University of Southern Queensland
Faculty of Health, Engineering & Sciences

Sensor Platform for Monitoring Conveyor Belt Rollers

A dissertation submitted by

David Cooper

in fulfilment of the requirements of

ENG4111 Research Project I

ENG4112 Research Project II

towards the degree of

Bachelor of Mechatronic Engineering

Submitted: October, 2015

Abstract

Conveyor belts are a critical part of many large mining and materials processing operations. Monitoring systems exist for almost every component of these belts except for the belt rollers themselves. In a typical conveyor system, there can be hundreds and often thousands of rollers in operation, and their performance is vital to the overall performance of the conveyor system.

This project developed a self-contained monitoring platform which can be integrated within each roller. The platform operates wirelessly and is self-powered, meaning it can be sealed inside the roller and does not require maintenance or external hardware.

The sensor platform monitors the RPM, vibration signature and temperature of the bearings within each roller. Temperature and vibration monitoring were successfully tested, however RPM measurement was not included in the final function set of the platform.

Using a wireless mesh, the sensor platform demonstrated a range of almost fifty metres when transmitting through a steel conveyor roller. The transmitted data correctly represented the state of the bearing in terms of its temperature and vibration.

By implementing real-time, online monitoring of the conveyor rollers in a materials transport system, the sensor platform is expected to reduce the costs of down time and improve safety for maintenance personnel.

University of Southern Queensland
Faculty of Health, Engineering & Sciences

ENG4111/2 *Research Project*

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering and Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering and Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Executive Dean

Faculty of Health, Engineering and Sciences

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

David Cooper

0050025878

_____

Signature

_____

Date

Acknowledgements

Thanks goes to my supervisor, Professor John Billingsley for his words of advice and invaluable perspective.

To make it to this point I have been fortunate enough to have had the support and understanding of friends and work colleagues, whom I thank for being great sounding boards for ideas.

The most thanks of all goes to my wife Amy, who has supported me on this journey, and has been wondering where I have been for quite some time.

David Cooper

University of Southern Queensland

October 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The many moving parts in conveyor systems make inspections and maintenance tedious as well as hazardous. In addition, the high rate of materials transfer of most conveyor belts means that any down time of the system translates into significant production losses.

While many parts of a conveyor system, such as the belt itself and the drive motors and gears, can be monitored from a single sensor location each, the conveyor rollers or "idlers" require many more sensors in many more locations. This is because the idlers are spaced out along the length of the belt, so to achieve individual monitoring of each idler, an independent, local sensor for monitoring is needed for each unit.

To make monitoring each conveyor roller feasible, this project aims to develop a zero maintenance device, designed to fit within a roller and monitor relevant performance metrics. The device will be self-powered and capable of wireless communication, with the intent that minimal external components are required to enable the system to function.

Rather than complicate or add extra management needs to an already complex conveyor system, this project aims to develop a low-cost add-on to applicable systems to provide a useful stream of data while operating invisibly within that system.

1.1 Objectives

To achieve the aims of the project and deliver an appropriate solution, a set of objectives were specified. Guided by the initial Project Specification, these objectives are:

- Document the inspection and maintenance challenges faced by conveyor system operators. This includes the cost of downtime and repairs when idler components fail.
- Research the current solutions available to monitor conveyor idlers.
- Document the common failure modes of conveyor idlers.
- Determine the performance metrics of conveyor idlers to monitor to enable performance measurement and failure prediction.
- Survey the market and select a sensor platform to monitor conveyor belt rollers. Given the wide range of embedded processing platforms, one should be selected to satisfy the power and space restrictions of this project, while offering enough flexibility and capacity for future upgrades.
- Design and construct a suitable power supply system. The power system should be able to support the operation of the sensor and charge a backup battery when in normal running mode, as well as draw power from the battery in standby mode.
- Select and implement a wireless communication system. This system must be capable of wireless meshing, self-configuration and healing, and support potentially hundreds of wireless transmitters/receivers.
- Test and evaluate the performance of the system by analysing the output data.
- Discuss the financial viability of the monitoring solution, by estimating the initial investment costs and maintenance costs compared to the benefits of improved belt roller monitoring.

1.2 Project Scope

This project involves the development of a remote monitoring system applicable to belt conveyors, and the hardware and software for the sensor platform only. A single pair of sensor platforms will be built for testing purposes.

1.3 Justification

Maintenance and monitoring are an important part of industrial operations, and although they can be expensive undertakings, the cost of unplanned downtime and the resulting loss of production can be even higher. After working on-site at many coal mines around Queensland and New South Wales, it is apparent that many conveyors are in a state of disrepair due to the cost of stopping production to inspect them, combined with the cost of labour to perform the inspection.

In terms of production at a typical mine, a major conveyor belt shutdown can cost millions of dollars per day. A copper mine, for example, can extract 5,670 tons of ore per hour, which equates to $320,000 of copper and molybdenum per hour (Schools 2015). A shutdown for just a couple of hours can vastly impact the mine's income. Another case study, undertaken by Fenner Dunlop (2015), investigates the cost of a single day of lost production at Anglo Coal's Moranbah North coal mine. It is estimated that such an event would equate to over one million dollars in lost production.

An additional concern is personnel safety. Conveyors have many moving parts, known as "pinch-points", which are a hazard to employees inspecting or working nearby an operational belt. The statistics claim that 57 percent of conveyor-related accidents occur during cleaning or maintenance tasks, and not through regular work routines (Giraud, Massé & Schreiber 2004).

By developing a system which enables remote, unmanned monitoring, and early fault detection of conveyor roller failure, the costs of maintenance and monitoring may be reduced.

Chapter 2

Literature Review

In this section, the knowledge gained in the process of developing a conveyor belt health monitoring solution has been assembled. It represents a summary of the "state of the art" of conveyor monitoring and in particular, Bearing Condition Monitoring (BCM).

BCM is an important component of this project as this field covers the real-time monitoring of bearings and the diagnosis of their health by comparing measured data with manufacturer supplied data or other known values. To better understand this topic, research into the important aspects of bearing monitoring systems, the approaches used and the challenges faced was conducted.

The first section, over page, includes background information to give the project context and illustrate the conveyor belt systems which could benefit from this work.

2.1 Background

The process of transporting objects or materials from one place to another is fundamental to many commercial operations and is a well-established, mature industry. This section reviews the conveyor types relevant to this project, the problems those conveyors face and the health monitoring solutions currently available.

Not all conveyor systems use rollers, therefore the project's uses (and this review) are limited to Belt-On-Roller (BOR) systems. Despite this limitation, it should be noted that each BOR system tends to use at least 50-200 rollers, and systems using thousands of rollers are common. Therefore, the project has significant scope for deployment within this subset of the conveyor industry alone. In the following section, the components of BOR systems are covered in more depth.

2.1.1 Configurations of belt conveyor systems

Of the many types of conveyors, the belt-on-roller arrangement is the type used most commonly for materials transport applications. They have remained virtually unchanged for 150 years and are typically used for carrying materials long distances with a single motor (McGuire 2010).

The main types of BOR systems are flat-belt, trough-belt and pipe-belt, and in each type, combinations of rollers are used to support and shape the belt. The belt itself is typically a flexible sandwich of rubber, Teflon, PVC and urethane; the exact combination of these is dependent on the material being transported.

Depending on the length and power requirement of the conveyor system, the belt may be driven by a single motor, or a motor at either end, and sometimes there may be driven rollers along the belt's length. Typically, a belt conveyor consists of load bearing rollers which support the belt and material in the direction of transport, and another set of rollers located above or below the first set support the belt as it travels, un-laden, in the opposite or "return" direction.

To aid in differentiating types of belt conveyors, diagrams have been included below. Figure 2.1, below, shows the arrangement of a flat-belt conveyor. In this arrangement, the load bearing belt slides along a set of rollers, and below this, the belt returns on another set of rollers (which are most likely fewer and further between than the top set, as they are not load bearing).

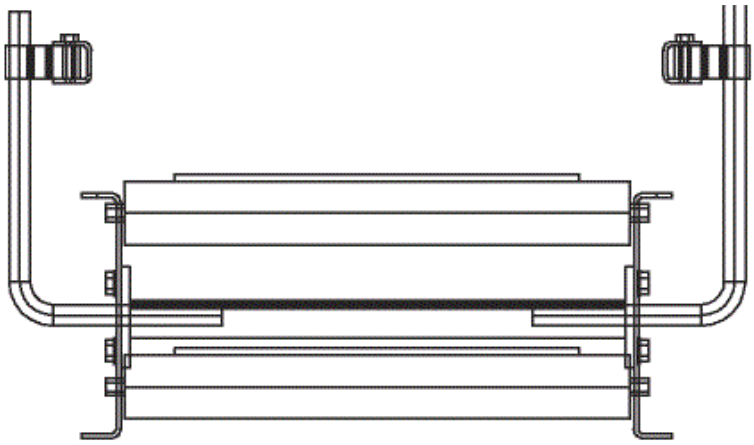

Figure 2.1: Cross-section of belt-on-roller conveyor (McGuire 2010, p. 37)

The most common variation on the above arrangement, and the type most often used for conveying loose materials (McGuire 2010), is the troughed-belt conveyor, illustrated below in Figure 2.2. Instead of a flat area for material to ride on, this conveyor type relies on another set of rollers at each edge of the top belt to bend the belt up to create a trough. The main advantage of troughed-belt conveyors is that the belt can be installed on uneven ground, enabling them to turn corners and travel up and down hills.

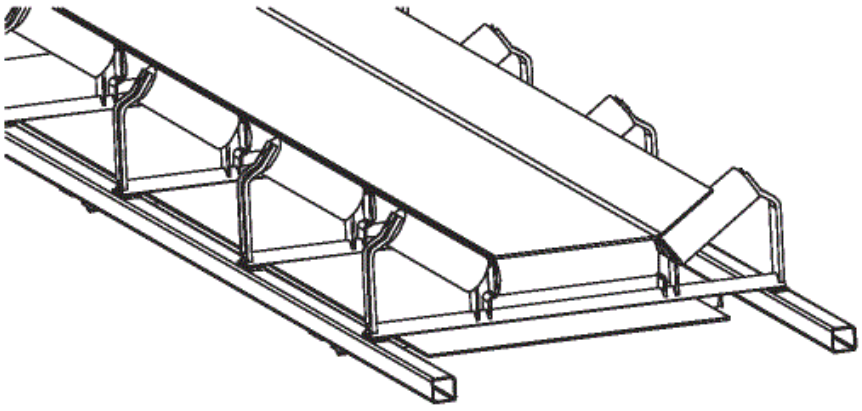

Figure 2.2: Simplified troughed belt conveyor design (McGuire 2010, p. 63)

The image below (Figure 2.3) shows a trough conveyor's ability to be curved as well as travel up and down an incline. This image also highlights the complexity involved in manually inspecting many conveyor systems, as they can be several kilometres long, on elevated platforms, and be hard to access for visual assessment.

Figure 2.3: Troughed belt conveyor (BEUMER 2012)

The other, less common variation on the trough conveyor is the pipe conveyor. A cross-section of the pipe conveyor is shown by Figure 2.4, in which the rollers can be seen on all sides, causing the belt edges to overlap and form a pipe. McGuire (2010) explains that this type of conveyor, although being more expensive to install and maintain, can climb steeper angles than the standard troughed-belt and can navigate tighter horizontal and vertical curves.

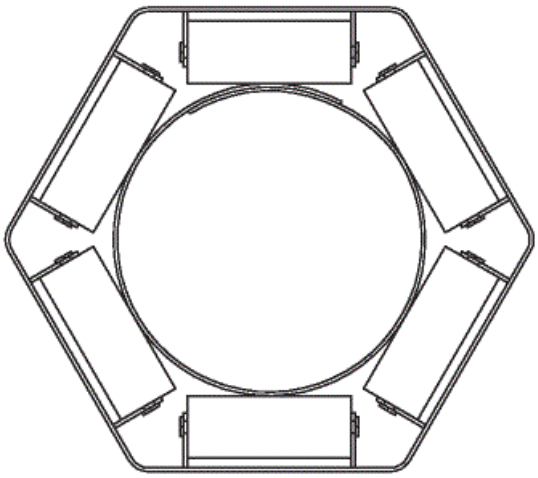

Figure 2.4: Cross section of a pipe conveyor (McGuire 2010, p. 66)

2.1.2 Roller Failure Modes

For this project, a roller is considered to have failed when it has either stopped rotating, i.e. experienced seizure, or has been externally worn enough to result in fracture and therefore be at risk of damaging the belt it supports. These are the extreme failure modes, and in reality, a roller causing excessive vibration or noise should be replaced before it reaches these stages of failure.

Conveyor belt rollers, being mechanically simple devices, experience failure in their only moving interfaces, those being the bearings and the outside of the roller. Located at either end of the central axle, a pair of bearings supports the outer shell or "drum" of the roller, so that it can be spun by the belt, which rests on the roller. Figure 2.5, below, presents a cut away view of one end of a typical roller. On the central shaft, a bearing is shown which attaches to the end cap. The steel outer shell is also attached to the end cap. As can be seen below, much of the centre of the roller is hollow, and it is in this space that the monitoring system is designed to operate.

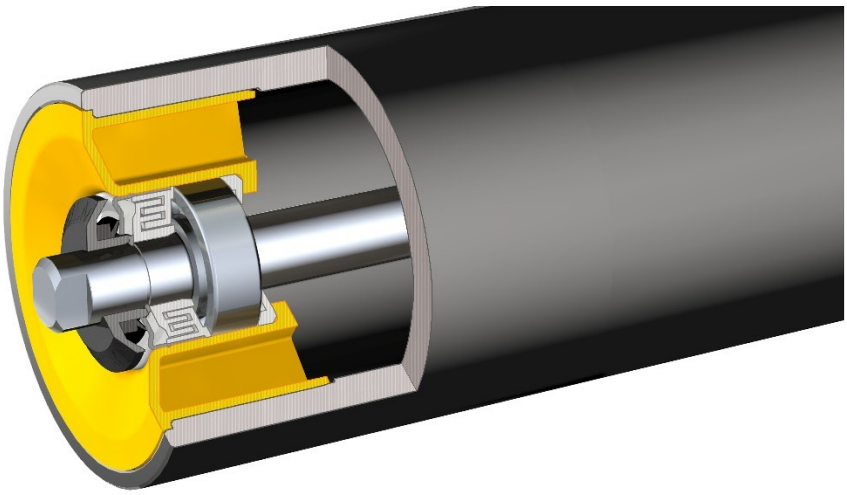

Figure 2.5: Cut away of a roller end (Rulmeca, 2013)

The first failure mode, which is the wearing through of the roller outer shell, leading to structural collapse, is the result of a combination of manufacturing error and in-service wear. Research by Watson and Niekerk (1989) discusses the wear issues caused by out-of-balance rollers, the measure of which is their Total Indicated Runout (TIR). This factor is not discussed by more modern day

research by Zhao (2011), however, which blames misalignment of the rollers with the belt as being a cause of failure. The resulting transverse movement of the belt across the roller creates vastly more wear than pure rolling friction would create. The difference in the conclusions of each paper is hypothesised to be due to the improvement in manufacturing techniques since 1989, leading to the likelihood of out-of-balance rollers diminishing (but the likelihood of manual misalignment staying the same).

The second failure mode, where the roller seizes completely, is attributable to the failure of its internal bearings. However, the cause of bearing failure is not due to the bearings simply reaching their end-of-life (known as their "service" life). As the SKF information page on "Bearing life and load ratings" explains, failures due to raceway spalling (which is more likely to be the result of a manufacturing defect) are very rare events (SKF 2015). According to SKF, bearing failure is significantly more likely to be the result of corrosion, improper mounting or tolerances, contamination, moisture or failure of the lubrication system. As a bearing manufacturer, this does not come as a surprise, that bearing issues are blamed on the implementation of the bearing and not the manufacturing.

Fortunately for SKF, the literature supports their claims. Articles by Zhao (2011) and Reicks (2006) point to breaching of the bearing seal, repeated impacts and contamination as common causes of conveyor roller failure. These articles only discuss the environmental aspects of roller wear, but the human factor is an important consideration as well. As found in a reliability investigation by Hughes (2004), at particular installation where the Maximum Time Between Failure (MTBF) had dropped to 2.5 weeks, the root cause was found to be over-lubrication and incorrect training of servicing personnel.

Evidently, ensuring continued operation of a conveyor system is no simple task, as it requires not only suitably designed and specified rollers, but a well-maintained installation which includes an understanding of the servicing required and trained workers capable of carrying out this maintenance. During normal operation the conveyor system must be regularly inspected to find rollers beginning to show signs of the failure modes discussed.

2.1.3 Existing roller monitoring techniques

Current solutions to monitoring conveyors are largely manual methods which require one or two personnel to be physically near the roller under investigation. The methods used rely on hand-held tools which measure and analyse heat, vibration, acoustic emissions or a combination of these.

An example of such a device is the BearingChecker, made by SPM Instruments. This handheld unit is specifically designed for bearing inspections and it uses the Shock Pulse Method (SPM) to detect bearing faults. It also includes an infrared sensor for "hot spot" detection and can be used as an "electronic stethoscope" to detect acoustic anomalies (SPM 2015). The product is shown in Figure 2.6.

Figure 2.6: The handheld BearingChecker (SPM 2015)

In many cases, conveyor rollers are difficult to access (Janse van Rensburg 2013) and so for a completely non-contact approach, several products are available which use infrared (IR) thermography to capture "hot spots". These are areas of the roller which are significantly higher than the surrounding areas. Figure 2.7 shows a product made by FLIR, the E6, which has a thermal sensitivity of less than 0.06 °C and an accuracy of ± 2 °C (FLIR 2015). When using such a device, the relative temperatures of nearby rollers must also be measured, to determine whether the "hot spot" is due to environmental conditions or is a sign of early bearing failure.

Figure 2.7: FLIR E6 handheld IR camera (FLIR 2015)

Until very recently, there were no known inspection solutions which offered remote monitoring of individual rollers. This changed in August 2015 when a Sydney-based company named Vayeron released their "Smart-Idler" product. The product has the same goals as the Roller Sensor Platform and is also intended to be a self-powered, wireless monitoring system which is integrated inside the roller. The one advantage the Smart Idler (Figure 2.8) has over this project is the inclusion of Radio Frequency IDentification (RFID) within the roller to enable its serial number and service history to be extracted directly (without using the wireless mesh network).

Figure 2.8: The Smart Idler from Vayeron (AIMEX 2015)

2.1.4 Benefits of remote monitoring

There are many benefits to implementing the remote monitoring of processes and plant. The benefits applicable to conveyor belts are highlighted below:

- Statistics about conveyor operations, e.g. tonnage/hour

  A remotely monitored roller network could maintain current and past records of the RPMs of each roller. From this record, the belt speed can be determined and estimates of production rates calculated.

- Improved safety; no physical access required

  The majority of current inspection techniques require the inspector to be located within reach of the conveyor. As mentioned in Section 1.3, these non-standard practices are the most likely to cause injuries. By keeping personnel physically separated from the conveyer belt but still able to access its health and performance information, less chance of injury exists.

- Up to date data

  Rather than sorting through hundreds or often thousands of conveyor roller maintenance records, a digital system can monitor roller age and determine expected lifespan remaining. Alerts could be set up to provide warnings about ageing rollers.

- Prevents damage to the belt and reduces the risk of fire

  By catching developing faults early, planned maintenance could be carried out, well before the risk of the roller either seizing or fracturing and causing belt wear and/or tearing.

- Maintains efficiency for lower overall power use

  Fewer seized rollers translates into less energy used to drive the belt and lower environmental impact, especially in terms of noise (Yusong & Lodewijks 2011).

2.2 Bearing Condition Monitoring

Found in almost all machinery with moving and rotating parts, bearings are fundamental to the operation of many machines. When used and protected appropriately, they can have long service lives, but the consequences of their failure can often be equipment damage and production losses.

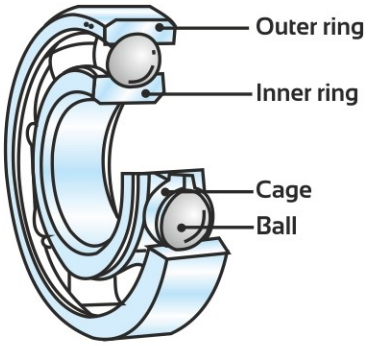

Figure 2.9: Ball bearing (SKF 2010)

Bearings generally consist of four components, the inner race, outer race, roller and cage. Each race is a ring, and between the inner and outer races sit the rollers, which are rolling elements (steel balls, in the case of non-thrust bearings) held in place by the cage. By placing rollers between two rings, bearings can carry load and eliminate friction (El-Thalji & Jantunen 2015).

Whether due to design error or manufacturing error, bearings are known to fail. A past study conducted by the Motor Reliability Working Group found that bearing failure (in electrical motors) was the cause of 41% of all break downs (IEEE 1985). Although the data in this study was collected in 1982, it is still relevant today as the mechanical construction of rotating machinery relies on the same components. What has changed, however, is the availability of electronic, digital monitoring systems in miniaturised packages which can be deployed at low cost.

2.2.1 Collecting bearing data

In recent years, new approaches have been developed for Bearing Condition Monitoring (BCM). The monitoring techniques relevant to conveyor idlers are based on either acoustic, vibration, or temperature inspection. Although Zhou, Habetler and Harley (2007) present other monitoring techniques such as chemical analysis, laser monitoring and electrical current monitoring, these are outside the scope of this review. For rolling element bearings, vibration and temperature are the most direct indicators of bearing failure (Yusong & Lodewijks 2011); these methods will be explored further.

When in operation, bearings should maintain a constant temperature. If damaged or worn, the moving part of the bearing will often experience increased friction, resulting in higher temperature (Scott et al. 2011). Using this principle, a healthy temperature range can be established and variations from this would be considered signs of trouble.

Much of the literature is focused on the application of BCM to existing machines, which have been designed without consideration for BCM. Due to the need for sensors to be tightly mounted to, and integrated with, monitored bearings, and because bearings are often sealed or hard to access, BCM is difficult to implement as an afterthought.

Of the considered monitoring approaches, vibration monitoring is most affected by mounting location. A review by Zhou, Habetler and Harley (2007) is particularly critical of this monitoring technique. In their review, they list the need for internal access to the machine, the need for many sensors, the need for expert mounting of sensors and the fact that sensors fail as disadvantages of vibration monitoring. However, if a machine was designed by the Original Equipment Manufacturer (OEM) to be compatible with vibration sensors, this would negate most of their criticisms except for the claim that sensors fail (faster than bearings). Once again, these sensors would be easier to access and replace if their location had been planned early in the machine's design phase.

In reality, many candidate machines were operating long before BCM became a viable option, and retrofitting these machines is a difficult task. Vibration sensors need to be mounted essentially on the bearing race itself. Typically, this

is not possible, so sensors are mounted on the bearing housing. Acoustic emissions sensors, while similar in operating principle to vibration sensors, have slightly more mounting flexibility, as they do not have to be mechanically linked to the bearing under measurement, but mounting them in close proximity to the bearing is still vital.

Sensor location is also important for temperature sensing. When considering temperature monitoring using a thermocouple, for example, mounting on the bearing housing makes the sensor slow to respond to changes within the bearing itself (Scott et al. 2011). Where possible, direct contact with the bearing race is preferable. It could be inferred then, that in all cases, mounting a sensor further away from the bearing race itself would reduce the responsiveness of the sensor to bearing emissions and make it more affected by other signal sources, i.e. "noise".

By approaching the monitored bearing from the inside, rather than the outside, this project expects to be able to circumvent most of the aforementioned challenges. The sensor platform is located within the sealed or "clean" side of the roller enclosure, which should mean that most existing roller designs could easily be modified to give direct access to the bearing races. Additionally, vibration sensors can mount directly to the shaft supporting the bearing, which should yield a signal relatively unpolluted by other sources of vibration.

Even with a clean signal source, sensor data is cannot be analysed in its raw form. Vibration signal data, especially, requires some processing before analysis can begin. This processing is covered in the following section.

2.3 Signal processing

Simply collecting and storing the data produced by sensors does not constitute a BCM solution. The data must be processed, to yield features of interest, and this signal processing is one of the major, challenging aspects of machine condition monitoring (Goyal & Pabla 2015).

Data processing can be a complex link in the monitoring system chain. As shown below, in Figure 2.10, this step is one of four main steps in the system. From left to right, the system begins with raw sensor input, which for this project is from a temperature or vibration sensor. This data must then be acquired and stored for enough time to collect a dataset to be processed. To perform this function, a microcontroller (with on-board memory) would be used. The next step is the processing of the data, which includes filtering, sampling, and can involve transformation to the frequency domain. Once a suitable data set is obtained, this needs to be interpreted to diagnose the state of the bearing. This can be as simple as comparison with a pre-calculated value (model-based approach), or it can involve Artificial Neural Networks (ANNs), expert systems, fuzzy logic or Support Vector Machines (SVMs) (El-Thalji & Jantunen 2015).

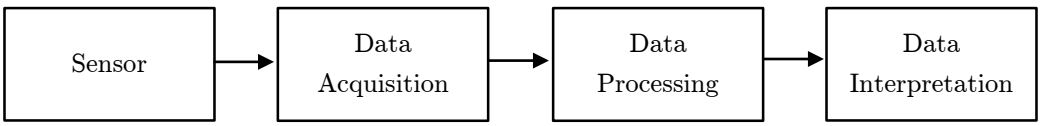

Figure 2.10: Vibration data analysis process

Much literature has been produced discussing signal processing techniques, and all the reviewed articles agreed on the advantages and drawbacks of each technique. Only literature discussing vibration sensor data processing was reviewed, as although temperature data will require some processing as well, a time-based average measurement is expected to be sufficient. A short overview of the common signal processing techniques is included in the next section.

2.3.1 Statistical Models

This category of signal processing methods includes the oldest and therefore computationally simplest analysis techniques. These methods are based in the time domain (as opposed to the frequency domain) and include parameters such as average, Root-Mean-Square (RMS), crest factor and kurtosis. Both El-Thalji and Jantunen (2015) and Goyal and Pabla (2015) claim that these are the methods most commonly used by current vibration monitoring systems.

By trending, or storing computed values over an extended period of time, a data set is produced which can then be analysed for signs of bearing failure. For example, if the RMS vibration speed increased over time but bearing rotation speed remained constant, this may signify a developing fault.

More involved measures, such as kurtosis, analyse the shape of the waveform in terms of its "spikiness". Kurtosis of a given dataset is defined as the ratio of the fourth moment and the second moment (variance) squared. Sawalhi and Randall (2011) found that the kurtosis of a vibration signature increased linearly as the defect size grew, so this can be a good method to use for early fault detection. However, the kurtosis value begins to plateau as the fault grows and so this technique is best for narrow bandwidth use at high frequencies, where the fault can be caught early. (El-Thalji & Jantunen 2015).

2.3.2 Fast Fourier Transform

Using vibration data collected in the time domain, analysis can be performed on the peak frequencies present in the data by using a Fast Fourier Transform (FFT) to view the signal in the frequency domain. As discussed by Norton and Karczub (2003), bearing element rotations generate a set of discrete frequencies, based on the bearing's geometry and its rotational speed. By analysing the vibration data in the frequency domain, it can be seen whether these discrete frequencies are present. If present, each frequency can represent a developing fault in the inner or outer races, the bearing cage or the rolling elements themselves. Once the peak frequencies are known, the data interpretation stage can be as straightforward as matching these frequencies to the bearing manufacturer supplied data to determine the location of the fault.

Like the other processing methods, a pure FFT computation is not ideal for extracting features of interest from real-world signals. Multiple authors point out that an FFT is unsuitable for non-stationary signals (i.e. noisy signals) and it results in a loss of time information, so that signal change over time cannot be determined. A simple improvement which addresses these issues is the Short Time Fourier Transform (STFT), which breaks up the data into much smaller time windows and performs a standard FFT on each window (Goyal & Pabla 2015), thereby capturing the variation in signal over time.

2.3.3 Wavelet Transform

Operating in a similar fashion to FFT transforms, Wavelet transforms use a family of functions which satisfy certain requirements (the wavelets) instead of sine and cosine functions (as in an FFT) to transform the signal. The surveyed literature, from Vidakovic and Mueller (1991), Goyal and Pabla (2015) and El-Thalji and Jantunen (2015) agrees that this approach is a good compromise between the time and frequency views of the signal. Additionally, the wavelet transform can represent many classes of function with much fewer basis functions than sine and cosine basis functions, which is very useful for data compression.

There exist two methods for calculating a wavelet transform, which are Discrete Wavelet Transform (DWT) and Continuous Wavelet Transform (CWT). The difference in each approach relates to how the scaling factor is chosen.

2.3.4 Hilbert-Huang Transform

Developed specifically to analyse real-world data, i.e. nonlinear and nonstationary processes, the Hilbert-Huang Transform (HHT) gives sharper results than the traditional methods to obtain time-frequency-energy representation (Huang, Wu & Long 2008). The HHT is a convolution between Hilbert Spectral Analysis and Empirical Mode Decomposition.

Many authors agree that some of the other advantages of the HHT are its computational efficiency as well as avoidance of issues relating to time and frequency resolution. However, Peng, Tse and Chu (2005) noted that there are some shortcomings in the HHT approach, which can be minimised by using a Wavelet Packet Transform (WPT) before applying the HHT.

2.3.5 Sampling Rate Requirements

In order to capture features of interest in the vibration data signal, a minimum sampling rate must be achieved. Too low a sampling rate results in an aliased sample output which does not accurately represent the input.

Typically, designers turn to the Nyquist-Shannon sampling theorem, but this guideline is often misused (Wescott 2015). The theorem states that as long as you sample faster than double the highest frequency component of the spectrum, then an accurate representation of the signal can be obtained. While this is a good starting point, it is preferable to start with a much higher sampling rate and, based on the nature of the signal, reduce the rate as much as possible while still satisfying the demands on the output signal. For this project, and the bearings sizes used, there are unlikely to be signals greater than 500 Hz (SKF 2010) of interest, therefore a sample rate of over 1 kHz will be desirable.

2.4 International Standards

In the field of vibration monitoring, many engineering standards have been produced that describe the sensor types to use, sensor mounting locations and data processing techniques. The primary standards are discussed in this section.

The first series of standards are the ISO 10816:1995 series. These cover the vibration measurement of machines on non-rotating parts. Most of the guidelines in this standard are not applicable to this project, as they are intended for steam turbines and generators above 15 kW. The "general guidelines", which is the first standard in the set, is much too general to be much use. It does mention the recommended sensor mounting locations but only for the measurement on non-rotating parts of horizontal machines. The diagram of the mounting locations is included over page, as Figure 2.11. Although basic, this diagram is duplicated by several other standards.

A more thorough standard on the topic of vibration monitoring is ISO 13373:2005. This series covers the approaches to signal measurement and processing in greater detail, and provides a more up-to-date summary of the current analysis techniques in use. The standard also includes a guide to help choose an analysis technique, based on the machine type and size to be monitored.

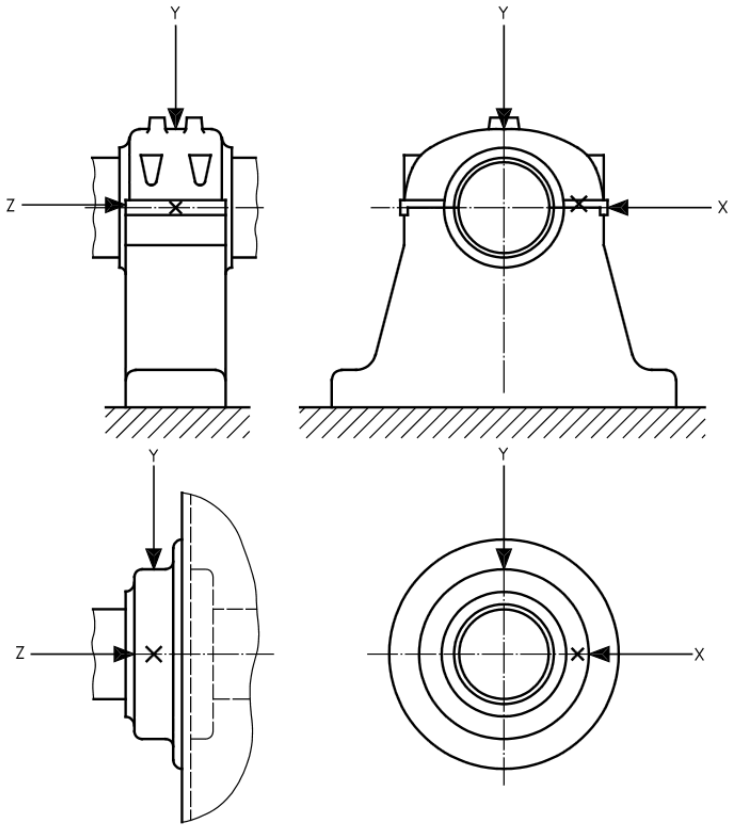

Figure 2.11: Transducer locations for vibration measurements (ISO 1995)

Another relevant standard is ISO 18431:2007, which covers the major approaches to signal processing in depth. This standard was an informative guide to the different analyses that can be performed on a data set.

Compared to other literature, the reviewed standards do not contribute much useful information or direction which is relevant to this project. This is largely because the standards are intended for large machines, which (traditionally) would have been the only size for which it was worth deploying an expensive commercial monitoring system.

Therefore, the standards have been used as a guideline rather than a rule, and the optimum locations of sensors will be determined experimentally to suit this specific application of vibration monitoring.

Chapter 3

Methodology

In this chapter, the approach to satisfying the project objectives is detailed. These objectives are:

1. Establish the characteristics of conveyor rollers which indicate their health.
2. Design a sensor platform to collect and broadcast these characteristics.
3. Test the performance of the system by analysing output data.

In essence, the Roller Sensor Platform (RSP) project can be broken into four stages: research, design, build and finally, test. At each project stage, a particular methodology was adopted, and the details of each approach will be explored in further detail.

Additionally, project considerations such as the project timeline and the potential risks during project execution are included.

3.1 Research Methodology

The first stage of the project involved an investigation into the components, operation and health indicators of conveyor belts, which was undertaken in the previous chapter.

When conducting this research, efforts were made to ensure the referenced material was suitable for use in a formal research capacity. This meant using research that was typically not more than fifty years old, and if older than this, scrutinizing it carefully to ensure it was still relevant. This relatively long time period was chosen based on the largely mechanical nature of conveyors and their components, and the expectation that little has changed in the construction of bearings over this length of time. This assertion was supported by McGuire (2010), who wrote that little had changed in belt conveying systems for 150 years. For the other research topics, e.g. (digital) wireless sensors and (digital) signal processing, much of the research has only been undertaken in the last three decades, which makes old, superseded research unlikely to be a concern.

The next criterion when evaluating research was its level of professionalism, in terms of its quality and peer reviews. Many articles were found which appeared to have been published in a journal but which were poorly written and had no evidence of a peer review process. Such articles were discarded and instead, articles which were either written by well known, prolific authors in the field or articles found in respected publications were referenced.

Finally, local research material was sought (whenever possible) to aim to meet the project objectives with a solution which is applicable to local needs. The exact requirements of any solution would vary with region and industry, and with a view to implementing this product in an Australian coal mine, the issues and environment of these operations were kept in mind when evaluating the relevancy of research material.

3.2 Design Methodology

As a proof-of-concept device, the RSP will be built using Commercial Off The Shelf (COTS) components as opposed to custom made assemblies of electrical components. Although this design approach leads to a solution which is not as integrated or compact as it could be, nor ready for field trials, it reduces the development and troubleshooting time. A large component of this project is the software development to poll data from connected sensors and transmit it wirelessly, and for this programming work it is important to have a hardware platform which is stable, well tested and well documented. A processing platform and/or a power supply built from scratch is unlikely to provide this so a COTS based approach was chosen instead.

To solve the problem of remote signal monitoring, data processing can either be performed locally on each RSP or centrally, by broadcasting the raw data to a central server. Both topologies have their merits, however, the decision was made to perform as much processing on-board as possible. This distributed processing approach reduces the strain on the wireless network and the central server. The decision was also influenced by the claim by Huang et al. (2015) that there is currently no bandwidth-efficient method to broadcast raw vibration data across a wireless network.

As the RSP is a new concept, it is important that its design can be accessed and improved on by others. This design goal meant that a commonly used hardware platform was preferred over an obscure one, and a programming language in widespread use was sought over a language with a niche market or specific application.

The following sections detail the design decision process and the resulting components chosen for the RSP.

3.3 Processing Platform Selection

To select an appropriate Micro Controller Unit (MCU) board on which to base the test platform, the following factors were taken into consideration:

- Accessibility: the platform should be easy to use and program, be thoroughly documented, and easy to obtain.
- Compatibility: the platform should be compatible with a wide range of readily available sensors, wireless transceivers and other peripherals to enable straightforward integration.
- Cost: The platform should be low cost as it is for an early proof of concept investigation only.
- Temperature requirements: Within a conveyor belt roller, it is assumed to be a sealed and therefore dry and dust free environment, so the MCU does not need to be waterproof. However, it may have to handle high operating temperatures. The maximum possible temperature (before the bearing begins to fail, and hence damage to the roller starts to occur) is typically 120 °C (SKF 2010).
- Physical size: The platform must fit within a mock-up of a mid-size roller, between the roller and the shaft. Ideally, size should be as small as possible.
- Power use: Due to limited battery capacity, the platform should draw as little power as possible. This in turn reduces load on the dynamo and the conveyor system as a whole. A platform which accepts 5 Volt input will be suitable.
- Clock Speed: Conveyor belts can reach speeds of 9 m/s (Yusong & Lodewijks 2011). Therefore, to measure RPM, the MCU will have to read a sensor at a minimum rate per second (Hertz). For example, for a conveyor speed v = 9 m/s, and a roller diameter of D = 100 mm (worst-case, i.e. smaller and faster than many conveyor rollers), the roller will spin at:

$$n_{ROLLER} = \frac{v_{CONVEYOR}}{\pi \; x \; D} = \frac{9}{\pi \; x \; 0.1} = 28.6 \; rev. \; per \; second \qquad (3.1)$$

If the roller rotates at up to 28.6 revolutions per second, this equates to a read rate of 28.6 Hz. The other speed requirement is for vibration frequency detection. Using the Nyquist-Shannon sampling theorem as a guideline, the sample rate should be at least double the frequency of interest. If the commonly used set of roller bearing tend to generate frequency signals at less than 500 Hz (SKF 2015), then the sample rate must be at least 1 kHz. As most modern MCUs operate in the MHz range, these clock speed goals should be easily achievable.

- Vibration requirements: The board may have to survive long-term impacts and shocks, and should be able to be securely mounted to minimise the potential for damage to occur.
- SRAM: In order to read, store and compute a high-resolution set of vibration measurements, significant amount of SRAM may be required. For the initial prototype, the program may not be fully optimised and may use more memory than necessary.
- EEPROM: Only very few variables will be required to be stored in non-volatile memory. This may be parameters such as the roller ID, and possibly a status indicator so that if powered on individually, the roller's state could be determined without a link to the network. In summary, the available EEPROM on most MCUs is unlikely to be restrictive.
- IO: The input/output requirements of the RSP are simple. Assuming a worst-case requirement of 3 vibration sensors with 1 analogue channel each, 3 temperature sensors with 1 analogue channel each, 2 digital channels for wireless interface, and 2 digital channels for LED indication, the total requirements are 5 analogue and 4 digital channels.

Based on these requirements, the chosen MCU board is the Arduino Fio. It is based on the ATmega328 MCU, has a compact, minimalist design with few extra components, yet it can accept 3.3-12 Volts input power supply and includes sufficient interface channels. The Fio also includes a lithium polymer battery charging circuit, and a connector for XBee devices, which makes attaching a range of wireless transceivers a straightforward task.

Below is a summary of the MCU board requirements, compared to the Fio's features:

Table 3.1: MCU Requirements vs Features

Requirement	Capability
Greater than 1 kHz clock speed	8 MHz
Operation in up to 120 degrees C environment	-40 to 85 degrees (MCU only) (Atmel 2014)
Low power	Up to 360 mW (MCU only) (Atmel 2014)
5 Volt power input	3.3-12 V (Arduino.cc 2014)
6 analogue IO pins	8
4 digital IO pins	14
Potentially large amount of SRAM	2 kB
Small amount of EEPROM	1 kB
Small size	28 x 65 x 5 mm (L x W x H)
Easy to program MCU	ATmega328 programmable in C/Arduino

3.4 Peripheral Component Selection

To support the MCU platform, a range of input devices (sensors), output devices (RF transceiver) and power sources (battery and dynamo) are required. The process behind the selection of each process is detailed below.

3.4.1 Dynamo selection

To power the RSP during normal operation, a dynamo will be used to remove the dependency on batteries. During normal RSP operation, the dynamo should be able to both power the platform and at trickle charge the battery. A small, 6 V DC motor (made by Mabuchi Motor Co.) was chosen to be used as a dynamo, as it is listed as being a 6 V motor which can output 0.8 W to 4.5 W. As the MCU requires only 360 mW, and the MAX1555 battery charging circuit requires up to 1 W, this dynamo should be able to support devices both easily. Due to the output voltage of the dynamo varying according to RPM, a voltage regulator circuit will be required to provide stable power for the MCU.

3.4.2 Power regulation

Under normal operating conditions, the RPM of the roller and the dynamo will be constant, resulting in a constant voltage supply from the dynamo. However, in over-speed or under-speed conditions, the voltage will vary and will need regulation, which can be performed by a Zener diode. Using a voltage regulator and a circuit to control power delivery, the RSP will be able to be powered without interruption, whether using generated power or power from a battery.

3.4.3 Battery selection

The range of battery technologies on the market are many and varied, but for simplicity and reliability, the optimal choice is a Lithium-Ion cell. Their light weight, good energy density, stable voltage output, simple charging process makes them an easy technology to implement. However, they can fail dangerously under certain conditions (like overcharging and overheating), and in certain industrial operations (e.g. underground coal mines) they are banned

outright. For production use, more investigation into a Sealed Lead Acid (SLA) rechargeable battery would be undertaken. Although not the most compact (i.e. energy-dense) type of battery, these batteries are easy to charge using a "trickle" system, and are not known to explode or catch fire when malfunctioning. Since an MCU board with a built-in Lithium-Ion charger was found, this battery technology was chosen for the prototype.

3.4.4 Wireless module selection

The concept behind this sensor platform describes a form of wireless network topology known as a mesh. In this arrangement, each roller system can talk to each neighbouring roller, in order to pass data through and on to the mesh gateway, which connects all the rollers to the outside world. To demonstrate this concept, the chosen wireless module is the XBee PRO Series 2. This module comes with inbuilt meshing capability and requires minimal configuration to create a resilient, wireless mesh network. The module operates at 2.4 GHz, is listed as cable of transmitting over distances up to 120 m at a power output of 2 mW, and can scale to networks of thousands of nodes.

3.4.5 Temperature sensors

When choosing a temperature sensor, several considerations were made. Factors such as measurement range, accuracy, interface complexity and cost were considered in the decision to use a digital IC sensor, instead of an analogue thermistor, thermocouple or a Resistance Temperature Detector (RTD). This was primarily because of the advantages a digital interface has over analogue, the main advantage being that the Analog-to-Digital Convertor (ADC) of the MCU is not required. Since using the ADC is a relatively slow task, this frees the MCU to continue with the rest of the program. Using a digital interface also reduces MCU channel requirements, as the chosen IC, the DS18B20 from Maxim Integrated, works on a "one-wire" bus, allowing multiple sensors to be connected to the same digital channel. With less pins required, and no ADC needed, a simpler MCU could be used in future RSP versions. The selected IC can read temperatures from -55 °C to +125 °C with an accuracy of ± 0.5 °C.

3.4.6 Vibration sensors

The type of vibration sensor to be used is a Micro Electrical-Mechanical System (MEMS) accelerometer. This type has advantages over other vibration sensor technologies as it is very compact, can measure a wide range of frequencies, is reliable and can have high sensitivity (Goyal & Pabla 2015). Accelerometers with analogue or digital interfaces are available, and a digital interface was chosen as it is more resilient against signal interference and again, does not require the ADC of the MCU. The chosen accelerometer, the ADXL345 from Analog Devices, offers three axes of measurement, with the ability to measure ± 16 g along each axis, and can survive a shock of up to 10,000 g. The sensor returns data in a 16-bit, twos complement format. Some tuning and adjustment of mounting location and orientation will be required in order to detect the suitable range of frequencies for bearing health monitoring.

3.4.7 Drive motor

To simulate conveyor operation, the roller will be driven by an external motor. The motor used is from a vacuum cleaner, which is a "universal" motor. This type of motor is advantageous as it can be powered by AC or DC current, at a range of voltages. Rotational speed must be limited by an external load, to prevent motor "runaway", and the motor requires significant airflow for cooling. Using the speed control (also taken from a vacuum cleaner) the RPM of the roller will be able to be adjusted to simulate different operating speeds.

3.4.8 Indicator LEDs

The proof-of-concept system includes externally visible indicator LEDs which can be used to diagnose the health of the roller. These LEDs will be controlled by the RSP to show a red light if the roller is determined to be unhealthy, and a green light when operating normally.

3.5 Software Selection

The majority of modern, general purpose microcontrollers are designed to be programmed using the C/C++ programming language. Its widespread use makes learning and sharing work easy and with this in mind, an MCU designed to be programmed in C was selected (the Atmel ATmega328). Many examples of the Arduino project are based on this series of MCUs, and include a bootloader and an Integrated Development Environment (IDE) which simplifies the coding, debugging, compiling and loading of the program onto the MCU. With this in mind, an MCU with an Arduino bootloader was found, due to its simplicity and full compatibility with C programming.

In the long-term, a "pure" C program would be preferred, but this would be an easy conversion from the Arduino approach. Existing libraries which make interfacing with sensors easier are available, and these will be used so that data collection and evaluation of the sensors' suitability can begin as early as possible.

3.6 Software Design

The approach taken to programming an embedded system should be carefully considered to ensure the system can meet performance, responsiveness and reliability requirements. Fortunately, in the case of the RSP, program requirements are simple, and there are no user interaction considerations to make. Additionally, the system does not have high performance demands, and can sample and process data to provide regular, but not necessarily real-time, information.

While some embedded systems must be programmed to operate in a time-dependant way, and must avoid delays in program loops (e.g. in a system flying an aircraft, where program delays could result in control command delays and therefore a bumpy or jerky response), the RSP does not have this requirement. Therefore, each program action can be programmed as a separate function, which returns the data it gathers when it is ready. In effect, the central program passes control to each sub-routine, which passes control back when it is ready.

3.6.1 Program PDL

Initially, the operation of the program can be mapped out using Program Descriptive Language (PDL), which uses plain English to summarise the steps involved in running the program. The parts of the program are shown below in Figure 3.1.

```
BEGIN
    Initialise ports,serial
    ID = roller identification number
    DO FOREVER
         RPM = CALL getRPM
         FREQ = CALL calcFREQ
         TEMP1, TEMP2 =  readTEMP
         UPTIME = current time
         OUTPUT = CALL formatDATA
         TRANSMIT OUTPUT
         WAIT 10 seconds
    ENDDO
END

BEGIN/getRPM
    WHILE no RPM tick is detected
         wait for tick
    ENDWHILE
    previous time = current time
    WHILE no RPM tick is detected
         wait for tick
    ENDWHILE
    time elapsed = current time - previous time
    RPM = 3600 / time elapsed
    RETURN RPM
END/getRPM

BEGIN/calcFREQ
    FOR i = 1 to i = n
         READ X axis from sensor
         store value to data
         increment i
    ENDFOR
    FREQ = calculated vibration parameter from data
    RETURN FREQ
END/calcFREQ

BEGIN/readTEMP
    TEMP1 = READ TEMP SENSOR 1
    TEMP2 = READ TEMP SENSOR 2
    RETURN TEMP1 and TEMP2
END/readTEMP

BEGIN/formatDATA
    OUTPUT = START, ID, FREQ, TEMP1, TEMP2, RPM, UPTIME, END
    RETURN OUTPUT
END/formatDATA
```

Figure 3.1: PDL for monitoring platform

3.6.2 Output Format

In order for the data from the sensor platforms to be recorded in a central location, the format of the transmitted data must be defined and must remain consistent from platform to platform. Essentially, the data from the RSPs must be broadcast in packets that follow the conventions common to other communications protocols. Typically, a packet contains a start header, and end header, and the data is located in-between. Many protocols also make use of a checksum to determine if the data has been transmitted without error. However, transmission error handling is performed by the wireless radios, and for this project, this will suffice.

The transmitted packet will contain the values shown below by Table 3.2. As shown, the transmission begins with a START command, to inform the receiving program that data is to follow. Next, the ROLLER ID is included, to assign the following values to the correct entity in the central database. The measured data is included in the next fields, and the packet is terminated with an END command.

Table 3.2: Output data packet structure

START	ROLLER ID	VIBRATION DATA	TEMP1	TEMP2	RPM	UPTIME	END

3.6.3 Vibration Analysis Approach

Several techniques of analysing vibration data were discussed in Chapter 2, however, not all of these techniques are feasible on the hardware selected for the sensor platform. Generally, the time domain-based approaches are computationally simpler and have lower memory requirements. However, the literature review highlighted the significant advantages of signal analysis in the frequency domain. Therefore, multiple approaches will be attempted. In order from most desirable to least desirable, these are the FFT, kurtosis, or a simple RMS energy measurement of the vibration data.

3.7 Performance validation

Rather than be based entirely in theory, this project includes practical testing and measurement of the Roller Sensor Platform in a simulated environment. The tests will measure the RSP's capability in terms of the accuracy and resolution of its measurements, as well as the range of its wireless component. Due to the scope of this project, a full scale test with tens of wireless nodes is not possible, neither is an evaluation of the RSP's long-term reliability. However, by undertaking the following measurements, its initial suitability can be determined, as can the direction further testing should take.

3.7.1 Test setup

Using readily available construction materials, a mock-up of the environment within a conveyor belt roller will be constructed. The design will attempt to simulate the conditions and signals that the RSP would experience if installed in an operational conveyor system.

To view the transmitted data, a computer will be setup with a wireless transceiver configured as the Coordinator (gateway), and the serial port of the transceiver will be monitored. The input sensors will then be connected to the MCU to verify their operation, and confirm the program is correctly reading and transmitting data. Once the RSP is programmed and functioning, it will be installed on the roller mock-up. This platform consists of a tube rotating around an axle, driven externally by a motor. The RSP will be setup to monitor one bearing on the axle as well as the RPM of the roller shell. As the vibration sensor will be attached to the axle rather than in the plane of the bearing itself, it does not strictly adhere to international standards for vibration measurement. However, these standards are intended for machines with stationary outer shells and rotating axles, whereas in the case of the roller, this relative movement is reversed. Hence the standard is partially applicable, as in accordance with the standards, measurements are being taken from non-rotating parts. Experimental results will determine whether this arrangement is suitable.

Below, in Figure 3.2, is a diagram of the test platform design. The design consists of a steel roller shell (grey), mounted to a steel shaft with a bearing at either end. The components of the RSP are shown located within the roller, and the roller itself is driven by an external motor.

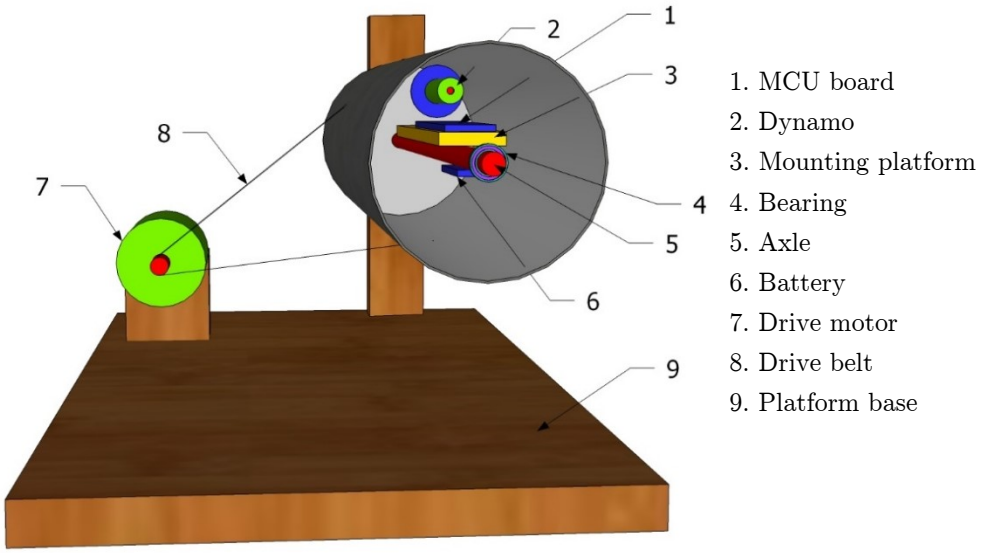

Figure 3.2: Test Platform Design (with one end removed)

3.7.2 RPM Accuracy Test

This test will use an external, commercially available laser tachometer to verify the data output by the RSP. By setting the drive motor to several different speeds, the readings can be compared to confirm that the RSP accurately reports RPM across a range of rotational speeds.

3.7.3 Heat Test

Due to high speed rotation, the inner bearing race is expected to exhibit a rise in temperature. For this test, the bearing may have to be attached to a shaft rotating at higher speeds than the roller could rotate at, in order to view a temperature response. By comparing the ambient temperature to the bearing temperature, the operation and placement of the sensors can be tested. Using an infrared thermometer, the ambient temperature will be compared to the sensor reading to confirm its accuracy.

3.7.4 Vibration test

Live data from the RSP will be collected while in operation, including data recorded from the on-board vibration sensors. To determine if these sensors are effectively tuned and installed, a range of faults will be simulated. These could include attaching a weight to one side of the roller to simulate an imbalance while in operation. Depending on bearing selection, it may be possible to intentionally damage the inside race of a bearing to determine the system's ability to detect such a defect.

Detailed analysis of vibration data is outside the scope of this project but it is important that the vibration sensors are fit for purpose and capable of supplying the raw data required for fault analysis. The main benchmarks of their performance are their ability to measure at a sufficiently high resolution, and a mounting location which accurately reflects the behaviour of the bearing.

3.7.5 Wireless range test

The final test will consist of estimating the wireless range of the RSP. During normal operation, each roller is unlikely to be further away than two meters from the roller next to it. However, the impact of the roller's construction material as well as the spacing of network receivers needs to be gauged. Firstly, as the RSP transceiver is located within the roller shell, the radio signals will have to travel through the roller (and most likely through the shell of another roller) before they will reach another transceiver. Whether the roller is constructed from steel or a composite material will significantly change wireless propagation range. Secondly, the roller network will require at least one bridge transceiver, connecting the rollers to the outside world. The maximum transmission distance from roller to receiving bridge will indicate the ratio of bridges to rollers required. Although, in theory, the network should easily scale to thousands of nodes, there is still an ideal limit to how many rollers should connect to one bridge and how many rollers the data from one roller will have to travel through in order to reach the bridge. By gauging the performance of the wireless modules, ideal design limitations can be determined to ensure the system performs as expected.

Chapter 4

Hardware Implementation

Using the specified parts and interfaces from the previous chapter, the sensor platform was assembled according to the requirements of each device. This section details the process of defining the electrical requirements of the circuitry so that the different components could be interfaced with one another, as well as the construction of the test platform.

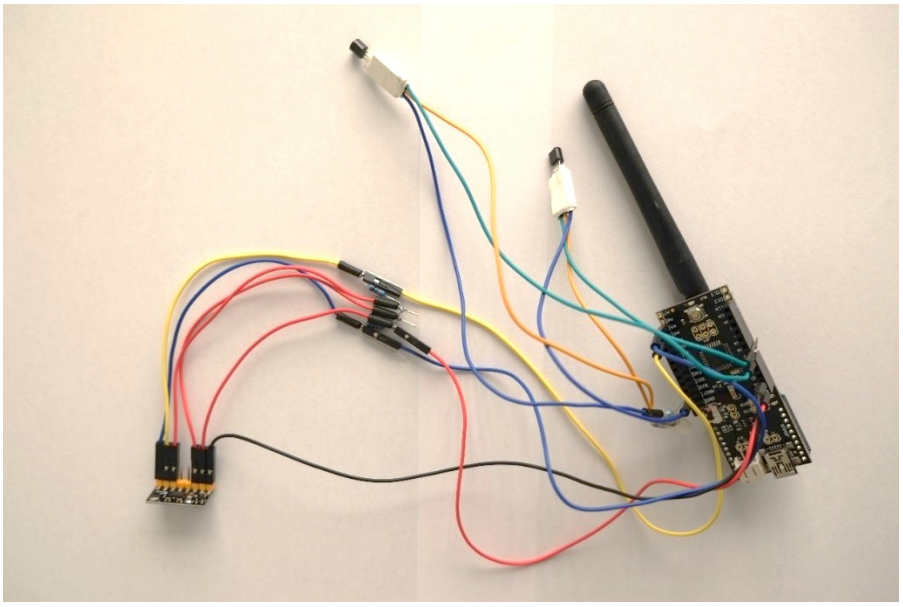

Figure 4.1: RSP sensors and main board, pre-installation

4.1 Input Devices

Several sensors were connected to the MCU to enable it to measure the environment around it. All sensors (apart from the Light Dependant Resistor) used a digital bi-directional interface to exchange data with the MCU, and in order for these data buses to function, some external components are required.

4.1.1 Accelerometer

The digital accelerometer (ADXL345) can use either Serial Peripheral Interface (SPI) or the I²C standard to communicate. For this project, I²C was chosen as it requires less wiring than SPI, and although its effective throughput is less than SPI it easily meets the needs of the intended application.

For proper operation, external pull-up resistors are required (NXP 2014). Their values are based on the capacitance of the electrical bus. For this design, the capacitance of the wires, pins and connections is considered low, as according to their respective datasheets, the capacitances for the ATmega328 and the ADXL345 are both 10 pF. Using NXP's *UM10204 I²C-bus specification and user manual* (2014) a safe R_P value of 10 kΩ was chosen. A diagram from the ADXL345 user manual is included below (Figure 4.2), which shows how to interface the sensor with the MCU.

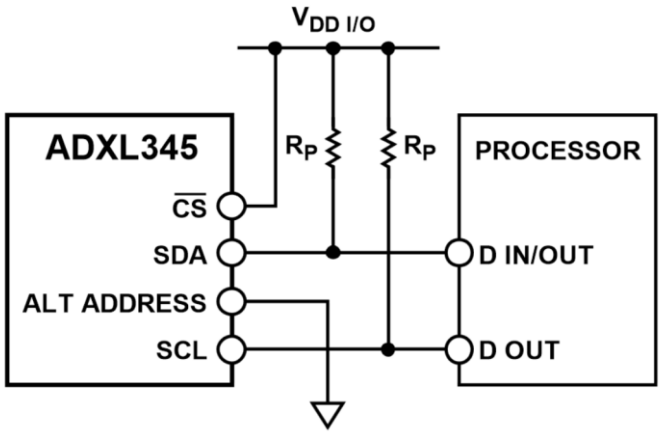

Figure 4.2: I²C Connection Diagram (Analog Devices 2015)

4.1.2 Temperature Sensors

The digital temperature sensors use Maxim's "1-Wire Bus System" to exchange data with the MCU. The advantage of this communications bus is that it only requires one wire connection to the sensor, along with a ground connection. The sensor is cable of powering itself off the bus connection, known as "parasitic power". For this project, the single-wire approach was not necessary, and to avoid the limitations it places on the operation of the sensor, a connection to the power rail was included.

To function correctly, a pull-up resistor is required between VDD (the supply rail) and the DQ (data pin) of the DS18B20. The value recommended by Maxim is a 4.7 kΩ resistor. The connection method is shown below, in Figure 4.3.

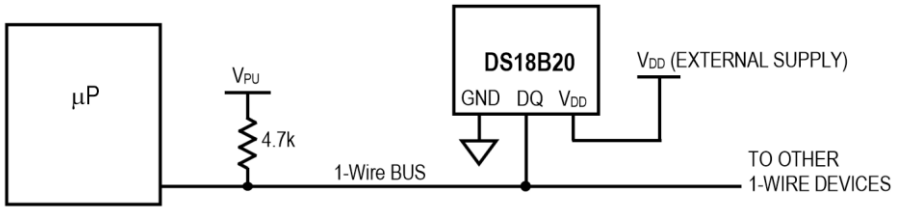

Figure 4.3: Powering the DS18B20 with an External Supply (Maxim 2008)

4.1.3 Light Dependant Resistor

This input device is being used as a simple beam-break detector. This means that light intensity information is not important, only full-on or full-off states. To use the Light Dependant Resistor (LDR) a simple voltage divider was setup using a pull-down resistor, i.e. the LDR connects to the +3.3 V rail, and the other side of the LDR connects to an analogue input on the MCU, as well as a 4.7 kΩ resistor which in turn connects to ground. Therefore in full-on (beam not broken) state the voltage will be about half 3.3 V, while in full-off state the detected voltage will be close to ground.

4.1.4 DC Generator

To harness power from the rotation of the conveyor belt roller, a small, DC motor was used as a generator by coupling its driveshaft to the inside wall of the spinning roller. However, DC motors operating as generators will output a voltage proportional to their RPM, which could exceed the voltage ratings of the digital circuitry being powered by the motor. In addition, generated power can be noisy, which could result in unstable behaviour if supplied directly to an MCU. Using the simple circuit shown below in Figure 4.4, the DC motor was connected to the Fio board which has a Low Drop Out (LDO) voltage regulator on-board as well as a battery charging IC.

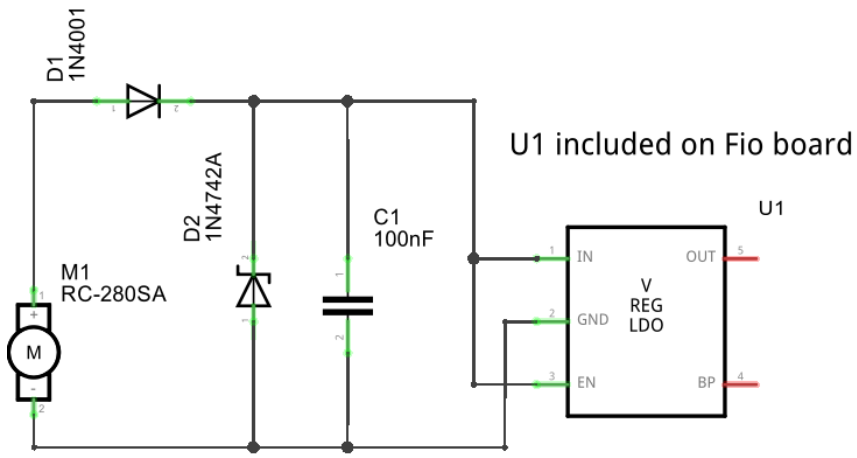

Figure 4.4: DC Generator circuit

The circuit above employs some basic components to protect the powered circuit from the DC motor, M1. First, a rectifier diode D1 is included, to prevent damage caused by the motor being driven in reverse, as this would reverse the polarity applied to U1, the voltage regulator. Next, a Zener diode D2 has been included, with a Zener voltage of 12 V. This prevents the motor from over-volting the regulator, which can only accept up to 15 V. Finally, a small capacitor has been included to reduce the electrical noise of the circuit, which is generated by the commutator brushes within the DC motor. By supplying this voltage range to the regulator, it will output a constant 3.3 V to the MCU, as well as supply the MAX1555 IC (not shown) which charges the battery.

4.2 Output Devices

The other side of the functions performed by the RSP consists of the devices which output data. This may be simply to measure RPM or to transmit streams of values wirelessly; the details of these devices are included below.

4.2.1 Wireless

In order to function as a remote sensor platform, the RSP must wirelessly transmit the data it collects to a receiver. This function is performed by the Digi XBee radio transceiver. By interfacing with the ATmega328 (the MCU) through its standard serial connection, the transceiver can act as a wireless serial link, which means that no special communication protocols are required in order to send and receive data. As the Fio board includes a header designed to fit the XBee radio, including connecting to its power and serial data pins, no special wiring was required either.

Using the Zigbee meshing protocol, the XBee radio communicates with neighbouring radios to exchange data in and out of the mesh network. This is done externally to the MCU, which has no direct control over this function. The Zigbee protocol defines three different possible roles for any single radio; each can either be a Coordinator, Router or End Device. Usually, there is only one Coordinator per network, as it functions as the bridge to the outside world (Digi International 2014). In Figure 4.5, below, an external data collection terminal is shown connecting to the mesh network Coordinator.

Other radios in the network function as Routers, which can not only send and receive their own data, but can forward the data from their neighbours on through the network, in order to reach the Coordinator. Finally, some radios can function as End Devices, which only access the network in order to transmit their own data; they do not route the data of other radios. In this project, all RSPs function as Routers in order to pass data up and down the length of the conveyor belt. At one end, or in the middle of the conveyor belt, a Coordinator radio would be situated, to provide the connection to remotely monitor the whole system.

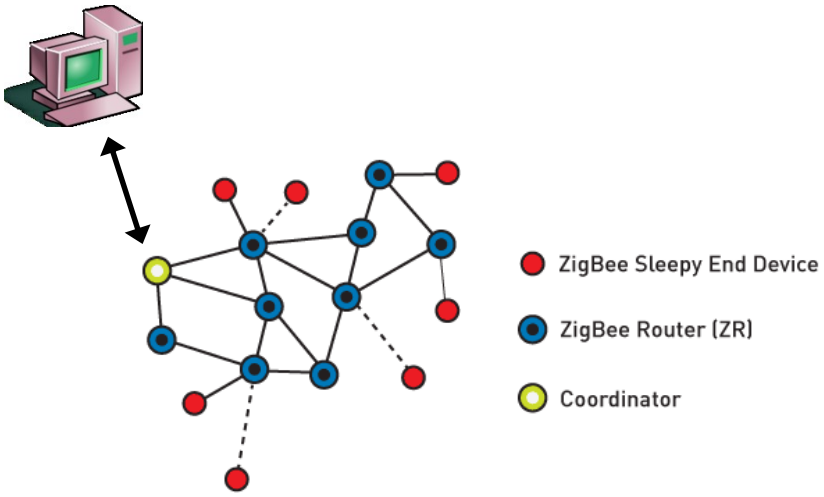

Figure 4.5: Zigbee Mesh Topology

To establish a wireless mesh network, configuration of the wireless radio hardware is minimal. The main requirements are that one radio functions as a Coordinator, and the other functions as a Router. This requires different firmware on each radio, which can be changed using Digi's XCTU software (shown below in Figure 4.6). The next requirement is that both radios share the same Personal Area Network IDentification (PAN ID), which was set to 11 as shown below. Once these settings are loaded, the radios establish a link and serial data can be transmitted (at the default baud rate of 9600 bps).

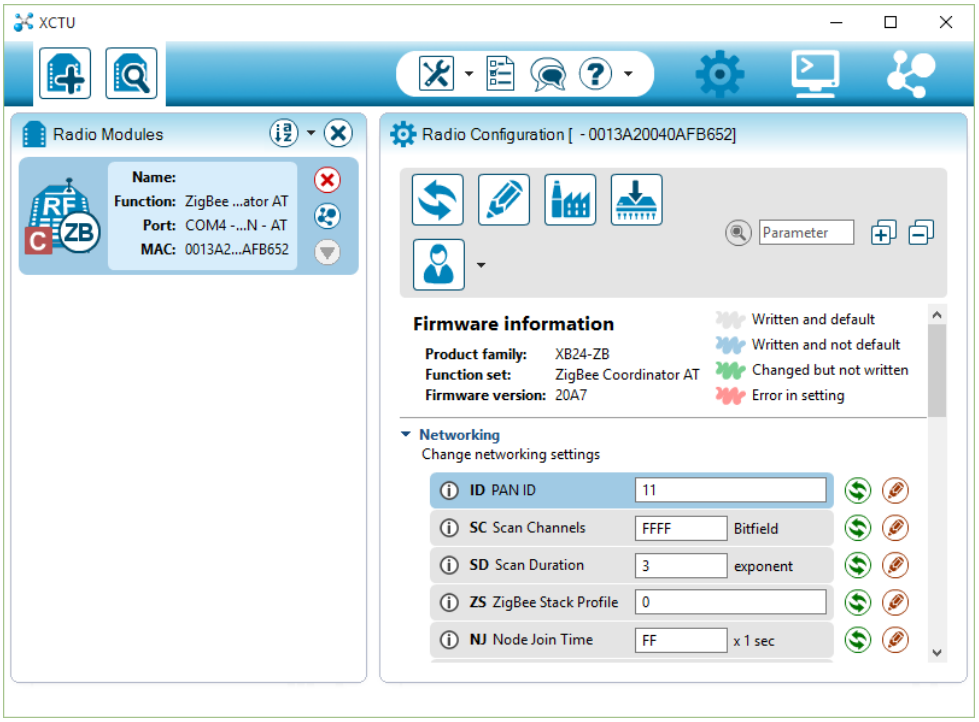

Figure 4.6: XCTU Software Interface (Coordinator radio shown)

4.2.2 Infra-Red LED

Although technically not outputting any data, the IR LED was connected to a digital output pin on the MCU so that the LED could be disabled when not in use, and to correctly size the current limiting resistor R6 the calculations below were necessary. By setting the output pin to a HIGH state, there will be very little voltage drop across the LED, resulting in no light emitted (reducing energy use when necessary).

To calculate the minimum value of R6, the MCU pin HIGH state was considered, as in this state the LED must be guaranteed to be off. From the datasheet, the MCU could output an output high (V_{OH}) of as low as 2.3 V, when using a supply voltage (V_S) of 3.3 V. To reduce power use, the current through the LED (I_{LED}) is limited to half of the MCU pin current to 20 mA. Therefore, to find the minimum value of R6, the calculation is:

$$\frac{(V_S - V_{OH})}{I_{LED}} = \frac{(3.3 - 2.3)}{20x10^{-3}} = 50\ \Omega$$

Next, the maximum value of R6 was found. This is important when the LED should be switched on, and in this state the MCU pin will be LOW. To fully illuminate, the IR LED requires a voltage of (V_{LED}) 1.5 V, and from the MCU datasheet, the output low (V_{OH}) could be as high as 0.6 V. The maximum value for R6 can be found by:

$$\frac{(V_S - V_{LED} - V_{OL})}{I_{LED}} = \frac{(3.3 - 1.5 - 0.6)}{20x10^{-3}} = 60\ \Omega$$

Therefore, a standard value from the E24 resistor series of 56 Ω was chosen.

4.2.3 Electrical Schematic

A complete circuit diagram has been included in Appendix B, Figure B.1.

4.3 Test Platform

The roller test platform was intended as a close approximation of a real-world conveyor belt roller to be used to evaluate the critical functions of the RSP. The original design of the test platform is shown in Figure 3.2. The first part of the test platform to be constructed was the roller itself, which consisted of the following components:

- 300 mm x 100 mm (OD) x 5 mm thick galvanised steel tube
- 10 mm galvanised steel rod (as the axle)
- 95 mm x 20 mm circular pine sections for the roller end caps
- Two 10 mm flange roller bearings

The assembled roller is shown below, in Figure 4.7. Note that the end cap has been reversed, in order to show the bearing. Under normal testing the sensors were located near the inner race of the bearing which is usually on the inside of the roller along with the rest of the RSP.

Figure 4.7: Mock-up of the test roller

Unfortunately, early testing of the roller showed that above a low rotation speed the roller was very unbalanced and would not only be likely to break any mount it was attached to, but the excessive vibration may affect the vibration sensor readings. Therefore the decision was made to test sensors individually, rather than as a whole system installed within the roller shell.

4.3.1 Temperature Sensor Mounting

Guided by the literature, the temperature sensors were installed as close to the inner race of the bearing as possible. This should give the quickest response to changes in the bearing temperature. As the sensors cannot be seen in the actual mounting (due to heat shrink shrouding), a sensor location has been drawn on the image of the bearings used (Figure 4.8, below). The second temperature sensor is in free air within the roller shell, so that it can be used as a reference to compare the increase in temperature at the bearing.

Figure 4.8: Mounting location of temperature sensor

4.3.2 Vibration Sensor Mounting

As previously mentioned in Section 3.7.1, the vibration sensor (accelerometer) was mounted to the roller axle. Using a rigid coupling, the sensor was mounted vertically so that the radial direction (outward from the axle) would align with the x-axis of the sensor. Figure 4.9, below, shows the sensor arrangement.

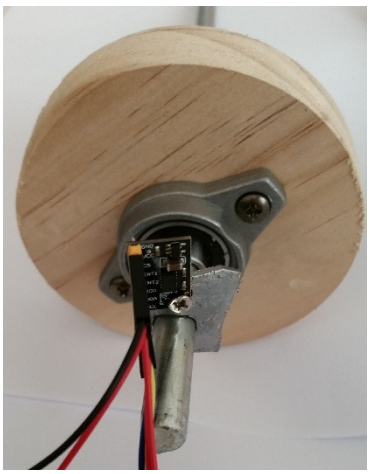

Figure 4.9: Vibration sensor mounting

Chapter 5

Software Implementation

The programming of the Roller Sensor Platform was a major component of this project. Not only does the software code poll sensors for data and then format and transmit this data, but it pre-processes some of it to reduce the load on the communications network.

To test the limits of the RSP, secondary programs were written to perform specific functions outside the normal operation of the system. These test programs targeted certain sub-systems of the Platform to evaluate their performance and find the limits of their operation. This helped identify weaknesses in the design and areas where re-configuration was needed, or future improvements could be made.

The details of the test programs are included in this chapter, as well as an overview of the main program.

5.1 Test Programs

The details of the test programs used to evaluate the real-world performance of the RSP are included in this section.

5.1.1 Wireless Range Test Program

The wireless range of the RSP is an important measure of the network performance which can be expected from the system. If reliable wireless communication can be demonstrated in similar conditions to those found in a real-world conveyor belt, the method of wireless communication used for this project can be considered a viable option for future use. However, if the effect of the steel shell of the roller results in intermittent data transmissions, then the design will require modification to account for this.

The test program listing is included in Appendix C, Figure C.2. This test represents one approach to evaluating the wireless capability of the XBee radios. As explained the listing description, the program transmits a steadily increasing number, and the results are recorded on the other end of the wireless link. A linear increase in value indicates a good connection, while sudden jumps in the transmitted value indicates that some transmissions were lost.

For this program to operate as expected, firmware settings of the XBee radios had to be modified to prevent the radio buffering data until the link was restored. Although this feature may be desirable in normal operation, it was disabled for this test so that the true nature of the wireless reliability could be determined.

5.1.2 Accelerometer Read Rate

As the RSP records vibration data from the vibration sensors, it is important that these readings occur at a regular, known rate. Without a reliable timescale, performing analyses on the data to extract frequency features would not be possible.

The rate at which readings can be taken must be more than twice as fast as the highest frequency component of interest. This is the guideline set out by the

Nyquist-Shannon sampling theorem, but as discussed in Section 2.3.5, a much higher sample rate is desirable. To find the limits of the current implementation of the ADXL345 accelerometers and the I2C bus, this program performs five hundred sequential reads of the accelerometer, and records the microseconds elapsed between starting each read and writing it to the storage array. The time elapsed between each sample is then stored next to the sample in the array, and the whole process repeats every two seconds. Each array is transmitted through the serial link in order to view the results. The full code listing of this program is included in Appendix C, Figure C.3 and Figure C.4.

5.2 Final Program

The program which runs the sensor platform is responsible for reading in data from the attached sensors, processing this data, and then transmitting it in the pre-defined data format through the serial port. As the serial port is connected to the XBee radio, this data is transmitted to the Coordinator radio of the wireless network. For the full code listing, refer to Appendix C, Figure C.5 to Figure C.7.

In order to function, the final program only makes use of three external libraries. These are "Wire.h", "OneWire.h" and "DallasTemperature.h". Respectively, these are used to control the I²C bus, Maxim's OneWire protocol bus, and the last library handles interfacing with the DS18B20 temperature sensors. Once each of the peripheral devices have been initialised, the program initiates a read from each sensor in turn. Before the vibration data is formatted and transmitted it is processed on the RSP itself.

Instead of transmitting the raw vibration data, the RSP processes it so that the nature of the vibration can be represented by a single number. This number is the kurtosis excess of the signal. Although using an FFT on the collected dataset was investigated, the memory requirements were more than the chosen MCU could handle. Kurtosis excess is calculated identically to kurtosis, except that it is normalised by subtracting three from the final value. The output then represents the difference from the normal distribution (which has a kurtosis value of three).

The kurtosis excess of the dataset is calculated by the function "VIB˙calc". This function is based on the work of Vogelaar (1995), who demonstrated a straightforward approach to calculating the kurtosis value iteratively. The function below (Figure 5.1) performs the calculation given by the formula for excess population kurtosis:

$$K = \frac{\sum_{i=0}^{n} \left(X_i - X_{avg}\right)^4}{\left(\sum_{i=0}^{n} \left(X_i - X_{avg}\right)^2\right)^2} \qquad (5.1)$$

Using the sampled dataset, which is an array of 500 values (each being 2 bytes in length), the first step is to calculate the mean of the dataset. Next, to find the variance, the square of the difference from the mean is divided by the sample size. The square root of this value is used to find the standard deviation. As the *kurtosis* variable contains the fourth moment about the mean, it is divided by the second moment squared to find the kurtosis, and finally, a value of three is subtracted to find the kurtosis excess.

```
// Returns Kurtosis excess of the data set
float VIB_calc() {
  // Declare variables
  float sum = 0.0;
  float sqr_diff = 0.0;
  float sdev = 0.0;
  float kurtosis = 0.0;
  // Find the mean of the data set
  for (int i = 0; i < SAMP_SIZE; i++){
    sum += roller_vib[i];
  }
  float mean = sum / (float) SAMP_SIZE;
  // Find the mean of the data set
  for (int i = 0; i < SAMP_SIZE; i++){
    sqr_diff = (roller_vib[i] - mean) * (roller_vib[i] - mean);
    sdev += sqr_diff;
    kurtosis += sqr_diff * sqr_diff;
  }
  // Use variance to find standard deviation
  sdev = sqrt(sdev / (float) (SAMP_SIZE - 1) );

  // Prevent a division by zero
  if (sdev == 0.0){
    kurtosis = 0.1;
  }
  else {
    // Calculate kurtosis value
    kurtosis = (kurtosis/(sdev*sdev*sdev*sdev* (double) SAMP_SIZE))-3.0;
  }
return kurtosis;
```

Figure 5.1: Code Listing of the Kurtosis function

For reference, the main program loop is shown below in Figure 5.2. The main loop is relatively simple, as all the sensor interfacing is completed by separate functions. The main activity of the loop below is the formatting of the serial output string. As specified in Section 3.6.2, the output string consists of a "START" command, followed by the sensor platform's unique identification, followed by the data and terminated with an "END" command. For the server to interpret the output string, it would look for the start command followed by space delimited values terminated with the end signal.

```
// Main program loop
void loop() {

  int RPM = 2400; // Sample RPM value
  int bearing_temp = read_temp(BEARING_SENS_NO);
  int internal_temp = read_temp(INTERNAL_SENS_NO);
  VIBdata();          // Read vibration sensors
  Serial.print("START ");
  Serial.print(ROLLER_ID);
  Serial.print(" ");
  Serial.print(VIB_calc());
  Serial.print(" ");
  Serial.print(bearing_temp);
  Serial.print(" ");
  Serial.print(internal_temp);
  Serial.print(" ");
  Serial.print(RPM);
  Serial.print(" ");
  Serial.print(millis());
  Serial.print(" ");
  Serial.println("END");
  // Output delay to conserve bandwidth
  delay(500);
}
```

Figure 5.2: Main program loop

Using the Arduino Integrated Development Environment (IDE), each program could be loaded into the MCU as necessary to undertake system testing. The results of this testing are detailed in the next section.

Chapter 6

Results

This chapter discusses the data gathered by the sensor platform, how this data was obtained and whether it met the requirements of the project. Using the pre-defined testing procedures and testing software, the performance of the hardware and software was evaluated.

Output from the RSP was recorded primarily using the Arduino IDE's built-in Serial Monitor function, which emulates a serial link with the MCU through the computer's USB port. On the other end of this link is an external USB to serial converter, as the Fio board has no USB capability, only the MCU's built-in Universal Synchronous Asynchronous Receiver Transmitter (USART) port. For wireless radio testing, this converter was removed and the XBee radio card installed. Likewise, a serial to USART adapter was use to connect the XBee Coordinator radio (Figure 6.1) directly to the host computer.

Test results are included in the next few sections.

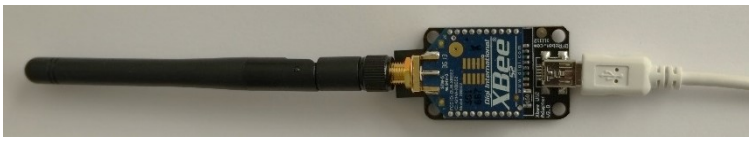

Figure 6.1: XBee Coordinator Radio

6.1 Temperature Response

In order to test the temperature sensors individually, their original mounting location had to be adjusted. Due to the low rotation speed of the roller assembly, a temperature response could not be observed. To see a temperature increase, the roller shaft was mounted to an electric power drill, so that the shaft could be spun at high speed. The drawback to this approach was that the sensor had to be relocated to the outer (stationary) race of the bearing, as shown below in Figure 6.2.

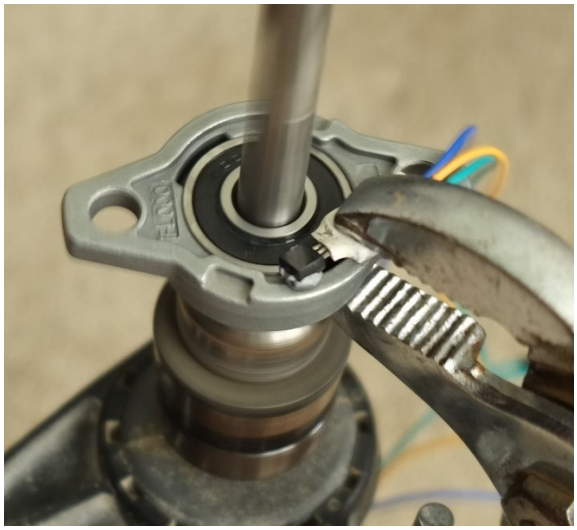

Figure 6.2: Actual location of temperature sensor

Initially, the bearing temperature response remained static. However, after 5 seconds of being run on a shaft at about 1400 RPM, the measured temperature started to rise. The results transmitted by the bearing temperature sensor have been graphed below, in Figure 6.3. While the ambient temperature remains almost constant, the bearing temperature rises and plateaus at 28 degrees Celsius. Using an IR thermometer, a matching reading was taken of the bearing outer race housing.

This test showed that the temperature sensors were performing as expected, and that the bearing sensor would respond to changes in bearing state. Under real-world conditions, the temperature response of the bearing is likely to be more

dramatic, as it would be bearing load, more friction would exist between the rolling elements and the races.

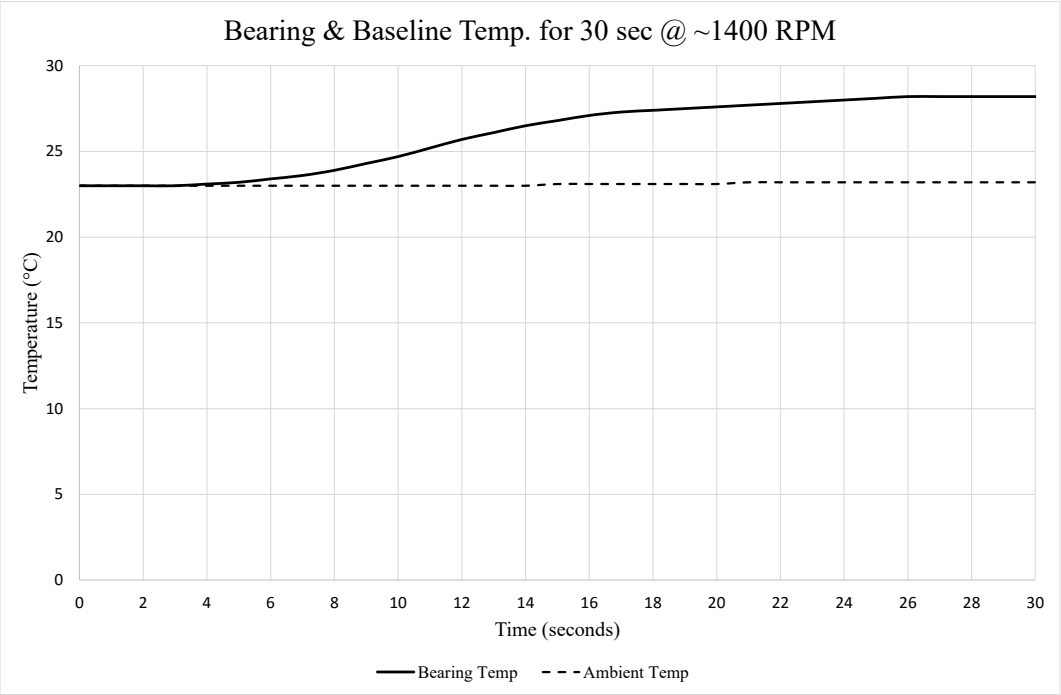

Figure 6.3: Temperature response of the bearing sensor

6.2 Accelerometer Read Rate

This test was performed to find the frequency sampling limitations of the current hardware setup. Using the program explained in Section 5.1.2, the accelerometer was polled 10 times, and the time taken for each reading was computed by the program and listed to next to each read value. The output is shown in Figure 6.4 (below).

```
116 512 122 504 124 512 126 504 128 504
-100 512 -98 504 -98 512 -100 520 -104 504
114 512 100 504 130 512 124 504 118 504
110 512 114 504 112 512 130 504 126 512
-210 504 -212 520 -224 512 -226 504 -220 504
-52 504 -52 504 -56 520 -56 512 -54 504
-4 504 -6 512 -8 504 0 512 -10 504
-48 520 -42 504 -48 504 -50 512 -52 520
-54 504 -52 512 -56 504 -50 512 -54 512
```

Figure 6.4: Read rate program output

As shown in the program output, the time between read rates is consistently about 500 microseconds. Even when the accelerometer is shaken or rotated (shown by larger, smaller and negative values) the read times remain almost the same.

With this fast read rate, it would appear the MCU can perform a read from the ADXL345 at a rate of 2 kHz. Unfortunately, this is not the case. According to the accelerometer datasheet, using the default I²C bus speed of 100 kHz results in an effective Output Data (OD) rate of 200 Hz, while using a faster bus speed of 400 kHz an OD of 800 Hz can be achieved (Analog Devices 2015). To achieve the faster data rate, the Arduino I²C interface speed was set to 400 kHz, and the ADXL345 was set to sample at 800 Hz accordingly.

Although the current method of polling the accelerometer can perform a read every 500 microseconds, it is actually reading the same data at times, as the ADXL345 has not yet stored a new measurement.

Despite the convenience of the I²C bus, the more mature SPI interface can perform significantly faster (Analog Devices 2015). This bus can run at clock speeds of 5 MHz, which enables an output data rate of 3200 Hz. For a more capable hardware setup, the SPI bus could be used to capture much higher resolution vibration data.

6.3 Vibration Detection

To gauge the implemented vibration detection technique's ability to monitor vibration, the installed accelerometer was measured in three states: at rest, under normal motor operation, and under heavy vibration. Using the kurtosis values being output by the RSP, a graph was generated of each case. A sample of the serial output is shown below, in Figure 6.5.

```
START 02A3C9 -0.09 29 29 2400 3805 END
START 02A3C9 -0.12 29 29 2400 4366 END
START 02A3C9 -0.02 29 29 2400 4927 END
START 02A3C9 -0.31 29 29 2400 5488 END
START 02A3C9 0.12 29 29 2400 6049 END
START 02A3C9 -0.13 29 29 2400 6610 END
START 02A3C9 0.00 29 29 2400 7172 END
```

Figure 6.5: Monitoring platform serial output

Due to the limitations of the testing equipment available, vibration verification tests were relatively basic, and the aim of the test was to verify that the kurtosis value increased with an increase in vibration. By spinning the roller end cap (while it was attached to the roller axle) with a smooth wheel, and next with a pitted wheel, radial vibration was simulated and the results collected.

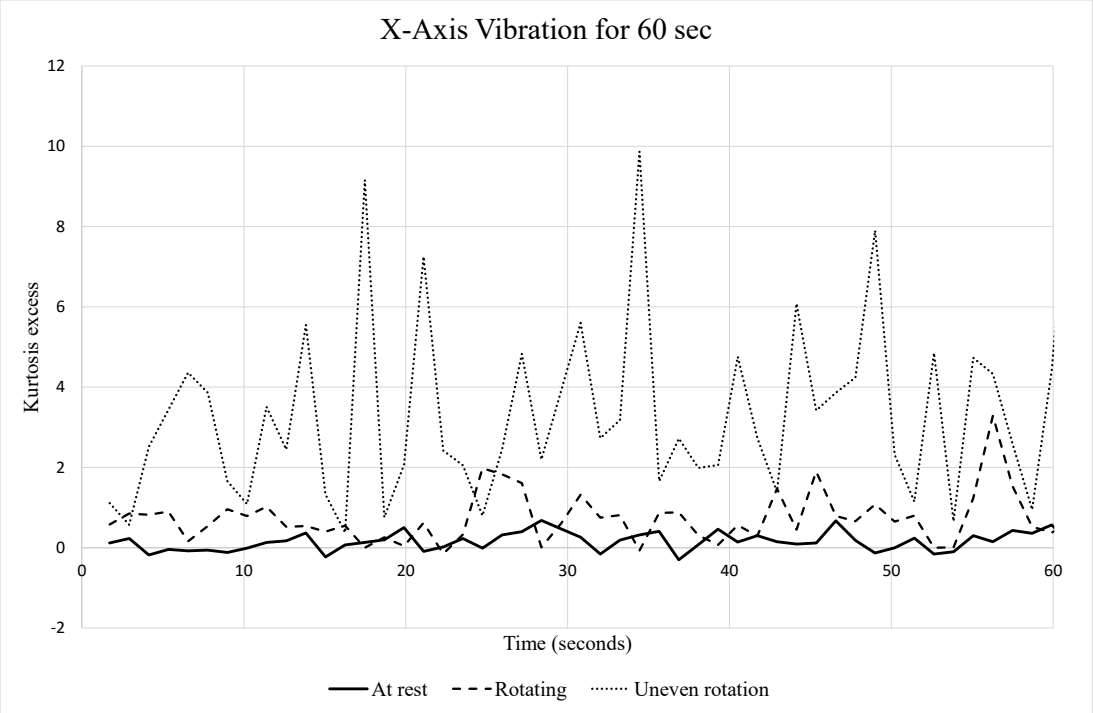

Figure 6.6: Vibration Measurement Results

The results of the vibration tests are shown in Figure 6.6 (above). With no movement of the accelerometer, a certain amount of noise is present. This noise has an average value of K = 0.16 (over one minute). When the bearing was rotating around the axle under normal operating conditions, the measured kurtosis value increased as shown by the dashed line. The average value increased to K = 0.93. When exposed to irregular rotation, and frequent impacts and bumps, the average increased to K = 3.51 and as shown by the dotted line, some peaks are much higher than this.

While the kurtosis of the signal seems to reflect the presence of vibration in the bearing, the memory requirements of the program limit each sample window to a short period. Each sample consists of 500 data points, sampled at 800 Hz, which equates to 625 milliseconds worth of data. For a full roller revolution to be captured, the roller would have to be rotating at 5760 RPM, which is highly unlikely. Without a longer sample time, some parts of the rotational cycle will

not be recorded. However, if the final values were averaged over time, the short window may eventually cover all parts of the roller cycle as long as it remains out of phase with the roller rotational frequency.

6.4 RPM Accuracy

The intent of this test was to validate the RSP's speed sensing capability. Originally, a hand-held laser tachometer was planned to be used to validate the results. Unfortunately, this piece of test equipment was unable to be obtained, and so the RPM component of the program was not completed. A DC motor can be used to sense RPM (as any DC motor will provide a voltage out proportional to its RPM) and this was investigated, however, in the time permitted this testing solution could not be explored further.

6.5 Wireless Range Test

The wireless performance of the XBee radios was tested to determine the reliable wireless range that could be expected. Using the program code described in Section 5.1.1, a pair of radios was tested at several distances apart. The test program is a simple function which transmits a steadily increasing value (the uptime of the MCU) and any sudden jumps in the value are a sign that transmissions have been missed. As there is no known way to obtain typical wireless performance benchmarks such as signal strength, transmission errors and retries or noise floor information from the XBee radios, this approach was a simple way to test the radios in practice.

To simulate communication from one conveyor belt roller to another, one radio was placed in the sealed steel tube. The tube was then located at varying distances from the receiver radio, and the value of the received serial data was recorded.

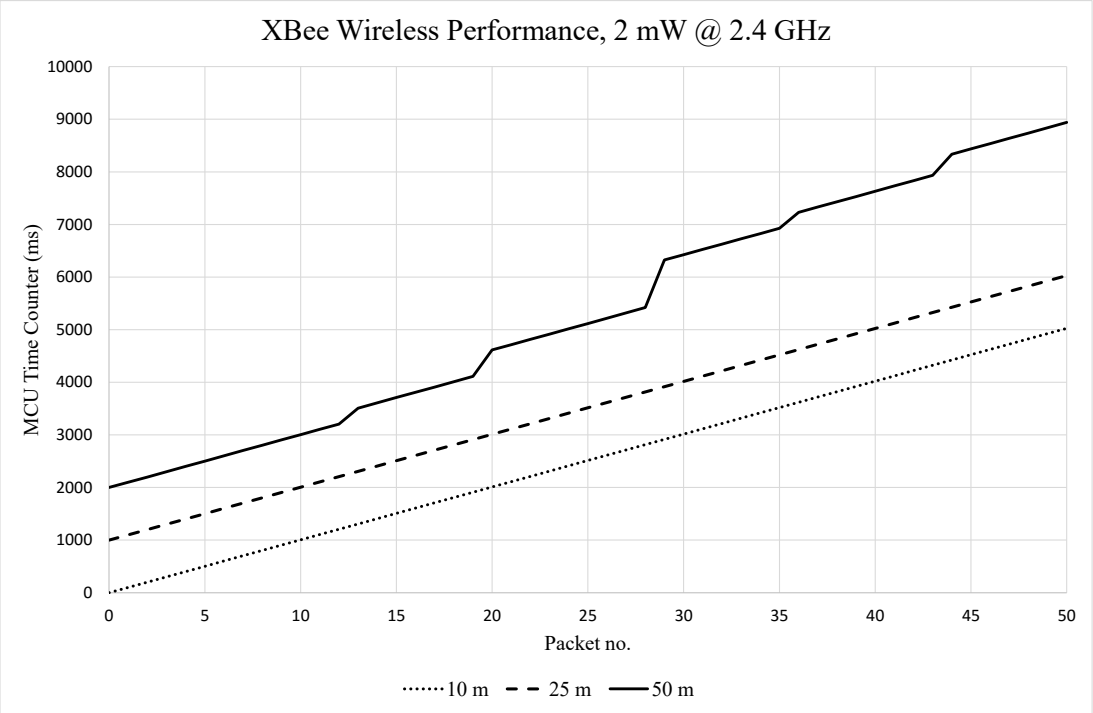

Figure 6.7: Wireless test results

The collected data is shown above, in Figure 6.7. Each dataset was plotted from a different start time, so that all three could be shown on one figure. Remarkably, it took a distance of 50 m between both radios before packet loss was evident. This is shown by the solid line, which does not increase linearly like the other two measurements but instead has steps along its length where the in-between packets have been lost.

Significantly worse performance was expected from the XBee radios. Not only do they operate at 2.4 GHz, which is a very crowded frequency band, full of interfering devices, but each radio is only rated for 2 mW. This low power was not expected to reach to the distances shown above, especially when broadcasting from within a solid steel roller.

Chapter 7

Conclusion

In this final section, the project outcomes are evaluated against the project objectives, and future work to improve the effectiveness of the solution is discussed.

7.1 Project Summary

To achieve the project objectives, the development process consisted of the research, design, build and test stages. In the first stage, a need for better conveyor belt monitoring was found, not only for financial reasons but for the safety of personnel. The current market solutions were found lacking and only one (very recent) product offered remote monitoring.

With literature to support an understanding of the failure modes of rollers and how to monitor for signs of failure, work began on a design capable of remotely, autonomously and cheaply collecting conveyor roller metrics and transmitting them to a central server. The system is based on the ZigBee mesh network protocol which enables each conveyor roller to pass along the data of its neighbouring rollers.

To ensure the Roller Sensor Platform (RSP) could operate in a maintenance-free way and in a sealed environment, a power harvesting system was designed. This system uses the rotational energy of the roller to generate electricity to power the RSP and its battery. This system was not tested as the RPM required to generate a significant amount of power was beyond the speed that the test bed roller could handle.

Results of tests performed on different aspects of the system were completed. Some results, like the effective read rate of the vibration sensor, highlighted significant room for improvement. However, the system detection of vibration was successful, and with more in-depth testing the full capability of the kurtosis method could be explored. An outstanding result in terms of the wireless signal range was achieved, as this was shown to reach much further than expected. With a transmission distance of just under 50 m before signal loss occurred (even when operating from within a steel tube), the XBee radios showed great potential to handle operation in a noisy, more interference prone environment.

In summary, this project only represents the beginning of the investigation into the potential for low cost, readily-available sensors and embedded systems to have a large impact in improving the safety and efficiency of industrial operations.

7.2 Future Work

To develop the RSP further into a more robust, capable and reliable system, there are many opportunities for future work. Given more time on the project, the very next step would be to optimise the data processing algorithm used to analyse vibration data. In its current form, the data processing demands more memory than the MCU can provide as it stores the entire data set before performing any analysis. Instead, a better approach is to manipulate the calculations so that they can be done incrementally, without needing the complete dataset. With the experience gained from optimising the current time-domain implementation, a frequency-domain approach could be investigated to determine if the hardware is capable of performing such an analysis.

Better computational efficiency would allow higher resolution sampling, which would push the limits of the I^2C bus used to interface with the accelerometer. A future development would see this replaced with the SPI bus which is fully compatible with the hardware and allows a much higher output data rate.

For a longer-term study, the reliability of MEMS based sensors would be worth further investigation. Plenty of literature discusses their suitability and all the applications this sensor type could have in the condition monitoring field, yet their long-term performance is yet to be evaluated.

Several improvements could be made to the communications protocol used between the RSPs and the central server. This project simply developed a 1-way data transfer. Ideally, the RSPs would be aware of whether their data had been received or not, and this would require implementing a type of ACKnowledge (ACK) signal from the server when it received transmitted data. Additionally, plenty of work could be completed on a software interface to monitor, control and store the data from the RSPs in a central location.

Finally, the concept of "crowd-sourcing" is worth investigating. Using a multitude of RSPs in the field, baselines for temperature, vibration signal and RPM could be established, and using this approach, error states diagnosed. This technique may be much easier than computing FFTs and other traditional signal analysis methods.

References

AIMEX 2015, 'AIMEX Preview: Conveyor monitoring', *Australian Mining*, 10/08/15.

Analog Devices 2015, *ADXL345*, Analog Devices, Norwood, MA USA, <http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>.

BEUMER 2012, *Troughed belt conveyors*, <http://www.mining-technology.com/contractors/materials/beumer/beumer2.html>.

Digi International 2014, *XBee and ZigBee basic concepts*, Digi International Inc., viewed 06/04/15, <http://ftp1.digi.com/support/documentation/html/90001399/90001399_A/Files/XBee-concepts.html>.

El-Thalji, I & Jantunen, E 2015, 'A summary of fault modelling and predictive health monitoring of rolling element bearings', *Mechanical Systems and Signal Processing*, vol. 60–61, no. 0, pp. 252-72, <http://www.sciencedirect.com/science/article/pii/S0888327015000813>.

Fenner Dunlop 2015, *Moranbah North Coal Mine*, Fenner Dunlop, viewed 18/09/15, <http://www.fennerdunlop.com.au/fenner-dunlop-conveyor-services-case-studies/moranbah-north-coal-mine-14>.

FLIR 2015, *Thermal Imaging for Manufacturing Industries*, viewed 10/9/15, <http://www.flir.com.au/instruments/electrical/display/?id=49526>.

Giraud, L, Massé, S & Schreiber, L 2004, 'Belt Conveyor Safety', *Professional Safety*, vol. 49, no. 11, pp. 20-6, <http://ezproxy.usq.edu.au/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=heh&AN=14899082&site=ehost-live>.

Goyal, D & Pabla, BS 2015, 'The Vibration Monitoring Methods and Signal Processing Techniques for Structural Health Monitoring: A Review', *Archives of Computational Methods in Engineering*, pp. 1-10, <http://dx.doi.org/10.1007/s11831-015-9145-0>.

Huang, N, Wu, Z & Long, S 2008, 'Hilbert-Huang transform', *Scholarpedia*, vol. 3, p. 2544, <http://www.scholarpedia.org/article/Hilbert-Huang_transform>.

Huang, Q, Tang, B, Deng, L & Wang, J 2015, 'A divide-and-compress lossless compression scheme for bearing vibration signals in wireless sensor networks', *Measurement*, vol. 67, no. 0, pp. 51-60, <http://www.sciencedirect.com/science/article/pii/S0263224115000755>.

Hughes, R 2004, *Conveyor Failure Roller Bearing Failures*, Reliability Center, Inc., Hopewell, Vancouver,<http://www.reliability.com/industry/pdf/proactreport_roller_bearing_failures.pdf>.

IEEE 1985, 'Report of Large Motor Reliability Survey of Industrial and Commercial Installations, Part I', *Industry Applications, IEEE Transactions on*, vol. IA-21, no. 4, pp. 853-64.

ISO 1995, *10816-1 Mechanical Vibration - Evaluation of machine vibration by measurements on non-rotating parts*, General guidelines, 10816-1, ISO.

Janse van Rensburg, B 2013, 'The development of a Light Weight Composite Conveyor Belt Idler Roller', USQ, Toowoomba.

Maxim 2008, 'DS18B20 Programmable Resolution 1-Wire Digital Thermometer ', no. 042208, <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.

McGuire, PM 2010, *Conveyors application, selection, and integration*, Industrial innovation series., CRC Press, Boca Raton FL, <http://ezproxy.usq.edu.au/login?url=http://www.USQ.eblib.com.au/EBLWeb/patron?target=patron&extendedid=P_566005_0>.

Norton, MP & Karczub, DG 2003, *Fundamentals of Noise and Vibration Analysis for Engineers*, Cambridge University Press.

NXP 2014, 'UM10204 I2C-bus specification and user manual', <http://www.nxp.com/documents/user_manual/UM10204.pdf>.

Peng, ZK, Tse, PW & Chu, FL 2005, 'An improved Hilbert–Huang transform and its application in vibration signal analysis', *Journal of Sound and Vibration*, vol. 286, no. 1–2, pp. 187-205, <http://www.sciencedirect.com/science/article/pii/S0022460X04007928>.

Reicks, AV 2006, *Belt Conveyor Idler Roll Behaviors*, Overland Conveyor Co., Lakewood, Colorado, <http://overlandconveyor.com/pdf/belt-idler-roll-behavior.pdf>.

Sawalhi, N & Randall, RB 2011, 'Vibration response of spalled rolling element bearings: Observations, simulations and signal processing techniques to track the spall size', *Mechanical Systems and Signal Processing*, vol. 25, no. 3, pp. 846-70, <http://www.sciencedirect.com/science/article/pii/S0888327010003146>.

Schools, T 2015, 'Condition Monitoring of Critical Mining Conveyors', *Engineering & Mining Journal (00958948)*, vol. 216, no. 3, pp. 50-4, <http://ezproxy.usq.edu.au/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=iih&AN=101635398&site=ehost-live>.

Scott, S, Kovacs, A, Gupta, L, Katz, J, Sadeghi, F & Peroulis, D 2011, 'Wireless temperature microsensors integrated on bearings for health monitoring applications', in *Micro Electro Mechanical Systems (MEMS), 2011 IEEE 24th International Conference on*: *proceedings of theMicro Electro Mechanical Systems (MEMS), 2011 IEEE 24th International Conference on* pp. 660-3.

SKF 2010, *SKF bearing maintenance handbook*, SKF Group, viewed 29/05/15, <http://skf.elanders.cn/media/custom/upload/File-1352446538.pdf>.

SKF 2015, *Bearing life and load ratings*, viewed 02/06/15, <http://www.skf.com/group/products/bearings-units-housings/roller-bearings/principles/selecting-bearing-size/bearing-life/index.html>.

SPM 2015, *BearingChecker*, viewed 13/05/15, <http://www.spminstrument.com/Products/Portable-instruments/BearingChecker/>.

Vidakovic, B & Mueller, P 1991, *Wavelets for kids: A tutorial introduction*, Duke University, Durham, NC, USA,<.

Vogelaar, MGR 1995, *moms.c*, 1.0, Kapteyn Laboratorium Groningen, <www.atnf.csiro.au/computing/software/gipsy/sub/moms.c>.

Watson, DR & Niekerk, JV 1989, 'High Speed Conveyor Idlers', in *International Materials Handling Conference*: *proceedings of the International Materials Handling Conference* Johannesburg SA.

Wescott, T 2015, 05/01/15, 'Sampling: What Nyquist Didn't Say, and What to Do About It', *Wescott Design Services*, viewed 17/10/15, <http://www.wescottdesign.com/articles/Sampling/sampling.pdf>.

Yusong, P & Lodewijks, G 2011, 'The application of RFID technology in large-scale dry bulk material transport system monitoring', in *Environmental Energy and Structural Monitoring Systems (EESMS), 2011 IEEE Workshop on*: *proceedings of theEnvironmental Energy and Structural Monitoring Systems (EESMS), 2011 IEEE Workshop on* pp. 1-5.

Zhao, L 2011, 'Typical Failure Analysis and Processing of Belt Conveyor', *Procedia Engineering*, vol. 26, no. 0, pp. 942-6, <http://www.sciencedirect.com/science/article/pii/S1877705811051034>.

Zhou, W, Habetler, TG & Harley, RG 2007, 'Bearing Condition Monitoring Methods for Electric Machines: A General Review', in *Diagnostics for Electric Machines, Power Electronics and Drives, 2007. SDEMPED 2007. IEEE International Symposium on*: *proceedings of theDiagnostics for Electric Machines, Power Electronics and Drives, 2007. SDEMPED 2007. IEEE International Symposium on* pp. 3-6.

Appendix A

Project Specification

University of Southern Queensland
ENG4111/ENG4112 Research Project

FACULTY OF HEALTH, ENGINEERING AND SCIENCES

PROJECT SPECIFICATION

FOR: DAVID COOPER Student No.: 0050025878

TOPIC: SENSOR PLATFORM FOR MONITORING CONVEYOR BELT ROLLERS

SUPERVISOR: Dr. John Billingsley

VERSION: Issue 1, 18 March 2015

PROJECT AIM: To develop a sensor platform to detect and communicate the health of a conveyor belt roller while it is in service, using pre-determined indicators of roller condition.

PROGRAMME

1. Research and report on the failure modes of conveyor belt rollers.
2. Research and report on the critical factors in the failure of belt roller bearings.
3. Select and implement a sensor platform to monitor conveyor belt rollers.
4. Design and construct a suitable power supply system.
5. Select and implement a wireless communication system.
6. Test and evaluate the performance of the system by analysing the output data.
7. Discuss the financial viability of the monitoring solution, by estimating the initial investment costs and maintenance costs compared to the benefits of improved belt roller monitoring.

As time permits:

8. Make contact with industry to facilitate performance testing of the solution in an industrial environment.
9. Develop software to analyse and intelligently report on the collected data.

AGREED

Student: _____ Supervisor: _____

Date: ___ / ___ / ___ Date: ___ / ___ / ___

Appendix B

Electrical Schematic

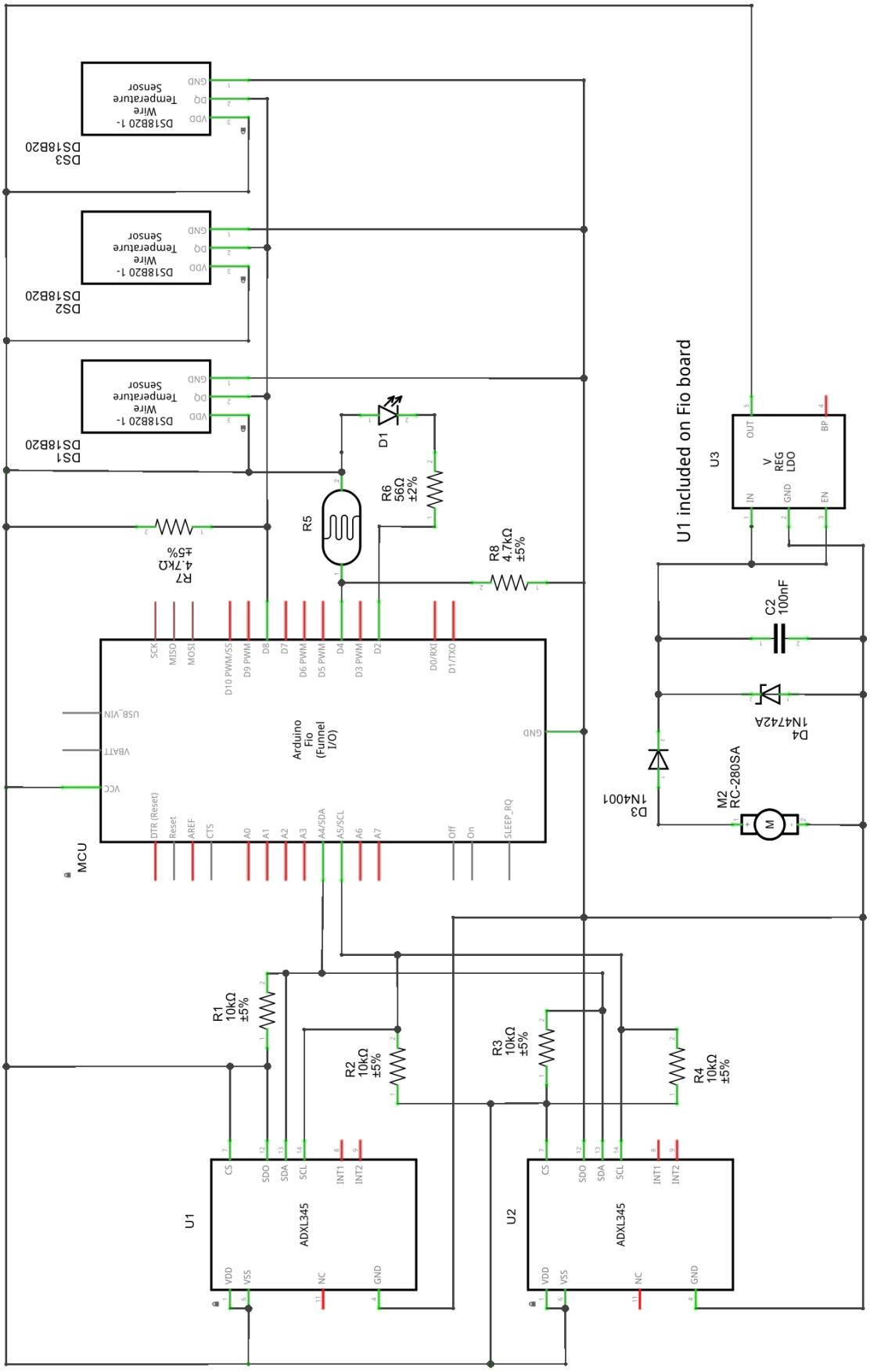

Figure B.1: Electrical Schematic of the Sensor Platform

Appendix C

Code Listing

C.1 Wireless Range Test Code

```
/*********************************************************************
                    ROLLER SENSOR PLATFORM - WIRELESS TEST

Author:       David Cooper
Student No.:  0050025878
Date:         20/09/15
Version:      1.4

The program below transmits an incrementing number through the serial
port. The number itself is not important, but its linear increase is.
Any irregularities signify lost data packets.  To stop the Xbee
buffers over-flowing, the program stops sending data if the
Clear-To-Send pin is set HIGH by the Xbee.  Output can be viewed a
serial terminal.

FILE: xbee_test2.ino
*********************************************************************/

// Declare variables
unsigned long time;     // Stores MCU runtime
const int CTSpin = 2;   // Set pin for CTS signal

// Main program loop
void setup(){
  pinMode(CTSpin, INPUT);
  digitalWrite(CTSpin, HIGH);
  Serial.begin(9600);
}
void loop(){
  time = millis(); // Assign current runtime to time variable
  if ((digitalRead(CTSpin)) == LOW) {
    Serial.println(time);         //Prints time since program started
  }
  if ((digitalRead(CTSpin)) == HIGH){ // Stop TX if buffer full
    delay(1000);
  }
  // Wait a tenth of a sec so the system has time to TX the last number
  delay(100);
}
```

Figure C.2: Code listing for wireless range testing

C.2 Accelerometer Test Code

```
/*********************************************************************
                ROLLER SENSOR PLATFORM - ACCELEROMETER TEST

Author:       David Cooper
Student No.:  0050025878
Date:         21/09/15
Version:      1.0

The program below makes a set of readings from 1 or 2 connected
accelerometers as fast as possible.  The time elapsed between each
read is stored next to the read value.  As the accelerometer is
expected to be mounted vertically, only the X axis data is captured.

FILE: accel_test1.ino
*********************************************************************/

#include <Wire.h>

#define VSENS_U1 (0x1D)     //first ADXL345 device address
#define VSENS_U2 (0x53)     //second ADXL345 device address
#define R_BYTES (6)         //num of bytes to read each time
#define SAMP_SIZE (10)     // Dataset size

// Define global variables

byte v_data[R_BYTES] ;      //Buffer for saving data read from a VSENS
int roller_vib[SAMP_SIZE];  //Create array for dataset
//Location of axis-acceleration-data register on the ADXL345
int regAddress = 0x32;

// Program setup
void setup() {
  Wire.begin();         // Start I2C bus
  Serial.begin(9600);   // Start serial bus
  delay(1000);          // Allow the accelerometer to stabilise
  VSENS_w(VSENS_U1, 0x2C, 15); // Set the sample rate to 800 Hz
  //Turn on measurement mode on the vibration sensors
  VSENS_w(VSENS_U1, 0x2D, 24);
 // VSENS_w(VSENS_U2, 0x2D, 24);
}

// Main program loop
void loop() {

  VIBdata();           // Read vibration sensors
  //Send the x y z values as a string to the serial port
  for (int i = 0; i < SAMP_SIZE; i++) {
    Serial.print(roller_vib[i]);
    Serial.print(" ");
    }
  Serial.println();
  //Output delay is needed in order not to clog the port
  delay(2000);
}
```

Figure C.3: Code Listing for Accelerometer Test, Part 1

```
//-------Functions-------

// Write value val to address register on VSENS
void VSENS_w(int device, byte address, byte val) {
   Wire.beginTransmission(device);      //start transmission to device
   Wire.write(address);                 // send register address
   Wire.write(val);                     // send value to write
   Wire.endTransmission();              //end transmission
}

// Read num bytes starting from address register on VSENS and stores
// in array v_data
void VSENS_r(int device, byte address, int num, byte v_data[]) {
  Wire.beginTransmission(device);     // start transmission to VSENS
  Wire.write(address);                // send address to read from
  Wire.endTransmission();             // end transmission
  Wire.beginTransmission(device);     // restart transmission to VSENS
  Wire.requestFrom(device, num);      // request num bytes from VSENS
  int i = 0;
  while(Wire.available())     // Repeat while data is available
  {
    v_data[i] = Wire.read();  // Receive a byte
    i++;
  }
  Wire.endTransmission();     // End transmission
}

//Reads accelerometers and formats values into x, y, and z
void VIBdata() {
  int x1 = 0, y1 = 0, z1 = 0;
  int x2 = 0, y2 = 0, z2 = 0;
  int i;
  for (i = 0; i < SAMP_SIZE; i++) {
    unsigned long starttime = micros(); // Record the start time
    VSENS_r(VSENS_U1, regAddress, R_BYTES, v_data); // Read accel 1
    x1 = (((int)v_data[1]) << 8) | v_data[0];
    y1 = (((int)v_data[3])<< 8) | v_data[2];
    z1 = (((int)v_data[5]) << 8) | v_data[4];

//  VSENS_r(VSENS_U2, regAddress, R_BYTES, v_data); // Read accel 2
//  x2 = (((int)v_data[1]) << 8) | v_data[0];
//  y2 = (((int)v_data[3])<< 8) | v_data[2];
//  z2 = (((int)v_data[5]) << 8) | v_data[4];

    roller_vib[i]=x1; // Only record the X axis data
    i++;
   // delayMicroseconds(1500);
    roller_vib[i]=micros()-starttime; // Calculate elapsed time
  }
}
```

Figure C.4: Code Listing for Accelerometer Test, Part 2

C.3 Monitoring Platform Code

```
/*********************************************************************
                ROLLER SENSOR PLATFORM - CONTROL CODE

Author:       David Cooper
Student No.:  0050025878
Date:         25/10/15
Version:      2.0

The program below reads data from the attached temperature and
acceleration sensors, and calculates the kurtosis of the vibration
data set.  This data is then transmitted using a pre-defined data
format to a central server.

FILE: rsp_v2.ino
*********************************************************************/

#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define VSENS_U1 (0x1D)       // first ADXL345 device address
#define VSENS_U2 (0x53)       // second ADXL345 device address
#define R_BYTES (6)           // num of bytes to read each time
#define SAMP_SIZE (500)       // Dataset size
#define ROLLER_ID ("02A3C9")  // Example ID of the roller
#define ONE_WIRE_BUS 8        // Define the temp sensors pin
#define BEARING_SENS_NO 0     // Index of the bearing temp sensor
#define INTERNAL_SENS_NO 1    // Index of the internal temp sensor

// Define global variables

byte v_data[R_BYTES] ;      // Buffer for saving data read from a VSENS
int roller_vib[SAMP_SIZE];  // Create array for dataset
char tx_buff[64];           // Buffer to format data before sending
//Location of axis-acceleration-data register on the ADXL345
int regAddress = 0x32;
OneWire oneWire(ONE_WIRE_BUS); // Setup a oneWire instance
DallasTemperature sensors(&oneWire); // Pass onewire to Dallas Temp.

// Program setup
void setup() {
  Wire.begin();        // Start I2C bus
  Serial.begin(9600);  // Start serial bus
  delay(1000);         // Allow the accelerometer to stabilise
  VSENS_w(VSENS_U1, 0x2C, 15); // Set the sample rate to 800 Hz
  //Turn on measurement mode on the vibration sensors
  VSENS_w(VSENS_U1, 0x2D, 24);
 // VSENS_w(VSENS_U2, 0x2D, 24);

  sensors.begin();   // Initialise the temp sensors
 }
```

Figure C.5: Monitoring Platform Code Listing, Part 1

```
// Main program loop
void loop() {

  int RPM = 2400; // Sample RPM value
  int bearing_temp = read_temp(BEARING_SENS_NO);
  int internal_temp = read_temp(INTERNAL_SENS_NO);
  VIBdata();          // Read vibration sensors
  Serial.print("START ");
  Serial.print(ROLLER_ID);
  Serial.print(" ");
  Serial.print(VIB_calc());
  Serial.print(" ");
  Serial.print(bearing_temp);
  Serial.print(" ");
  Serial.print(internal_temp);
  Serial.print(" ");
  Serial.print(RPM);
  Serial.print(" ");
  Serial.print(millis());
  Serial.print(" ");
  Serial.println("END");
  // Output delay to conserve bandwidth
  delay(500);
}

//-------Functions-------

// Write value val to address register on VSENS
void VSENS_w(int device, byte address, byte val) {
   Wire.beginTransmission(device);      //start transmission to device
   Wire.write(address);                 // send register address
   Wire.write(val);                     // send value to write
   Wire.endTransmission();              //end transmission
}

// Read num bytes starting from address register on VSENS and stores
// in array v_data
void VSENS_r(int device, byte address, int num, byte v_data[]) {
  Wire.beginTransmission(device);     // start transmission to VSENS
  Wire.write(address);                // send address to read from
  Wire.endTransmission();             // end transmission
  Wire.beginTransmission(device);     // restart transmission to VSENS
  Wire.requestFrom(device, num);      // request num bytes from VSENS
  int i = 0;
  while(Wire.available())     // Repeat while data is available
  {
    v_data[i] = Wire.read();  // Receive a byte
    i++;
  }
  Wire.endTransmission();     // End transmission
}
// Reads accelerometers and formats values into x, y, and z
void VIBdata() {
  int x1 = 0, y1 = 0, z1 = 0;
  int x2 = 0, y2 = 0, z2 = 0;
  int i;
  for (i = 0; i < SAMP_SIZE; i++) {
    VSENS_r(VSENS_U1, regAddress, R_BYTES, v_data); // Read accel 1
    x1 = (((int)v_data[1]) << 8) | v_data[0];
    y1 = (((int)v_data[3])<< 8) | v_data[2];
    z1 = (((int)v_data[5]) << 8) | v_data[4];
```

Figure C.6: Monitoring Platform Code Listing, Part 2

```
//  VSENS_r(VSENS_U2, regAddress, R_BYTES, v_data); // Read accel 2
//  x2 = (((int)v_data[1]) << 8) | v_data[0];
//  y2 = (((int)v_data[3])<< 8) | v_data[2];
//  z2 = (((int)v_data[5]) << 8) | v_data[4];

    // Delay to prevent same data being read again from the ADXL345
    delayMicroseconds(300);
    roller_vib[i]=x1; // Only record the X axis data
  }
}

// Returns Kurtosis of the data set
float VIB_calc() {
  // Declare variables
  float sum = 0.0;
  float sqr_diff = 0.0;
  float sdev = 0.0;
  float kurtosis = 0.0;
  // Find the mean of the data set
  for (int i = 0; i < SAMP_SIZE; i++){
    sum += roller_vib[i];
  }
  float mean = sum / (float) SAMP_SIZE;
  // Find the mean of the data set
  for (int i = 0; i < SAMP_SIZE; i++){
    sqr_diff = (roller_vib[i] - mean) * (roller_vib[i] - mean);
    sdev += sqr_diff;
    kurtosis += sqr_diff * sqr_diff;
  }
  // Use variance to find standard deviation
  sdev = sqrt(sdev / (float) (SAMP_SIZE - 1) );

  // Prevent a division by zero
  if (sdev == 0.0){
    kurtosis = 0.1;
  }
  else {
    // Calculate kurtosis value
    kurtosis = (kurtosis/(sdev*sdev*sdev*sdev* (double) SAMP_SIZE))-3.0;
  }

return kurtosis;

}
// Read temperature sensor values
float read_temp(byte index) {
  sensors.requestTemperatures(); // Start reading temperatures
  // Convert to degrees Celsius
  return ((sensors.getTempCByIndex(index)-32.0)/1.8);
}
```

Figure C.7: Monitoring Platform Code Listing, Part 3

Appendix D

Risk Analysis

A risk analysis was performed for the project task work. This includes constructing the prototype platform and operation during the testing phase.

Table D.1: Risk Analysis

Description of Hazard	**Risk Rating**	**Controls in place**	**Managed risk rating**
Electrical shock.	19	De-energise all circuits prior to work. Inspect wiring fully before connection to mains power. Keep body away from high voltage parts of the circuit.	15
High speed rotating cutting parts, i.e. power tools.	14	Plan task before beginning. Keep hands and body away from cutting tool edges. Unplug power tool when not in use. Wear eye protection at all times.	13
Entanglement with rotating parts.	12	Stand back from test platform while in operation. Only touch controls and on/off switch. Avoid loose clothing.	6
Projectiles thrown off spinning parts.	9	Wear eye protection when test platform is operating. Stand back along axis of rotation.	3
Hot motors and bearings.	16	Allow parts to cool before accessing them. Use IR heat sensor to check part has cooled.	4
Laser radiation.	10	Point laser tachometer away from face at all times.	4

Table D.2: Risk Analysis Legend

Consequence		**Likelihood**				
Keyword	Description Safety Health & Hygiene	Almost Certain	Likely	Possible	Unlikely	Rare
Catastrophic	-Fatality or Disability -Huge Financial Loss	Extreme 25	Extreme 24	Extreme 23	High 19	Medium 15
Major	-Lost time Injury -Loss of production -Major Financial Loss	Extreme 22	Extreme 21	High 18	Medium 14	Medium 13
Moderate	-Medical treated injury -High financial loss	Extreme 20	High 17	Medium 12	Medium 11	Low 6
Minor	-First Aid Injury -Medium financial loss	High 16	Medium 10	Medium 9	Low 5	Low 4
Insignificant	-No injuries -Low financial loss	Medium 8	Medium 7	Low 3	Low 2	Low 1

Appendix E

Project Timeline

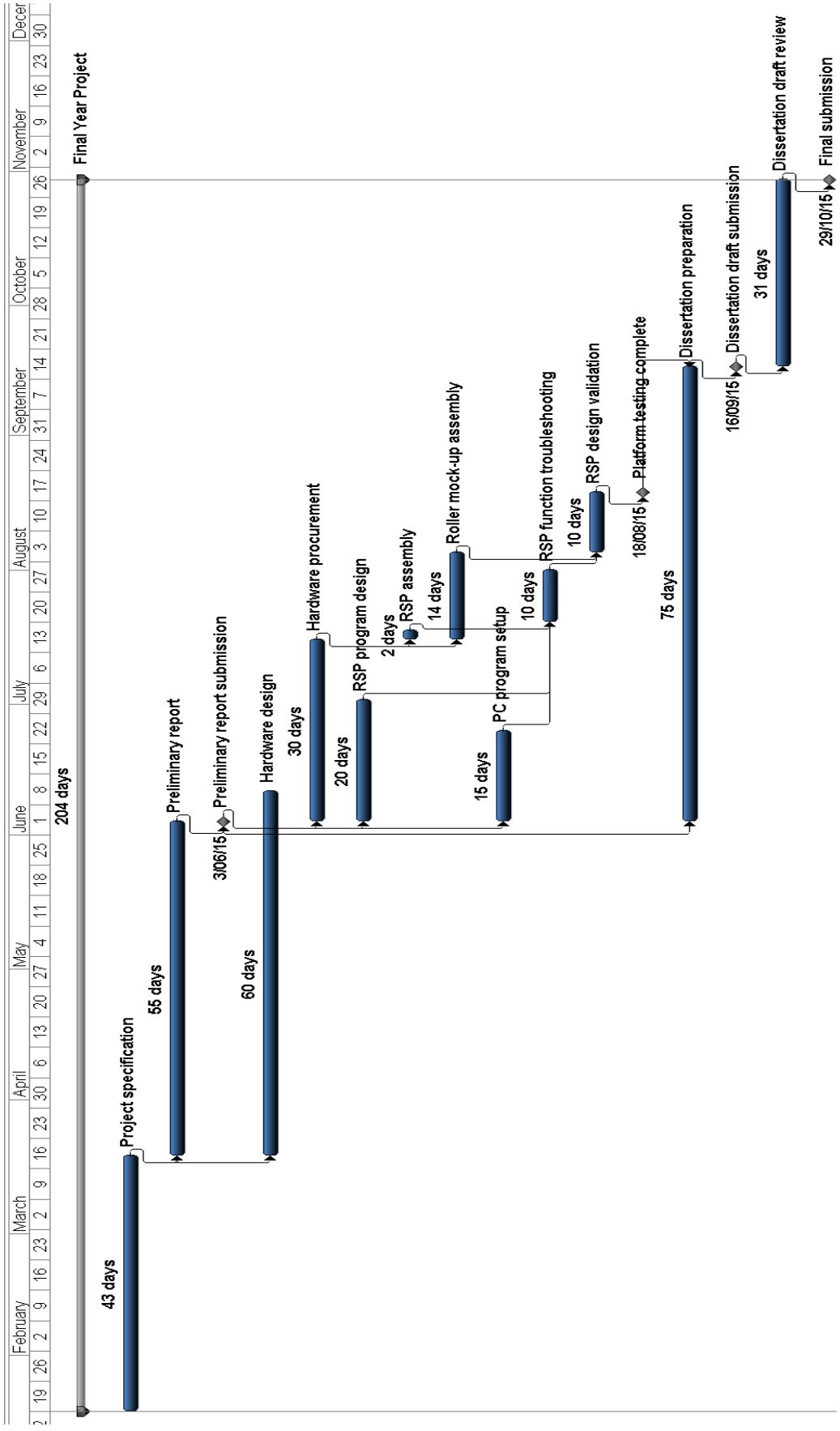

Figure E.8: RSP Project Timeline