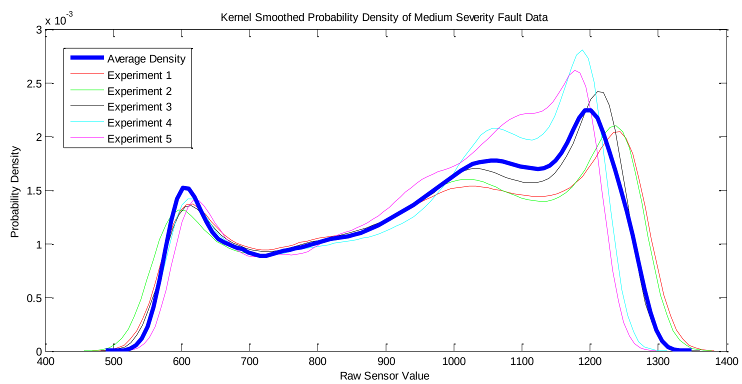

Figure 3.2 Example PDF of a Simulated Bearing Fault (5g Offset – Medium Severity)

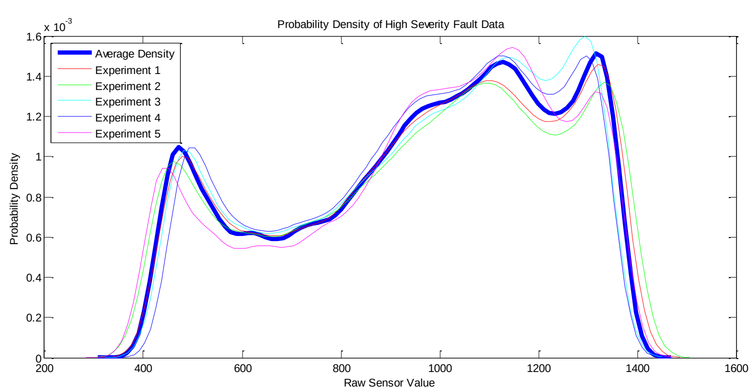

Figure 3.3 Example PDF of Simulated Bearing Fault (10g – High Severity)

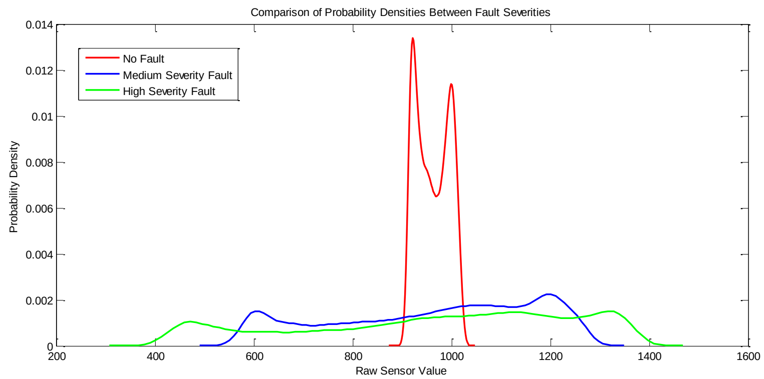

Figure 3.4 Comparison of Example PDFs Between Fault Conditions

3.1.3 Frequency Domain Analysis

3.1.3.1 Introduction

For periodic vibration signals of the type that is generated by rotating machinery or A/C electrical equipment, there is also a significant amount of useful diagnostic data that can be found in the frequency domain [5]. The signal coming from a piece of rotating equipment such as a transmission or robotic tool will have frequency components related to the various components of the system. A helicopter for example, might have an engine running at 30,000 RPM (500 Hz) driving a transmission that rotates two main blades at 260 RPM (4.3 Hz). In the time domain, this signal from the attached vibration sensor would look something like Figure 3.5. This time series represents a simplified model of a helicopter engine and transmission based on the description above with some added Gaussian noise. An analysis in the frequency domain of the vibrations coming from the helicopter would show frequency components with high magnitudes at 500 Hz due to the engine, 4.3 Hz due to the main shaft, and 8.6 Hz due to the two blades. Figure 3.7 is the plot of the FFT of the time-series used to create the plot in Figure 3.6. Note that the x-axis is logarithmically scaled in order to clearly show the wide-ranging frequency components. Figure 3.7 clearly shows that the signal contains a large amount of energy at the expected frequencies and relatively low amounts of energy at other frequencies. One would also expect to find frequency components related to the internal gearing of the transmission. Should the transmission begin to malfunction or if the helicopter experienced problems with one of its blades, it would be observable in the spectral analysis of the vibrations coming from the system.

3.1.3.2 Calculation of the Frequency Spectrum

In order to convert the time series to the frequency domain, it is necessary to use the Discrete Fourier Transform (DFT). This transform from the time-domain to the frequency-domain is defined mathematically as:

$$X(k) = \sum_{n=1}^{N} x(n) e^{-\frac{j2\pi(k-1)(n-1)}{N}} \quad k = 1,2,\dots,N \tag{3.8}$$

Modern Digital Signal Processing libraries found in workstation development software such as Matlab or in embedded DSP development frameworks provide methods of computing the DFT using a computationally efficient algorithm known as the Fast Fourier Transform (FFT). The plots in this section were generated using the Matlab FFT command.

If the system has frequency components that vary with time, it may be necessary to use a windowed FFT. Similar to calculating moving averages, this involves calculating multiple FFTs over the length of the data for a smaller number of samples. This can be done with or without overlap between the sample windows. When there is no overlap, this is referred to as a bin windowed FFT. Windowed FFTs are also referred to as Short Time Fourier Transforms (STFT). The resulting plot is a 3-dimensional plot of the multiple 2-dimension FFTs for each point in time. This allows for the change in the spectrum to be visualized across a period of time. Mathematically, the windowed FFT is a function of a frequency k and a time index N. It is described by the equation:

$$X(k, N) = \sum_{n=N-m+1}^{N} \frac{x(n) e^{-j2} \pi(k-1)(n-1)}{N} \tag{3.9}$$

A graphical example of a windowed FFT can be seen in Figure 3.5.

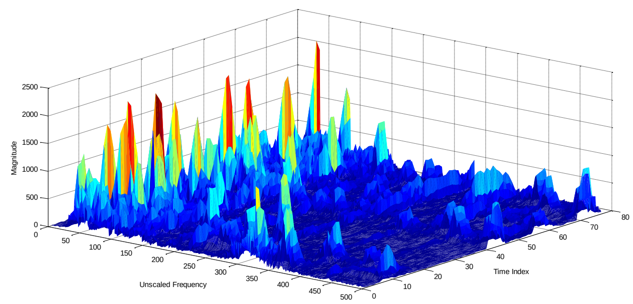

Figure 3.5 Windowed FFT

3.1.3.3 Power Spectral Density

In many cases with frequency analysis used in fault detection, a similar metric called the power spectral density is favored over the use of the simple FFT because it potentially provides greater detail about the frequency response in select cases [20]. The power spectral density (PSD) is actually the FFT of the autocorrelation of the data. Since the autocorrelation is simply the square of the data in the frequency domain, there is very little computation involved in computing the PSD beyond taking the FFT itself. In practice, there are more accurate methods of calculating the estimate of the PSD such as Welch's method, which is used by mathematical analysis software such as Matlab.

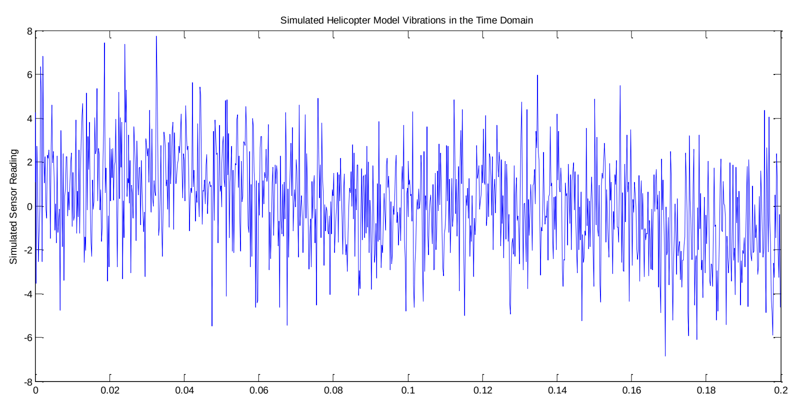

Figure 3.6 Simplified Simulation of Helicopter Vibration Data in the Time Domain

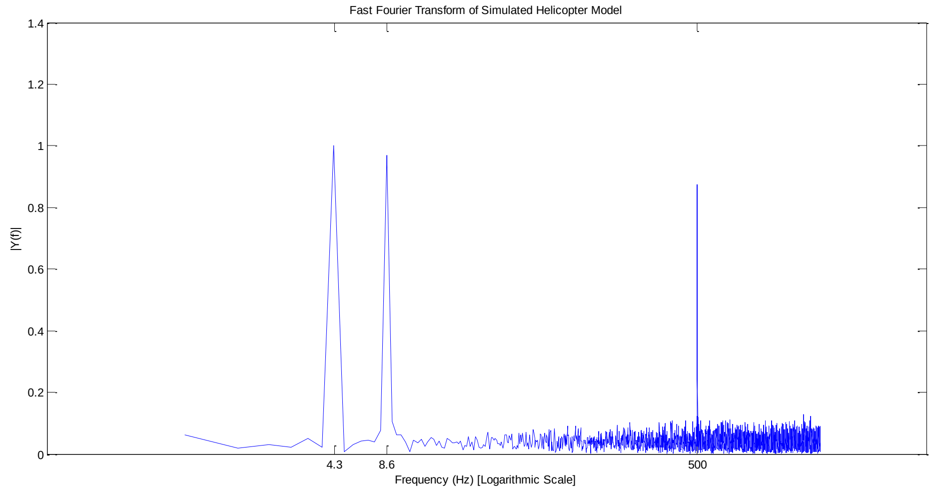

Figure 3.7 Simplified Simulation of Helicopter Vibration Data in the Frequency Domain

3.1.3.4 Analysis in the Frequency Domain

The fundamental task involved in detecting faults using frequency domain analysis is to search for frequency values that have unexpected energy based on the constructed model of the system during normal operation. In some cases, this may mean that a frequency that is expected to be present in this signal has either too much or too little energy, or may even be missing. In other cases, unexpected frequencies with significant energy content may be detected. If a motor with a shaft rotation speed of 30 Hz spins a gear with 16 teeth meshed with a gear with 8 teeth, it would be expected to see frequency components of 60 Hz, 480 Hz and 960 Hz. If a tooth breaks on the gear with 16 teeth, then the 480 Hz and 960 Hz signals would shift closer to 450 Hz and 900 Hz.

3.1.4 Fault Classification

3.1.4.1 Introduction to Classification

The time domain and frequency domain techniques above demonstrate techniques for collecting statistics and identifying features about the current operating conditioning of the machinery. In the context of industrial machinery which is typically not expendable, it is critical to use that data in order to specifically identify the fault so that it can be repaired. For fault detection and classification related to expendable agents undertaking a real-time mission, it is likely neither important nor possible to specifically identify the nature of the fault. Even if the fault were identified precisely, it is unlikely that it would merit the cost of retrieving the device and then repairing it. The important thing is simply to determine whether or not there is a fault significant enough to prevent the agent from completing its required tasks. This is a much more basic and coarse determination that simplifies the task of fault identification for this subset of problems. Once an existing fault classification method that can make this determination has been selected for the specific system in question, the output of the classifier can then be sent to the high-level discrete event controller that performs mission analysis and resource dispatching.

The nature of fault classification in different systems can be very different. It is not possible to determine a theoretical threshold that defines whether or not a system is functioning that is relevant across systems or even for a single system in different operation conditions. To go from feature vectors to the determination of a fault requires implementing a framework relevant to the specific system. This often involves some method of machine-learning or at least the creation of a database of failure modes. That database or knowledge base can then be queried using the current features of the machine to calculate the likelihood of a fault.

3.1.4.2 Definition of the Fault Feature Vector

The fault feature vector for a data-driven classification framework is a set of data that includes the features extracted using the analysis techniques in sections 3.1.2 and 3.1.3 including moments of the sensor readings as well as well as frequency data. For data-driven approaches, sensors are selected that respond in the cases of expected faults. [20] demonstrates the process of identifying a fault based on historical analysis of a system. It is possible to create a fault-feature vector $\varphi[x(t)]$ that is a sufficient statistic for the full state and history *x(t)* of the system that determines the fault/failure condition *z(t)*. The decomposition of the fault likelihood probability density function

$$f\left[\frac{z(t)}{x(t)}\right] = g[x(t)]h\{\varphi[x(t)], z(t)\} \tag{3.10}$$

so that knowledge of $\varphi[x(t)]$ is sufficient to diagnose the fault state z(t). Accordingly it is not necessary to know the full system state and history in order to diagnose a fault. Instead $\varphi[x(t)]$ is continuously updated from the incoming sensor data and compared against a fault pattern library to determine the presence of a fault. The fault pattern library (FPL) is a database of fault-feature vectors that maps those features to the presence of a specific fault.

3.1.4.3 Fault Pattern Libraries

Fault Pattern Libraries can be created numerous ways, including by learning based methods, model based methods, and a variety of other means. This thesis focused more on the framework of fault detection than on the specific means by which it is done. For the experiments in chapter 4, the fault pattern libraries were generated with the use of historical data from induced faults. In this method, known faults are introduced to the system and measurements are taken for each type of fault. Once the features are extracted, the features between faults are compared with each other in order to determine which features uniquely define a given fault. Those unique features are assembled into a fault-feature vector that is stored in the FPL. When the system is running online, its current fault-feature vector is compared against the feature vectors in the library. If the current fault-feature vector matches an entry in the library within some tolerance, then it is determined that related fault has occurred.

3.1.4.4 Bollinger Bands

[20] details another method of determining abnormal operation conditions through the use of Bollinger bands. This method requires no historical data but simply provides an online method for determining significant deviation of the operating characteristic of a system.

Bollinger “bands” represent the positive and negative difference between the moving average and the moving variance. When the actual signal deviates outside of these bands, then it can typically be concluded that the operation of the system has significantly changed denoting a probable fault condition.

3.1.4.5 Clustering Methods

More sophisticated methods use clustering techniques such as fuzzy inference systems and neural networks to group historically measured conditions and map them to specific faults. These methods are often critical when there is an abundance of historical data and a large number of fault modes, such that it is not easy for a person to accurately classify the fault using the data. This is especially true when fault information can only be derived by the simultaneous analysis of multiple features. [2,15] details the use of a two-stage neural network for precise fault identification using the wireless sensor based fault detection mechanism used in this thesis.

3.1.4.6 Fault Severity Determination

In the context of the framework presented in this thesis, the important end result of fault classification is to determine whether or not the fault precludes a given agent from performing its function related to the mission. This is a binary decision that can be handled by multiple methods and the choice is inevitably system dependent. It is dependent on the nature of the mission as well as the nature of the agents. If a fault pattern library is uses or some clustering method that maps the data to a specific fault, then it is necessary to determine if that specific fault precludes the use of that resource in the context of the mission. In the Bollinger band case, it must be determined whether or not the deviation of the operation from its normal operating range precludes the use of the resource. Ultimately a binary decision regarding the resources fitness for its mission must be passed to the discrete event controller (DEC) detailed in the next section. It may be necessary to create an intermediate mapping from the fault detection logic to the binary resource availability matrix that is part of the DEC.

3.2 Discrete Event Controller for Fault Response

Identifying the presence of faults in a networked team of agents is only the first step in building a robust mission-execution platform that can function in light of failed resources. As the teams grow, the diagnostic data would overwhelm the ability of a mission commander to respond in real-time to changes in the operating conditions of the team. Some high-level controller must be designed that can respond to detected faults in real-time to ensure the mission is completed if sufficient resources exist, or abandoned as gracefully as possible if there are not sufficient remaining resources. Researchers at the Automation and Robotics Research Institution pioneered and patented a general high-level mission controller called a Matrix Based Discrete Event Controller (DEC). This matrix-based DEC allows for intuitive formulation and implementation of function discrete event (DE) systems that are based on linguistic if-then rules. These rules can then be directly converted to a matrix form where they can be used for system analysis, simulation, and implementation using a computer. [6,14,16] demonstrate the suitability of this control method as the primary high-level controller for a networked team of agents that allows for flexible feedback response to external events and input from the mission commander. This thesis demonstrates that this controller also provides an ideal framework for responding to fault conditions.

3.2.1 Introduction and Definition of a Discrete Event Controller

A matrix-based Discrete Event Controller is a mathematically defined rule-based framework that allows for the analysis of and implementation of complex discrete event systems. This matrix-based formulation was initially detailed in [18]. The following mathematical definitions are fully detailed in [18,12]. A mission can be defined as a sequence of rules prescribing under what conditions each of its tasks can be fired. Thus a mission can be considered a discrete event system and can be controlled by a DEC. The functionally of the DEC can be decomposed into two phases. The first phase is the planning and programming phase that typically occurs offline. This phase involves the creation of the Task Sequencing Matrix and the Resource Assignment Matrix. These matrices are widely used in manufacturing and there are many software tools available to aid in the creation of the matrices beyond simple examples. The next section will detail the nature of these matrices and how to formulate them in terms of a rule base that allows for the functioning of the DEC. The second phase of the DEC is the operation phase in which the DEC dynamically responds to events and dispatches resources to complete tasks according to the defined rules towards the end of reaching a goal state that defines completion of the mission. This phase will be detailed in the next section.

3.2.1.1 Planning Phase Definitions

The first component of the planning and programming phase is the Task Sequencing Matrix (TSM). The TSM is a mapping from tasks to other tasks that specifies which tasks are prerequisites for other tasks. As such, the TSM describes a strict partial ordering on the set of tasks in terms of time. In formal terms, let P be the binary relation on the set of tasks $\{t_i\}$ that describes the temporal ordering of the tasks. In that case, $(t_{i,}t_j) \in P$ if task t_i is an immediate precursor to task t_j. It can be readily observed that this binary relation holds the properties of a strict partial ordering as the relation is irreflexive (a task cannot be an immediate precursor to itself), antisymmetric (two tasks cannot be immediate precursors to each other), and transitive (if task a is a precursor to b, and b is a precursor to c, then a is a precursor to c). The TSM only requires a partial ordering of the tasks which gives the mission planner a large amount of flexibility in programming the mission. The mission planner is only required to program the important causal relations between tasks. The on-line portion of the DEC does the difficult work of creating an on-line total ordering of the task sequence that abides by the rule set, available resources, and incoming events. This is crucial for its use in responding to faults as the DEC can determine if alternative orderings are available to complete the mission in light of failed resources without having every possible scenario determined ahead of time. Formally, the TSM is a matrix that has element (i, j) equal to 1 if task t_j is an immediate precursor to task t_i. The research for this thesis has also been concerned with using the properties of rule bases to guarantee a certain performance characteristics of the DEC. For this reason and for the DEC to practically operate in real-time, it is necessary to formulate the DEC and its components in terms of a rule base. This can be done by decomposing the TSM such that

$$TSM = S_v \cdot F_v \tag{3.11}$$

where F_v is the input TSM and S_v is the output TSM. F_v maps from the tasks to the rule set. An element (i, j) in F_v that has a 1 means that task t_j must immediately precede the firing of rule i. A row in F_v can have multiple 1 entries which means that multiple tasks are required in order to fire a rule. The S_v matrix maps from the rule set back to the tasks. An element (i, j) in S_v that has a 1 means that task t_i is to be started when rule j is fired. If there is only a single rule to start each task then S_v is similar to the identity matrix. It should be noted that because these matrices are binary logic matrices the operators used in the definition of the DEC do not refer to standard matrix algebra operators. Instead, these operators refer to Boolean or/and algebra where the dot multiplication operator means AND while the '+' operator means OR.

The second part of the planning phase is the creation of the resource assignment matrix (RAM). This matrix indicates which resources are needed to accomplish each task. If the RAM has element (i, j)equal to 1, it means that resource r_j may be used to complete task t_i. If a task requires multiple resources, there may be more than a single 1 in the row [12]. If there is more than a single 1 in any column, it means that a resource is a shared resource that is necessary for the completion of multiple tasks [12]. The concept of shared resources provides interesting challenges for the Discrete Event Controller because it introduces the possibility of bottlenecks and deadlock. Part of the research involved in this thesis involves using the properties of rule bases to prove that if the TSM is properly constructed, these problems will not occur. These proofs will be presented later in the paper. The shared resource case is also interesting from the perspective of fault recovery using a DEC. In fact, a resource r_i does not refer to a single resource, but rather a single type of resource. The DEC can handle cases of multiple instances of a given type of resource in cases where multiple instances of single resource may be necessary for either a single or multiple simultaneous tasks. This capability is also critical for handling the redundancy of resources so that the system can continue to function in the event of the loss of a particular resource. By using the RAM to do static analysis of which resources are shared and in high demand, it can allow resource managers to make sure sufficient resources of each type with a proper amount of redundancy are allocated towards missions.

The RAM itself is a mapping from the resources to the tasks. Similar to the TSM, the RAM must be formulated as a rule-base for analysis and practical operation. Similarly, the RAM can be decomposed as

$$RAM = S_v \cdot F_r \tag{3.12}$$

where v is the output TSM matrix previous defined and F_r is the input RAM -- a mapping from the resources to the rule base. An element (i, j) in F_r equal to 1 means that resource j is required to fire rule i. S_v is a mapping from the rule base to the set of tasks. As described earlier in the context of the TSM, an element (i, j) in S_v equal to 1 means that task i is to be started when rule j fires. There is also an output RAM S_r which is a mapping from the rule base to the resources. An element (i, j) in S_r equal to 1 means that resource i is released when rule j is fired.

3.2.1.2 Operation Phase Definitions

Once the TSM and RAM are created and formulated as a rule base, the DEC can use that rule base to dynamically dispatch the available resources in order to accomplish the mission tasks in accordance with those rules. The DEC can respond to external input and changes to the available resources while running.

To formally define the DEC, it is necessary to first define several key types of vectors:

a) Task vector v = $[t_1 \ldots t_N]$ where t is the set of N tasks
b) Resource vector r = $[r_1 \ldots r_N]$ where r is the set of N resources
c) State vector x = $[x_1 \ldots x_N]$ where x is the set of N rules in the rule base

The specific vector instances used in the DEC are:

a) Rule state vector x that has a '1' in index □ if rule □ is ready to fire
b) Task completion vector v_c which has a '1' in index i if task t_i has just completed
c) Resource available vector r_c which has a '1' in index i if resource r_i is currently available
d) External trigger event vector u which has a '1' in index i if trigger event u_i has just occurred.

It is now possible to define the controller state equation which is:

$$\bar{x} = F_v \cdot \bar{v}_c + F_r \cdot \bar{r}_c + F_u \cdot \bar{u} + F_D \cdot \bar{u}_D \tag{3.13}$$

where F_u is the input matrix that maps input events to rules, and F_D and u_D are the conflict resolution matrix and conflict resolution vector respectively. The remaining terms are previously defined. As with the other operations used in the DEC equations, the state equation operations specify Boolean algebra and not matrix algebra. The state equation runs for each step of the operation of the controller and updates the state vector x. This can then be used to calculate the output of the controller which is the next task to start and which resources can be freed. These first output equation is the calculation of the task start equation

$$v_s = S_v \cdot x \tag{3.14}$$

where $\square_{\square}$ is the task start vector. An entry of '1' in index i of v_s indicates that task $\square_{\square}$ can be started. The second output equation is the resource release equation

$$r_s = S_r \cdot x \tag{3.15}$$

where r_s is the resource release vector. An entry of '1' in index i of r_s means that the resource r_i is no longer being utilized and can be released. Once these dispatching vectors have been calculated, the binary results indicate what actions the actual agents need to perform. A middle layer controller can then instruct the specific agent controllers on what action to take.

3.2.2 Interfacing the Discrete Event Controller with Diagnostic Data

In [6,1,16], the authors discuss the application of a DEC to the control of missions using wireless sensor networks. There are a number of additional challenges in this environment that are not typically encountered in industrial environments that are the traditional home of DECs. Two major challenges relevant to this thesis are that related to resource availability and resource selection. In a wireless sensor network and especially one comprised of expendable mobile agents, the likelihood of a resource changing its availability status due to health condition or communications availability is significantly higher than in a more predictable industrial environment. Secondly, wireless sensor networks tend to be more dynamic and each resource may be able to perform tasks that overlap with other resources allowing for greater potential redundancy. The authors in [6,1] discuss the theoretical use of a dynamic priority assignment matrix (DPAM) that allows for on-line updating of the F_r in response to changes in the health and availability of resources. For this research, the DPAM was modified such that it is now a mapping from the rule-base to resources just like the F_r matrix. In the F_r matrix, multiple 1's in a single row mean that multiple resources are required to fire that task, denoting an AND relationship. The DPAM by contrasts has values that range from 0-1 for each (i,j). If a single row has multiple columns with a value that is greater than 0, that means either of those resources can accomplish the job representing an OR relationship. A '0' entry means that a given resource cannot be used to perform the tasks associated with a given rule and a '1' entry means that resource is the best resource to perform the tasks associate with that rule. Entries between 0 and 1 can be used to rate the relative performance of resources for performing the associated tasks. In this research, a greedy method is used to find the highest resource value for each rule. The F_r matrix is then updated on-line before the next dynamic update and the DEC responds to the updated resource statuses.

One other extension to the DEC was required to facilitate the functionality of this system. Because the DPAM is directly used to make a new F_r matrix, there has to be a method of indicating that no suitable resources exist to fire a given rule. In the DPAM, this would be represented by a row with no column values greater than zero. In the traditional DEC, there is no manner to indicate that no resources exist to service a certain task. With the R_c vector could be modified, this would cause many problems with other tasks while not accomplishing the functionality required. Instead, the DEC matrices (except DPAM) are updated to reflect the addition of one additional resource. When the DPAM is analyzed to update the F_r matrix, if no resources are available to service the relevant tasks for a given rule, F_r is updated to require the newly added resource in order for that task to fire. This prevents the affected task from firing while allowing the DEC to seek an alternate task ordering to reach the goal state. If there are no alternatives, then at the moment, the DEC will experience deadlock until a resource becomes available that can service the tasks. Because the DEC allows for simulation as well, the state of the DEC could be extracted at regular intervals to check for current or future deadlock given the current resource conditions. If the simulation indicates that the mission is not achievable, the commander can be alerted of this and choose to gracefully abandon the mission.

3.2.3 Discrete Event Controller Proof of Correctness

While the matrix-based DEC has been verified to work in both industrial and WSN environment in both software simulation and in cases where it actually controlled and dispatched hardware, it was also important to verify its theoretical functionality in order to locate any cases in which it might fail. Specifically, it was important to investigate whether a specific set of rules can cause problems of circularity or deadlock. Because the DEC is an expert system based on a set of rules, it's possible to analyze the DEC's performance using the properties of rule bases that are defined in the computer science literature. [8] details methods to determine whether a rule base is consistent, complete, and concise. A rule base that has these properties and is properly executed by the DEC is guaranteed to properly sequence the tasks and correctly dispatch resources [12]. In the following sections, it is shown that if each mission is properly defined, then a rule base that has these desirable properties is created and the proper operation of the DEC is guaranteed.

A rule-base is consistent if there is no way to derive a contradiction from valid input data [8]. A rule base is not consistent if it is compromised by circularity or conflicts. Conflicts are defined as having a condition where a condition or rule and its negation are simultaneously true. This compromises the logic of the rule parsing system and either halts the rule-based system or creates unpredictable and inconsistent operation. The matrix based DEC rule-base by its very nature cannot simultaneously capture a condition or rule and its negation. The matrix formulation only allows for a single case, thus the DEC cannot have conflicts due to its very means of representing the rules. Circularity in a rule-base is defined as a subset of the rules that are executed in an endless cycle. This can cause the controller to never reach its goal state. Again, the very means by which the DEC is created precludes this condition from happening. The F_v matrix is block diagonal and each block defines a strict partial ordering on the set of mission tasks. It is impossible to create conditions of circularity in this ordering [12].

A rule-base is concise if it does not contain any unnecessary or useless piece of knowledge. This means that a concise rule-base can contain no rules that are redundant or subsumed. Because the F_v matrix is block diagonal, it is not possible that rules in the different blocks can have the same antecedent and thus there are no redundant rules between blocks. Within a block, it is hypothesized that the mission planner will not create multiple identical rules or tasks. If this assumption holds, then the there are no redundant rules in the DEC [12]. Additionally, because the matrix-formulation only allows a single manner in which to specify the antecedent for firing a rule, it is impossible to have any rules that are subsumed by a similar rule with the same consequent but a strict subset of the antecedents

Finally, a rule-base is complete if it can cope up with all possible situations that can arise in its domain This means the rule-base contains no missing rules, unreachable conclusions, dead-end states, or dead-end if conditions [8]. The construction of the rule matrices for the DEC guarantee there will be no missing rules because the matrix implicitly defines a rule for each consequent. The other properties cannot be guaranteed by construction. It is the responsibility of the mission planner to create a set of rules that do not generate the undesirable conditions. There are a number of task planning software suites that can assist planners when specifying the rules for a given work-flow.

CHAPTER 4

IMPLEMENTATION AND EXPERIMENT DETAIL

4.1 Implementation of the Wireless Sensor Node for High-Performance Data Acquisition

4.1.1 Design Considerations for the Wireless Sensor Node

In order to predict and detect faults in a physical plant, it is necessary to engineer a system that can measure and process some feature of the plant. When the set of plants consist of many mobile agents, there are even additional requirements such as the ability for wireless operation. Meet all of these requirements including measuring, processing interpreting, and wirelessly sharing fault detection data necessitates the design and integration of hardware, firmware, and software. The implementation of a wireless fault detection system that forms the basis of this research underwent several iterations before its final form, but there are a number of common elements that must be a part of any such system. Namely, such system must have an analog or digital means of acquiring data from a sensor, a means of storing and processing the data, and a means of communication with a base station or other sensor nodes. In selecting these components however, there are many tradeoffs to be made in terms of size, performance, power consumption, and cost. [19] describes in detail the fundamental elements of creating an energy efficient WSN for condition-based maintenance. Optimal selection can only be made in the context of a specific physical plant to be monitored, but the device developed in this research was primarily intended to be as general purpose as possible.

4.1.1.1 Sensor Selection, Dynamic Range, and Bandwidth

The first step in detecting a fault in a physical plant is to determine what features must be analyzed and to select a sensor or an array of sensors capable of measuring those features. The selection of features, associated sensors, and the placement of such sensors is the topic of substantial bodies of research. The focus of this research is how to proceed once these selections have been made, though some general concepts relevant to the physical plants occurring in mobile networks will be discussed in the following chapters. Once the sensors have been selected and placed, the signals they output must be properly acquired. Most sensors relevant to the type of fault monitoring in this research, namely accelerometers and hall-effect current sensors are inherently analog, as is the world that is being measured. Analog sensors output a signal, generally current or voltage, which is analogous to the physical property being measured. It should be noted that analogous does not mean there is a linear proportional relationship between the property being measure and the output. Analog signals are also continuous and have infinite resolution, meaning that the signal can take any real number value at any infinitesimal division of time. Finally, analog signals have a certain bandwidth that describes the range of frequencies contained within the signal. In order to acquire these signals properly into a digital processing unit, these analog properties must be understood in order to maximize the usability and information content of the captured signal.

4.1.1.2 Digital to Analog Conversion

The processing and communication of the data in this research is all performed in the digital realm. While the signals that are being measured and evaluated are inherently analog, nearly all modern computation units are general purpose digital processors. It is necessary then that the analog signals in question are converted to the digital realm. This is accomplished through the process of sampling. A device that performs this sampling is called a Digital to Analog Converter. There are several architectures of Analog to Digital Converters available, but the fundamental concept is that the device measures an input signal at a regular interval and produces a binary digital output that represents the analog input with some finite resolution. Most ADCs output a binary output of a certain resolution that is linear to the voltage of the input signal over the voltage range of the ADC. Resolution refers to the amount of information used to store each sample, and thus the number of finite, discrete steps used to represent the level of the signal. The fundamentally important characteristics of an ADC are its sampling rate, resolution, and voltage reference and ranges, though there are a number of other important parameters that relate to the accuracy of the device. These parameters are selected based on a number of criteria related to what is being measured and what processing will be done on the measurements. There are a number of theorems on digital sampling, including the Nyquist theorem that specifies the minimum sampling rate necessary to properly sample a signal with a given bandwidth. The parameters chosen for the device used in this research are related to the specific task of fault detection.

4.1.1.3 Signal Conditioning

In many cases, the analog signal from the sensor output cannot be connected directly to the Analog to Digital Converter. At a minimum, the amplitude of the analog signal from the sensor, usually measured in volts, must not exceed the voltage input range for the ADC device, otherwise the ADC will be damaged. Ideally, the voltage output by the sensor should match the range of the ADC in order to maximize the dynamic range, or the amount of useful information collected. In order to match the signal levels, it is necessary to use analog integrated circuits called operational amplifiers to shift and scale the signals to the proper range. The ADC used in this research requires a voltage between 0-3 VDC, while the hall-effect current sensor being used outputs a signal between -15-+15 VDC. The fault monitoring device has an array of operational amplifiers that multiplies the signal by 0.1 and then adds 1.5 volts to the resulting signal so that the signal now matches the input of the ADC. This signal conditioning circuit can be easily tuned to work with sensors that have a wide variety of signal output levels.

Once the signal levels are properly matched, the ADC is technically able to collect data from the sensor, but other considerations must be taken into account in order to get a usable digitization of the signal with maximal information content. It is important that there is compatibility between the impedance of the ADC input and the output of the analog sensor. If the ADC does not have sufficiently high impedance, it can consume too much current from the analog sensor and distort the analog output. A high impedance voltage-follower constructed from an operational amplifier can be used for this impedance matching. Another major consideration is aliasing, a form of distortion that occurs when the frequency content of the signal is not properly matched to the sampling rate of the ADC. In order to avoid aliasing, it is necessary to use an anti-aliasing filter that attenuates frequencies in the input signal that are above half the sampling frequency. Generally, the chosen sampling frequency is chosen in relation to the expected bandwidth of the signal coming from the plant up to the bandwidth limit of the sensor itself. Most accelerometers used to measure vibration for example, have a maximum bandwidth of several kilohertz while a physical plant may only give of vibrations with frequency content in the hundreds of hertz. It is important to choose a sampling frequency at least twice the bandwidth of the relevant signal. Because the shape of the waveform is important for many fault detection algorithms, the rate chosen in this research is several times higher than the bandwidth of the sensor. All frequencies above the relevant frequencies to capture should then be filtered out using the anti-aliasing filter.

4.1.1.4 Control Flow and Logic of Firmware

Once an anti-aliased, impedance matched signal with a proper voltage range in relationship to the ADC is connected, the ADC can then begin taking samples of the data that have meaningful information content. A processor unit, usually a DSP or microcontroller connected to the ADC must then read the binary output from the ADC and do something with that value. The ADC used in this research samples at relatively high sample rates and resolutions and produces a substantial amount of data. Because the research system is designed for low-power operation using embedded components, it does not have the ability to process or stream this information in real-time over its communication device. As a result, the data is stored in attached memory and processed or sent as fast as possible. The use of a slower sample rate or lower resolution would potentially allow the data to be streamed real-time, however. The firmware on the device is what configures and controls all of the peripherals, performs the necessary processing, and then coordinates the communication of the resulting data to other devices. Software running on a computer with more horsepower can also connect to the wireless node and receive its data.

4.1.2 Implementation of the Wireless Sensor Node

4.1.2.1 General Specifications

To achieve this necessary functionality, a custom device and software was created for this research. The final device consists of a printed circuit board containing a number of mixed-signal integrated-circuits designed to condition, sample, process, and communicate data coming from sensors connected to a real physical plant. As mentioned previously, it took several iterations and designs to achieve a system capable of fully meeting the requirements for this research. A previous design is described in [2]. At the time of this research and writing, there were no off the shelf systems that had the necessary performance specifications needed for this project. Namely, the device had to be capable of relatively high-performance sampling at rates of at least 62.5 Ksps per channel for 4 channels at 12-bit resolution. Simultaneously, the device has to be compact and consume small amounts of power so that it can be used in deeply embedded applications without easy access to external power. The device can be seen below in Figure 4.1.

Figure 4.1 Final Iteration of the ARRI Wireless Sensor Node for Fault Data Collection

4.1.2.2 Power Supply Design

The most fundamental part of any hardware system is the power supply. This wireless data acquisition places a number of requirements on the power supply that required significant design effort. In order to meet the requirement that the overall system consume as little power as possible, it was necessary to use a combination of linear voltage regulators and switched-mode power supplies (SMPSs) to provide the primary voltages and currents needed by the integrated circuits components of the system. The majority of the power in the circuit is drawn by the digital signal controller, the WIFI module, and the signal conditioning circuitry. These circuits must be supplied by regulated supplies with various voltages. The digital signal core requires 1.9 volts for its core operation and 3.3 volts for its peripherals. The WIFI module and signal conditioning circuitry operates at 3.3 volts. Additionally, to preserve the integrity of the analog signals that are measured by the device, it is necessary to separate the digital and analog power supplies to prevent digital noise from corrupting the analog waveforms. Finally, all of the power for the circuitry must be provided from a battery that provides power at a higher voltage than any of the specific board requirements. To meet all of these requirements, the power supply for the total circuit includes two SMPSs and three low dropout linear voltage regulators (LDOs). The two SMPSs (1.9V and 3.3V) connect directly to the battery and supply the digital circuitry including the WIFI module and the digital signal controller. The 3.3V SMPS also provides input power to the 1.9V LDO that supplies the analog input of the digital signal controller. The SMPSs uses in this circuit are National Semiconductor LMZ1200 that achieve up to 92% efficiency in the power conversion. This topology reduces the amount of power wasted during the conversion process and thus reduces the total power consumption of the device. Finally there is a 5V LDO that connects directly to the battery that in turn supplies the input voltage to a 3.3V LDO, the onboard USB connection, as well as any 5V sensors that might be connected. The 3.3V provides the clean analog power supply used by the signal conditioning circuitry. In sum, the power supply topology is designed to balance the competing requirements of power efficiency with low-noise performance to maximize the signal integrity.

4.1.2.3 Analog Section Design

In order to properly interface a wide range of sensors, the circuit in this system has an analog section that handles signal conditioning and anti-aliasing. The device takes 4 simultaneous analog inputs and therefore has 4 identical analog circuit channels. For each channel, a sensor is connected to the circuit using a robust pluggable terminal block connector to ensure good electrical connection even in environments with lots of vibration. The signal is immediately fed to a single channel of a quad-channel, high-bandwidth, low-noise operational amplifier in a voltage-follower configuration. This is used to impedance match the device input with the output of many different types of sensors. The impedance-matched signal is then shifted and scaled so that the resulting signal is between 0-3 VDC as required by the Analog to Digital Converter. This shifting and scaling is accomplished using the remaining three channels of the quad-channel operational amplifier. The scaling and shifting factors can easily be set by changing the resistors on the PCB in order to accommodate different types of sensors. Once the signal has been properly conditioned, it is fed to an LTC1563 4^{th}-Order Active RC Low-pass filter with a Butterworth filter response in order to filter out any frequencies above the sensor bandwidth and/or half of the sampling frequency. The anti-aliased signal is then fed into the ADC peripheral of the Texas Instruments Digital Signal Controller.

4.1.2.4 Digital Signal Controller Specifications

The processing unit used in this device is a Texas Instruments TMS320F28334 Digital Signal Controller (DSC). This controller was chosen because it has a unique combination of features and peripherals necessary for this research project. To begin, the controller has a high-performance integrated ADC capable of sampling up to 12.5MSPS with 12 bit resolution. The controller also has an external parallel interface necessary for connecting a large static ram buffer used to hold the collected samples. Because the device is a microcontroller / digital signal processor hybrid, is also has a floating point unit and optimized DSP libraries crucial for executing fault detection algorithms on-board. Despite its wide array of features however, the device is highly integrated and consumes a relatively small amount of power, another fundamental requirement for some of the project's use in cases especially when on lightweight, expendable vehicles that have very limited power sources. The DSC executes compiled C code and has access to a number of optimized signal processing libraries that can assist with the time-domain and frequency-domain signal analysis. This allows for the DSC to process or pre-process the data and reduce the amount of data that must be transferred.

4.1.2.5 Wireless Communications

In order to communicate with other nodes or a base station, the device also has a wireless communications module. While previous iterations of the device used a module based on the 802.15.4 wireless standard, the final version of the device uses a Roving Networks 802.11g Wi-Fi module. This module allows for high-speed robust communications over long ranges while still consuming small amounts of power. Previous versions of the device required taxing communication protocols in order to ensure reliable data transmission that dramatically reduced effective bandwidth increasing transmission time and power consumption. The Wi-Fi device selected for the final version natively and efficiently uses the TCP protocol to accomplish such reliability preserving bandwidth and minimizing power consumption. The use of Wi-Fi and TCP/IP based communications means that no custom base-station is required. The device can immediately connect to existing 802.11 wireless infrastructures or create ad-hoc connections to other individual wifi devices.

4.1.2.6 Printed Circuit Board and Mechanical Design

All of the components of the device are surface mount integrated circuits that have been integrated onto a 4 layer printed circuit board (PCB). The design of the PCB itself is no trivial effort requiring careful attention when selecting signal routes between components in order to preserve the integrity of the analog signals. The PCB is also mechanically designed in order to fit inside of an aluminum box. This box protects the board from physical damage but also electrically shields the circuit from external electrical noise. A CNC mill was used to create the faceplates for the box that allows access to the sensor inputs, antenna, button, led, and power input.

4.2 Fault Simulation Experimental Setup

4.2.1 Introduction

The previously described wireless sensor node was tested using two experimental fault generation experiments. The first experiment is a mechanical fault generation test bed that is designed to reproduce the types of vibrations that would occur should a bearing start to fail in a motor. The second experiment is an electrical fault generation test bed that is designed to simulate the failure of insulation in the stator windings of a motor and the resulting internal shorting that would occur. The author of this thesis participated in SBIR grant work along with Signal Processing Inc., in order to collect data from these fault simulators in order to test the ability of the hardware in conjunction with software algorithms to properly indentify and classify the faults. These experiments and more results are fully detailed in [11].

4.2.2 Mechanical Experiment

The mechanical experiment showin in figure 4.1 consists of a 3-phase, 2 pole, 1HP squirrel cage induction motor (SCIM). A custom hub was machined that would attach to the shaft of the motor and allowed for various weights to be attached. This offset weight would create an imbalance on the rotor similar to what would be expected during a bearing fault. Additionally, and ADXL-330 3-axis accelerometer was epoxied to the motor frame in order to pick up the resulting vibrations. The x, y, and z outputs of the accelerometer were attached to the inputs of the wireless sensor node for data acquisition and analysis. The entire setup was located inside of a wooden enclosure for safety.

Figure 4.2 3-phase Motor and Rotating Hub (Mechanical Experiment)

4.2.3 Electrical Experiment

The electrical experiment shown in Figure 4.2 consists of a number of inductors connected in series used to simulate the stator windings in an induction motor. An arc-generator was created that could be placed at various points in the inductor series to simulate the transient shorts that would be expected had the insulation of the stator windings partially or completely failed. A hall-effect current sensor was attached to the current source feeding the experiment. The output of this sensor was then connected to the wireless sensor node for data acquisition and analysis, with the goal being to determine at which location the short was in the windings.

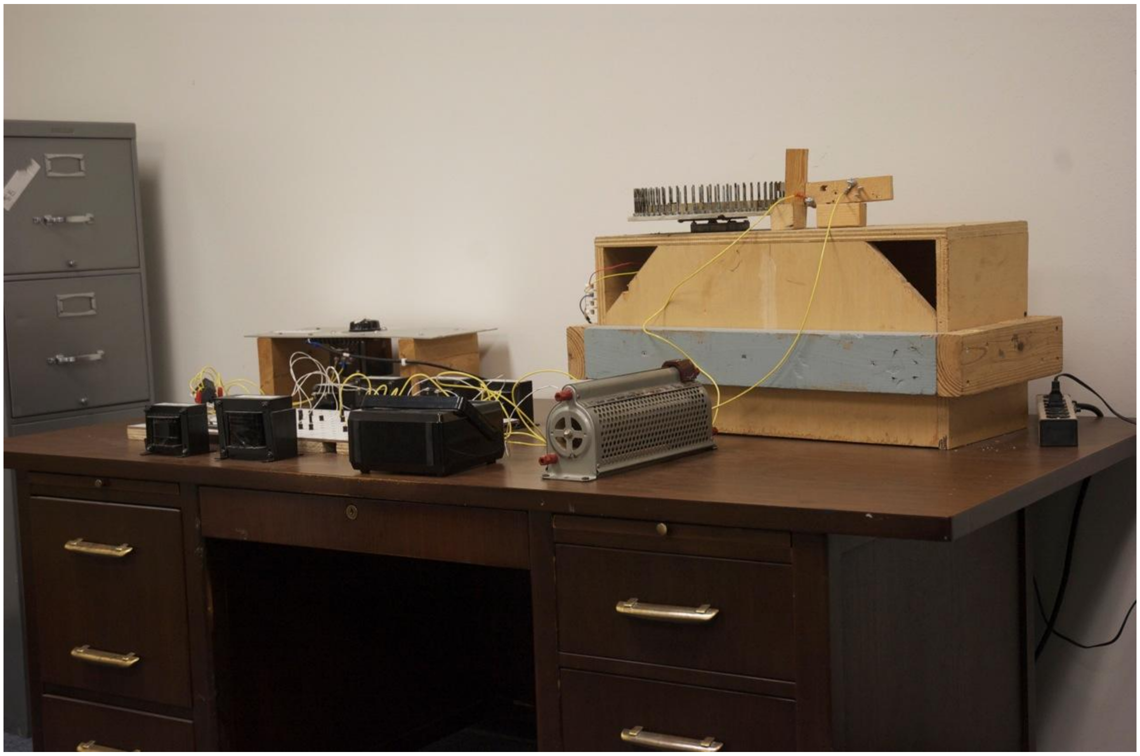

Figure 4.3 Components of the Electrical Experiment

4.3 Discrete Event Controller Simulation

The final component to perform the experiment was the creation of a functioning Discrete Event Controller (DEC) and intermediate software capable of responding to the updated diagnostic information. A new simulation was created in Matlab to simulate a theoretical firefighting mission. The simulation consists of creating and entering the matrices specified in section 3.2 and performing the state update equation at a regular interval. Specific information on simulating a DEC can be found in [18]. The results of this simulation are shown below in 4.4.3.

4.4 Experimental Results

The previous sections detail the physical implementations of the components that make up the complete fault diagnosis and response framework. The components were tested individually and the interactions were simulated. It was not possible as part of this thesis research to create a fully functioning large scale network of mobile agents with fault detection capability.

4.4.1 Wireless Sensor Node Experimental Results

The first tests conducted were on the system used to collect the data for fault analysis. The device had to be thoroughly tested throughout its several iterations in order to achieve the performance that it needed in order to operate in the environment that it was created for. It was first necessary to evaluate the functionality and noise characteristics of the analog section to ensure integrity of the signal. Once this was assured, it was necessary to confirm that the analog to digital converter properly sampled the data with a sufficient dynamic range, low noise, and precisely at a regular interval. It was also necessary to validate the firmware running on the device in order to make sure no logic flaws existed in the operation of the device. Finally, it was necessary to test and ensure the integrity of the data as it was transmitted from the DSC, to the Wi-Fi module, and eventually to the remote node. The device has been rigorously tested and fully meets the specifications it was designed for. It is capable of properly conditioning the incoming analog signals, digitizing them, processing them, and reliably communication the data to remote nodes as quickly as possible for a low-powered embedded network device.

4.4.2 Fault Diagnosis Experimental Results

The verified wireless sensor node was then connected to the fault simulation experiments detailed previously and an algorithm designed by Signal Processing Incorporated was used to take the data and classify the fault. For the bearing fault simulator, the algorithm was able to properly classify the relative bearing fault severity in all of the tested cases. This is a somewhat trivial result as the fault is easy to classify based on a number of individual features including the power at given frequencies, variance characteristics, and kurtosis statistics. More complex faults often can only be detected by simultaneous analysis of a higher-dimensional feature vector and in these more complex cases, it is not typical to expect 100% proper classification. Typical methods provide a best estimate and a confidence value for that estimate. Continuing work is being done on detecting more complex faults, though this thesis is only concerned with the ability of the hardware to collect fault-detection data as well as the ability of the high-level framework to respond to detected faults. The electrical experiment results were not available at the time of publication of this thesis. Initial data collection results indicated a malfunction of the experiment and the malfunction was not rectified before publication. The important conclusion is that the infrastructure is in place to allow for accurate data collection from the experiments and the possibility of making proper fault classifications using the system.

4.4.3 Discrete Event Control Simulation Results

As previously mentioned, the Discrete Event Controller defined in chapter 3 was extended to allow it to respond to changes in the health of the resources in real-time. The DEC simulation is written in Matlab and additional functionality was added to simulate the receipt of data from the fault diagnosis system indicating that a particular resource no longer had the capability of servicing a specific task. In this case, the dynamic priority assignment matrix (DPAM) was updated to reflect the resource health and the F_r matrix was updated at each iteration. In this case, the DEC was able to adequately respond to failures of given resources and continue the mission if possible. Figure 4.3 and 4.4 show the binary states of the discrete event controller as it progresses through the mission. For the task lines, when the task is activated, the line is in the high position and returns to the low position when the task is completed. For the resource lines, a line in the high position indicates the resource is available and in the low position indicates the resource is currently used by a task. In these cases, the tasks are programmed to fire sequentially in order to demonstrate a simple example of the fault response characteristics of the framework. Substantially more complex missions can be programmed for real supervisory control cases. Figure 4.3 shows a case in which for task 4, resource 2 is the preferred resource. It can be seen that when task 2 is fired, resource 4 is unavailable.

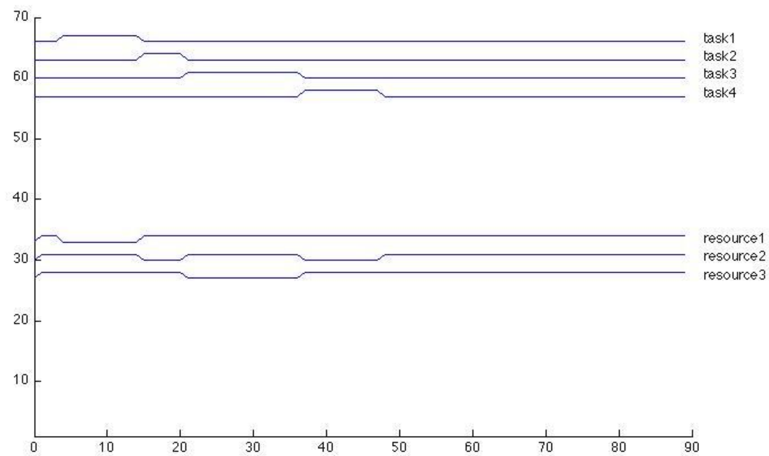

Figure 4.4 Discrete Event Control Simulation Case 1 (Resource 2 Available)

In figure 4.4, the DPAM has been updated to indicate that resource 2 is not available for task 4 due to a detected fault in that resource that prevents it from functioning properly. In this case, the greedy algorithm searches the DPAM to find an alternative resource and in this case chooses alternate resource R1. In 4.4, it can be seen that when task 4 fires, resource 1 is used in place of the failed resource R2.

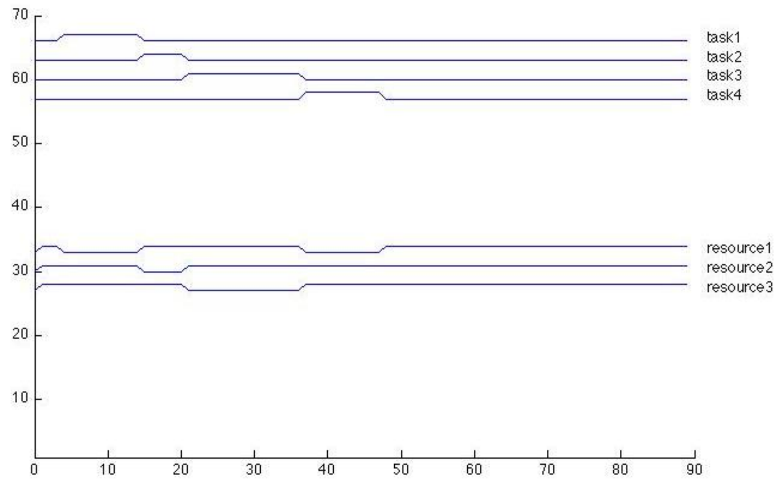

Figure 4.5 Discrete Event Control Simulation Case 2 (Resource 2 Failure)

While the necessary functionality has been achieved via the DPAM extensions, further research is still needed in dynamically determining the S_r matrix online. This matrix determines which resources are released when a given rule is fired. For this research, a book-keeping method was employed that kept track of which resources were dispatched according to the DPAM at each time iteration. This method does not fit in well with the matrix-based DEC formulation and does not allow for clear matrix-operations for proper resource release.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

This thesis introduced a framework that allows for a supervisory controller of a large team of agents in the process of completing a mission to adapt to inevitable mechanical failures that arise during mission execution. Furthermore, it details the design and creation of a hardware and software infrastructure that can be installed on each agent to allow for the collection of signals used in fault analysis, identification and classification. The design of the hardware, firmware, and software components for data acquisition were verified to operate correctly through experimentation and to meet the specifications mandated by the grant that funded their creation. Through simulation, it was also verified that the discrete event control supervisory controller can successfully be extended in order to take account of the real-time health conditions of the agents and still properly dispatch the necessary tasks and resources. The DEC itself has been the subject of a number of papers and its own functionality has been proven by experimentation and rigorous mathematical analysis of its functionality. This thesis also showed that by the nature of its construction, it also has several inherent characteristics that guarantee a certain level of performance and operational correctness.

5.2 Future Work

While this thesis introduces a functional framework, there are a number of areas in which the functionality of the framework can be improved. In the area of fault identification, a particularly useful extension of this framework would be to include a model-based method of detecting faults in the mobile agents. The methods in this paper rely on a data-driven approach that looks for signals indicating anticipated faults. In reality, mobile agents are likely to experience many unanticipated faults as well. Moreover, mobile agents typically have a model-based low level controller for their navigation such as a Kalman filter. By comparing the expected output of the model given by the Kalman filter observer vs. the actual output as measured by the sensors, a binary decision of whether or not a fault exists can be easy to discern. A second important area for research includes determining methods for implementing the extensions to the supervisory controller such that they fit within the matrix-based discrete event control framework. The matrix based framework has a number of advantages including its ability to be easily analyzed and implemented. The current framework requires comparatively more convoluted means of keeping track of the resource release information.

REFERENCES

[1] P Ballal and V Giordano…, "Deadlock free dynamic resource assignment in multi-robot systems with multiple missions: a matrix-based approach," *Control and Automation*, Jan 2007

[2] P Ballal et al., "Mechanical fault diagnosis using wireless sensor networks and a two-stage neural network classifier," , 2009, pp. 1-10

[3] P Chen, "Bearing condition monitoring and fault diagnosis," *dspace.ucalgary.ca*, Jan 2000

[4] AN Das, DO Popa, PM Ballal, and FL Lewis, "Data-logging and supervisory control in wireless sensor networks," *International Journal of Sensor Networks*, vol. 6, no. 1, pp. 13-27, 2009

[5] R. C. Eisenmann Sr. and R. C. Eisenmann Jr., *Machinery Malfunction Diagnosis and Correction*. Englewood Cliffs, NJ: Prentice Hall, 1998

[6] V Giordano, P Ballal, F Lewis, B Turchiano, and JB Zhang, "Supervisory control of mobile sensor networks: math formulation, simulation, and implementation," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 4, pp. 806-819, 2006

[7] V Giordano, F Lewis, and B Turchiano…, "Matrix computational framework for discrete event control of wireless sensor networks with some mobile agents," *Intelligent Control*, Jan 2006

[8] G. S. Gursaran, S. Kanungo, and A. K. Sinha, "Rule-base content verification using a digraph-based modelling approach," *Artificial Intelligence in Engineering* , vol. 13, no. 3, pp. 321–336, 1999

[9] R Heng…, "Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition," *Applied Acoustics*, Jan 1998

[10] CH Kuo and JW Siao, "Petri Net Based Reconfigurable Wireless Sensor Networks for Intelligent Monitoring Systems," , vol. 2, 2009, pp. 897-902

[11] C Kwan et al., "Real-time system condition monitoring using wireless sensors," *Aerospace conference, 2009 IEEE*, pp. 1 - 8, 2009, 10.1109/AERO.2009.4839681

[12] F Lewis, G Hudas, C Pang, and M Middleton…, "Discrete Event Command & Control for Networked Teams with Multiple Missions," *oai.dtic.mil*, Jan 2009

[13] H Martin…, "Application of statistical moments to bearing failure detection," *Applied Acoustics*, Jan 1995

[14] D Popa and M Mysorewala…, "Deployment algorithms and indoor experimental vehicles for studying mobile wireless sensor networks," *International Journal of Sensor …*, Jan 2009

[15] A Ramani, "Diagnosis And Prognosis Of Electrical And Mechanical Faults Using Wireless Sensor Networks And A Two-stage Neural Network Classifier," *dspace.uta.edu*, Jan 2008

[16] V Schiraldi, V Giordano, and D Naso…, "Matrix-based scheduling and control of a mobile sensor network," *17th IFAC World …*, Jan 2008

[17] A Shukla, H Garg, G Varshneya, and A Srivastava, "Real time acquisition of vehicle diagnostic data using wireless sensor network," *Wireless Communication and Sensor*

Networks (WCSN), 2009 Fifth IEEE Conference on, pp. 1 - 5, 2009, 10.1109/WCSN.2009.5434800

[18] D Tacconi…, "A new matrix model for discrete event systems: application to simulation," *Control Systems Magazine*, Jan 2002

[19] A Tiwari and P Ballal…, "Energy-efficient wireless sensor network design and implementation for condition-based maintenance," *ACM Transactions on Sensor …*, Jan 2007

[20] G Vachtsevanos, F Lewis, M Roemer, and A Hess…, "Intelligent fault diagnosis and prognosis for engineering systems," *Wiley Online Library*, Jan 2006

BIOGRAPHICAL INFORMATION

Matthew Benjamin Middleton received a Bachelor of Science in Foreign Service from Georgetown University in Washington, DC in 2005. Having a longstanding interest in computers and electronics, he returned to school in 2006 to begin formal study of those disciplines and entered the Masters program at the University of Texas at Arlington in 2008. Matthew is particularly interested in autonomous vehicles and in the many components that make them work, including the electronics, control systems, and software algorithms. During his tenure at UTA, he received a Research Assistantship under Dr. Frank Lewis at the Automation and Robotics Research Institute where he worked on all facts of autonomous vehicle design and implementation.