Abstract

Slusser, Timothy James. The Application of MEMS Accelerometers for Accurately Finding Warp Yarn Breaks in Textile Machinery
(Under the direction of Dr. Edward Grant.)

The textile industry, particularly in the weaving areas, needs sensors to monitor for faults and to aid the automation of warp yarn repair. As MEMS (MicroElectroMechanical Systems) technology advances, sensors and actuators get smaller. MEMS sensors are very powerful and are highly accurate. These sensors are inexpensive and are readily available. Currently, in the textile machinery, drop wires are used to monitor the tension of the warp yarns in the weaving process. These drop wires are abrasive to the warp yarns and can lead to more warp yarn breaks. Therefore, it would be beneficial to develop a system that does not contact the warp yarn in any way, such that extra warp yarns are not broken because of the sensor. This research has led to the development of a sensor system that has no contact with the warp yarn. The main purpose was to show proof of concept for applying a MEMS sensor to the textile machinery, specifically the Jacquard Loom, in order to develop a sensor system having no contact with the warp yarns. A MEMS Accelerometer, available from Analog Devices, was used to monitor the motion of the heddle, whose acceleration properties change based on the presence of the warp yarn. Matlab was used to interpret the data and analyze for broken warp yarns (using recorded data) based on the change in acceleration. Once a warp yarn is determined to have been broken, Matlab would notify the user.

The Application of MEMS Accelerometers for Accurately Finding Warp Yarn Breaks in Textile Machinery

by
Timothy J. Slusser

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Electrical and Computer Engineering

Raleigh

March 2003

Approved By:

____
Chair of Advisory Committee

________________________________ ________________________________

Biography

Timothy Slusser was born August 9, 1978 in West Palm Beach, Florida to Deborah Ohl and James Slusser and raised by Deborah and Larry Ohl. He received his Bachelor of Science in Electrical Engineering from North Carolina State University, Raleigh, NC in 2001. He married Sarah Winters on June 9, 2001. He currently is a candidate to receive his Master of Science in Electrical Engineering from North Carolina State University, Raleigh, NC in 2003.

Acknowledgements

I would like to thank my committee members, Dr. George Hodge and Dr. Mark White, for their time, and my advisor Dr. Edward Grant for his patience and advice. I would like to extend special thanks to David Lassiter for his help with building the test frame. I also would like to thank everyone in the CRIM lab, especially Andrew Nelson for his contributions to the Matlab code, and Kyle Luthy for help with the editing. Lastly, I'd like to thank my family, my wife Sarah, and my parents Deborah and Larry Ohl, for their support.

Table of Contents

List of Tables .... **vi**

List of Figures .... **vii**

Chapter 1. Introduction **1**
Section 1.1 Motivation 2
Section 1.2 Thesis Goals 2
Section 1.3 Outline of Thesis 3

Chapter 2. Literature Review **4**

Chapter 3. A MEMS Sensor-Based Fault Diagnosis Jacquard Loom Test-Bed **12**
Section 3.1 An Overview of the Test-bed 13
Section 3.2 The Harnesses 14
Section 3.3 Design Specifications for Driving the Harnesses 14
Section 3.3.1 The Solenoid System 15
Section 3.3.1.1 Driving the Solenoid 15
Section 3.3.1.2 Generating the PWM signal 16
Section 3.3.1.3 Drawbacks to using a solenoid and other considerations 17
Section 3.3.2 The Cam Lever System 18
Section 3.3.2.1 The Cam 19
Section 3.3.2.2 The Cam Shaft 20
Section 3.3.2.3 The Cam Levers 21
Section 3.3.2.4 The Cam Follower 24
Section 3.3.2.5 Driving the Cam-Shaft 24
Section 3.3.2.6 Controlling the Harness path 25

Chapter 4. Sensing and Communications **27**
Section 4.1 System Analysis 27
Section 4.2 Harness Properties 28
Section 4.3 Sensor Requirements 30
Section 4.4 Proposed Sensors 31
Section 4.4.1 An LVDT sensor in the Jacquard Loom 31
Section 4.4.2 A Load Cell sensor in the Jacquard Loom 34
Section 4.4.3 An Accelerometer sensor in the Jacquard Loom 36
Section 4.4.4 Conclusions about the sensors 37
Section 4.5 The Accelerometer 37
Section 4.5.1 Accelerometer Theory 32
Section 4.5.2 Building a circuit for the Accelerometer 32
Section 4.5.2.1 Pin Configuration for the ADXL202AE 32
Section 4.5.2.2 Circuit Design 34
Section 4.5.2.3 Circuit construction 35
Section 4.5.3 Mounting the circuit on the harness 36

Section 4.5.3.1 Attaching the circuit to the harness ... 36
Section 4.5.3.2 The accelerometer mounted ... 38
Section 4.6 Acquiring data from the accelerometer ... 39
Section 4.7 Acquiring the Data ... 41
Section 4.7.1 Calibrating the Accelerometer ... 42
Section 4.7.2 Decoding the output using the microcontroller ... 45
Section 4.7.3 Analyzing the output ... 46

Chapter 5. Experimentation and Results ... 47
Section 5.1 No load, different speeds ... 47
Section 5.1.1 Constant load, different speeds ... 48
Section 5.1.2 Removing the load during operation ... 49
Section 5.2 The Results ... 49
Section 5.2.1 Load vs. No Load ... 50
Section 5.2.2 Removing the Load ... 52
Section 5.3 Analyzing the Results ... 53
Section 5.3.1 Using Matlab to search for breaks ... 54
Section 5.3.2 Simulated Data ... 54
Section 5.3.3 Real Data ... 56

Chapter 6. Conclusion and Future Research ... 58

Chapter 7. References ... 60

Chapter 8. Appendices ... 63
Section 8.1 Accelerometer output plots ... 63
Section 8.1.1 Output plots when there is no motion ... 63
Section 8.1.2 Output plots at different speeds with no load ... 64
Section 8.1.3 Output plots at different speeds (same motor input) with the same load ... 66
Section 8.1.4 Output plots at different speeds (adjusted motor input) with the same load ... 69
Section 8.1.5 Output plots at the same speed with different loads ... 71
Section 8.2 BasicX BX-24 Accelerometer code ... 73
Section 8.3 Matlab Source Code ... 91
Section 8.4 Accelerometer Data Sheet ... 97
Section 8.5 Motor Specifications ... 98
Section 8.6 Frame Dimensions ... 99
Section 8.7 Cam Dimensions ... 100
Section 8.8 Lever Dimensions ... 101

List of Tables

Table 4.1 Properties of the Harness-Spring system 29

Table 4.2 Pin-outs for the ADXL202AE 40

Table 5.1 Input/Output of the motor under no load conditions... 55

Table 5.2 Input/Output of the motor under no load conditions with the same Armature input, but lower output speed 56

Table 5.3 Input/Output of the motor under constant load conditions with a higher Armature input, but with the same output speed.. 56

Table 8.1 Motor Characteristics from Data Sheet.. 106

Table 8.2 Frame Dimensions 107

Table 8.3 Lever Dimensions based on lever position 109

List of Figures

Figure 2.1 Drop wire schematic (Gahide, 2001) 6

Figure 2.2 Strain Gauge Sensors developed by Severine Gahide 10

Figure 3.1 The completer test apparatus 12

Figure 3.2 A view of the system 13

Figure 3.3 A PWM signal 14

Figure 3.4 A 555 Timer setup to control frequency 16

Figure 3.5 A 555 Timer setup to control duty cycle 16

Figure 3.6 Layout for the DRV101 solenoid driver circuit 17

Figure 3.7 Solenoid Lever setup. 18

Figure 3.8 The Cam Lever setup 19

Figure 3.9 Machined Cam 19

Figure 3.10 The Cam Shaft 20

Figure 3.11 A view of the levers 21

Figure 3.12 Geometry setup for computing lever angles 22

Figure 3.13 The follower is the small black wheel at the end of the lever 24

Figure 3.14 The motor (center), the Cam-Shaft (left), and the tachometer (right) 25

Figure 3.15 Plexiglas guide for harness path 25

Figure 4.1 Force Diagram and System Analysis 27

Figure 4.2 Spring Constant of the Harness 29

Figure 4.3 A cross section of a LVDT 32

Figure 4.4 LVDT placement on the harness 32

Figure 4.5 The Strain Gauge 33

Figure 4.6 Load cell under applied uni-axial strain 34

Figure 4.7 Load Cell placement on the harness 34

Figure 4.8 Accelerometer placement on the harness.. 35

Figure 4.9 The ADXL202AE MEMS accelerometer.. 36

Figure 4.10 Functional block diagram of the ADXL202AE Accelerometer 37

Figure 4.11 A description of common accelerometer properties.................................... 38

Figure 4.12 Working range for an accelerometer 39

Figure 4.13 Pin Layout.. 40

Figure 4.14 Circuit Layout 42

Figure 4.15 CirCAD drawing (left) and picture (right) of the Accelerometer circuit....... 43

Figure 4.16 Axes of the accelerometer.. 44

Figure 4.17 A view of the harness.. 45

Figure 4.18 The accelerometer circuit mounted... 46

Figure 4.19 The BasicX BX-24 Microcontroller 47

Figure 4.20 Connecting the ADXL202 to the BX-24 evaluation board 48

Figure 4.21 Interface board for the Accelerometers 48

Figure 4.22 Acquiring time constants from the PWM signal 49

Figure 4.23 Oscilloscope output when X axis is parallel to the plane of gravity 50

Figure 4.24 Oscilloscope output when X axis is perpendicular to the plane of gravity in the positive X direction 51

Figure 4.25 Oscilloscope output when X axis is perpendicular to the plane of gravity in the negative X direction 52

Figure 4.26 Flowchart for decoding the output... 53

Figure 4.27 Excel output of 4 accelerometers perpendicular to the plane of gravity. Output is approximately 1 g. 54
Figure 5.1 Heddle Accelerations of accelerometer 1 seen at conditions 1 (100 rpm), 2 (100 rpm), and 3 (86 rpm) 58
Figure 5.2 Heddle Accelerations of accelerometer 1 seen at conditions 1 (150 rpm), 2 (150 rpm), and 3 (136 rpm) 59
Figure 5.3 Heddle Accelerations of accelerometer 1 seen at conditions 1 (200 rpm), 2 (200 rpm), and 3 (186 rpm) 60
Figure 5.4 Accelerometer output of accelerometer 1 seen when a 200 g load is removed, speed begins at 100 rpm and ends at 120 rpm........ 60
Figure 5.5 Accelerometer output of accelerometer 1 seen when a 200 g load is removed, speed begins at 150 rpm and ends at 186 rpm........ 61
Figure 5.6 Matlab output of 4 accelerometers. The red plot shows a break occurring on accelerometer 4 63
Figure 5.7 Matlab output of 4 accelerometers when the load is removed from all 4 accelerometers at the same time, starting at 100 rpm loaded........ 64
Figure 5.8 Matlab output of 4 accelerometers when the load is removed from all 4 accelerometers at the same time, starting at 150 rpm loaded........ 65
Figure 8.1 Accelerometer output with no motion 71
Figure 8.2 Accelerometer output at 100 rpm 72
Figure 8.3 Accelerometer output at 125 rpm 72
Figure 8.4 Accelerometer output at 150 rpm 73
Figure 8.5 Accelerometer output at 175 rpm 73

Figure 8.6 Accelerometer output at 200 rpm 74

Figure 8.7 Accelerometer output at 86 rpm (same motor input)...................................... 74

Figure 8.8 Accelerometer output at 113.5 rpm 75

Figure 8.9 Accelerometer output at 136 rpm 75

Figure 8.10 Accelerometer output at 163.5 rpm 76

Figure 8.11 Accelerometer output at 186 rpm 76

Figure 8.12 Accelerometer output at 100 rpm with a load of 25 g 77

Figure 8.13 Accelerometer output at 125 rpm with a load of 25 g 77

Figure 8.14 Accelerometer output at 150 rpm with a load of 25 g 78

Figure 8.15 Accelerometer output at 175 rpm with a load of 25 g 78

Figure 8.16 Accelerometer output at 200 rpm with a load of 25 g 79

Figure 8.17 Accelerometer output at 200 rpm with a load of 25 g 79

Figure 8.18 Accelerometer output at 200 rpm with a load of 35 g 80

Figure 8.19 Accelerometer output at 200 rpm with a load of 50 g 80

Figure 8.20 ADXL202AE Top page of Data sheet.. 105

Figure 8.21 Motor picture from data sheet... 106

Figure 8.22 Original sketch of the frame 107

Figure 8.23 Cam Dimensions... 108

Figure 8.24 Lever Dimensions and diagram. Dimensions A, B, and C vary depending on lever position 109

List of Abbreviations and Definitions

Shed - The gap formed when warp threads are separated during the weaving action.

Weft – The horizontal warp yarns that interlace at right angles with the vertical warp threads.

Warp - The lengthways threads in a woven fabric.

Shuttle - The mechanism on a loom that carries the weft thread through the shed to interlace with the warp.

Heddle - A flat steel strip, looped cord, or shaped wire with an eye in the centre through which warp yarn is threaded.

Pick - A weft thread passing through the warp in weaving.

Reed - Comb-like feature of a loom through which the warp ends pass.

Jacquard (weaving) - A shedding mechanism attached to the loom that gives individual control of up to several hundred warp threads and thus enables large complex designs to be produced.

Jacquard Harness (weaving) - A group of cords and their attachments, from the hooks of the machine downwards that control the lifting of the warp threads.

PWM – Pulse Width Modulated signal

Duty Cycle – percent of PWM which is “high”

Accelerometer – device used to monitor accelerations, vibrations, and tilt.

Sensor – device used to monitor for changing conditions and convert it into an electrical signal.

MEMS – MicroElectroMechanical Systems

Oscilloscope – Device used to monitor wave forms and voltages.

CRIM – Center for Robotics and Intelligent Machines

Chapter 1. Introduction

The location and repair of broken warp yarns is one of the largest time consumptions in the textile industry. Most factories employ weavers whose sole purpose is to monitor the machinery and repair the broken warp yarns. When a warp yarn breaks, the machine stops because a drop wire is attached to the warp yarn. The drop wire activates the machine shut down. However, all warp yarns have a drop wire, so it can be difficult to locate the broken warp yarn when there are several thousand warp yarns in a single machine.

This is where the application of Micro-Electro-Mechanical Systems (MEMS) Sensors and Actuators can be useful in the textile industry. MEMS sensors are becoming smaller and less expensive every year and they are very simple and easy to use. The application of MEMS sensors here is to detect warp yarn breaks through monitoring the acceleration of the weaving heddles within the weaving machine. Since each warp yarn is threaded through a heddle, each warp yarn will be monitored by an accelerometer. When a warp yarn breaks, the change in acceleration, monitored by the sensor attached to the heddle, will indicate that break. Matlab is used to analyze the data for breaks and notify the user when a break has occurred.

Monitoring the heddles for the warp yarn breaks can reduce the time that it takes a weaver to find the broken warp yarn. By notifying the weaver exactly where the broken warp yarn is, the monitoring system can save minutes of searching per break. The machines run 24 hours a day and money is lost for every extra second that the weaver takes to find and repair the broken warp yarn.

One future use of this monitoring system is to relay the information to a database and mine the data for common occurrences of broken warp yarns. This data can be used to find problems with the machinery and yarn quality to prevent broken warp yarns in the future. Furthermore, this system can be used in conjunction with an automated tying machine that will find the exact location of the broken warp yarn and then retie it within seconds of the machine going down.

Section 1.1 Motivation

The main goal of this project was to confirm that MEMS sensors could be used to monitor for broken warp yarns in a Jacquard Loom system. The sensor was constructed and tested on a test apparatus built in the CRIM lab. Experiments were conducted on the sensors to prove that they could be used in a commercial Jacquard Loom. The sensors were tested at different speeds and with different tensions. The signals generated from the tests were used to model the heddle behavior based on the presence of the warp yarn.

Based on the initial results from the accelerometers tested, it was confirmed that the change in acceleration could be measured using Matlab. However, the accelerometer sensors have not been tested in a commercial loom and it is unknown how they would perform.

Section 1.2 Thesis Goals

The objectives of this thesis are to describe the:

- Design and construction of a test apparatus to replicate the motion of a Jacquard Loom.
- Steps taken to choose the right sensor for the application.
- Development and design of a sensor circuit for placement on a heddle.
- Results and experiments that were conducted.

- Use of Matlab to identify Warp breaks.
- Future of MEMS sensors in the textile industry.

Section 1.3 Outline of Thesis

Chapter 2 outlines literature that was reviewed. This chapter includes previous work by other researchers and their ideas for sensing warp yarn breaks.

Chapter 3 describes the test apparatus used to test the concept of MEMS sensors. This chapter gives an in-depth description of the steps taken to build a test apparatus to replicate the Jacquard Loom.

Chapter 4 describes the steps taken to choose the right sensor for the Jacquard Loom application and the construction and use of the chosen sensor.

Chapter 5 describes the experiments conducted with the sensors and the results and the use of Matlab to identify warp breaks.

Chapter 6 provides conclusions and future research avenues for this idea.

The Appendices provide source code for the BasicX and Matlab as well as extra data for the sensors and the test apparatus. All dimensions for construction of the apparatus are included.

Chapter 2. Literature Review

In 1801, Joseph Jacquard invented a machine that would revolutionize the textile industry and create the foundation for modern computers (Terkoski, 2001). The Jacquard Loom, as it is known today, was the first automated machine. This machine would use punch cards to recreate patterns in weaving. The punch cards consisted of hard cardboard with holes, as the weaver lifted the hooks within the loom, only the hooks located at the holes would be raised, allowing the hook to insert another thread, thus creating the pattern. This eliminated the need for two people to run the machine since the weaver could both run the shuttle and control the pattern. Another advantage of using the punch cards is that the pattern could be reproduced multiple times without flaw (Terkoski, 2001).

The automation of the Jacquard Loom advanced the textile industry quite dramatically (Terkoski, 2001). Not only did it reduce the amount of labor needed to create a product, it also reduced the cost and time. The punch card system used by the Jacquard Loom was the first example of memory. Patterns could be saved and reproduced using the punch cards. This ability to store information helped to start the computer revolution. Charles Babbage would later use the idea of punched cards from Jacquard and design one of the first computers, the Analytical Engine (Murray, 2000).

Standard looms consist of the warp beam, heddles, the harness, the reed, the breast beam, the cloth beam, and a weft insertion device (“loom”, 2002). The warp beam is a cylinder on which the warp threads are wound. The warp threads pass through the heddles (rods or cords), each of which has an eye for the thread to be drawn. The heddles are controlled by the harness, which is a rectangular frame set to raise or lower the heddle based on the desired pattern. Raising the heddles forms a shed between the warp threads for insertion of weft threads. The weft threads are commonly inserted using a shuttle, but there are shuttle-less methods (the dummy shuttle, air or water jets, and rapier). The reed is a comb-like frame that pushes the filling warp yarn firmly against the finished cloth after each pick, or row. The cloth is wound over the breast beam to create tension with the warp beam. The woven cloth is then rolled onto the cloth beam.

Modern looms can be controlled directly by a computer instead of using punch cards. Designs can be entered using software or scanned from other sources (Noonan, 2000). Looms have also gotten much larger and faster. For example, the Jacquard machine LX3201 from Stäubli can control up to 12,288 hooks and can operate at very high speeds ("Jacquard Machine LX3201", 2001). This particular loom is digitally controlled; it uses a single cam drive to control the motion of the lifting knives. The computer determines the position of the lifting knives, such that each individual heddle can be raised or lowered depending on the pattern. Complex patterns and even pictures imported into the computer can be woven by the loom. Once a pattern has been programmed into the loom, the loom is almost completely automated. The loom can weave continuously until a warp yarn breaks. When a warp yarn breaks, the machine stops and the warp yarn has to be re-tied so that the weaving can continue.

There are two main ways to monitor for a warp yarn break within the Jacquard Loom. The first way is to create a system that maintains contact with the warp yarn itself. One way is the use of drop wires (See Figure 2.1). Each warp yarn is threaded through the legs of a U-shaped piece of metal, which is known as a drop wire. The drop wire is supported on each individual warp yarn as long as the warp yarn is under tension or unbroken. When a warp yarn breaks, the drop wire falls across two terminals. Connecting the terminals with a drop wire acts like a switch that causes the machine to stop. The drop wire system has been used in the Jacquard Loom system for many years. This is mainly due to the low cost and low maintenance of using the drop wires. For every warp yarn in a particular loom, there needs to be a stop sensor, and the drop wires are very inexpensive.

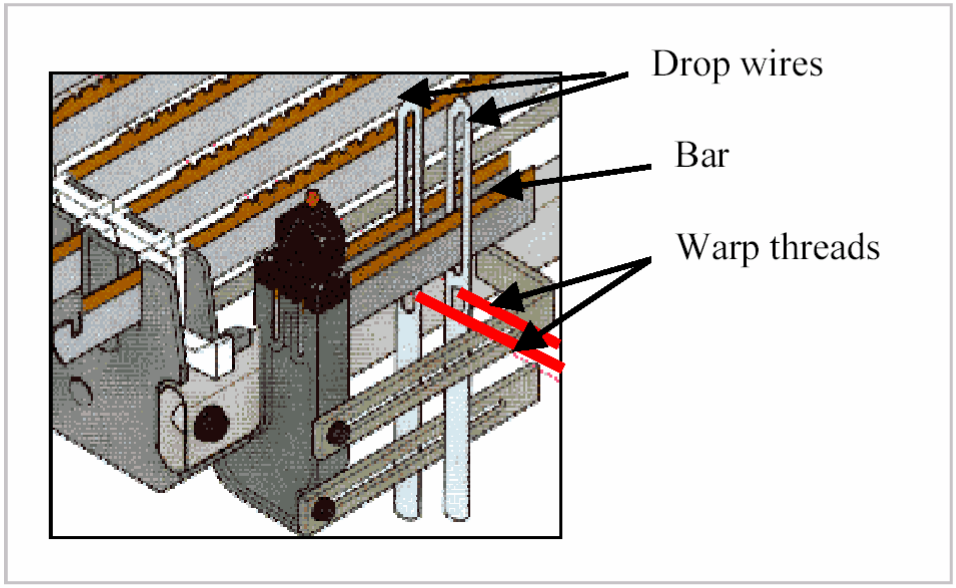


Figure 2.1 Drop wire schematic (Gahide, 2001)

Even though drop wires are an accepted system for detecting broken warp yarns, they have several disadvantages. Since the drop wire rests on the warp yarn, it can cause abrasion or vibration which can lead to more broken warp yarns (Gahide, 2001). The drop wire problem becomes worse as the warp yarns become more delicate. Another problem that the weavers face occurs when the machine stops, any one of up to 12,000 warp yarns may be broken and it can be difficult to locate the broken warp yarn. Replacing the drop wire system in the loom with another type of sensor becomes more expensive as the number of warp yarns increase. By detecting a change in tension, strain gauges, another example of a contact system, could also be used to determine whether the warp yarn has broken.

The other way to monitor a warp yarn breakage is to use a non-contact system. Unlike the contact system, the non-contact system does not cause extra stress and abrasion on the warp yarns. There have been several attempts to create a new non-contact system to detect the broken warp yarns. One method, that is available from Eltex, is called the Eltex Signal Giver. The Eltex Signal Giver senses the movement of the warp yarn by detecting the vibrations of the warp yarn ("Working Principle of Eltex Systems", 2001). The system consists of a unit which can be placed on the loom. Up to eight warp yarns pass through holes in the unit, which contain sensors for determining the number of warp yarns in a single hole, the vibrations of the warp yarns, and the movement of the warp yarns. The system can interface with the loom and notify the user when a warp yarn breaks, when it does not move, or if the warp yarn is moving when it should not be moving ("Working Principle of Eltex Systems", 2001). Since the Eltex Signal Giver uses a piezoelectric system, all types of moving strands can be monitored ("Working Principle of Eltex Systems", 2001).

The KFW 4600/4800, from Grob Horgen AG, is a new type of warp stop motion device that can be mounted on an existing Jacquard Loom. The device consists of two bars that are used as electrodes connected to a transformer and a magnetic knock-off device. When a warp yarn breaks, the wire drops and completes the electrical circuit. The current passing through the circuit operates the magnetic knock-off device, which stops the loom ("Grob Horgen Latest Infos", 2003). In order to speed up the location of the broken warp yarn, Grob Horgen AG also has added an optical scanner to the KFW 4600/4800 that has the ability to locate the quadrant of the broken warp yarn ("Grob Horgen Latest Infos", 2003).

Izumi International has developed a warp yarn breakage sensor that uses light to detect a broken warp yarn. The system consists of high accuracy lasers that are placed beneath the warp yarns. The laser generates a beam of light that is aimed at a light sensor at the other end of the machine. When a warp yarn breaks, the warp yarn should pass through the path of the beam and stop the loom ("Warp yarn Breakage Sensor", 2003). The system is a good idea because it does not involve any contact with the warp yarns and is very easy to implement. However, sometimes when a warp yarn breaks, it sticks to other warp yarns and does not fall, leading to a missed warp yarn break.

The Eltex Signal Giver and the KFW 4600/4800 can be very accurate in detecting the breakage of a warp yarn. Both units can narrow down the location of the broken warp yarn, but neither can tell the specific location of the warp yarn that is broken, which would be necessary for repair automation. Finding the exact location of the broken warp yarn requires that every warp yarn within the loom be continuously monitored for a break. To monitor every warp yarn, it is important that the break monitoring system is inexpensive and easily adapted to an existing Jacquard Loom. For each warp yarn to have a sensor monitoring it, the sensor would have to be very small, otherwise the sensor could interfere with the operation of the loom.

MEMS (Micro-Electro-Mechanical Systems) is the integration of mechanical elements, sensors, actuators, and electronics on a common silicon substrate through micro-fabrication technology (Weinberg, 2000). MEMS sensors are the smallest commercially available sensors. These sensors are both inexpensive and high performance (Travis, 1998). At an increasing rate, MEMS are coming out of the labs and into production (Travis, 1998). They are rapidly emerging as the sensor of choice for automobile safety systems (Weinberg, 2002). Since MEMS devices are so new, both designers and users of MEMS have a steep learning curve to overcome (Weinberg, 2002).

The electronic design of MEMS sensors is very challenging (Weinberg, 2002). Most MEMS sensors use variable capacitance to perform their sensing, including accelerometers and strain gauges (Weinberg, 2002). The MEMS devices need to convert the variable capacitance to a variable voltage or current, and then filter the signal in a way that will be useful to the end user (Weinberg, 2002). Since the idea behind MEMS is to keep the technology small, fitting all of the measuring and filtering into a small device is very challenging.

Because the MEMS technology is new, there is not much information about using the technology located in text books (Weinberg, 2002), therefore, in order to accurately use the MEMS, the end user has to employ trial and error techniques . Most of the time, the company providing the MEMS sensors or actuators has to maintain up to the minute detailed instructions about each particular MEMS product (Weinberg, 2002). To help the user properly utilize the sensor, Analog Devices developed a web site with tools, software and application notes all specific to their MEMS accelerometers.

MEMS designers do not have access to standard design software for the micro sensors and actuators (Weinberg, 2002). Also, the fabrication process used to create the MEMS varies based on the particular company. This leads to diversity in the MEMS market (Buss, 2001) and many researchers are calling for standardization of both the design and the interface of MEMS sensors and actuators. Standardization would lead to a wider acceptance of MEMS because it would help to speed up the learning curve involved (Buss, 2001). For MEMS to become standardized, the companies producing MEMS would have to cooperatively develop a new design and fabrication system (Buss, 2001).

Partly due to the popularity of the Analog Devices accelerometers, the most common MEMS sensor is an accelerometer. MEMS accelerometers are quickly replacing conventional accelerometers for crash air-bag deployment systems in automobiles. Conventional accelerometers can be much larger and up to 5 times more expensive than MEMS accelerometers.

Accelerometers have long been used to monitor the vibration of large machinery to provide information on its health and condition. Accelerometers that measure all ranges of acceleration are available, whether the machine is high speed (greater frequency) or low speed (smaller frequency). MEMS accelerometers are no different, except that they are easier to integrate because of their small size and low cost, therefore allowing them to be integrated without being obtrusive or unattractive. The sensors could also be mounted in places that conventional sensors are too large to be mounted.

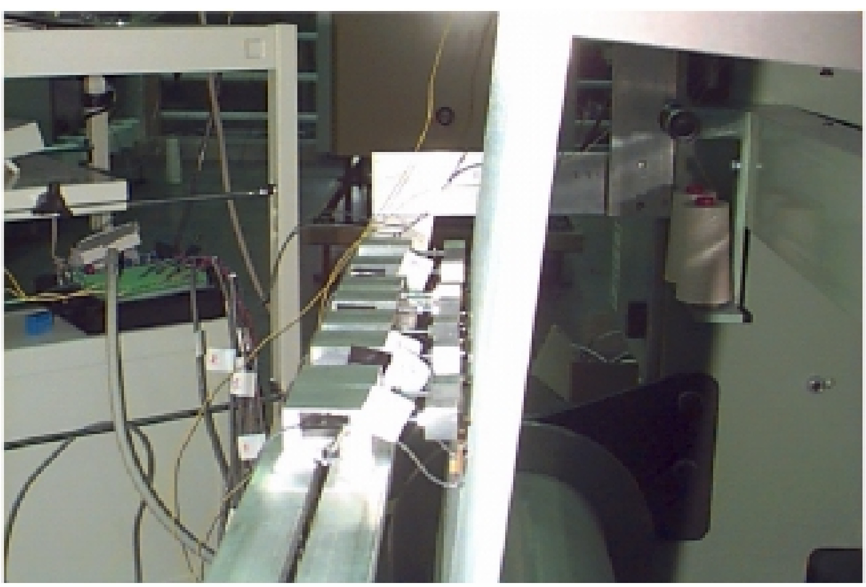


Figure 2.2 Strain Gauge Sensors developed by Severine Gahide.

In a research project for the Textiles Department at NCSU, Severine Gahide began attempting to integrate MEMS sensors into the Textiles Industry. Her idea was to create a warp break detection system for a Jacquard Loom using a warp yarn contact strain gauge sensor (Gahide, 2001). The sensor maintained contact with the warp yarn and monitored the tension by measuring the force of the warp yarn against the sensor (See Figure 2.2). When a warp yarn would break or lose tension, her system would cause the machine to stop (Gahide, 2001). While the project never used an actual MEMS sensor, it proved the concept that a strain gauge could monitor the warp yarns and could later be replaced by a MEMS strain gauge. Gahide's research laid the ground work for the research conducted in this project.

The literature search has shown that MEMS seem to be the ideal choice for detecting the warp breaks in a Jacquard Loom. The high speed of the loom and size constraints show the need for smaller and more accurate sensors. Due to their small size, MEMS accelerometers have been found to be more accurate than conventional accelerometers. Interfacing the MEMS sensors with the current Jacquard Loom should be feasible.

Chapter 3. A MEMS Sensor-Based Fault Diagnosis Jacquard Loom Test-Bed

In order to properly test the accelerometers, a decision was made to construct a test apparatus which replicated the Jacquard Loom (See Figure 3.1). The test bed was modeled after a Jacquard Loom; however, it only contains those parts of the loom that are to be monitored in this project. The stand consists of four harnesses, a harness drive system, and a means of positioning the threads. Measurements were taken from the actual Jacquard Loom so that the placement of the threads would be correct. Each thread passes through a harness, which will be attached to a fixed point to simulate the comb, while the other end of the thread passes over a pulley which is attached to a weight. The weight provides variable tensions in order to test the performance of the MEMS-based accelerometer for detecting warp yarn breaks.

Figure 3.1 The complete test apparatus.

Section 3.1 An Overview of the Test-bed

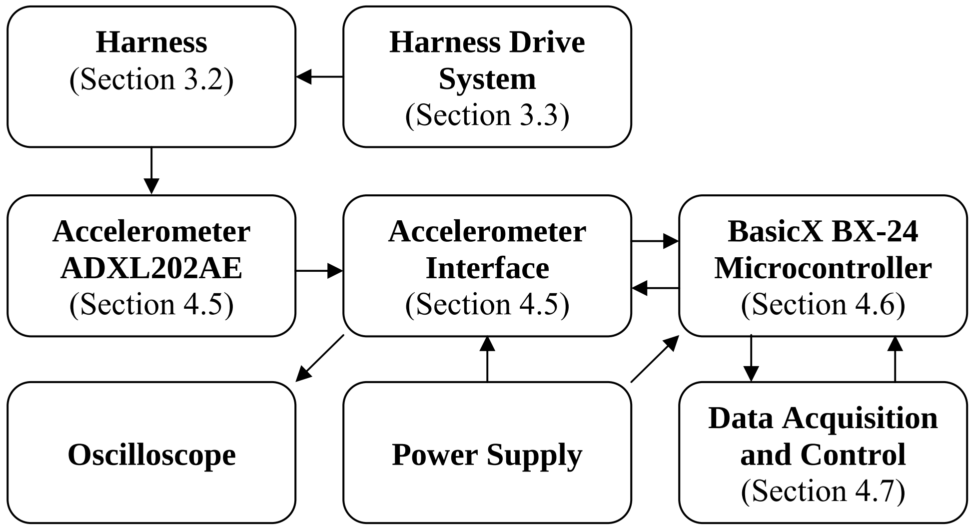


Figure 3.2 A view of the system

The system is composed of the following:

- The Harness Drive System – this is the system used to drive the harnesses, it can be either solenoids or a cam-follower system.
- The Harness – the harness from a Jacquard Loom, there are four in this setup.
- The ADXL202AE MEMS Accelerometer – the sensor attached to the harness, measures acceleration.
- The Accelerometer Interface – the interface from the accelerometer to the microcontroller.
- The BasicX BX-24 Microcontroller – a microcontroller relays the accelerometer information back to the computer.
- An Oscilloscope – used to monitor the signal from the accelerometer for testing purposes.

- A Power Supply – supplies power for all four accelerometers and the microcontroller.
- A Data Acquisition and Controller – programs the microcontroller and records the data from the accelerometer via the microcontroller.

Section 3.2 The Harnesses

In the Jacquard Loom, the harness is defined as the part that attaches the heddle to the Jacquard machine. The harness is used to raise and lower the heddle to form the shed in which the shuttle moves to insert fill warp yarn and create a pattern. The heddle is a vertical cord or wire that has a lope or eye in the middle which receives a warp yarn.

The heddles used in this project were obtained from Staubli and can be used in different types of Jacquard Looms. The heddle system consists of a tension spring, a wire containing an eye for warp yarn, an orange guidance sleeve, and a harness hook at the top. The heddle system (harness, sleeve, heddle, spring) will be referred to as the harness throughout the rest of this paper.

Section 3.3 Design Specifications for Driving the Harnesses

The test apparatus consists of four separate wire harnesses that are set to oscillate at a certain frequency, mimicking the motion of a Jacquard Loom. The period of oscillation of the harnesses is set such that only two harnesses are up during each oscillation period. To replicate the harness positions in the actual loom on the test apparatus, the harnesses are arranged diagonally with ½” between each harness. The drive system is required to displace each harness 4” vertically.

Section 3.3.1 The Solenoid System

A solenoid is basically a wire wrapped around a hollow cylinder to create an inductive coil. A magnetic bore is given free motion so that it may move within the cylinder. When a current is applied, the inductive coil acts upon the bore, either pushing the bore out or pulling it in. There are three different types of solenoids. There is a push solenoid where the current causes the bore to push out. Applying current to a pull solenoid causes the bore to retract within the inductor. The last solenoid both pushes and pulls, depending on the direction of current within the inductor.

In a commercial Jacquard Loom, the harness motion is computer controlled by solenoids. Using solenoids in this test rig would be the easiest option, since it doesn't require much engineering and solenoids are very common. These solenoids would be mounted to the top of the frame and controlled using a Pulse Width Modulated (PWM) signal.

Section 3.3.1.1 Driving the Solenoid

The solenoids can easily be controlled using a PWM signal (See Figure 3.3), which allows the frequency at which the solenoids oscillate up and down to be adjusted. PWM signals consist of two controlling factors, the duty cycle and the frequency. The duty cycle is the width of the pulse and the frequency is how often the pulse occurs. Adjusting the duty cycle will control how long the solenoid is on. When the pulse is high, a 5 volt signal is sent

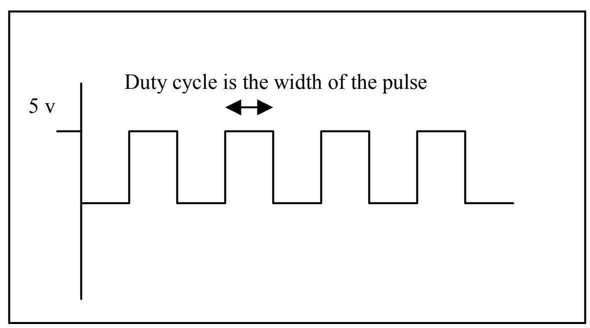


Figure 3.3 A PWM signal

to the solenoid, causing the bore to either pull or push. Setting the frequency controls the speed at which the harness oscillates.

Section 3.3.1.2 Generating the PWM signal

There are several options to generating a PWM signal. The easiest option to implement is to generate the PWM using a microcontroller. The microcontroller can digitally change both the frequency and the duty cycle of the PWM. The drawback to using a microcontroller to supply the PWM input to the solenoid is that the solenoid has a large current draw that cannot be supplied by the microcontroller. This can be fixed by applying the PWM output to a transistor. The transistor will act as a switch that is active whenever the output is high.

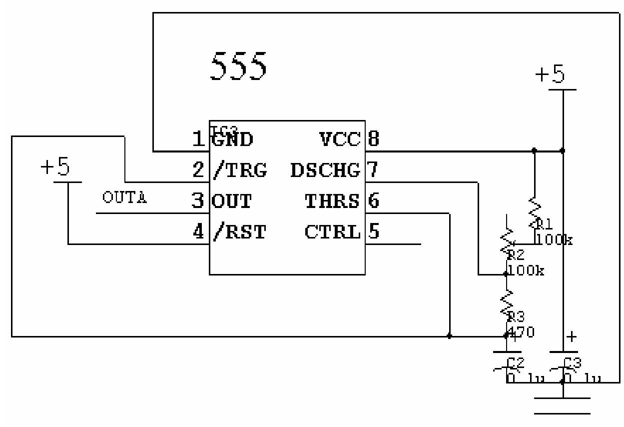


Figure 3.4 A 555 Timer setup to control frequency.

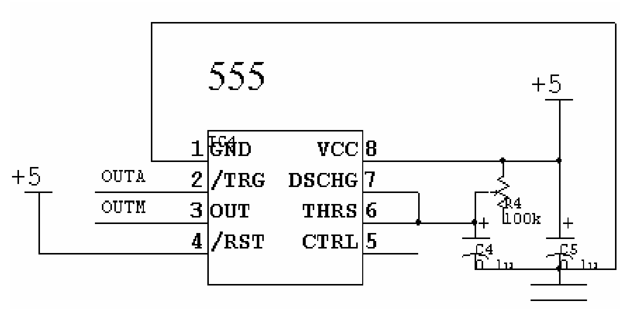


Figure 3.5 A 555 Timer setup to control duty cycle.

A PWM signal suitable for solenoid input can be generated using a double timer circuit. The circuit consists of two 555 timers (See Figures 3.4 and 3.5). The first 555 timer will set the frequency of the PWM by tuning the potentiometer located at R1. The output of the first 555 timer (Pin 3 in Figure 3.4) will be the input for the second 555 timer (Pin 2 in Figure 3.5).

The second 555 timer will set the duty cycle of the PWM signal. The output of the second 555 timer (Pin 3 in Figure 3.5) will provide the PWM signal used to control the solenoids.

Another way to generate a PWM signal using hardware is to use an integrated chip designed specifically to drive a solenoid. This chip (DRV101) generates a PWM signal at a stead frequency. The drawback of the chip is that the frequency is not adjustable, only the duty cycle is adjustable. The circuit layout (See Figure 3.6) is much simpler than the previous option. There is one IC with four discrete components compared to two IC's with five discrete components for each IC.

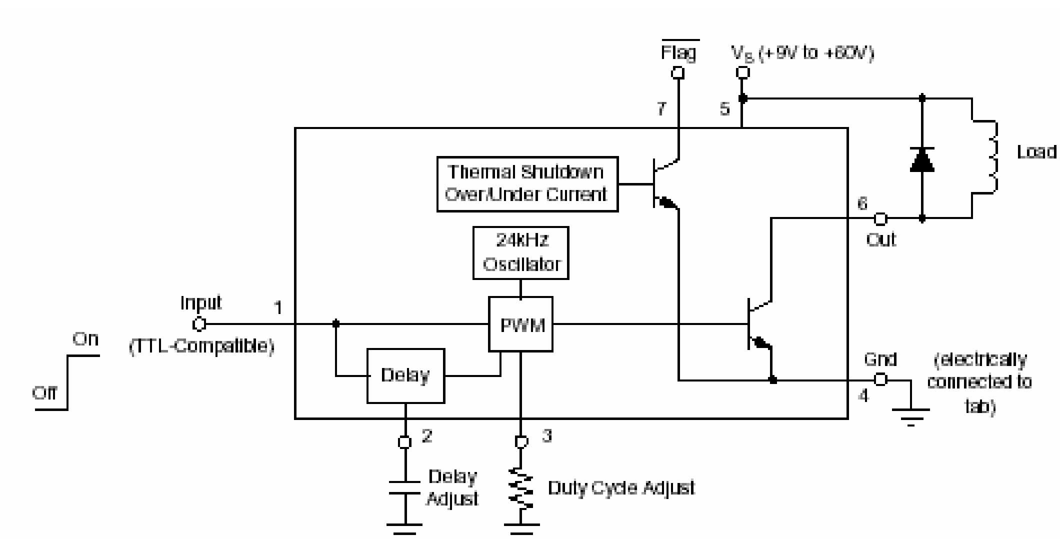


Figure 3.6 Layout for the DRV101 solenoid driver circuit

Section 3.3.1.3 Drawbacks to using a solenoid and other considerations

The Jacquard Loom system requires that the harness displace a total of 4". Most solenoids do not displace more than 2" and are output very little force. The 2" solenoid was not strong enough to displace the harness. The solenoids used on some Jacquard Looms are very expensive and are not available for commercial purchase. An alternate method is to use a push solenoid in conjunction with a lever. The lever would increase the strength of the

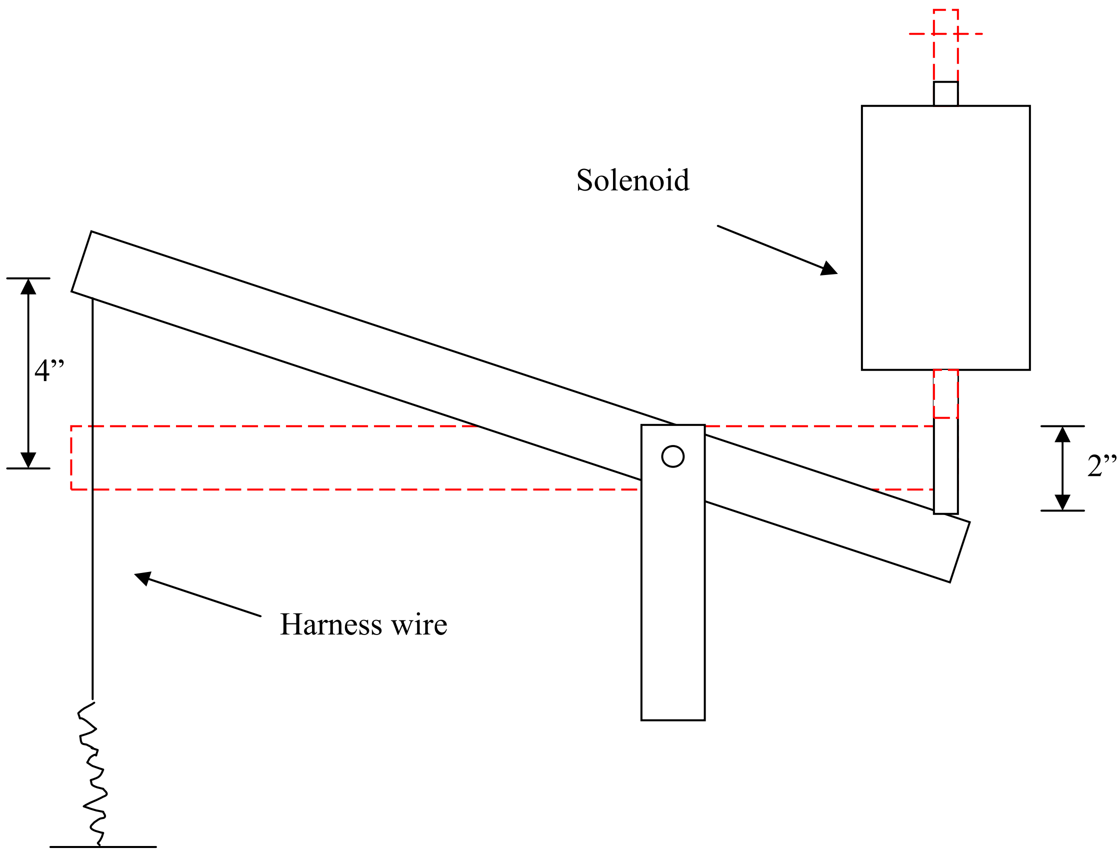


Figure 3.7 Solenoid Lever setup. The initial position of the lever is denoted by a dashed line and the final position is a solid.

solenoid and allow a greater displacement (See Figure 3.7).

Section 3.3.2 The Cam Lever System

The cam system consists of a shaft with four opposing cams that operate four different levers. As the end of the lever following the cam goes down, the opposite end displaces the harness a maximum of 4 inches. The lever will be set up such that the spring on the harness will keep the following end of the lever connected to the cam at all times. As the cam rotates, the lever raises and lowers causing the harness to raise and lower (See Figure 3.8).

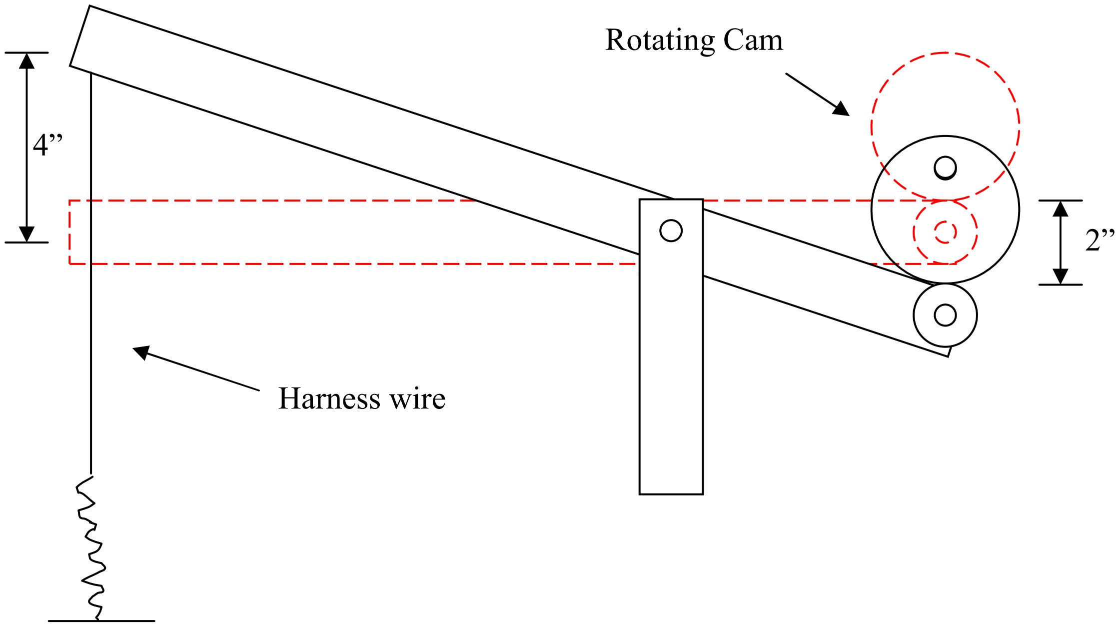


Figure 3.8 The cam lever Setup. The initial position of the lever is denoted by a dashed line and the final position is a solid line.

Section 3.3.2.1 The Cam

The cams were machined out of ½” aluminum plate and have a 2 ½” diameter (See Figure 3.9). There are set screws inside the cam that allow the cam to stay fixed to the shaft. A shaft will be placed off center of the cam in order to create a sinusoidal motion. As the follower end of the lever reaches the point

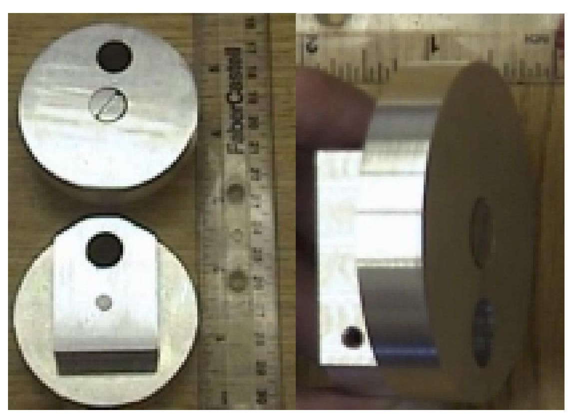


Figure 3.9 Machined Cam

on the cam closest to the shaft, the harness will be at its lowest position. When the lever is at the point on the cam furthest from the shaft the harness will be displaced the full four inches.

Section 3.3.2.2 The Cam Shaft

The shaft (See Figure 3.10) is supported at either end of the frame using bearing mounts. The shaft is a ½" diameter carbon steel rod with a belt wheel attached to one end to interface the motor. The shaft was machined to prevent the belt wheel from sliding. This end of the shaft passes freely through a ½" bearing mount, while the other end of the shaft was machined down to 1/8" to pass through a bearing mount that is 1/8". Machining one end of the shaft to be smaller prevents the shaft from sliding laterally within the bearing mounts.

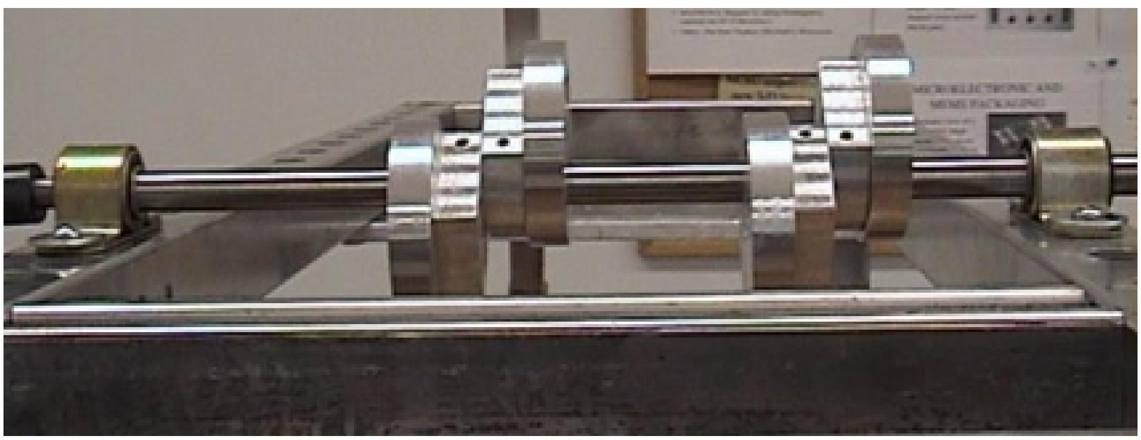


Figure 3.10 The Cam Shaft

The cams slide freely along the shaft allowing them to be adjusted and positioned such that the cam maintains contact with the follower. Once the cam is positioned the set screw within the cam is tightened to prevent the cam from slipping on the shaft. The cams are placed such that each cam opposes another cam, such that the lever motion will be opposing, raising two levers while lower the other two. Opposing the cams also balances the cam shaft, thus creating less of a load on the motor and keeping the system stable.

Section 3.3.2.3 The Cam Levers

Figure 3.11 shows the cam shaft in contact with the levers. The figure shows two levers that are in the up position, and one lever in the down position. The levers are 4 different lengths because of the position of the wire harnesses. The layout of the harnesses is very close to the layout of the actual Jacquard Loom, which places the harnesses diagonally with an offset of ½". The lever has a follower at one end and is attached to a harness at the other end. For simplicity, the follower ends of each lever line up along the same axis so that a single shaft can drive all four. To achieve this, the levers had to be of different lengths (9",

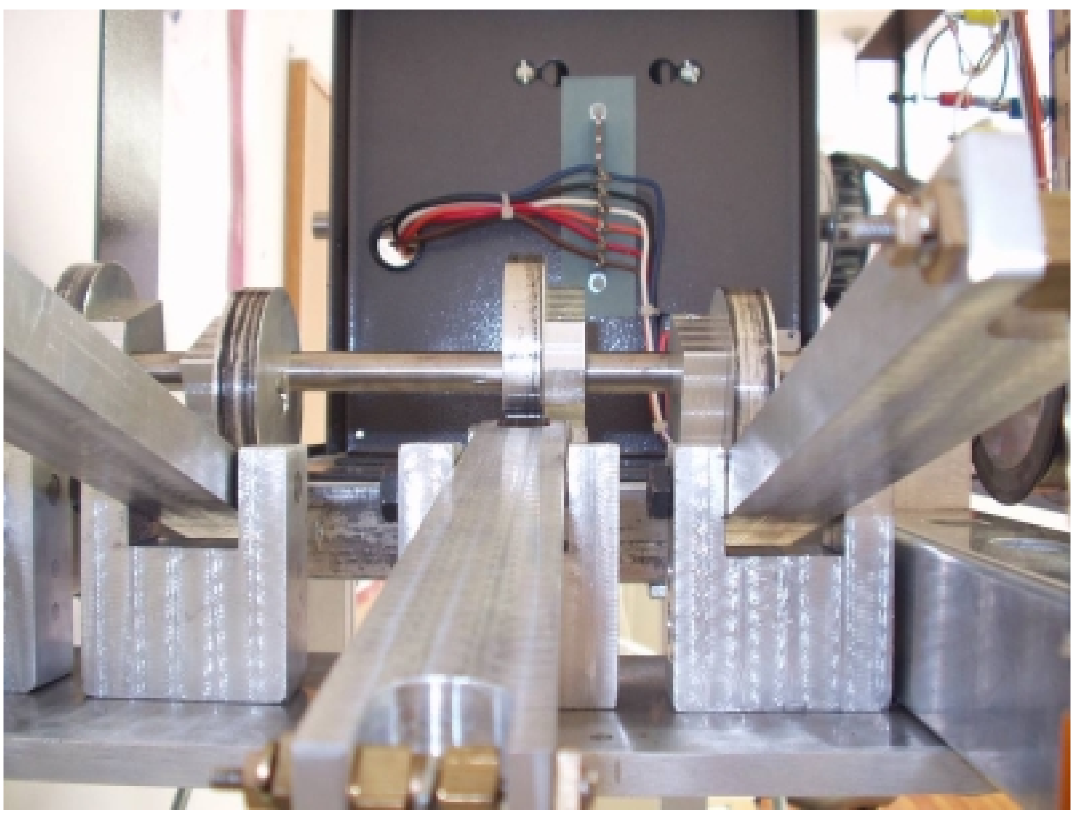


Figure 3.11 A view of the levers.

9.5”, 10”, and 10.5”). Also with different length levers the pivot points of the levers are different. Each lever displaces the harness 4” and is displaced by the cam 2”.

Formulas for determining the lever lengths and pivot point:

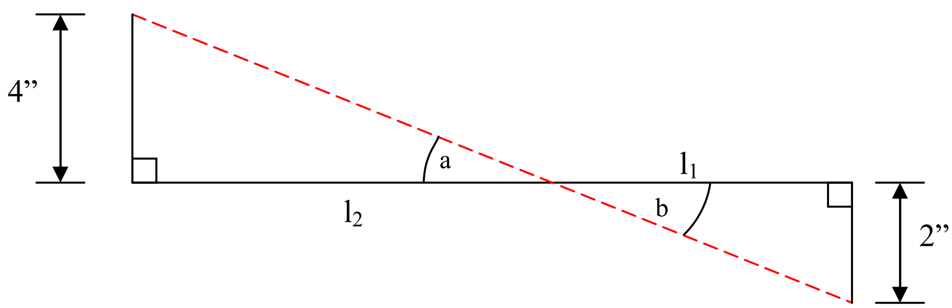


Figure 3.12 Geometry setup for computing lever angles.

From geometry:

Assume: Angle a = Angle b = 45°

The larger triangle (left) is twice as large as the smaller triangle (right):

$l_2 = 2*l_1$

First lever:

Find the minimum length of l_2 needed to displace 4”:

$$l_2 = \frac{4"}{\sin(45°)} = 5.66" \quad \& \quad l_1 = \frac{l_2}{2} = \frac{5.66"}{2} = 2.83"$$

$$l_1 + l_2 = 5.66"+2.83" = 8.49"$$

So, for the first lever, length from follower to harness is approx. 8.5” and length for pivot is 2.83” from the follower.

Second lever:

Since the harness position is ½" further than the first position, $l_1 + l_2$ should equal 9".

$l_1 + l_2 = 9"$

$2 * l_1 + l_2 = 9" \Rightarrow l_1 = 3" \ \& \ l_2 = 6"$

The angle changes:

$$l_2 = 6" \Rightarrow \sin(x) = \frac{4"}{6"} \Rightarrow x = 42^o$$

Third lever:

Since the harness position is ½" further than the second position, $l_1 + l_2$ should equal 9.5".

$l_1 + l_2 = 9.5"$

$2 * l_1 + l_2 = 9.5" \Rightarrow l_1 = 3.1667" \ \& \ l_2 = 6.3333"$

The angle changes:

$$l_2 = 6.3333" \Rightarrow \sin(x) = \frac{4"}{6.3333"} \Rightarrow x = 39^o$$

Fourth lever:

Since the harness position is ½" further than the third position, $l_1 + l_2$ should equal 10".

$l_1 + l_2 = 10"$

$2 * l_1 + l_2 = 10" \Rightarrow l_1 = 3.3333" \ \& \ l_2 = 6.6666"$

The angle changes:

$$l_2 = 6.6666" \Rightarrow \sin(x) = \frac{4"}{6.6666"} \Rightarrow x = 37^o$$

The angles for each lever are different, but the ratio of displacement is the same. The pivot points do move, but the followers are all in line with each other.

Section 3.3.2.4 The Cam Follower

The cam follower (See Figure 3.13) is one of the more important aspects of this system. The follower is a small bearing wheel located at the cam end of the levers. The follower maintains a smooth contact point between the lever and the cam. The follower and the cam must both be perfectly smooth for the system to perform accurately; otherwise there will be small disturbances in the motion.

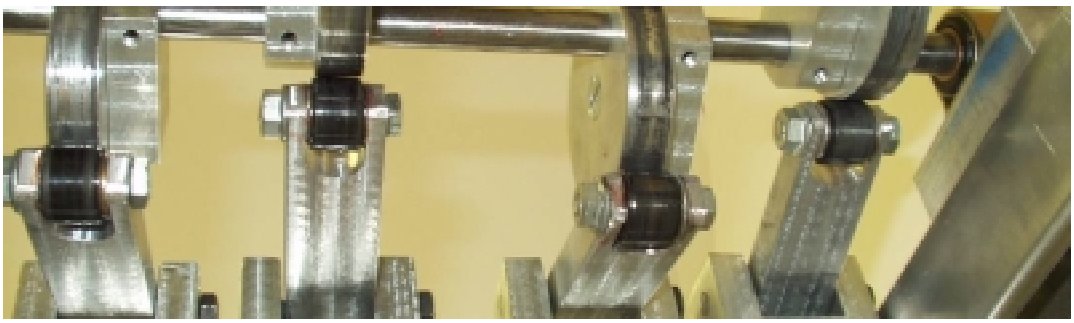


Figure 3.13 The follower is the small black wheel at the end of the lever.

Section 3.3.2.5 Driving the Cam-Shaft

The motor that is used to drive the cams is a DC motor with a maximum input of 120V/3A (See Figure 3.14). The motor is capable of ¼ horsepower and a max rpm (revolutions per minute) of 3600. To couple the motor to the Cam-Shaft a belt system was used, so that the motor can mount directly to the frame instead of hanging off to the right of the frame. The motor has sufficient power to drive the Cam-Shaft with the harnesses attached. Attached to the shaft of the motor is a Tachometer. The Tachometer has an analog voltage output that is proportional to the speed of the motor and has a needle display in rpm. A single revolution of the motor translates to a single cycle (one up and one down) of the harness. The system is considered saturated whenever the cams are rotating too fast for the lever to follow and can lead to skipped cycles, this occurs at speeds over 200 rpm.

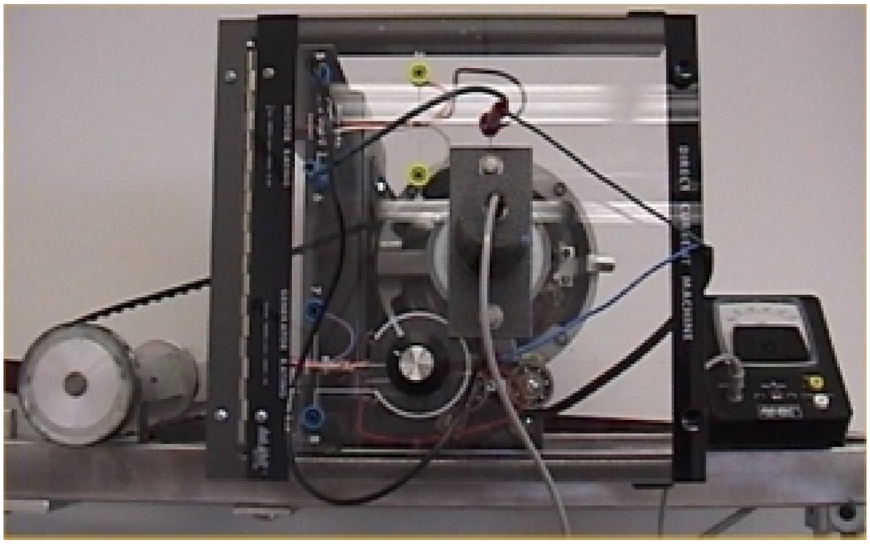


Figure 3.14 The motor (center), the Cam-Shaft (left), and the tachometer (right).

Section 3.3.2.6 Controlling the Harness path

Using levers, the harnesses do not move in a direct straight line up and down. They actually move slightly to the right when the lever is in the top position and slightly to the left when the lever is in the bottom position. This causes an oscillation from the left to the right as the harness moves up and down within the system. This motion can be corrected with a path guide.

The harnesses have an orange sleeve above the eye. This sleeve is used to guide the motion of the harness within the loom. To control the path of the harness, a ½ inch plate of Plexiglas has been installed (See Figure 3.15). The

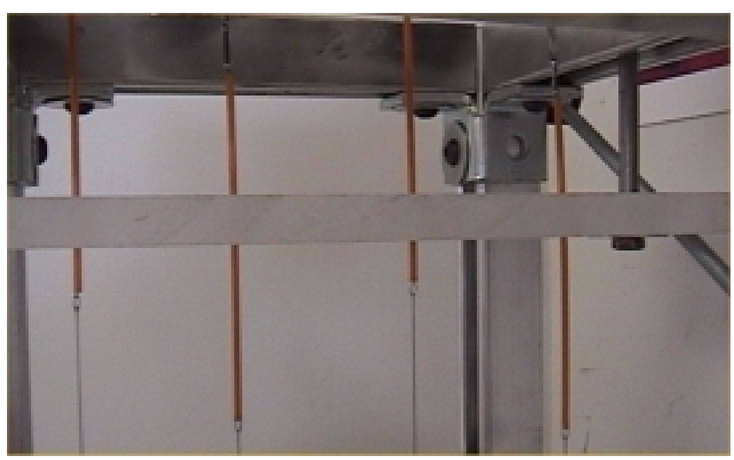


Figure 3.15 Plexiglas guide for harness path.

plate of Plexiglas has holes that are just large enough for the orange sleeve to pass through without any sideways motion. This allows for the section of the harness below the plate, including the eye and the accelerometer, to travel in a straight line and reduces the lateral oscillations.

Chapter 4. Sensing and Communications

Before choosing the sensor, it was important to know which property of the system is to be measured. The given system is contained within a Jacquard loom which consists of a harness which has mass, a warp yarn which has two angles and a spring constant, and a spring attached to the bottom of the harness which has a spring constant. It is assumed that when the warp yarn is removed, such as when a break occurs, the acceleration in the upward motion changes and that the spring constant of the warp yarn will no longer be a part of the system.

Section 4.1 System Analysis

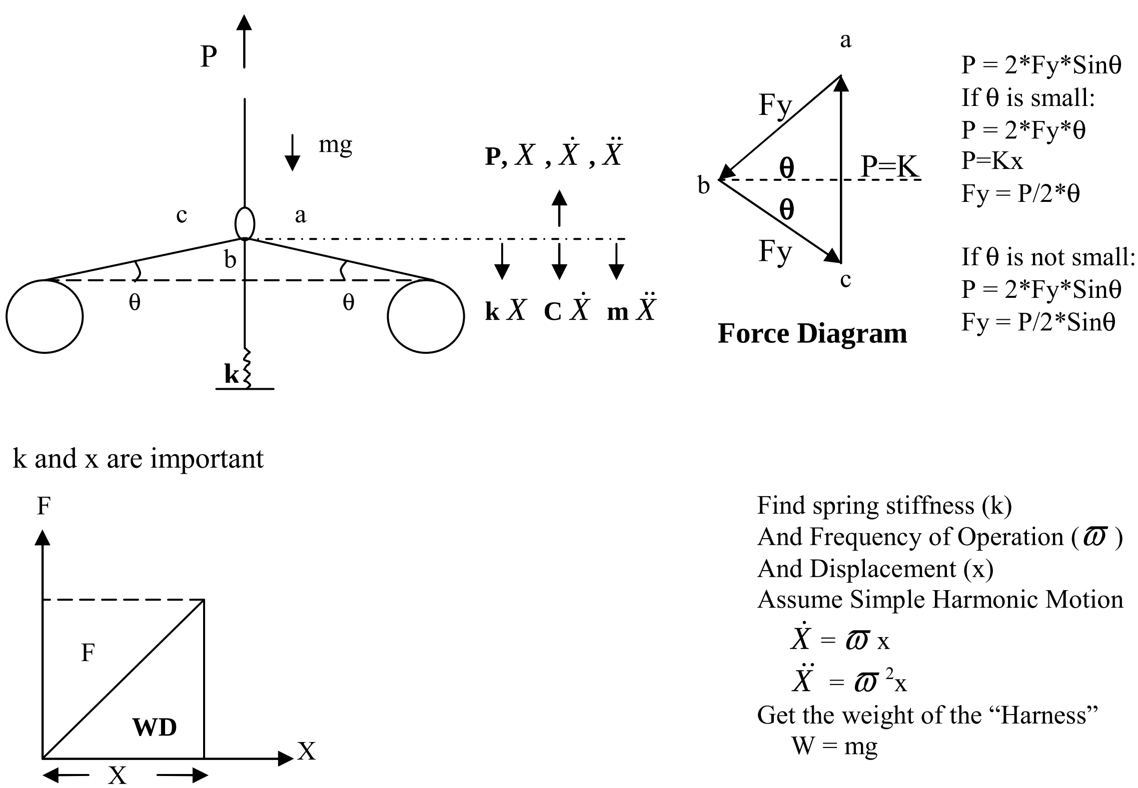


Figure 4.1 Force Diagram and System Analysis

Newton’s 2nd Law:

$$m X + C \dot{X} + k \ddot{X} = P$$

Using Newton’s Law to find the order of the system:

1. If m is small:

   $C \dot{X} + k X = P$
   $(Cs + k) X = P$

   X/P = 1/ (Cs + k)

   The result is a first order system

2. If m is large or if m is small, but X is large:

   (ms2 + Cs + k) X = P

   X/P = 1/ (ms2 + Cs + k)

   The result is a second order system

From the system analysis, it has been determined that acceleration is the most viable attribute for determining if the warp yarn has been broken. Some sensors can measure acceleration directly, and others can derive acceleration from position and/or velocity.

Section 4.2 Harness Properties

The measured properties of the harness are heddle weight (m = 1.6 g) and relaxed spring length (X_0 = .255 m).

From dynamics, the following equations are true:

- Force is F = Load * 9.81 m/s^2
- Displacement of the spring (X_{SPRING}) is $X_1 - X_0$
- Spring Constant (K) is Force/Displacement

Table 4.1 Data for finding the spring constant

Mass	Force	Spring Length (X_1)	Displacement	Spring Constant (K)
0.3 kg	2.943 N	0.034 m	0.0085 m	346.23 N/m
0.4 kg	3.924 N	0.0415 m	0.016 m	245.25 N/m
0.5 kg	4.905 N	0.0485 m	0.023 m	213.26 N/m
0.6 kg	5.886 N	0.0565 m	0.031 m	189.87 N/m
0.7 kg	6.867 N	0.0665 m	0.041 m	167.49 N/m
0.8 kg	7.848 N	0.0775 m	0.052 m	150.92 N/m
0.9 kg	8.829 N	0.094 m	0.0685 m	128.89 N/m

$$K = Slope = \left|\frac{\Delta F}{\Delta D}\right| = \left|\frac{2.943 - 7.848}{.0085 - .052}\right| = 112.76$$

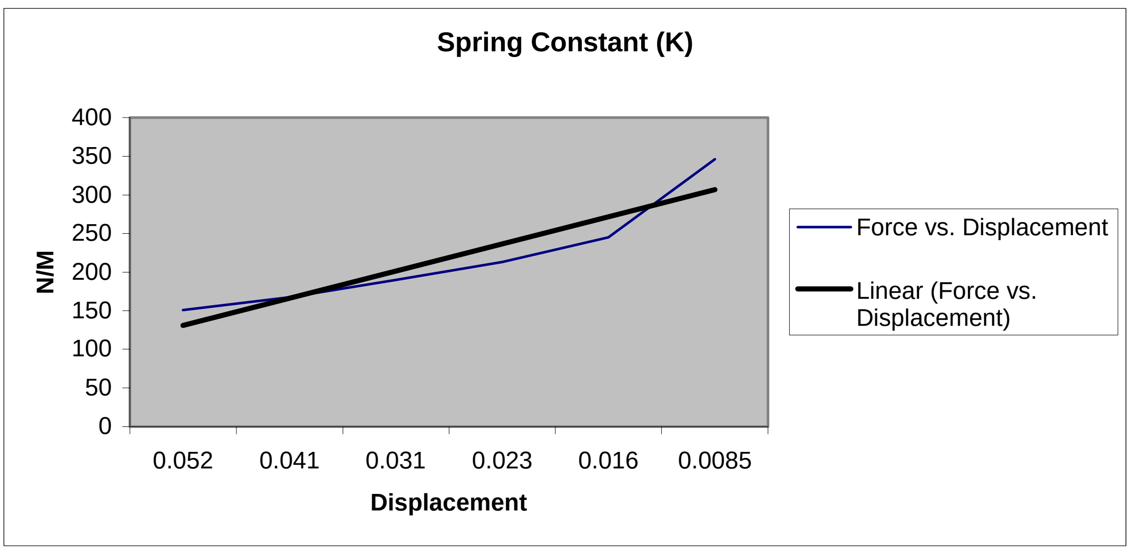


Figure 4.2 Spring Constant of the Harness

Section 4.3 Sensor Requirements

As mentioned in Chapter 3, drop-wires are the most used sensors for detecting a warp break in the Jacquard Loom. Unfortunately, drop-wires are abrasive and heavy and can lead to more warp yarn breaks. Drop-wires are also not addressable, therefore when a warp yarn breaks, a drop-wire will only notify the weaver that a warp yarn has broken, not which warp yarn has broken.

At the beginning of this project, the main desire was to develop a sensor system that is capable of determining when a warp yarn has broken and then report the exact position of the broken warp yarn. This would help speed up the repair process and allow for automated repair. The loom contains many moving parts and warp yarns, leading to certain guidelines for our sensor:

- must be small and unobtrusive, i.e. MEMS
- must be inexpensive
- must be addressable and able to relay position
- must be simple to acquire data

The size of the sensor is of great concern, particularly because it should not interfere with the operation of the loom. The proposed sensor position and design, which will be covered later in this chapter, calls for the sensor to be mounted on the harness. This is difficult since the harness oscillates up and down within the machine and is very close to other harnesses. The sensor must not catch on to the warp yarn nor can it catch on to another harness.

Cost is a big issue in the textiles industry. Is it more beneficial to spend money on replacing a weaver than to just keep the weaver and not update the equipment? There are several thousand separate warp yarns in the loom. Each warp yarn would need a sensor to accurately monitor the loom for warp breaks.

The loom has so many warp yarns to monitor, it can take a while to find a broken warp yarn, especially if the warp yarn has not fallen. A warp yarn can break and stick to the other warp yarns, such that it doesn't appear to be broken, yet the drop-wire still falls and the machine stops. Adding a sensor to every harness would lead to addressability so that whenever a warp yarn breaks, the individual sensor would be able to relay the exact warp yarn position, reducing the time that it takes to find a broken end. This can also lead to an automated tie-in system.

In order to reduce the complexity of the system the sensor must be easy to access and acquire real time data. It is unacceptable if the sensor takes a few extra seconds to determine if the warp yarn has broken, because the loom will continue to run with broken warp yarns.

Section 4.4 Proposed Sensors

With such a large variety of sensors on the market, both large and small, the choice for a sensor becomes very difficult. There are several options for placement of the sensor. One option is to have a sensor that is not connected to the system in any way, i.e. an optical sensor or an LVDT. Another option is to have a sensor that pushes up against the thread, i.e. the drop wires or strain gauges. And the last option to consider is to mount a sensor on the harness, i.e. accelerometer or strain gauge.

Section 4.4.1 An LVDT sensor in the Jacquard Loom

An LVDT is basically a series of inductors in a hollow cylindrical shaft. A solid cylindrical core passes through the shaft (See Figure 4.3). The movement of the core within the shaft produces an electrical output proportional to the position of the core. LVDT's are used in many different types of measuring devices that need to convert changes in physical position to an electrical output. The lack of friction between the hollow shaft and the core prolong the life of the LVDT and enable very good resolution. In addition, the small mass of the core allows for good sensitivity in dynamic tests.

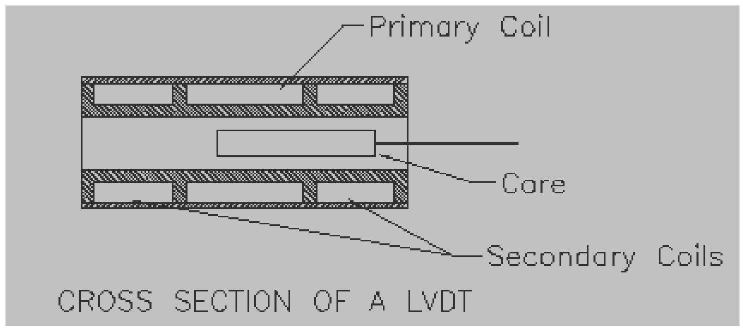


Figure 4.3 A cross section of a LVDT

The LVDT may also be calibrated by varying the position of the core and measuring the corresponding output voltages. Then a calibration curve or calibration constant may be determined and applied to give a change in position. Given the change in position and the time for the change, velocity can be obtained. Take the derivative of the velocity and get acceleration. The acceleration of the

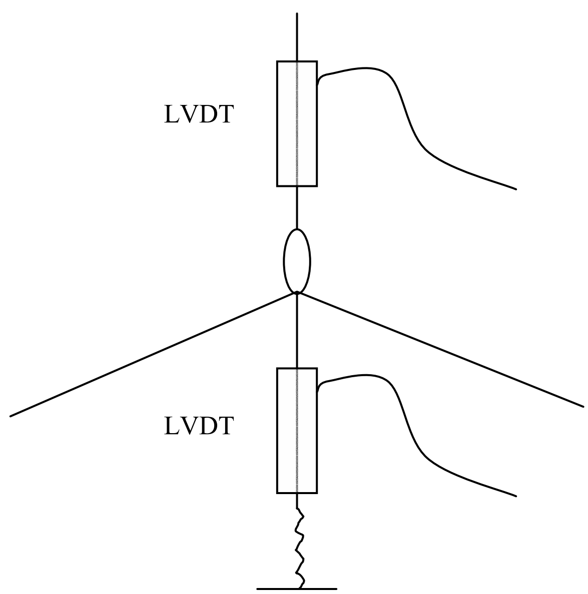


Figure 4.4 LVDT placement on the harness.

harness can be used to determine if a warp yarn has broken or not.

The core of the LVDT can be mounted on the harness and the hollow shaft can be mounted in the loom such that the core is allowed to pass through (See Figure 4.4). The placement of the LVDT along the harness can be either above or below the eye.

Section 4.4.2 A Load Cell sensor in the Jacquard Loom

A common load measuring device is the load cell. Load cells have been used to measure both strain and weight. This sensor commonly consists of a set of strain gauges arranged strategically to measure strain from different directions. A strain gauge uses a grid of fine wire or metal foil contained in a thin resin backing (See Figure 4.5). The gauge is then attached using a thin layer of epoxy which acts as a carrier matrix to transfer the strain in the specimen to the strain gauge. The cross-sectional area of the gauge will increase for

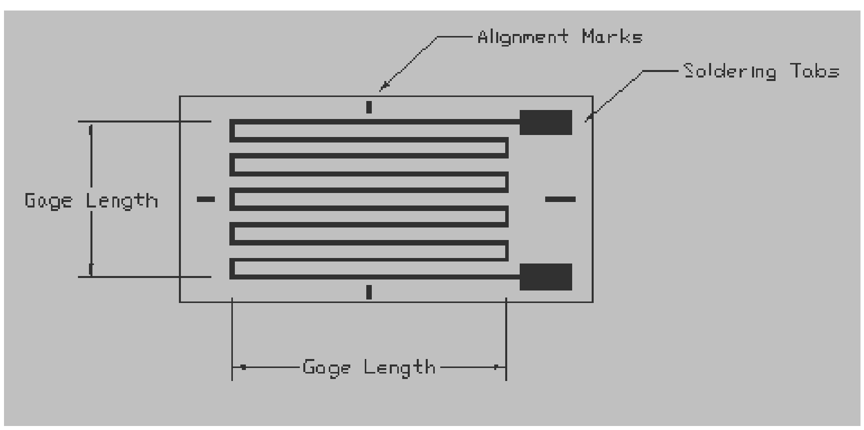


Figure 4.5 The Strain Gauge

compression and decrease in tension. Because the wire has an electrical resistance that is proportional to the inverse of the cross-sectional area, $R \alpha 1/A$, a measure of the change in resistance will produce the strain in the material.

There are two common types of load cells, a beam type and an axial type. The beam type load cell consists of strain gauges attached to a beam that is allowed to bend. The strain gauges are measuring the bending stresses of the beam which is proportional to the load. The axial type load cell consists of a hollow or solid cylindrical shaft and four strain gauges mounted around the circumference to measure the shear stress along the shaft. The basic relationships of: Stress = Load/Area (σ=P/A) and Strain = Stress/Young's Modulus ($\varepsilon = \sigma/E$) can be used to determine the strain under different loads (See Figure 4.6).

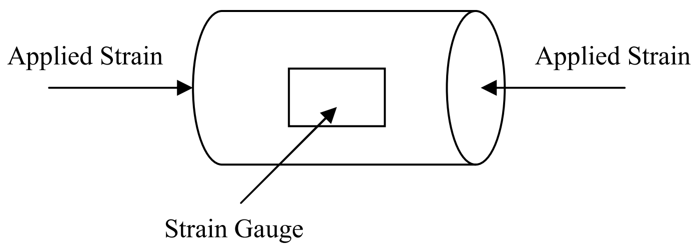


Figure 4.6 Load cell under applied uni-axial strain

There are a few possible ways to use the load cell or a strain gauge in the Jacquard Loom to determine if a thread has broken. The first and most obvious is to place the load cell such that the thread is pressing against it. This would measure the compression of the thread against the load cell and relate the tension of the thread (Gahide, 2001). This approach has already been tried and did not seem a viable solution. Another approach is to place the load cell in line with the harness such that the

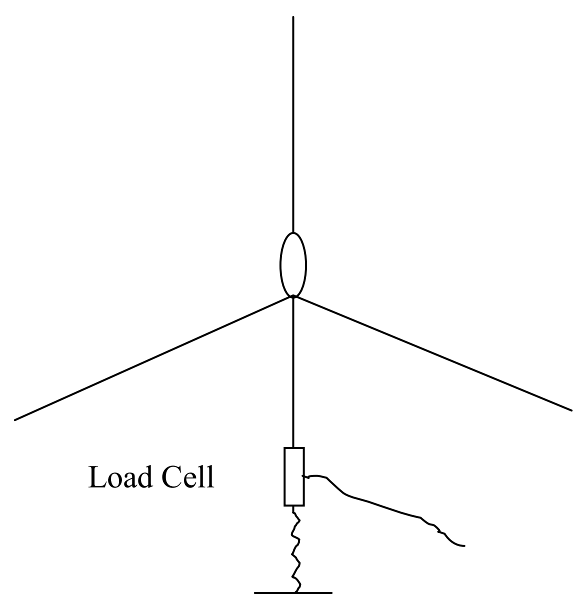


Figure 4.7 Load Cell placement on the harness.

harness will be applying tension to the cell. The load cell would be located in line with the spring (See Figure 4.7). This placement would put the load cell in a position to measure the tension of the spring. The spring tension will change when a thread is not present.

Section 4.4.3 An Accelerometer sensor in the Jacquard Loom

An accelerometer is typically viewed as a mass-spring transducer housed in a sensor case. The sensor case is attached to a moving part whose motion is inferred from the relative motion between the mass and the sensor case. It turns out that the displacement of the mass is directly proportional to the acceleration of the case and therefore the moving part.

At its most basic level, an accelerometer can be viewed as a classical second order mechanical system, which is a damped mass-spring system under an applied force. When the accelerometer experiences acceleration with a component parallel to its sensitive axis, the accelerometer's proof mass develops a corresponding inertial force $f = ma$. This force acts on and displaces the spring a distance $x = f/k$ where k is the spring constant. The sensor's output is related either to the spring's displacement or to the spring's internal force, both of which are proportional to the applied acceleration.

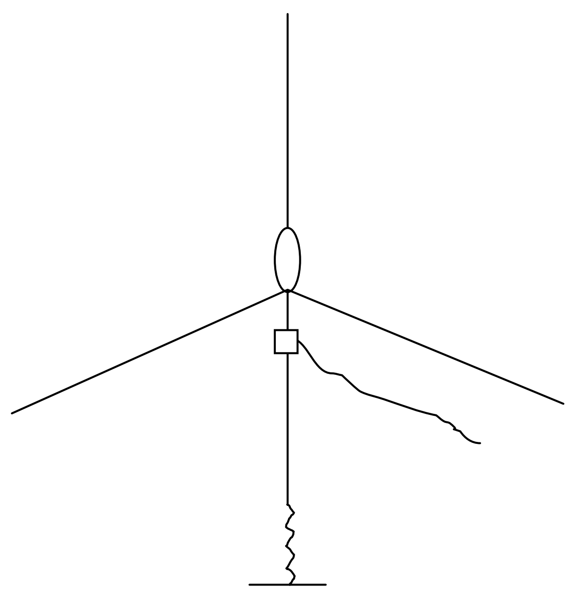


Figure 4.8 Accelerometer placement on the harness.

An accelerometer could be placed on the harness such that the upward acceleration could be measured (See Figure 4.8). The acceleration will change based on the spring constant k, which differs depending on whether or not a thread is present. The change can be measured and a determination about the status of the thread can be made.

Section 4.4.4 Conclusions about the sensors

Due to its small size and relatively inexpensive cost, the accelerometer seems like the best choice for the Jacquard Loom system. LVDT's would be difficult to implement in the system and are very expensive. Load Cells have been tried before (Gahide, 2001) and may be too difficult to implement in the current system.

Section 4.5 The Accelerometer

The chosen accelerometer is an ADXL202AE dual-axis accelerometer developed by Analog Devices (See Figure 4.9). Analog Devices have been known to create high-performance low-cost accelerometers for years. The ADXL202AE is their first MEMS accelerometer and at the beginning of this project was available only as a prototype. This accelerometer has been determined to be the best for the harness system because if it's characteristic size and light weight, so it will not affect the dynamics of the harness. The ADXL202AE is available in 5mm X 5mm X 2mm 8-lead hermetic LCC package.

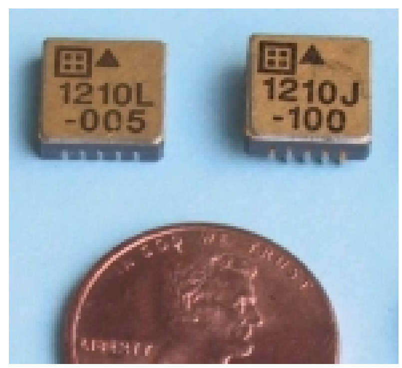


Figure 4.9 The ADXL202AE MEMS accelerometer

The ADXL202AE is a very versatile sensor in that both acceleration and tilt can be measured using the same chip. This sensor has low power consumption and both analog and digital output. The ADXL202AE can measure both static acceleration (gravity) and dynamic acceleration (vibration). The full-scale range of the accelerometer is +/- 2 g.

The bandwidth and duty-cycle of this accelerometer are fully adjustable by adding filter capacitors. This filtering improves measurement resolution and helps prevent aliasing, meaning that the sensor resolution can be adjusted to reduce noise. The adjustable resolution also allows the sensor to either filter or measure low accelerations due to vibration. Figure 4.10 shows a functional block diagram of the system within the ADXL202AE accelerometer.

The ADXL202AE consists of a series of beams micro-machined onto a piece of silicon wafer. The acceleration is directly proportional to the deflection of the beams. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and central plates attached to the moving mass. The fixed plates are driven by 180° out of phase square waves. Acceleration will deflect the beam and unbalance the differential capacitor, resulting in an output square wave whose amplitude is proportional to acceleration.

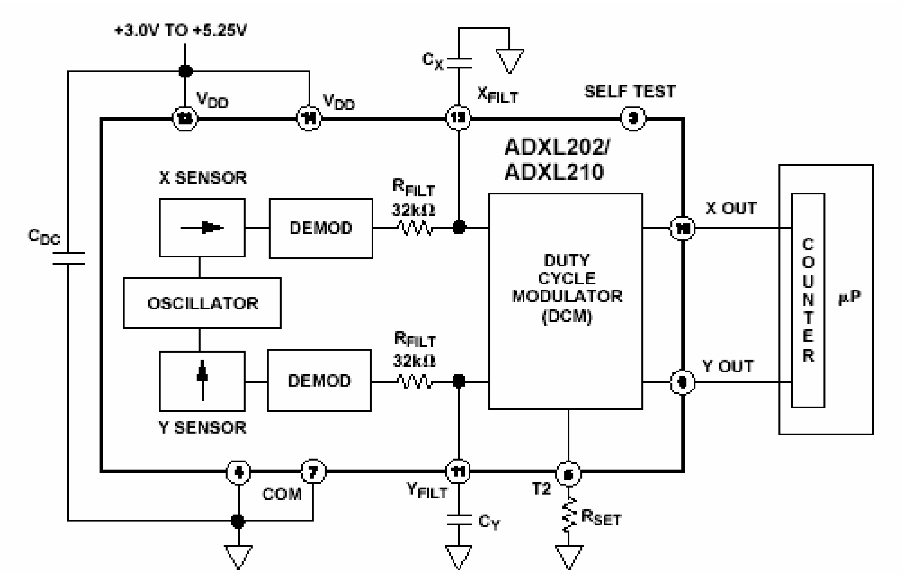


Figure 4.10 Functional block diagram of the ADXL202AE Accelerometer. (ADXL202AE Data Sheet)

Section 4.5.1 Accelerometer Theory

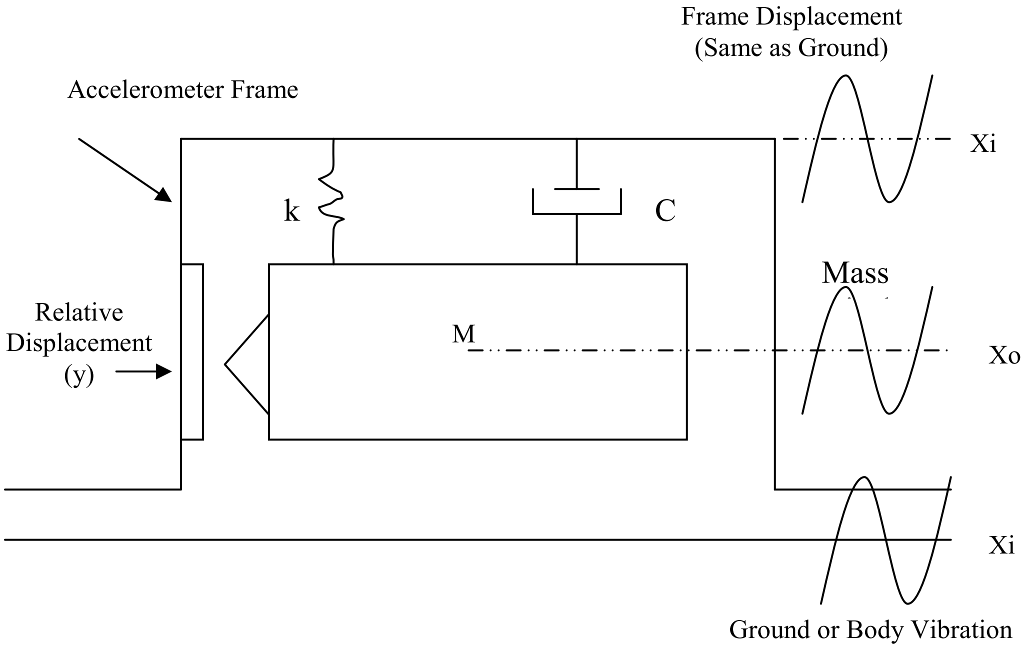


Figure 4.11 A description of common accelerometer properties.

Xi = Ground Displacement
Xo = Mass Displacement
Y = Scale Reading = Relative Displacement = (Xo – Xi)
Mass M

From Newton's 2nd Law:

ΣForces = 0

$$Mo + C(\ddot{X}o - \dot{X}i) + k(Xo - Xi) = 0$$

Substitute for Y

$$M\left(\ddot{Y} + \ddot{X}\right) + \dot{Y}C + KY = 0$$

Assume Simple Harmonic Motion

$$\left[\left(k - \omega^2 M\right) + i\omega C\right]\hat{Y} = \omega^2 M\hat{X}i$$

$$\hat{Y} \big/ \hat{X}i = \frac{\omega^2 M}{(k - \omega^2 M) + i\omega C}$$

Divide M

$$\hat{Y}/\hat{X}i = \frac{\omega^2}{(\omega n^2 - \omega^2) + i2\zeta\omega n\omega}$$

Divide ω_n^2

$$\hat{Y}/\hat{X}i = \frac{(\omega/\omega n)^2}{(1-(\omega/\omega n)^2)^2 + i2\zeta\varpi/\omega n}$$

Take Modulus

$$\left|\hat{Y}/\hat{X}i\right| = \frac{(\omega/\omega n)^2}{\sqrt{(1-(\omega/\omega n)^2)^2 + 4\zeta^2(\omega/\omega n)^2}}$$

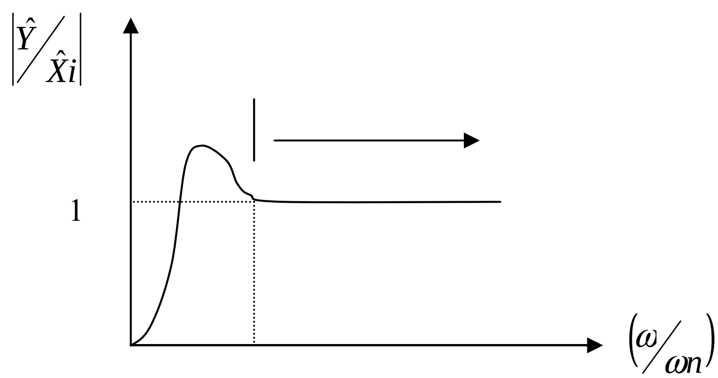


Figure 4.12 Working range for an accelerometer

Section 4.5.2 Building a circuit for the Accelerometer

Before designing a circuit for the ADXL202AE accelerometer, it is important to note certain properties. First, the accelerometer is so sensitive that it can pick up vibrations due to through-hole components on the circuit board, such as resistors and capacitors. This can be corrected by only using surface mount components, which are ideal because they will reduce the overall size of the circuits. Also weight needs to be kept down so the circuit should only have the components that need to be close to the accelerometer. There will be wires leaving the circuit and some components can be at a remote location, but increasing the distance of the components from the circuit can cause the accelerometer circuit to be noisier.

Section 4.5.2.1 Pin Configuration for the ADXL202AE

The following is a description of the pin out for the ADXL202AE which was included in the data sheet (See Figure 4.13 and Table 4.2).

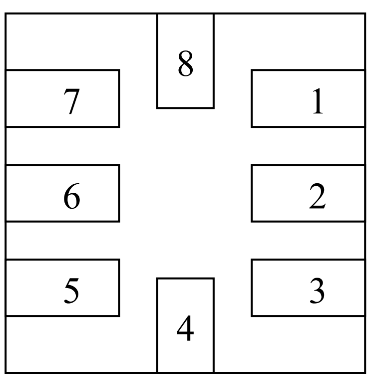


Figure 4.13 Pin Layout

Table 4.2 Pin-outs for the ADXL202AE

Pin No.	Name	Description
1	ST	Self-Test
2	T2	Connect RSet to Set T2 Period
3	COM	Common or Ground
4	YOut	Y-Channel Duty Cycle Output
5	XOut	X-Channel Duty Cycle Output
6	YFilt	Y-Channel Filter Pin
7	XFilt	X-Channel Filter Pin
8	VDD	3 V to 5.25 V

Pin 1, or ST, is a Self-Test pin. This pin is there to test the functionality of the accelerometer chip. Normally this pin is left either open or grounded. Applying a logic "1" or 5 volts to the ST pin causes the chip to apply a deflection voltage to the inner structure of the accelerometer which is equal to approximately -5 g of force. Activating the ST pin and measuring the resulting output, a user can verify that the electronics within the accelerometer are working correctly.

Pin 2, or T2, is there to set the duty cycle of the accelerometer. There are two important advantages to changing the duty cycle. The first is the ability to reduce the noise level of the output. The second is to change the resolution of the output based on the input. Changing the resolution can lead to faster measurements of acceleration. To ensure that it is capable of taking measurements at the desired speed, it is important to know the limitation of the microprocessor before choosing a duty cycle. The duty cycle is changed by adding a resistor connected from ground to the T2 pin.

Pins 4 and 5 are the outputs from the device. Using these pins, the output will be in a duty cycle format.

Pins 6 and 7 serve two functions. The first is to set the bandwidth of the output. To set the bandwidth a capacitor is connected from ground to the pin. Having a separate pin for both X and Y allows the user to set different bandwidths for each output, but they will both still have the same duty cycle. Decreasing the bandwidth reduces noise and improves the resolution, but smaller accelerations may be filtered out. The second function involves the use of the analog output. Leaving Pins 4 and 5 unconnected and connecting the output to these pins results in an analog output such that the voltage level increases as the acceleration increases. Using the analog option still requires the filter capacitors.

Section 4.5.2.2 Circuit Design

The circuit design is based on the circuit used for the Analog Devices ADXL202EB, which is the evaluation board for the ADXL202 series of accelerometers. The board itself allows a user to swap resistor and capacitor values to determine the best results for their particular application. The evaluation board also provides a direct serial interface to a computer and software to analyze the data from the accelerometer. Unfortunately the evaluation board is quite large and cannot be mounted on the harness. This led to the design of a small circuit that could be mounted. The circuit interfaces with a microprocessor that relays the data back to a computer.

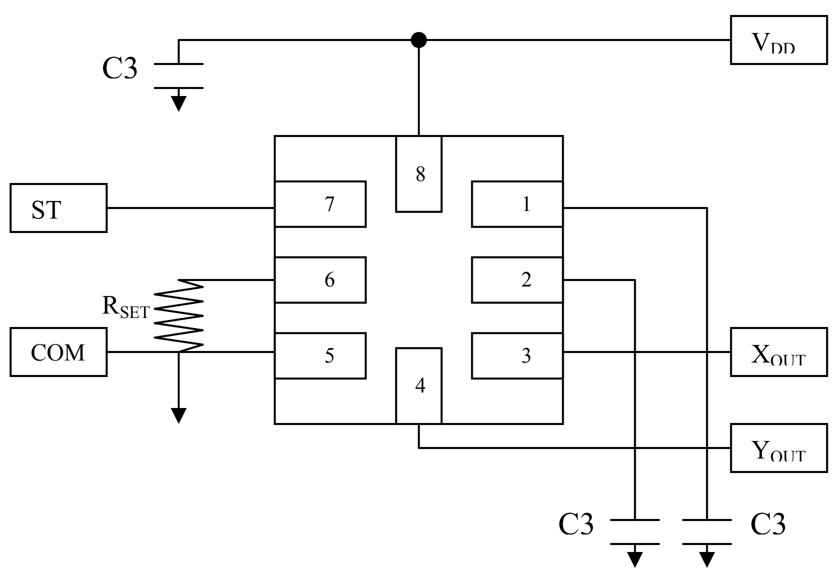


Figure 4.14 Circuit Layout

The circuit was designed and prototyped based on Figure 4.14. The circuit is also set up to give a duty cycle output rather than the analog output. The values for R, C1, C2, and C3 were chosen based on the high speed data acquisition needed for the Jacquard Loom system. R was chosen to be 250 KΩ so that a 2 ms duty cycle would be output by the accelerometer. C3 is a voltage source noise filter, set to the standard value of .47 μF. C1 and C2 were both set to a value of .47 μF to give a bandwidth of 10 Hz and lower noise.

Section 4.5.2.3 Circuit construction

After testing the circuit, the design was put into CirCAD (See Figure 4.15), a circuit authoring program that allows users to design circuits to be developed professionally. The red tabs at the top allow access to the following pins (from left to right): ST, Common (GND), VDD (+5V), Yout, and Xout. The tabs will be connected to the microprocessor using stranded wire. The stranded wire is very flexible and less susceptible to breaks.

The size of the circuit is 8.6 mm X 12 mm. Depicted in Figure 4.15, C1 represents the voltage source filter, C2 and C3 are the X and Y filters and R1 represents the Rset. The yellow box in the lower center of the drawing represents the ADXL202AE chip. The blue

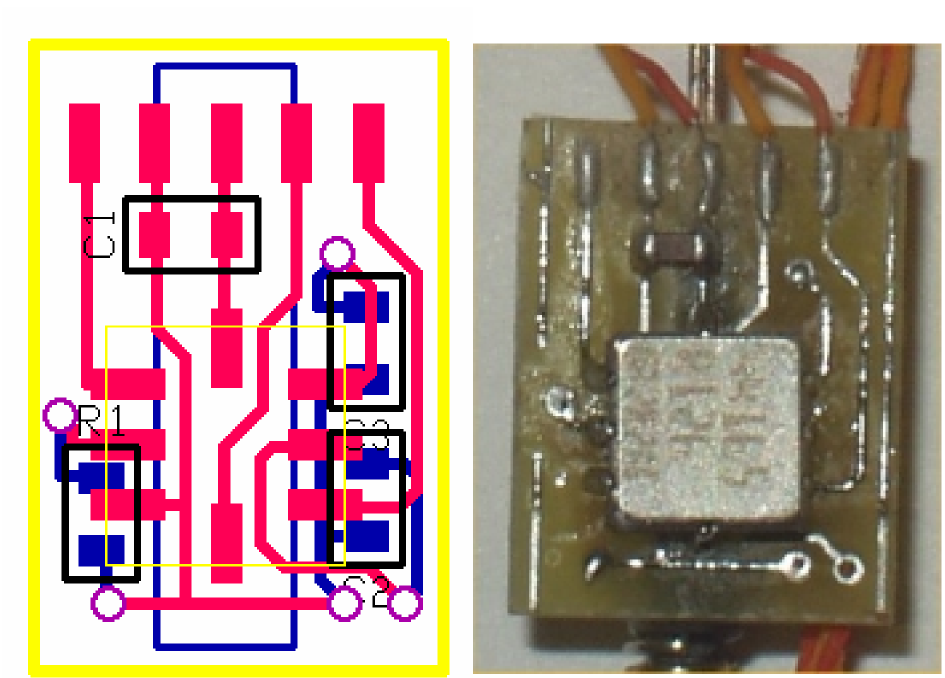


Figure 4.15 CirCAD drawing (left) and picture (right) of the Accelerometer circuit

rectangle in the center represents the place on the harness where the circuit will be mounted. This area of the circuit will be hollowed out to allow a better contact with the harness.

Section 4.5.3 Mounting the circuit on the harness

Since the accelerometer has two axes (See Figure 4.16), it is important to align at least one axis where the axis is perpendicular to the plane of gravity. The circuit is placed such that the X axis is in line with the harness and is perpendicular to the plane of gravity.

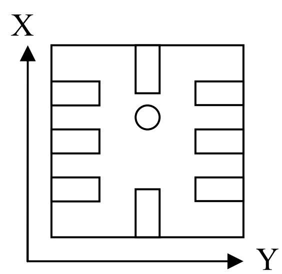


Figure 4.16 Axes of the Accelerometer

The X output seen from the accelerometer in gravities will be approximately 1 g and the Y output will be approximately 0 g. This output is because the gravity of the Earth will be acting on the mass within the accelerometer.

Section 4.5.3.1 Attaching the circuit to the harness

As with all accelerometers, optimum performance depends on proper mounting of the device. It must be mounted such that the sensor is properly coupled to the object for which acceleration is to be measured. Improper mounting can increase the effects of mechanical resonance. Careful design consideration must be applied or some resonance from the components may be introduced resulting in false data. In Section 4.5.2.3, a small circuit was described. This circuit is large compared to the ADXL202AE chip, but can be mounted very easily. Mounting the ADXL202AE chip (i.e. separate from the circuit) to the harness is ideal as far as size goes, however that would double the number of wires leaving the harness.

The best place to mount the circuit on the harness is at the plastic connector just above the spring (See Figure 4.17). This is the best place because it is below the eye and is fixed to the harness, unlike the plastic sleeve. The spring itself is not a good place to mount because the spring is expanding and contracting. Below the spring is not a good place because it does not move relative to the machine.

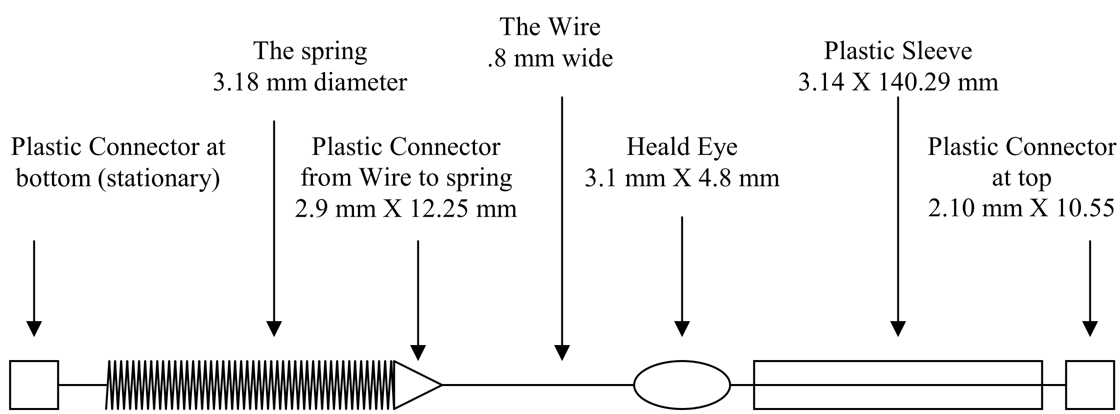


Figure 4.17 A view of the harness

Section 4.5.3.2 The accelerometer mounted

The circuit is attached to the plastic section of the harness just above the spring using super glue (See Figure 4.18). Once attached, the circuit will be immovable compared to the harness. This connection will cause the frame of the chip to be part of the harness and the accelerometer will get an accurate reading.

Since the sensor is not wireless, wires are required to provide power to the sensor and relay the data back to the microcontroller. To prevent extra oscillations from the wires in the Y direction, the wires are wrapped around the spring section in a “coil” fashion. When the harnesses displace upward 4 inches, it is best to have a “coil” so that the wire may extend like a spring. The wires lead from the top of the circuit and are wrapped in electrical tape around the harness, this allows for a much stronger hold than just the solder connection. This will help prevent the wires from detaching if they were to snag on something within the machine. The wires lead from the circuit to a header that can be directly connected to the microprocessor.

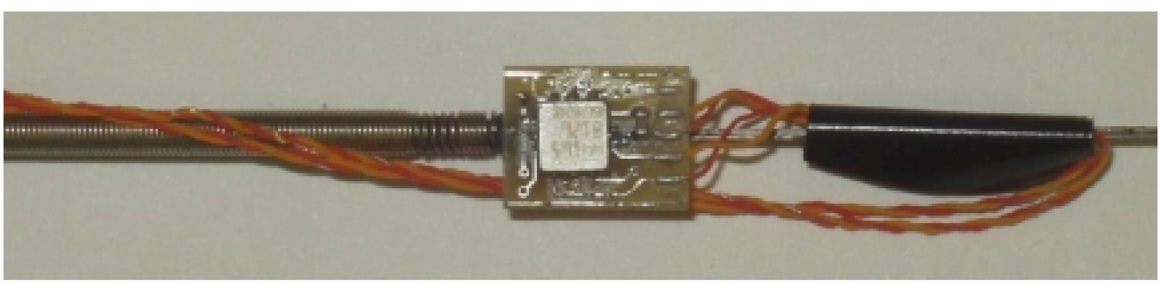


Figure 4.18 The accelerometer circuit mounted

Section 4.6 Acquiring data from the accelerometer

The ADXL202AE accelerometer was specifically designed for use with low cost microcontrollers. After a search on the internet, it was determined that the best microcontroller for use with the accelerometer was the BasicX BX-24 (See Figure 4.19). The BasicX website provided several pieces of sample code, including programs to display the acceleration and tilt of a single ADXL202 accelerometer.

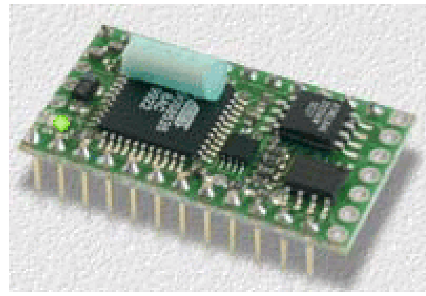


Figure 4.19 The BasicX BX-24 Microcontroller

The BasicX BX-24 microcontroller has several outstanding features for such a small low cost microcontroller. The BX-24 has a total of 16 standard I/O pins, 8 of which can be used as Analog to Digital Converters (ADC). There are 400 bytes of available memory and the processor can handle up to 65,000 lines of code per second. A 5 V regulator is also included on board to allow the use of a range of voltages from 3.3 V to 18 V. The BX-24 microcontroller can be used on an evaluation board providing easy access to the BasicX's I/O pins as well as power and ground (See Figure 4.20). The evaluation board also provides a direct serial connection to a desktop PC which is necessary to program the BX-24 and receive data.

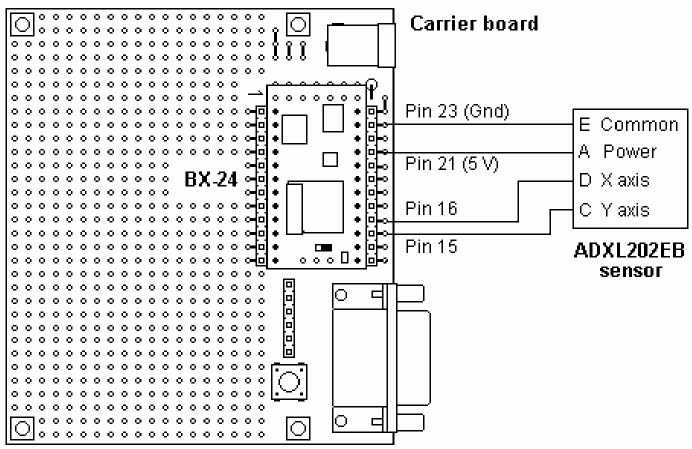


Figure 4.20 Connecting the ADXL202 to the BX-24 evaluation board

For the harness system, only the X output is desired, so one BX-24 chip can be used with 16 separate accelerometers. There are a total of 4 wires leading from the accelerometer to an interface board (See Figure 4.21). On the interface board are two separate headers, one header has screw-down fasteners to allow wire leads to be attached and detached and the other header allows the interface board to be plugged into a prototype board (See Figure 4.20). The prototype board has headers that are connected by ribbon cables to the headers on the evaluation board of the microcontroller. The quick detachment and attachment is necessary for the sensor to be easily replaced or repaired.

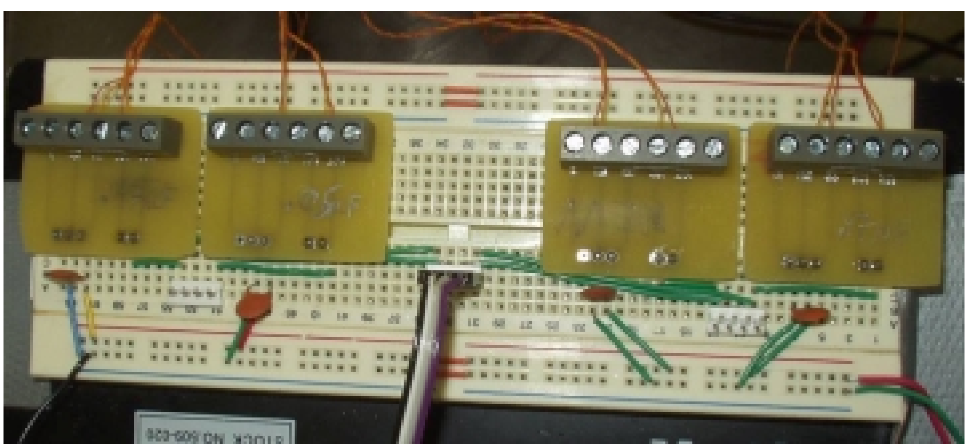


Figure 4.21 Interface board for the Accelerometers

In Figure 4.21, the light green circuit boards are the individual interface boards. The orange and red wires leading in from the top of the pictures are from the accelerometers. The interface boards are placed on a prototype board and power is supplied from a nearby power supply. The 4-wire ribbon connector in the center of the board attaches the four interfaces to the BX-24 chip.

Section 4.7 Acquiring the Data

Acceleration at the ADXL202AE may be calculated using the following formula:

$$Acceleration\,(in\ g) = \frac{Duty\ Cycle - Duty\ Cycle\ at\ Zero\ g}{Duty\ Cycle\ per\ g}$$

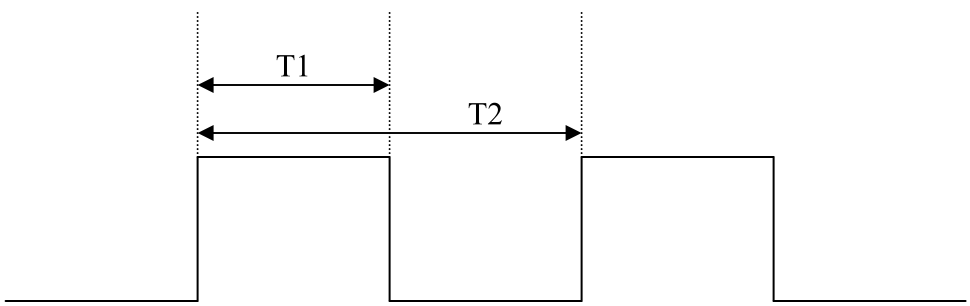


Figure 4.22 Acquiring time constants from the PWM signal.

In a PWM signal (See Figure 4.22), the duty cycle is the ratio of time that the pulse is high (T1) compared to the period of the pulse (T2). So, duty cycle is (T1/T2). The nominal values for an acceleration of 0g's the duty cycle is approximately 50% and for each g of acceleration the duty cycle changes 12.5%. These values are obtained when the accelerometer is placed with both the X and the Y axis parallel to the plane of gravity. Using these as constants, the above equation changes to the following:

$$Acceleration\,(in\ g) = \frac{(\mathrm{T1/T2})\ - 50\%}{12.5\%}$$

Section 4.7.1 Calibrating the Accelerometer

The constants determined above were supplied in the technical data sheets for the ADXL202AE accelerometer. These values are acquired when the accelerometer is laid flat such that the X and Y axis are parallel to the plane of gravity. For the harness system, the X axis is perpendicular to the plane of gravity. Calibration of the sensor is based on the 0 g, 1 g, and the -1 g duty cycles. To measure acceleration accurately, the duty cycle must be accurate. An oscilloscope was used to determine the duty cycle of the pulse.

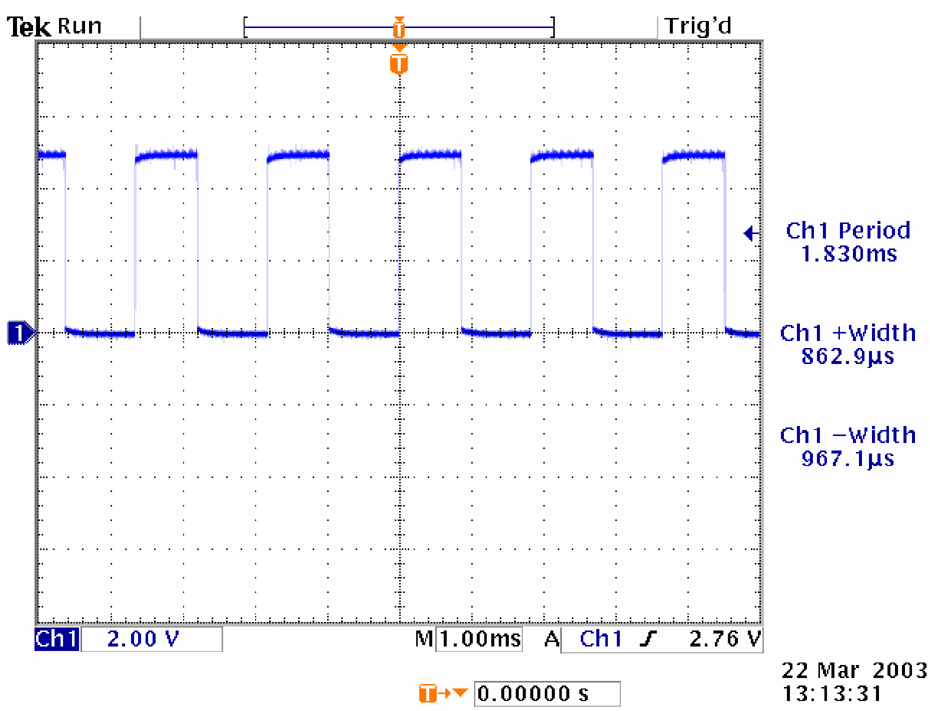


Figure 4.23 Oscilloscope output when X axis is parallel to the plane of gravity.

In Figure 4.23, in the oscilloscope output, the X axis is parallel to the plane of gravity and it can be calculated that the duty cycle (+width/period) is 48%, which is very close to the estimated value of 50%. The period can be found two ways, using the oscilloscope period (1.830 ms) or adding the + width (862.9 µs) to the - width (967.1 µs) to get the same result. Derived from the oscilloscope:

The 50% duty cycle is approx. 48%.

T2 or the period is 1.830 ms.

T1 or the + pulse width is 862.9 µs.

To calculate the gravities seen at this orientation of the sensor use the above equations, and the calculated value of 48% and the nominal value of 12.5 %:

((.8629/1.830) - .48)/.125 = -.07 g ~ 0 g

Since the X axis is parallel to gravity, the output is approximately 0 g, as expected.

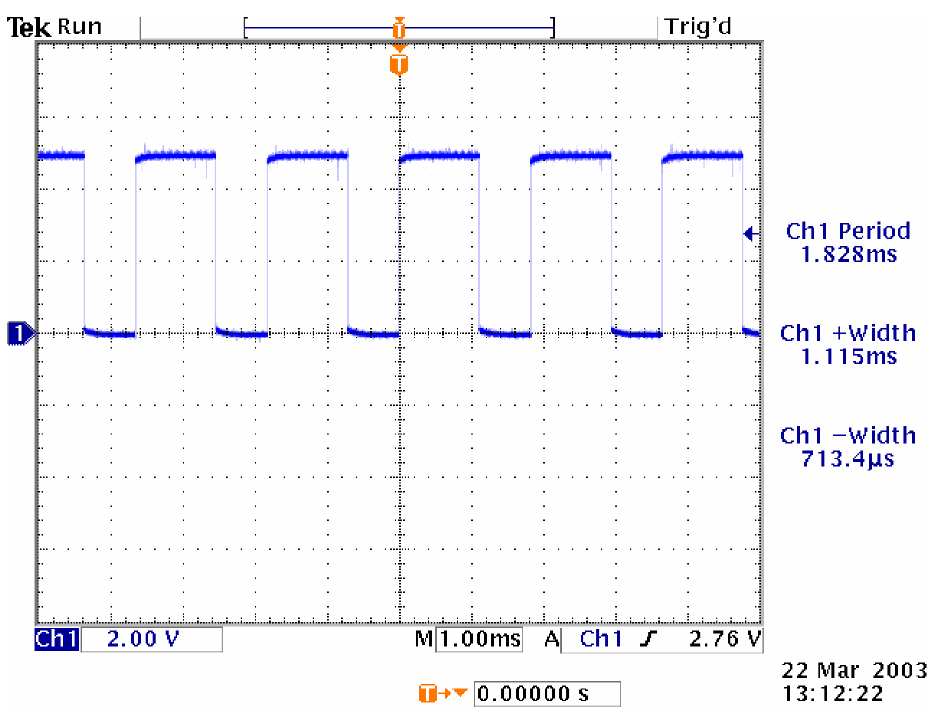


Figure 4.24 Oscilloscope output when X axis is perpendicular to the plane of gravity in the positive X direction.

In Figure 4.24, the X axis is now perpendicular to the plane of gravity and the duty cycle is now about 61%. To calculate the gravities seen at the orientation, we use 48% for the 50% duty cycle and the nominal value of 12.5%.

Derived from the oscilloscope:

T2 or the period is 1.828 ms.

T1 or the + pulse width has become 1.115 ms.

To calculate the gravities seen at this orientation of the sensor use the above equations:

((1.115/1.825) - .48)/.125 = 1.13 g

This should equal 1 g, but the nominal value of 12.5% may not apply here. Since the nominal value of 12.5% doesn't give the expected output, the value must be determined. The output of the accelerometer when X axis is in the positive direction should be 1 g, so the output when the X axis is in the negative direction should be -1 g.

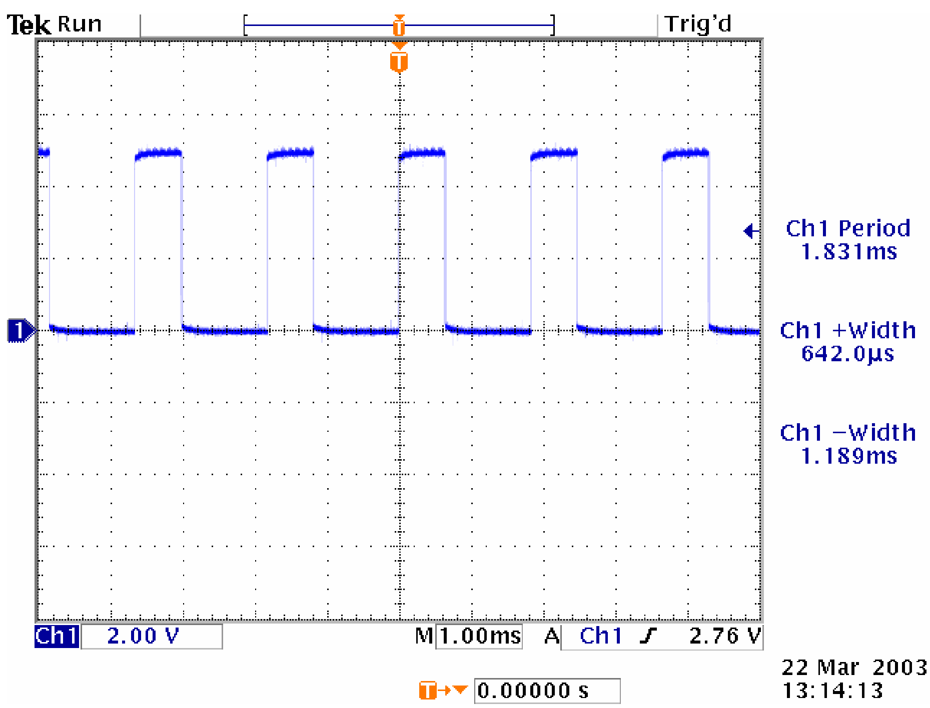


Figure 4.25 Oscilloscope output when X axis is perpendicular to the plane of gravity in the negative X direction.

Using the duty cycles for 1 g (Figure 4.24) and -1 g (Figure 4.25), the change per gravity can be calculated by subtracting the negative X axis duty cycle from the positive axis duty cycle and then dividing by 2.

$((+T_1/T_2)-(-T_1/T_2))*.5 = \text{change per g}$

$((1.115/1.828)-(.642/1.831))*.5 = .129$

This means that at 1 g, the change is 12.9 %. Recalculate to get the output in g:

$((1.115/1.828) - .48)/.129 = 1.01 \text{ g} \sim 1 \text{ g}$

This calibration is done for each accelerometer and can be tested by monitoring the output of the accelerometer when it is not moving. The alignment within the loom should generate an output of approximately 1 g.

Section 4.7.2 Decoding the output using the microcontroller

The BasicX has a very specific command (PulseIn) to measure the pulse of an incoming signal. PulseIn measures the time that the pulse is either "high" or "low". A strategy (See Figure 4.26) to measure the change in acceleration using the BX-24 is to measure the period (T2) first and then measure the time that the pulse is "high" (T1). PulseIn can be used to measure the period of the pulses received from the accelerometer. First find the time that the pulse is "high" and then find the time that the pulse is "low". Add them to get the period of the pulse. The

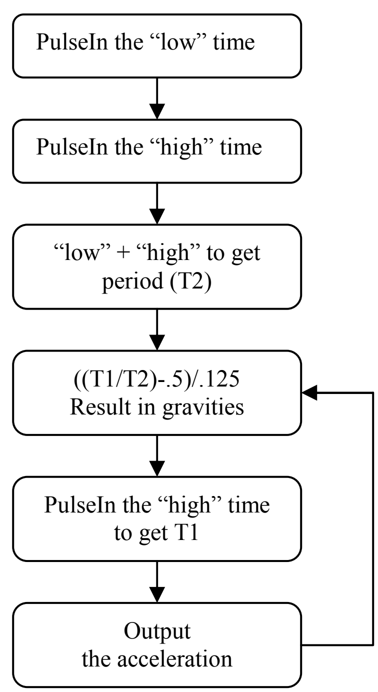


Figure 4.26 Flowchart for decoding the output

period doesn't change unless the temperature changes, so this does not have to be monitored after the first measurement unless the room temperature changes. The BX-24 can then be set to continuously monitor the "high" pulses. Take the "high" pulses and use the constants derived from the oscilloscope in the equations for calculating the gravities. The BX-24 will then output the values in the terminal window. The values will be output almost in real-time so that the outputs correspond to what is happening in the loom. Whenever there is a fault, the controller can stop the machine.

Section 4.7.3 Analyzing the output

The values from the BasicX Microcontroller are saved into a text file. The text file is then imported into Microsoft Excel to generate wave forms. The Excel plot will show the amplitude of acceleration in g's versus time. Using the plot, the change in acceleration based on the presence of the thread can be monitored. Figure 4.27 shows an Excel plot of four accelerometers at rest within the test apparatus. The X axes of the accelerometers are perpendicular to the plane of gravity. This gives an output of 1 g.

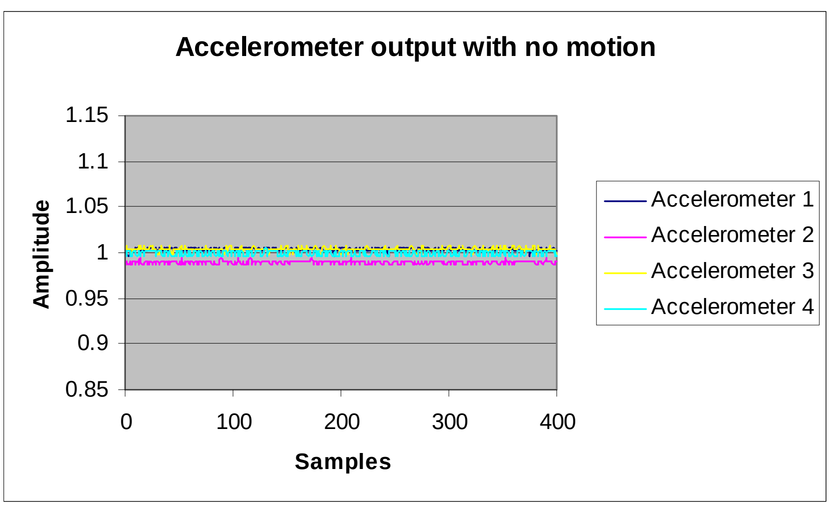


Figure 4.27 Excel output of 4 accelerometers perpendicular to the plane of gravity. Output is

Chapter 5. Experimentation and Results

The experiments were conducted using the test bed. The motor set to drive the cam shaft was supplied an armature voltage from the power supply. The shaft of the motor was coupled to a tachometer that measured the revolutions per minute (rpm) of the motor. The tachometer has an analog output, the output data was measured using the oscilloscope with a conversion factor of 2 V per 1000 rpm. An inertial flywheel was added to the motor to give a more constant rpm at lower speeds, since the motor was not meant to run at such low speeds. In order to properly test the accelerometers the experiments were conducted:

- No load, different speeds
- Constant load, different speeds
- Removing the load during operation

Section 5.1 No load, different speeds

The no load, different speeds experiment represents the heddles without a thread. Experiments were not conducted under 100 rpm. At speeds greater than 200 rpm, the levers bounce off of the cams causing undesirable oscillates, this speed was the upper limit. In the table below (See Table 5.1) the armature input represents the voltage provided to the motor and the motor speed is the calculated rpm based on this tachometer voltage.

Table 5.1 Input/Output of the motor under no load conditions.

Armature input (no load)	Tachometer voltage	Motor Speed
16.4 V	.2 V	100 rpm
17.8 V	.25 V	125 rpm
19.1 V	.3 V	150 rpm
20.5 V	.35 V	175 rpm
22.1 V	.4 V	200 rpm

Section 5.1.1 Constant load, different speeds

The constant load in this experiment is a 100 g weight. This applies a constant load to all four threads simultaneously. Each thread is tied to the rod, so at the point where all the heddles are in line they experience a 25 g load each, assuming that the load is distributed evenly. During one revolution of the motor, 2 heddles are up and 2 heddles are down. The heddles in the up position are experiencing 50 g load each, again assuming that the load is distributed evenly. Adding a load decreases the speed of the motor slightly so tests were run at both the same armature input from the no load condition and at the same output speed with an increased input. The outputs based on the same input speed are detailed in the table below (See Table 5.2)

Table 5.2 Input/Output of the motor under load conditions with the same Armature input, but lower output speed.

Armature input (no load)	Tachometer voltage	Motor Speed
16.4 V	.172 V	86 rpm
17.8 V	.227 V	113.5 rpm
19.1 V	.272 V	136 rpm
20.5 V	.327 V	163.5 rpm
22.1 V	.372 V	186 rpm

On applying a load the voltage required to achieve a constant motor speed will increase, so the armature input increases to compensate for the added load (See Table 5.3).

Table 5.3 Input/Output of the motor under constant load conditions with a higher Armature input, but with the same output speed.

Armature input (load)	Tachometer voltage	Motor Speed
17.2 V	.2 V	100 rpm
18.3 V	.25 V	125 rpm
19.6 V	.3 V	150 rpm
21 V	.35 V	175 rpm
22.6 V	.4 V	200 rpm

Section 5.1.2 Removing the load during operation

As in the previous experiment, a load was applied to a rod connected to all four threads. Each thread experiences ¼ of the load applied when the heddles are in the same position, assuming an even distribution of load. When 2 heddles are raised, those heddles each experience ½ of the load applied, again assuming even distribution. This experiment involved starting the machine with a particular load applied to the threads, and then removing the load suddenly and recording the changes in data. This test showed that if the motor speed starts at 100 rpm with the load applied, when the load is removed increases the motor speed increases to 120 rpm. The second set of experiments was carried out where the motor speed starts at 150 rpm with the load applied and the motor speed increases to 168 rpm after removing the load. The load used in both tests was 200 g.

Section 5.2 The Results

The data was acquired from all 4 accelerometers and imported into Microsoft Excel. The data obtained from the above experiments can be reviewed in Chapter 8, Section 1. Here, the data has been arranged to overview the system behavior can be seen based on the experiments carried out above. The outputs of the accelerometers are displayed separately, such that all the outputs from Accelerometer 1 can be shown on the same plot, i.e. showing the result of No Load at 100 rpm vs. the result of Constant Load at 100 rpm. The results of the first two experiments (No Load, different speeds and Constant Load, different speeds) are discussed together, while the results of the last two experiments (Constant speed, different load and Load removed) will be discussed later.

Section 5.2.1 Load vs. No Load

The experiments showed that the outputs from the accelerometers are essentially identical, so only Accelerometer 1 will be discussed here. The data from all the accelerometers is cataloged in Chapter 8. The following plot (See Figures 5.1) shows three outputs of Accelerometer 1 under different conditions. The conditions are:

1. No Load 16.4 V input and 100 rpm output
2. 50g Load and 100 rpm output – 17.2 V input
3. 50g Load and 16.4 V input – 86 rpm output

The plot shows that the accelerations of conditions 1 and 2 are almost identical; this is expected since the velocity is the same. However, condition 3 is noticeably lower. The conclusion can be made that if the system starts in condition 3, when the threads are removed, the output will increase to 100 rpm. The transition from 3 to 1 is shown later in Section 5.2.3.

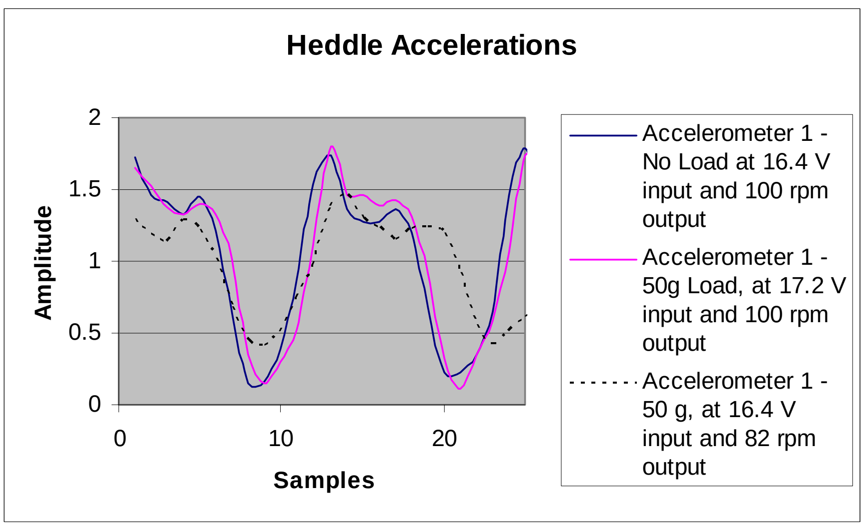


Figure 5.1 Heddle Accelerations of accelerometer 1 seen at conditions 1 (100 rpm), 2 (100 rpm), and 3 (86 rpm).

The next figure (See Figure 5.2) shows the differences based on a larger input. Notice that the width of the acceleration wave is wider as the rpm decreases, this leads to a phase difference between the higher rpm signal and the lower rpm signal.

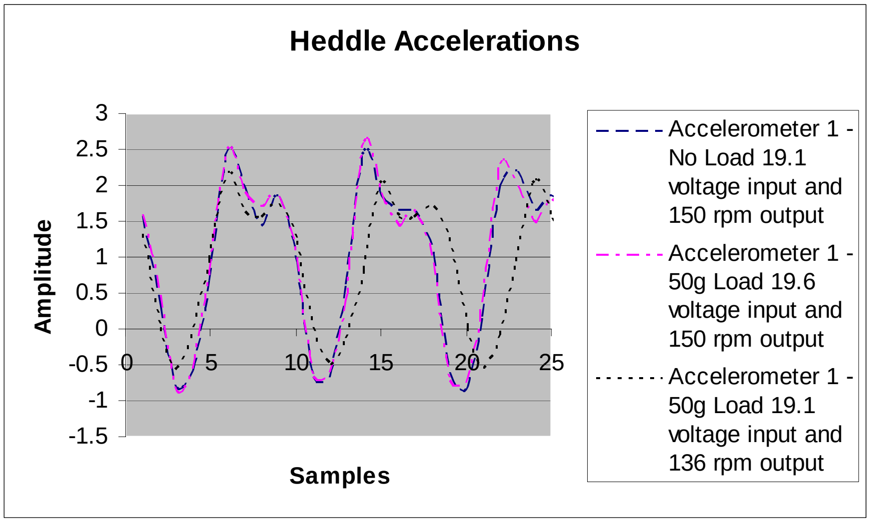


Figure 5.2 Heddle Accelerations of accelerometer 1 seen at conditions 1 (150 rpm), 2 (150 rpm), and 3 (136 rpm).

Figure 5.3 shows the difference of the load and no load signals based on a 200 rpm no load input. The loaded output is 186 rpm, the displayed results show that this gives a phase difference and a lower acceleration compared to the no load case.

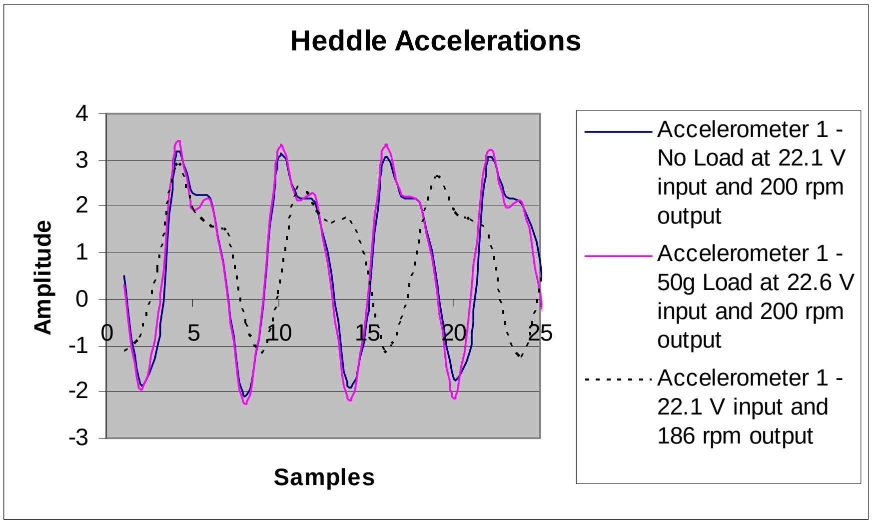


Figure 5.3 Heddle Accelerations of accelerometer 1 seen at conditions 1 (200 rpm), 2 (200 rpm), and 3 (186 rpm).

Section 5.2.2 Removing the Load

The first plot (Figure 5.4) shows the output of the accelerometers when the load is suddenly removed during operation. The speed of the motor was 100 rpm while loaded, and increased to 120 rpm after removing the load. The load was removed just after 1200 samples, on the plot, the increase in acceleration of the heddle can be seen clearly. This would clearly indicate a warp yarn break if it occurred in a weaving machine.

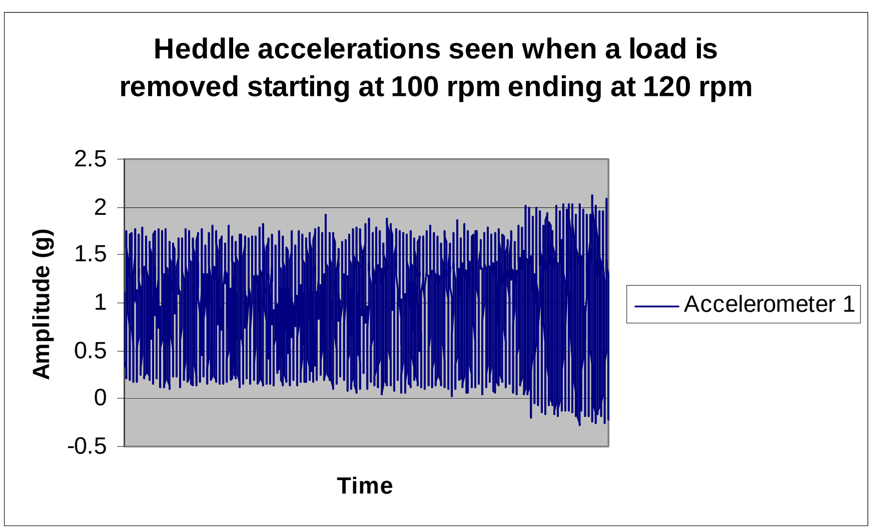


Figure 5.4 Accelerometer output of accelerometer 1 seen when a 200 g load is removed, speed begins at 100 rpm and

Figure 5.5 shows the difference in the output signal between a starting rpm, when loaded, and the ending rpm, when unloaded. The difference isn't as noticeable as it was in Figure 5.4, but is still visible. Only through further experimentation and data analysis would we be able to determine if the change in the signal is significant enough to indicate a warp yarn break.

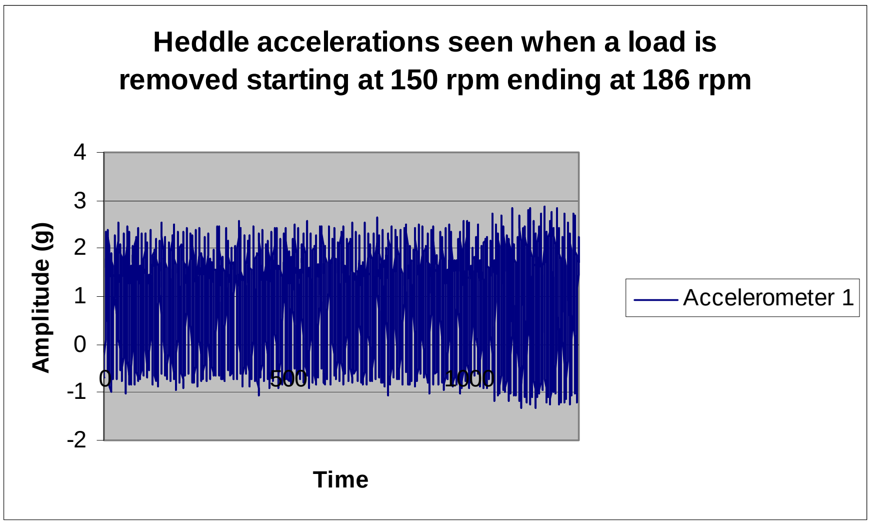


Figure 5.5 Accelerometer output of accelerometer 1 when a 200g load is removed, speed begins at 150 rpm and ends at

Section 5.3 Analyzing the Results

The set of Excel plots (5.1 – 5.5) show that the presence of a load in the system affects its dynamics. Looking at the plots, a difference can be clearly seen, therefore a program was written to automatically detect warp yarn breaks. The simplest way to analyze the data was using Matlab, since Matlab allows text files to be imported, e.g. Excel data files from the experiments in Section 5.2.

Section 5.3.1 Using Matlab to search for breaks

The data from the accelerometers was analyzed using Matlab. Matlab source code was written to search for increases in signal amplitude, indicating a broken warp yarn. The data for each accelerometer is extrapolated into 4 separate data sets. The data for accelerometer *i* (*i* representing 1, 2, 3, or 4) is then separated into segments. Each segment contains 100 data points. If the last segment has less than 100 data points, it is dropped. Segments 1 and 2 are then compared to find the maximum value between the two segments. Both segments are then divided by the maximum value to normalize the data. After normalizing the data, the max and min value from each segment is then found. The max value from Segment 1 is compared to the max value from Segment 2. If they differ by more than a specified amount, then the min value from Segment 1 is compared to the min value of Segment 2. If this value is also differs by a specified amount, the inference is there may be a break. If there is no break in these two segments, the code moves on to compare Segment 2 and 3, and then 3 and 4, and so on. It is assumed that the data does not contain a break in the first Segment, so the machine starts with no breaks. The software requires calibration, this is based on the speed of the heddle. This calibration involves specifying certain thresholds in the software to determine system sensitivity.

Section 5.3.2 Simulated Data

In order to test the software, data from section 5.2.1 was adjusted to simulate a break occurring. In this experiment, the data from two different cases is combined to simulate a break. The armature voltage required to drive the heddles at 100 rpm, was 16.4 V. At this voltage, when the heddles are experiencing a load, the rpm drops to 86 rpm. To simulate a break occurring on a particular accelerometer, the data from the load case (86 rpm) of that particular accelerometer was combined with the data from the no load case (100 rpm). After combining the data, signals show that there is no break for period of the signal, later in the signal a break will be indicated on accelerometer 4 (See Figure 5.5). In this figure, the graphs on the left are the first segment, and the graphs on the right are the second segment. Each row represents one accelerometer (1, 2, 3, and 4).

In the bottom right hand plot, the graph has turned red. This indicates that a break occurred from within the signal. This break is confirmed visually through observing that the peaks of the curves increased in amplitude after 60, these eventually leveled off at 1. The other accelerometer output plots remain blue, indicating no warp yarn breaks.

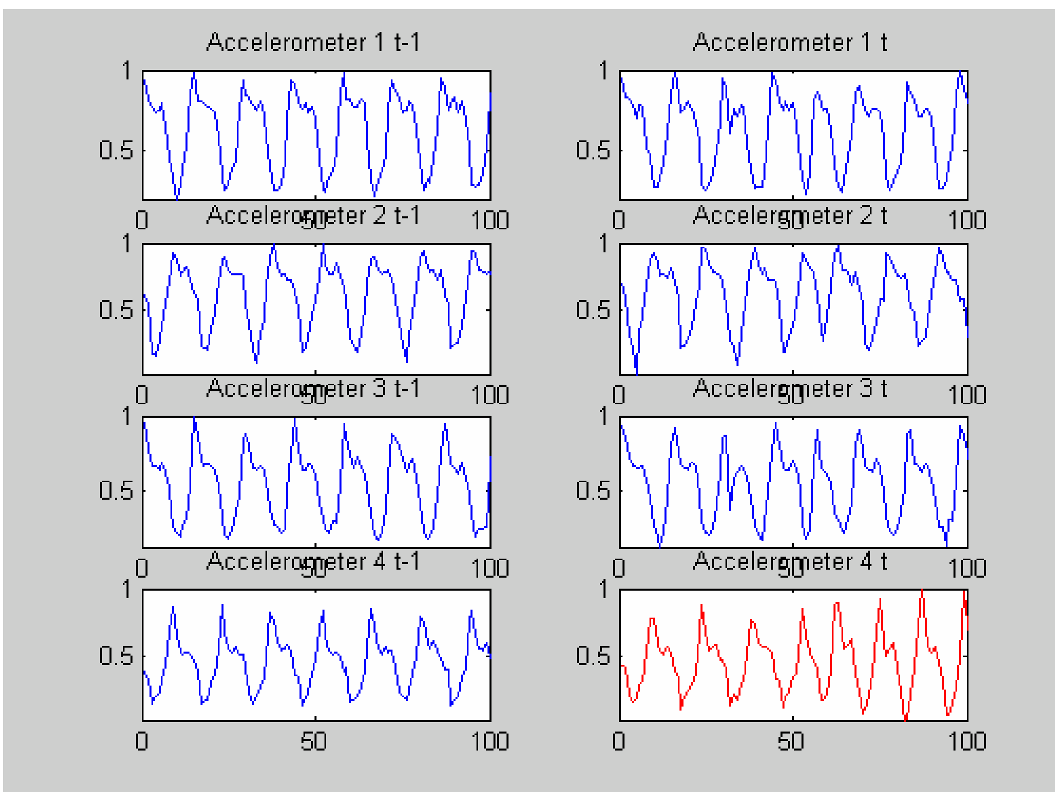


Figure 5.6 MatLab output of 4 accelerometers. The red plot shows a break occurring on accelerometer 4.

Section 5.3.3 Real Data

The data in Figure 5.7 and Figure 5.8 represents real data taken from the Jacquard Loom test bed system. At the "break point", where the load is removed, the accelerations increase. Unlike in the simulated data, where the speed increase was instantaneous, here the speed gradually increases to compensate for the removed load. Since this cam lever design is a rigid system, a decreased load on one heddle affects the accelerations of all heddles so the experiments involved removing the load from all four heddles simulataneously. In a commercial Jacquard Loom, removing a thread from a single heddle does not affect the other heddles because the system is not rigid so the absence of a warp yarn on a single heddle in a commercial Jacquard Loom can be found.

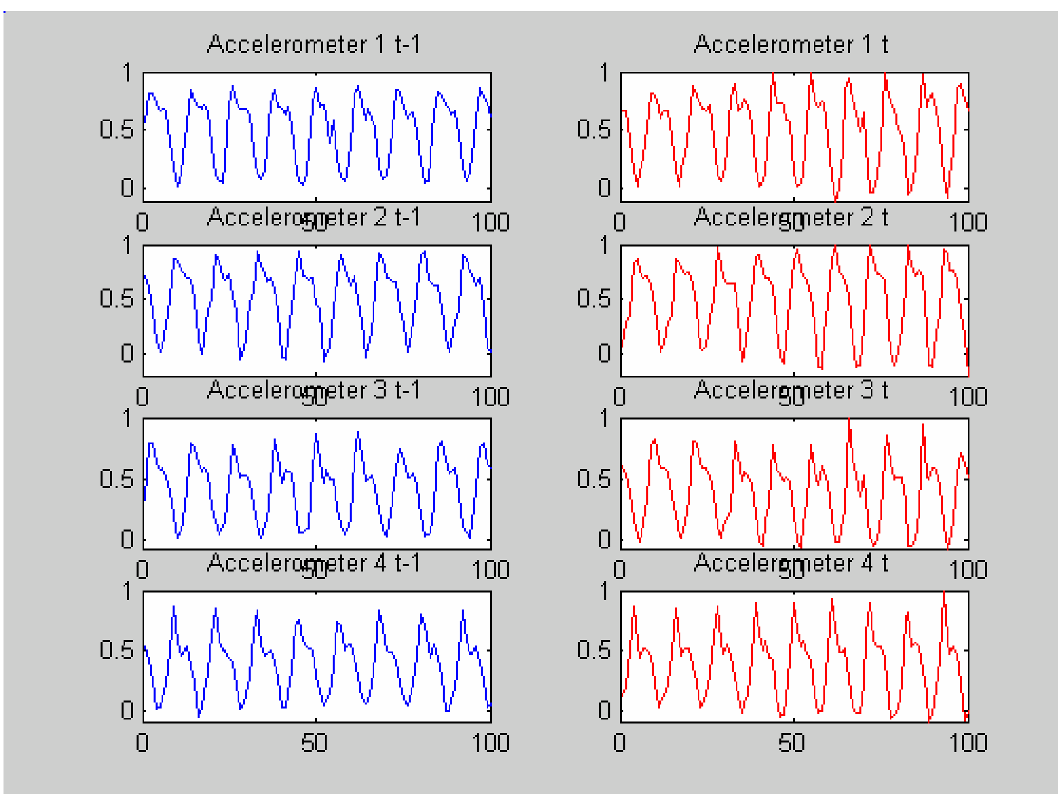


Figure 5.7 MatLab output of 4 accelerometers when the load is removed from all 4 accelerometers at the same time, starting at 100 rpm loaded.

Figure 5.7 shows the outputs of the accelerometers when the load was removed at the lower speed of 100 rpm, this data corresponds to the data in Figure 5.4. The plots show that Matlab was able to determine that there was a change, it is the same point in Figure 5.4 where the accelerometer output began to increase.

Figure 5.8 shows the outputs of the accelerometers when the load was removed at the higher speed of 150 rpm, this data corresponds to the data in Figure 5.5. The plots show that Matlab was able to determine that there was a change at the same point in Figure 5.5 where the accelerometer output detected an increase in acceleration.

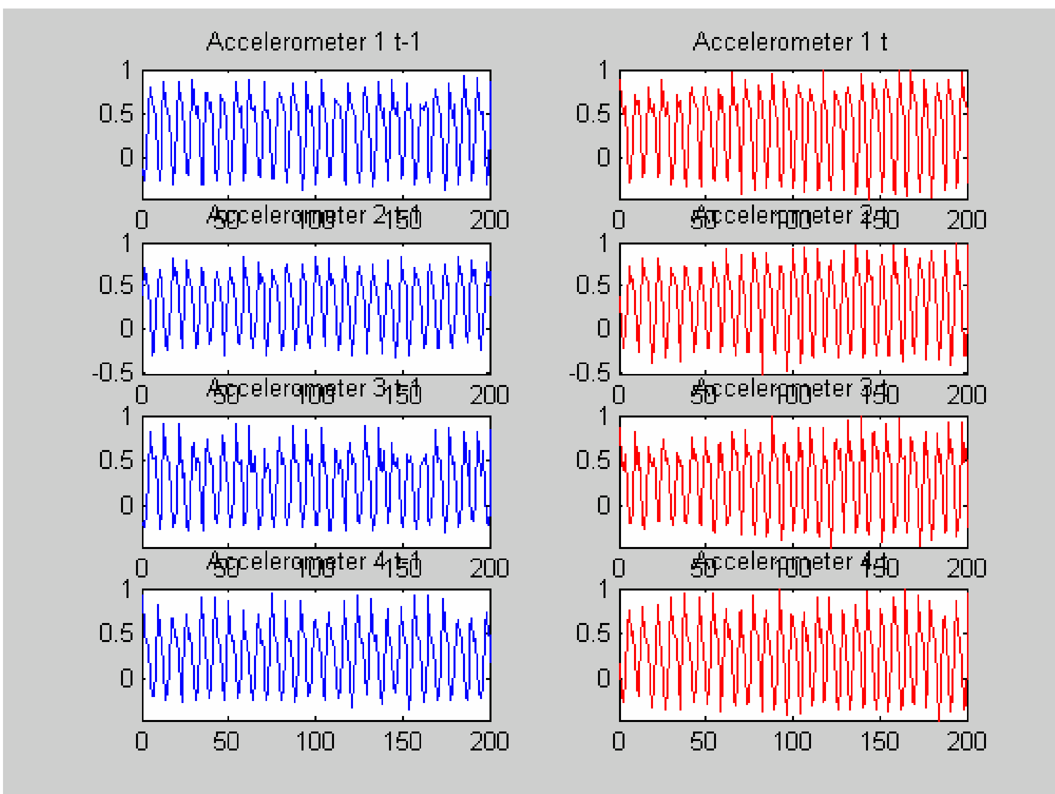


Figure 5.8 MatLab output of 4 accelerometers when the load is removed from all 4 accelerometers at the same time, starting at 150 rpm loaded.

Chapter 6. Conclusion and Future Research

The current Jacquard Loom fault detection system, the drop wire system, is out of date. The MEMS accelerometer could be the answer to finding a non-contact fault detection system. By placing MEMS accelerometers on the individual heddles, each heddle can be monitored for changes in acceleration based on the presence or absence of a warp yarn. This could lead to faster warp yarn repair and maybe even automated repair.

In order to test the MEMS accelerometers, a test frame, modeled after the Jacquard Loom system, was constructed in the CRIM lab. A total of four heddles were mounted in the test frame and monitored using the accelerometers. A motor drove a cam-lever system connected to the heddles to give the basic motion of a heddle within a Jacquard Loom. The top speed of the system was approximately 200 rpm, which corresponds to about 200 picks per minute. Some commercial looms are capable of over 1000 picks per minute and the accelerometer should be able to handle the higher frequency.

The MEMS accelerometers are provided by Analog Devices and were monitored using a BasicX BX-24 microprocessor. The output of the microprocessor was sent to a computer terminal window and saved into a text file. This text file was then imported into Microsoft Excel and plotted to model the behavior curves. The behavior of the heddle changed based on speed and on load, showing that the accelerometer can monitor slight differences in acceleration. After modeling the behavior in Excel, the output was then imported into Matlab. Software for Matlab was written that was able to detect the change in acceleration based on the absence of the load.

This research has shown that the MEMS accelerometer can monitor acceleration at the heddle and even see differences based on the absence of the warp yarn. The accelerometers were successful in the test environment provided, which may lead to success in a commercial environment.

Some future research that could be conducted involves placing the accelerometers on a commercial Jacquard Loom, since the accelerometers were not tested using actual Jacquard machinery. This research could show that the accelerometers would be useful in commercial Textile machinery.

If these sensors are to be integrated into Jacquard machinery, the Matlab software would need to be changed to monitor for a break in real time. The software could be implemented to allow for some sort of reaction system, such as automatically re-tying the broken warp yarns using an existing re-tying machine.

Wireless MEMS accelerometers could also be developed for use in the Jacquard Loom. These sensors are already becoming available on the market (G-Link, 2002). Currently the wireless sensors are slightly too large for mounting on the heddles within the loom, but the MEMS technology is continually getting smaller.

Chapter 7. References

Adanur, Sabit. "Weaving Speeds Up." Textile Industries Media Group. May 2000. <http://www.textileindustries.com/News.htm?CD=027&ID=85> 3 March 2003.

Buss, J and Szu, H. "MEMS Sensors Interface Standardization." International Collaboration through ChemBio Conferences. April 2001 <http://www.cse.cuhk.edu.hk/~apnna/proceedings/iconip2001/papers/303a.pdf> 3 March 2003.

Gahide, Severine. (2001) "Exploration of Micromachines to Textiles: Monitoring Warp Tension and Breaks during the formation of woven fabrics." PhD Thesis. North Carolina State University.

Murray, Kevin. "Weaving in a post-human domain." Loom November 2000. <http://www.craftvic.asn.au/exhibit/2001/loom/posthum.htm> 3 March 2003.

Noonan, Karl. "CAD/CAM's Effect on the Jacquard Weaving Industry and What Can Be Expected in the Future." TechExchange. January 2000. <http://www.techexchange.com/thelibrary/jacquard_weave.html> 3 March 2003.

Terkoski, Erin. "Joseph Marie Jacquard (1752-1854)." Ancient History of Computer Science. January 2001. <http://kzoo.edu/~k00et01/jacquard.html> 3 March 2003.

Travis, Bill. "MEMS Sensors and Mechanical Gadgets enter the mainstream." EDN ACCESS. May 1998. < http://www.e-insite.net/ednmag/archives/1998/052198/11df_01.htm> 3 March 2003.

Weinberg, Harvey. "MEMS Sensors: Are Driving the Automotive Industry." Sensors Express. February 2002. <http://www.sensorsmag.com/articles/0202/36/main.shtml> 3 March 2003.

Weinberg, Harvey. "MEMS (Micro Electro-Mechanical Systems) Technology" Sensorland. June 2000. < http://www.sensorland.com/HowPage023.html> 3 March 2003.

"About MEMS Technology." MEMS Clearinghouse. January 2002. <http://www.memsnet.org/mems/what-is.html> 3 March 2003.

"G-Link Wireless Accelerometer System." MicroStrain. Spring 2001 <http://www.microstrain.com/pdf/G-Linkflier6.pdf> 3 March 2003

"Grob Horgen Latest Infos." Grob Horgen AG. March 2003. < http://www.grob-horgen.ch/english/latest_infos.html> 3 March 2003.

"Jacquard Machine LX 3201." Stäubli. January 2001.
<http://www.staubli.com/web/textile/division.nsf> 3 March 2003.

"loom." The Columbia Encyclopedia, Sixth Edition. August 2002.
<http://www.bartleby.com/65/lo/loom.html> 3 March 2003.

"Textile Terms." The Textile Guide. March 2002
<http://www.thetextileguide.com/tgterms.htm> 3 March 2003

"Warp yarn Breakage Sensor." Izumi International. March 2003.
<http://www.izumiinternational.com/carbon/laserstop.html> 3 March 2003.

"Working Principle of Eltex Systems." Eltex of Sweden AB. January 2001.
<http://www.eltex.se/eng/index-e.htm> 3 March 2003.

Appendices

Chapter 8. Appendices

Section 8.1 Accelerometer output plots

The following plots are data that was taken during the experiment stage. The first section shows data that was taken when the heddles were at rest. The second section shows data taken when there was no load at the heddle, representing the absence of a thread. The third section shows data taken at different speeds with the same load, representing the presence of a thread at approximately 25 g tension. The fourth section shows data taken at the same speed with different loads.

Section 8.1.1 Output plots when there is no motion

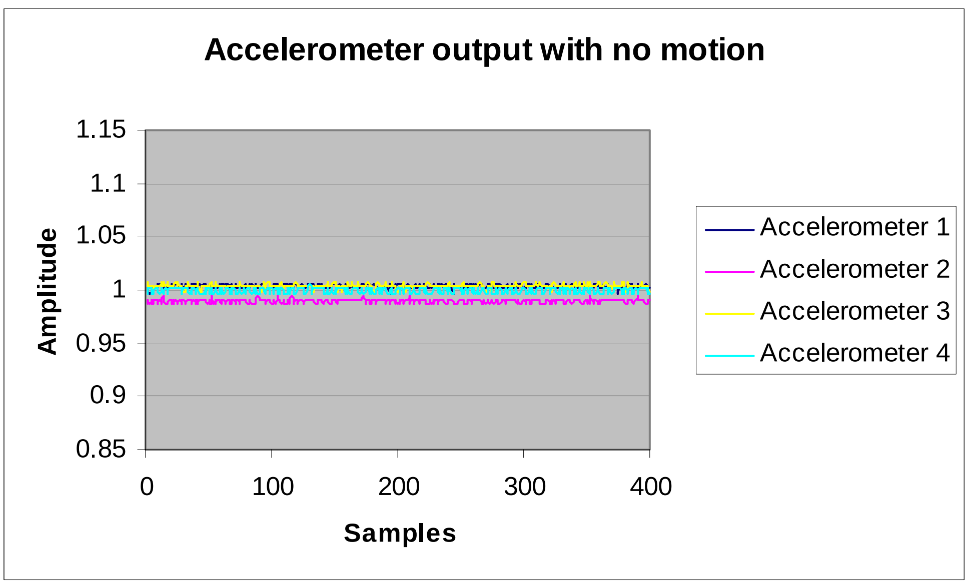


Figure 8.1 Accelerometer output with no motion.

Section 8.1.2 Output plots at different speeds with no load

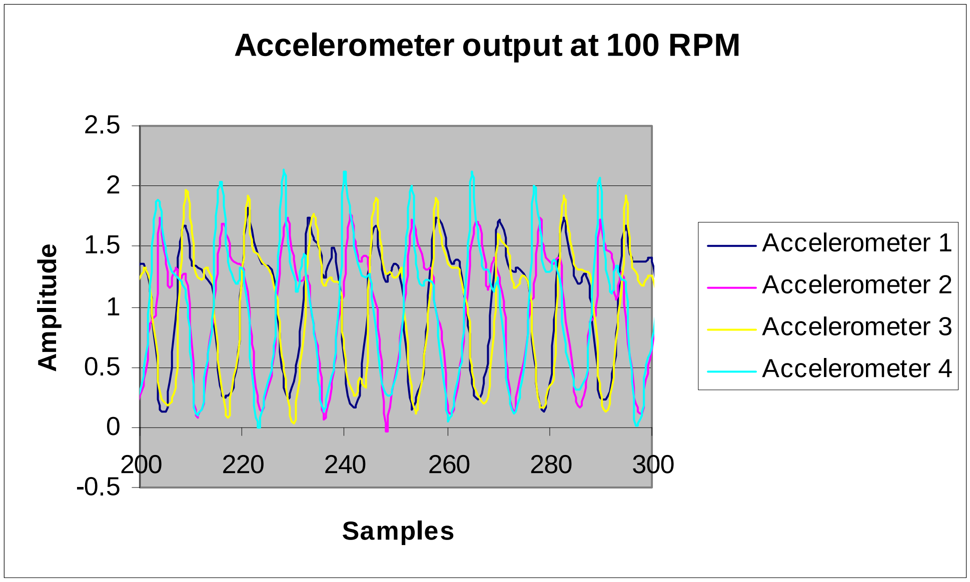


Figure 8.2 Accelerometer output at 100 RPM.

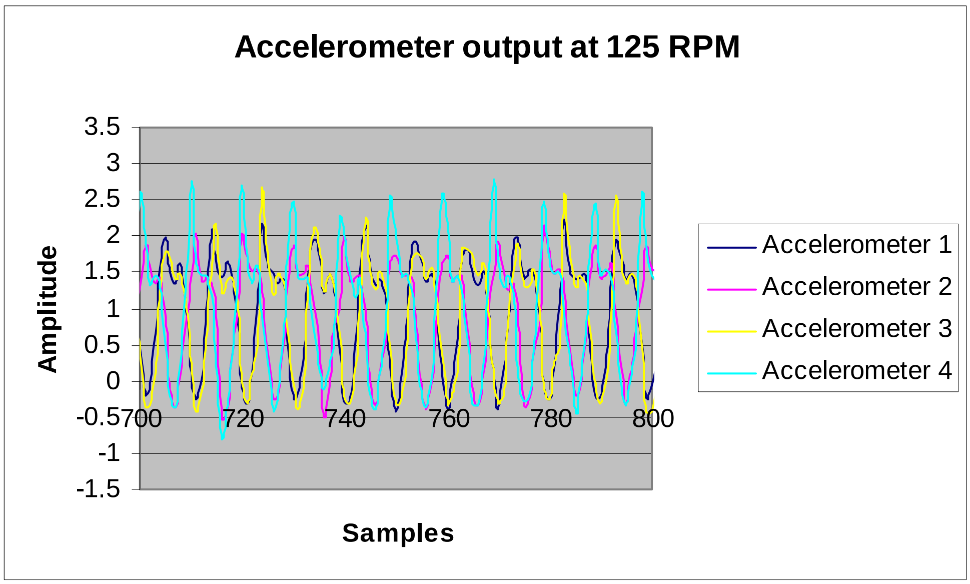


Figure 8.3 Accelerometer output at 125 RPM.

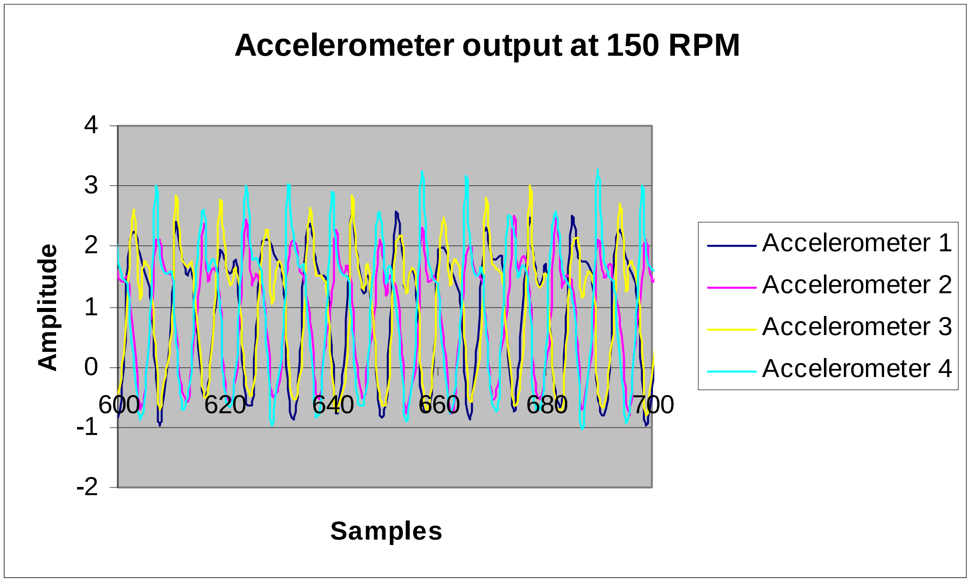


Figure 8.4 Accelerometer output at 150 RPM.

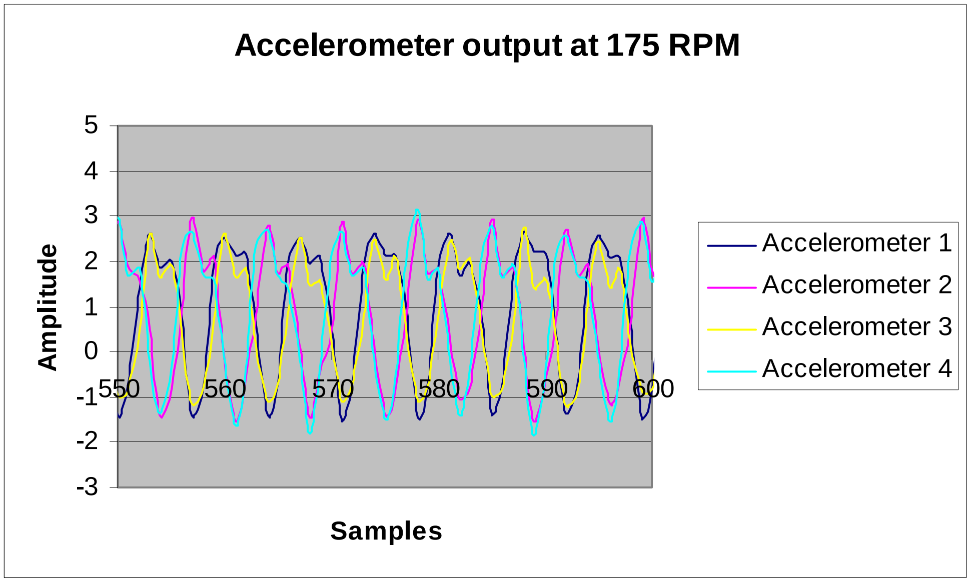


Figure 8.5 Accelerometer output at 175 RPM.

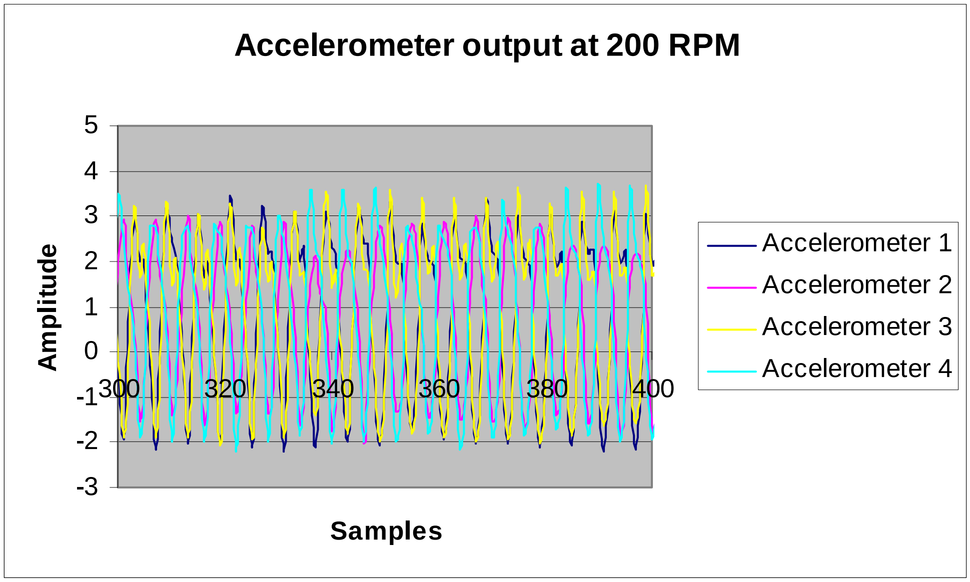


Figure 8.6 Accelerometer output at 200 RPM.

Section 8.1.3 Output plots at different speeds (same motor input) with the same load

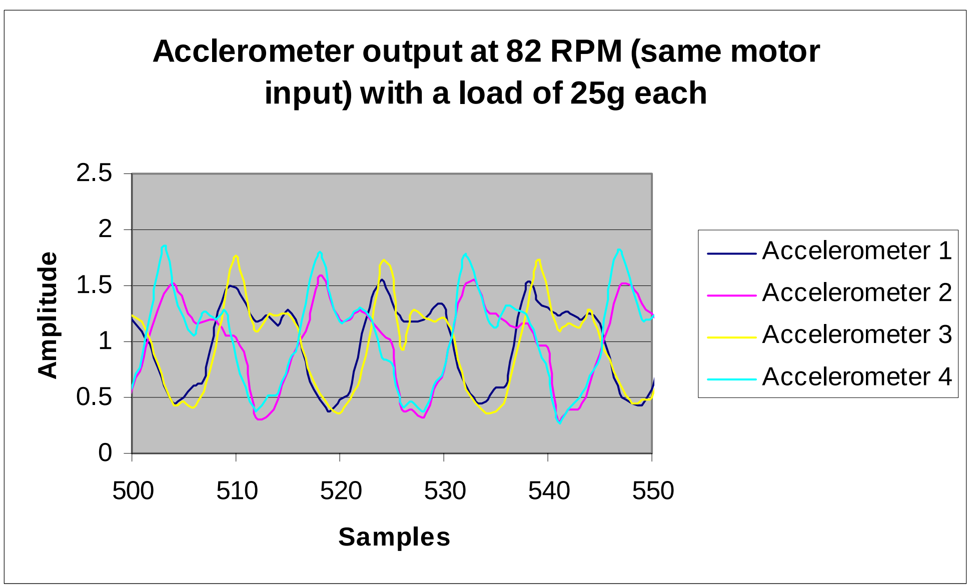


Figure 8.7 Accelerometer output at 86 RPM (same motor input).

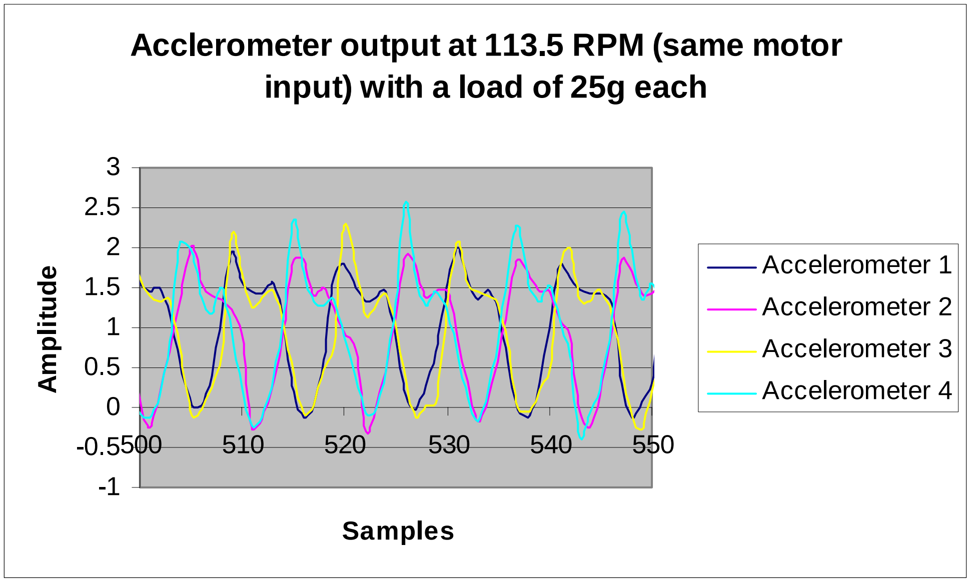


Figure 8.8 Accelerometer output at 113.5 RPM.

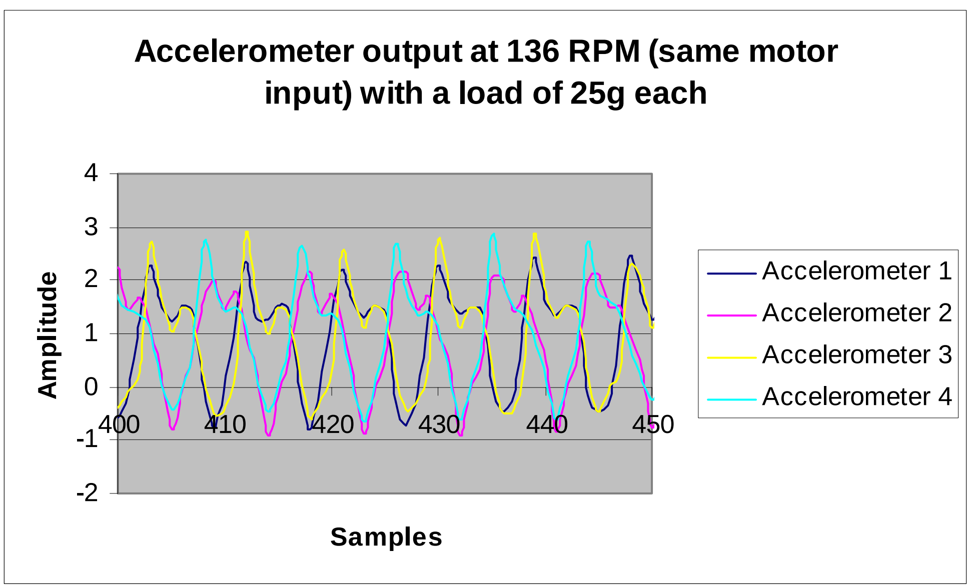


Figure 8.9 Accelerometer output at 136 RPM.

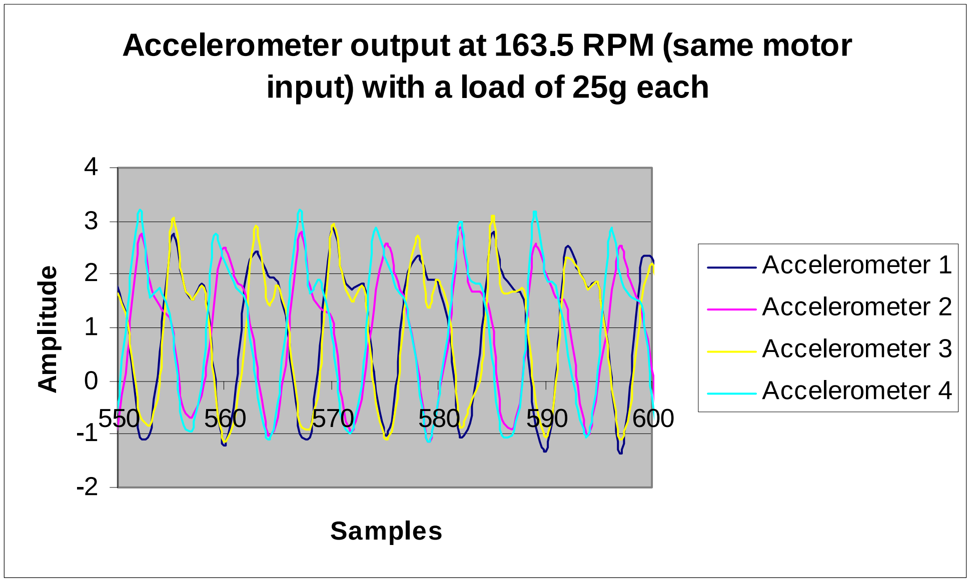


Figure 8.10 Accelerometer output at 163.5 RPM.

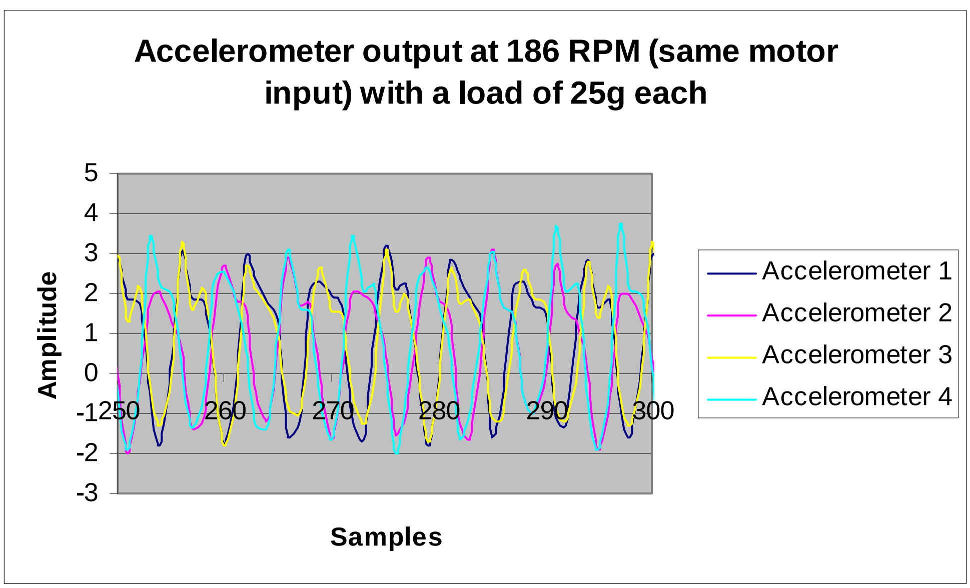


Figure 8.11 Accelerometer output at 186 RPM.

Section 8.1.4 Output plots at different speeds (adjusted motor input) with the same load

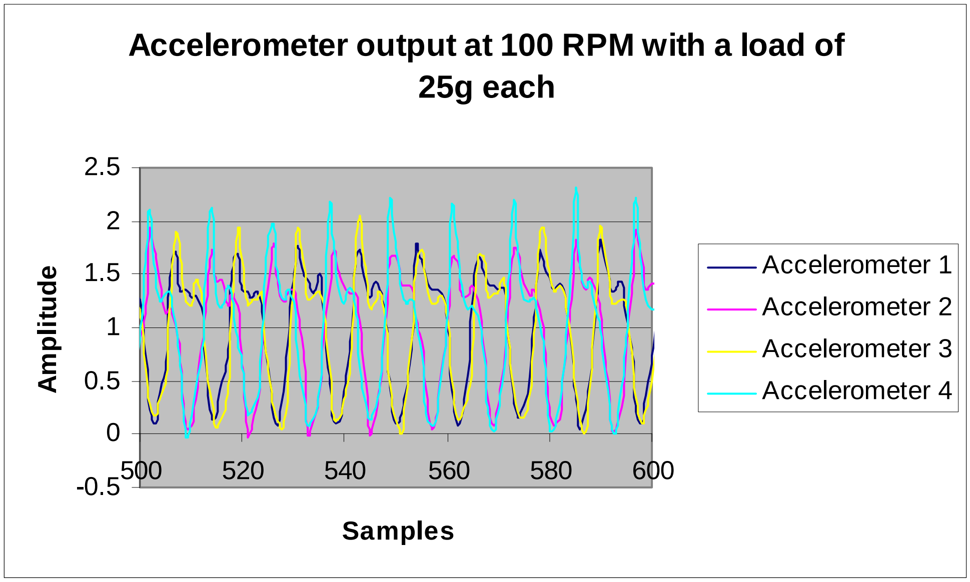


Figure 8.12 Accelerometer output at 100 RPM with a load of 25g.

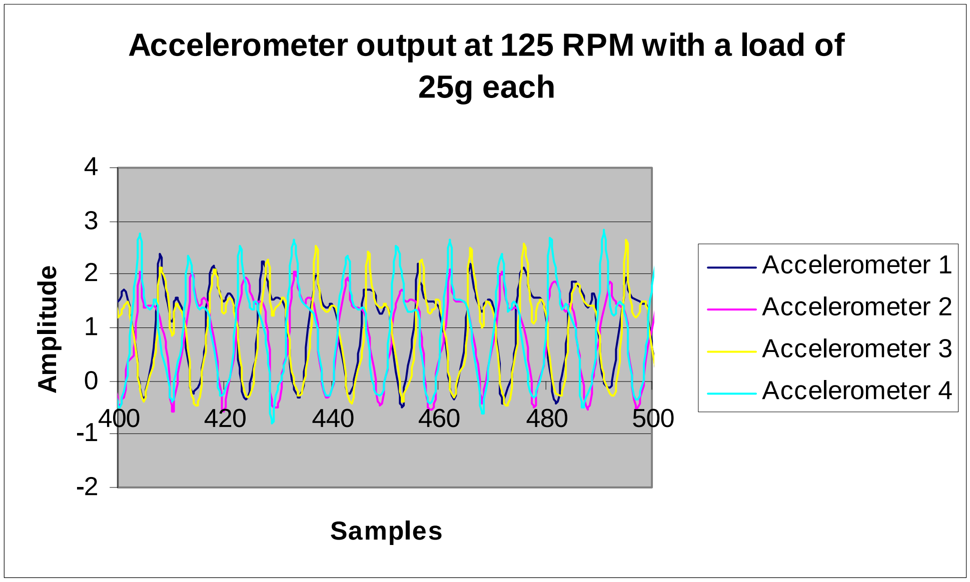


Figure 8.13 Accelerometer output at 125 RPM with a load of 25g.

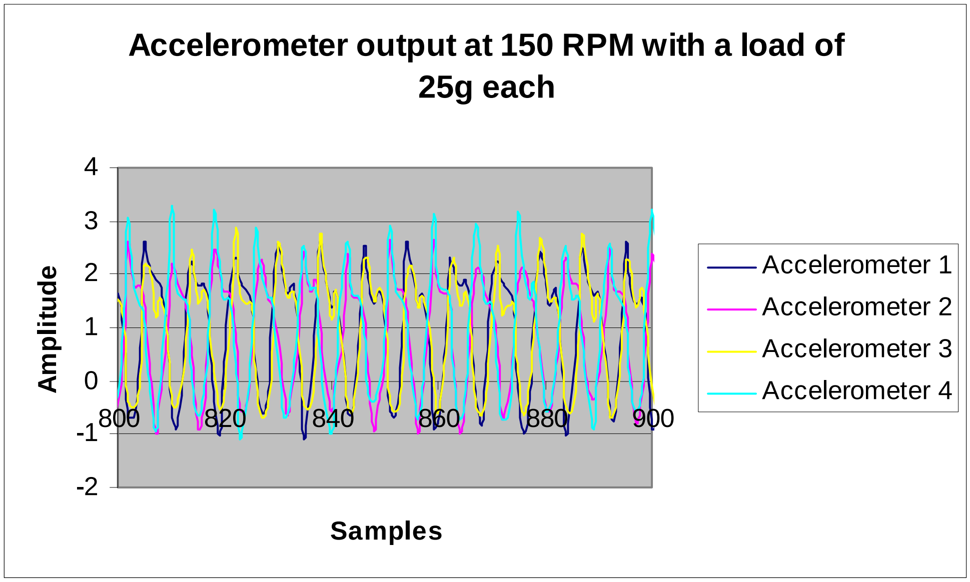


Figure 8.14 Accelerometer output at 150 RPM with a load of 25g.

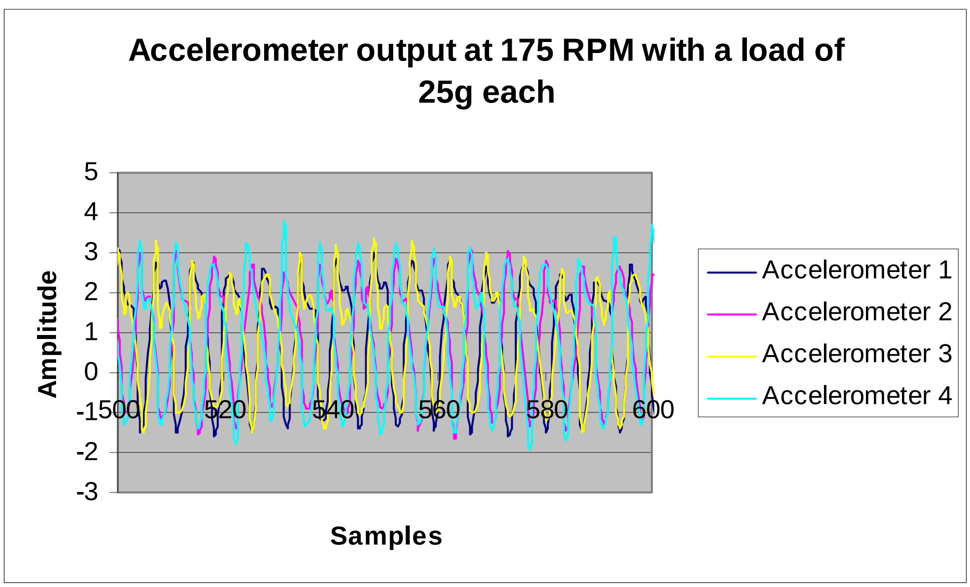


Figure 8.15 Accelerometer output at 175 RPM with a load of 25g.

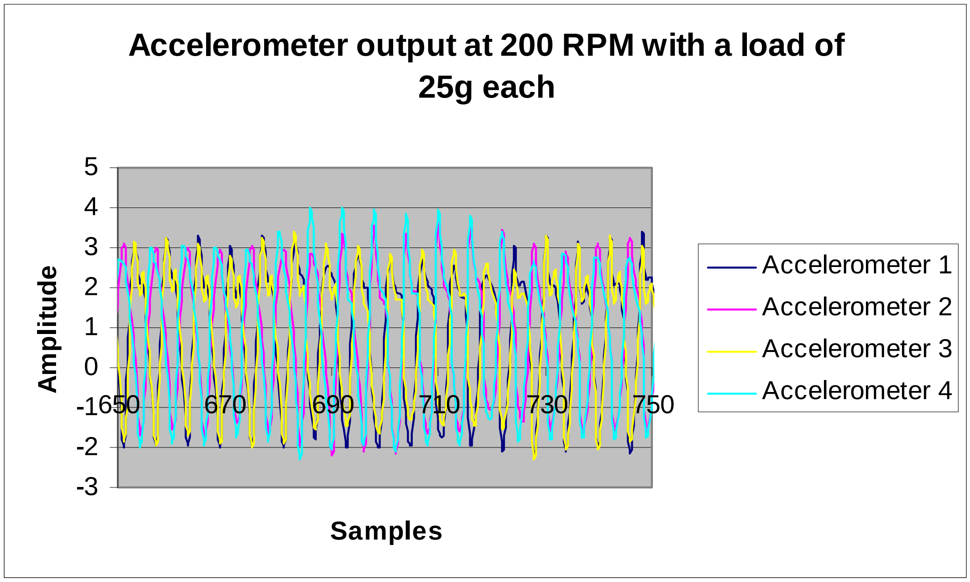


Figure 8.16 Accelerometer output at 200 RPM with a load of 25g.

Section 8.1.5 Output plots at the same speed with different loads

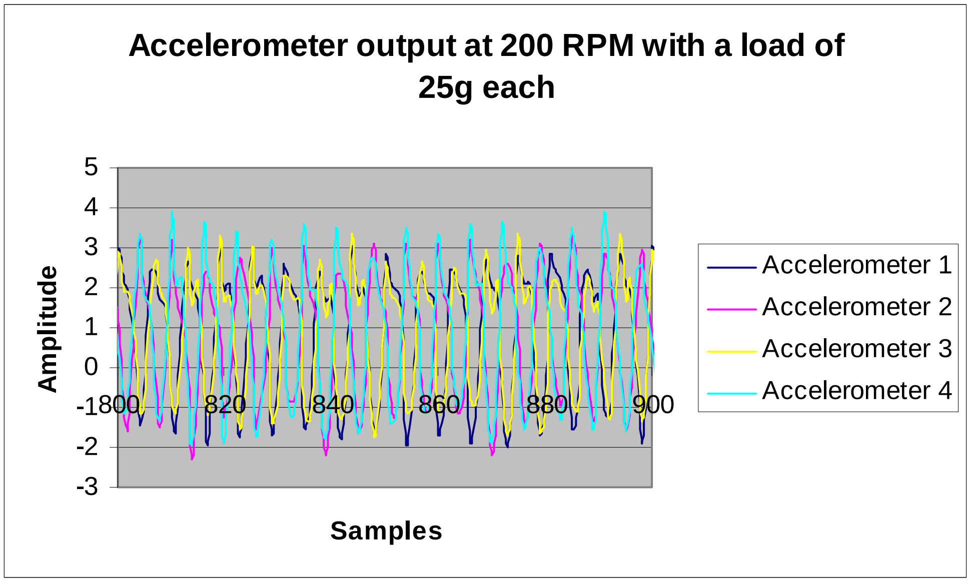


Figure 8.17 Accelerometer output at 200 RPM with a load of 25g.

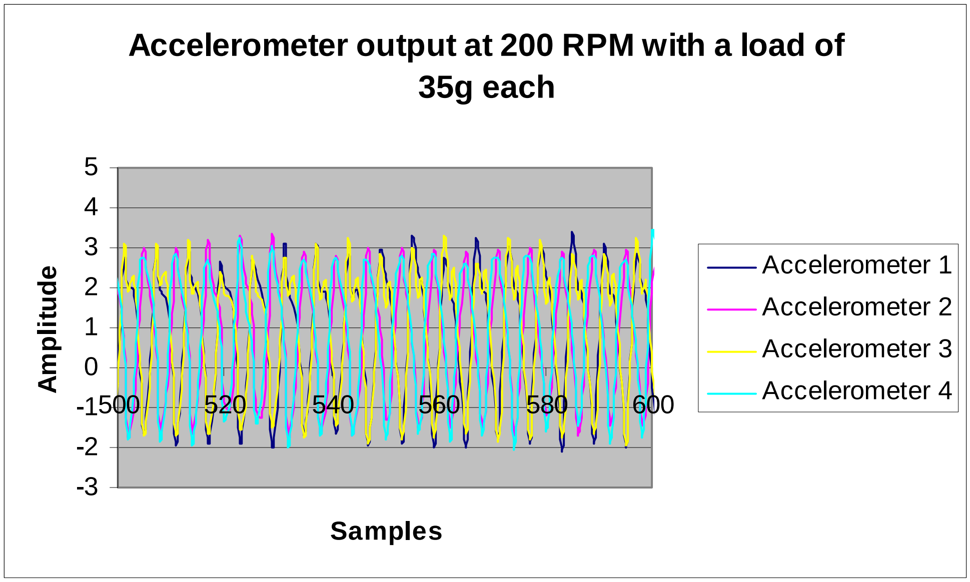


Figure 8.18 Accelerometer output at 200 RPM with a load of 35g.

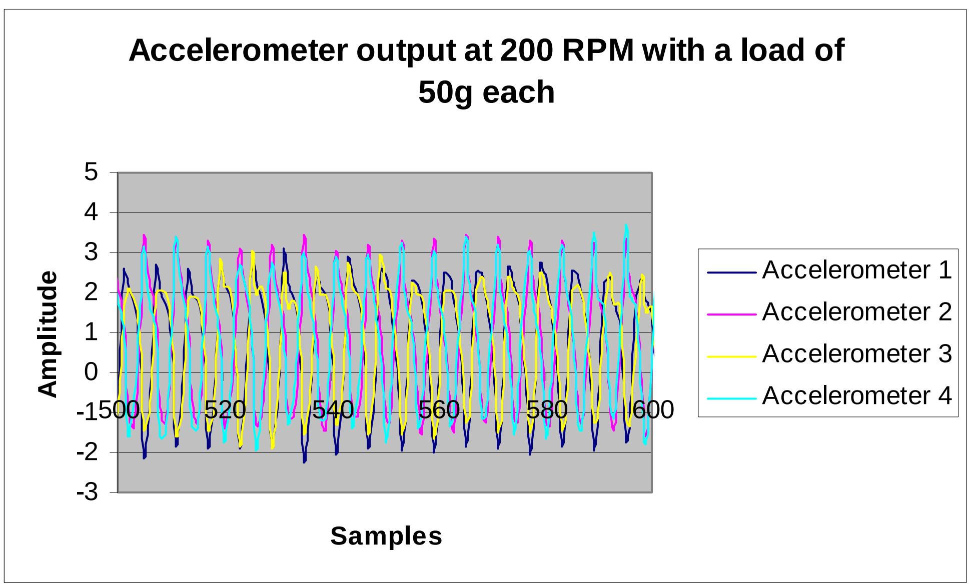


Figure 8.19 Accelerometer output at 200 RPM with a load of 50g.

Section 8.2 BasicX BX-24 Accelerometer code

To read all four accelerometers, the BX-24 needs 3 sets of source code (DisplayAcceleration4.bas, Accelerometer4.bas, and SerialPort.bas). Accelerometer4 reads the output from each accelerometer, DisplayAcceleration4 uses SerialPort to display the acceleration on the computer. The BasicX software allows exportation into a text file. To use, run the module containing these three files and using the BasicX software output to a file. Divide the values by 1000 to get acceleration in g. Each value needs to be divided by 1000 because the output looks best in integer form, and a resolution of 3 digits past the decimal were required.

Filename: DisplayAcceleration4.bas

```
'-------------------------------------------------------------------------------
Option Explicit

'-------------------------------------------------------------------------------
Public Sub Main()

' This program reads 4 Analog Devices ADXL202 accelerometers and
' displays the X component of the acceleration vector.
' The output is an integer and is multiplied by 1000.
' When using the output, divide by 1000 to get an accurate representation
' of the acceleration.  It is multipled by 1000 as an integer to drop
' the extra bits, and still retain 3 digits after the decimal.

   Dim Ax1 As Single
   Dim Ax2 As Single
   Dim Ax3 As Single
   Dim Ax4 As Single

   Dim Tx As String
   Dim Ax_int As Integer

   Call Init

   Tx = " , "

   Do
      ' Read acceleration vectors.
      Call GetAccelerations(Ax1,Ax2,Ax3,Ax4)

      Ax_int = CInt(Ax1*1000.0)
      Call PutI(Ax_int)
      Call PutStr(Tx)

      Ax_int = CInt(Ax2*1000.0)
      Call PutI(Ax_int)
      Call PutStr(Tx)

      Ax_int = CInt(Ax3*1000.0)
      Call PutI(Ax_int)
      Call PutStr(Tx)

      Ax_int = CInt(Ax4*1000.0)
      Call PutI(Ax_int)
```

```
        Call NewLine
    Loop

End Sub
'-----------------------------------------------------------------------------------------
Private Sub Init()

    Call OpenSerialPort(1, 19200)
    ' For calibration
    ' Call GetPeriod

End Sub
'-----------------------------------------------------------------------------------------
```

Filename: Accelerometer4.bas

```
'-------------------------------------------------------------------------------
Option Explicit

' This module is for reading the output of 4 Analog Devices ADXL202
' 2-axis accelerometers.
'
' These values depend on calibration.
' CenterX is the 50% duty cycle.
' DeltaX is the percent change of duty cycle per g.
' Period is the period of the PWM signal.

Private Const CenterX1 As Single = 0.478535
Private Const DeltaX1 As Single = 0.128731
Private Const Period1 As Single = 0.001774

Private Const CenterX2 As Single = 0.481075
Private Const DeltaX2 As Single = 0.130229
Private Const Period2 As Single = 0.001825

Private Const CenterX3 As Single = 0.479381
Private Const DeltaX3 As Single = 0.127642
Private Const Period3 As Single = 0.001879

Private Const CenterX4 As Single = 0.483252
Private Const DeltaX4 As Single = 0.128666
Private Const Period4 As Single = 0.001796


' Defining the input pins on the BasicX BX-24

Private Const PinX1 As Byte = 15
Private Const PinX2 As Byte = 16
Private Const PinX3 As Byte = 17
Private Const PinX4 As Byte = 18

'-------------------------------------------------------------------------------
Public Sub GetAccelerations( _
   ByRef Ax1 As Single, _
   ByRef Ax2 As Single, _
   ByRef Ax3 As Single, _
   ByRef Ax4 As Single)

' This procedure reads the X axis of 4 ADXL202 accelerometers.
' The acceleration vectors are returned. Units are in gravities.
```

```
'
'     <------ Period ----->
'     +--------+              +----
'      |          |               |
'  ---+          +------------+
'     <--T1-->
'
' Acceleration = (T1/Period - 0.5)/0.125

   Dim T1X as single

   Dim SumX1 As Single
   Dim SumX2 As Single
   Dim SumX3 As Single
   Dim SumX4 As Single

   Dim i As Byte, T1 As Single

   SumX1 = 0.0
   SumX2 = 0.0
   SumX3 = 0.0
   SumX4 = 0.0

   Call PulseIn(PinX1, 1, T1)
   SumX1 = SumX1 + T1

   Call PulseIn(PinX2, 1, T1)
   SumX2 = SumX2 + T1

   Call PulseIn(PinX3, 1, T1)
   SumX3 = SumX3 + T1

   Call PulseIn(PinX4, 1, T1)
   SumX4 = SumX4 + T1

   Ax1 = ((SumX1 / Period1) - CenterX1) / DeltaX1
   Ax2 = ((SumX2 / Period2) - CenterX2) / DeltaX2
   Ax3 = ((SumX3 / Period3) - CenterX3) / DeltaX3
   Ax4 = ((SumX4 / Period4) - CenterX4) / DeltaX4

End Sub
'-------------------------------------------------------------------------------
Public Sub GetPeriod()

' This procedure determines the average period generated by the X axis
' channel of an ADXL202 accelerometer.
```

```
'
' This procedure would replace the constant period set up top, use this
' if you have the accelerometers in a variable temperature environment.
' Comment out the constants defined above and uncomment the Period statements
' below. The function call is made in DisplayAcceleration4 and also need to be
' uncommented out.
'
' The period is also denoted as T2 in Analog Devices documentation.
'
'      <------ Period ----->
'     +--------+           +----
'     |        |           |
'  ---+        +-----------+
'      <--T1--> <--- T3 --->

    Dim SumT1 As Single, SumT3 As Single
    Dim T1 As Single, T3 As Single, i As Integer
    Dim AvgT1 As Single, AvgT3 As Single
    Const NSamples As Integer = 20

    SumT1 = 0.0
    SumT3 = 0.0
    For i = 1 to NSamples
        Call PulseIn(PinX1, 1, T1)
        Call PulseIn(PinX1, 0, T3)

        SumT1 = SumT1 + T1
        SumT3 = SumT3 + T3
    Next

    AvgT1 = SumT1 / CSng(NSamples)
    AvgT3 = SumT3 / CSng(NSamples)

'   Period1 = AvgT1 + AvgT3


    SumT1 = 0.0
    SumT3 = 0.0
    For i = 1 to NSamples
        Call PulseIn(PinX2, 1, T1)
        Call PulseIn(PinX2, 0, T3)

        SumT1 = SumT1 + T1
        SumT3 = SumT3 + T3
    Next
    AvgT1 = SumT1 / CSng(NSamples)
```

```
   AvgT3 = SumT3 / CSng(NSamples)

'   Period2 = AvgT1 + AvgT3


   SumT1 = 0.0
   SumT3 = 0.0
   For i = 1 to NSamples
      Call PulseIn(PinX3, 1, T1)
      Call PulseIn(PinX3, 0, T3)

      SumT1 = SumT1 + T1
      SumT3 = SumT3 + T3
   Next

   AvgT1 = SumT1 / CSng(NSamples)
   AvgT3 = SumT3 / CSng(NSamples)

'   Period3 = AvgT1 + AvgT3


   SumT1 = 0.0
   SumT3 = 0.0
   For i = 1 to NSamples
      Call PulseIn(PinX4, 1, T1)
      Call PulseIn(PinX4, 0, T3)

      SumT1 = SumT1 + T1
      SumT3 = SumT3 + T3
   Next

   AvgT1 = SumT1 / CSng(NSamples)
   AvgT3 = SumT3 / CSng(NSamples)

'   Period4 = AvgT1 + AvgT3


End Sub
'-------------------------------------------------------------------------------
```

Filename: SerialPort.bas

```
Attribute VB_Name = "SerialPort"
'-------------------------------------------------------------------------------
Option Explicit

' This module is used to transfer data to and from the serial port.

Private Const InputBufferSize As Integer = 13   ' 4-byte buffer.
Private Const OutputBufferSize As Integer = 10  ' 1-byte buffer.

Private InputBuffer(1 To InputBufferSize) As Byte
Private OutputBuffer(1 To OutputBufferSize) As Byte

Private Const ASCII_LF     As Byte = 10
Private Const ASCII_CR     As Byte = 13
Private Const ASCIIplus    As Byte = 43
Private Const ASCIIminus   As Byte = 45
Private Const ASCIIdecimal As Byte = 46
Private Const ASCIIzero    As Byte = 48
'-------------------------------------------------------------------------------
Public Sub OpenSerialPort( _
   ByVal PortNumber As Byte, _
   ByVal BaudRate As Long)

' Opens a serial port at the specified baud rate.

   ' Com1 requires that the network be disabled. On the BasicX-01
   ' Developer Board, it may be necessary to raise pin 14, which can be
   ' done here or in the chip I/O initialization.
   '>>If (PortNumber = 1) Then
   '>>    Call PutPin(14, bxOutputHigh)
   '>>End If

   Call OpenQueue(InputBuffer, InputBufferSize)

   Call OpenQueue(OutputBuffer, OutputBufferSize)

   Call OpenCom(PortNumber, BaudRate, InputBuffer, OutputBuffer)

End Sub
'-------------------------------------------------------------------------------
Public Sub PutByte( _
   ByVal Value As Byte)

' Sends one byte of binary data to the serial port. The byte is sent
```

```
' directly without translating it to a string.

   Call PutQueue(OutputBuffer, Value, 1)

End Sub
'-------------------------------------------------------------------------------
Public Sub GetByte( _
   ByRef Value As Byte, _
   ByRef Success As Boolean)

' Inputs a byte from the serial port, if available. Returns regardless.  The
' Success flag is set depending on whether a byte is available.
'
' The byte is in direct binary format -- it is not in string format.

   ' Find out if anything is in the queue.
   Success = StatusQueue(InputBuffer)

   ' If data is in the queue, extract it.
   If (Success) Then
      Call GetQueue(InputBuffer, Value, 1)
   Else
      Value = 0
   End If

End Sub
'-------------------------------------------------------------------------------
Public Sub NewLine()

' Outputs a <CR> <LF> to the serial port.

   Call PutByte(ASCII_CR)
   Call PutByte(ASCII_LF)

End Sub
'-------------------------------------------------------------------------------
Public Sub PutLine( _
   ByRef Tx As String)

' Outputs a String type, followed by <CR> <LF>. Output is to the serial
' port.

   Call PutStr(Tx)

   Call NewLine
```

```
End Sub
'-------------------------------------------------------------------------------
Public Sub PutStr( _
   ByRef Tx As String)

' Outputs a String type to the serial port.

   Dim Length As Integer, Ch As String * 1, bCh As Byte
   Dim I As Integer

   Length = Len(Tx)

   For I = 1 To Length
      Ch = Mid(Tx, I, 1)
      bCh = Asc(Ch)
      Call PutByte(bCh)
   Next

End Sub
'-------------------------------------------------------------------------------
Public Sub PutB( _
   ByVal Value As Byte)

' Outputs a Byte type to the serial port.

   Dim Digit(1 To 3) As Byte
   Dim i As Integer, NDigits As Integer
   Const Base As Byte = 10

   NDigits = 0

   Do
      NDigits = NDigits + 1
      Digit(NDigits) = Value Mod Base
      Value = Value \ Base
   Loop Until (Value = 0)

   For i = NDigits To 1 Step -1
      Call PutByte(Digit(i) + ASCIIzero)
   Next

End Sub
'-------------------------------------------------------------------------------
Public Sub PutHexB( _
   ByVal Value As Byte)
```

```
' Outputs a Byte type to the serial port. Hexadecimal format is used.

   Dim Digit(1 To 2) As Byte, D As Byte
   Dim i As Integer, NDigits As Integer
   Const Base As Byte = 16
   Const ASCIIhexBias As Byte = 55

   NDigits = 0

   Do
      NDigits = NDigits + 1

      D = Value Mod Base
      If (D < 10) Then
         D = D + ASCIIzero
      Else
         D = D + ASCIIhexBias
      End If

      Digit(NDigits) = D

      Value = Value \ Base
   Loop Until (Value = 0)

   For i = NDigits To 1 Step -1
      Call PutByte(Digit(i))
   Next

End Sub
'-------------------------------------------------------------------------------
Public Sub PutI( _
   ByVal Value As Integer)

' Outputs an Integer type to the serial port.

   Call PutL(CLng(Value))

End Sub
'-------------------------------------------------------------------------------
Public Sub PutUI( _
   ByVal Value As UnsignedInteger)

' Outputs an UnsignedInteger type to the serial port.

   Dim L As Long, Tmp As New UnsignedInteger
```

```
    Tmp = Value

    L = 0

    ' Copy Value into the lower two bytes of L.
    Call BlockMove(2, MemAddress(Tmp), MemAddress(L))

    Call PutL(L)

End Sub
'-------------------------------------------------------------------------------
Public Sub PutUL( _
    ByVal Value As UnsignedLong)

' Outputs an UnsignedLong type to the serial port.

    Dim UL As New UnsignedLong, L As Long, Digit As New UnsignedLong
    Dim I As Integer, Temp As New UnsignedLong

    ' If the top bit is clear, the number is ready to go.
    If ((Value And &H80000000) = 0) Then
        Call PutL(CLng(Value))
        Exit Sub
    End If

    ' Divide by 10 is done by a right shift followed by a divide by 5.
    ' First clear top bit so we can do a signed divide.
    UL = Value
    UL = UL And &H7FFFFFFF

    ' Shift to the right 1 bit.
    L = CLng(UL)
    L = L \ 2

    ' Put the top bit back, except shifted to the right 1 bit.
    UL = CuLng(L)
    UL = UL Or &H40000000

    ' The number now fits in a signed long.
    L = CLng(UL)

    L = L \ 5

    Call PutL(L)

    ' Multiply by 10. Since multiply is not implemented for UnsignedLong, we
```

```
    ' do the equivalent addition.
    Temp = CuLng(L)
    UL = 0
    For I = 1 To 10
        UL = UL + Temp
    Next

    ' Find the rightmost digit.
    Digit = Value - UL
    Call PutL(CLng(Digit))

End Sub
'-----------------------------------------------------------------------------
Public Sub PutL( _
    ByVal Value As Long)

' Outputs a Long type to the serial port.

    ' Reserve space for "2147483648".
    Dim Digit(1 To 10) As Byte
    Dim NDigits As Integer
    Dim i As Integer
    Const Base As Long = 10

    ' The working number must be zero or negative. Otherwise the negative
    ' limit will cause overflow if we take its absolute value.
    If (Value < 0) Then
        Call PutByte(ASCIIminus)
    Else
        Value = -Value
    End If

    NDigits = 0

    Do
        NDigits = NDigits + 1
        Digit(NDigits) = CByte( Abs(Value Mod Base) )
        Value = Value \ Base
    Loop Until (Value = 0)

    ' Digits are stored in reverse order of display.
    For i = NDigits To 1 Step -1
        Call PutByte(Digit(i) + ASCIIzero)
    Next

End Sub
```

```
'-----------------------------------------------------------------------------
Public Sub PutSci( _
   ByVal Value As Single)

' Outputs floating point number in scientific notation format. The format
' is such that 13 characters are always generated. Sign characters are
' included for both mantissa and exponent. Exponents have 2 digits,
' including a leading zero if necessary.
'
' Example Formats:  "+1.234567E+00"
'                   "-7.654321E-20"
'                   "+3.141593E+05"
'                   "+0.000000E+00"

   Dim Mantissa As Single, Exponent As Integer, LMant As Long

   Call SplitFloat(Value, Mantissa, Exponent)

   ' Sign.
   If (Mantissa < 0.0) Then
      Call PutByte(ASCIIminus)
   Else
      Call PutByte(ASCIIplus)
   End If

   ' Convert mantissa to a 7-digit integer.
   LMant = FixL((Abs(Mantissa) * 1000000.0) + 0.5)

   ' Correct for roundoff error. Mantissa can't be > 9.999999
   If (LMant > 9999999) Then
      LMant = 9999999
   End If

   ' First digit of mantissa.
   Call PutByte( CByte(LMant \ 1000000) + ASCIIzero)

   ' Decimal point.
   Call PutByte(ASCIIdecimal)

   ' Remaining digits of mantissa.
   LMant = LMant Mod 1000000

   Call InsertZeros(LMant)

   Call PutL(LMant)
```

```
    ' Exponent.
    Call PutByte(69)  ' E

    If (Exponent < 0) Then
       Call PutByte(ASCIIminus)
    Else
       Call PutByte(ASCIIplus)
    End If

    ' A 2-digit exponent has a leading zero.
    If (Abs(Exponent) < 10) Then
       Call PutByte(ASCIIzero)
    End If

    Call PutI(Abs(Exponent))

End Sub
'------------------------------------------------------------------------------
Private Sub InsertZeros( _
    ByVal X As Long)

    Dim NumZeros As Byte, I As Byte

    If (X >= 100000) Then
       Exit Sub               ' 100 000 <= X
    ElseIf (X >= 10000) Then
       NumZeros = 1           '  10 000 <= X <= 99 999
    ElseIf (X >= 1000) Then
       NumZeros = 2           '   1 000 <= X <=  9 999
    ElseIf (X >= 100) Then
       NumZeros = 3           '     100 <= X <=    999
    ElseIf (X >= 10) Then
       NumZeros = 4           '      10 <= X <=     99
    Else
       NumZeros = 5           '       0 <= X <=      9
    End If

    For I = 1 To NumZeros
       Call PutByte(ASCIIzero)
    Next

End Sub
'------------------------------------------------------------------------------
Public Sub PutS( _
    ByVal Value As Single)
```

```
' Outputs a floating point number to the serial port. If the number can be
' displayed without using scientific notation, it is. Otherwise scientific
' notation is used.

    Dim X As Single, DecimalPlace As Integer, Mantissa As Single
    Dim Exponent As Integer, DigitPosition As Integer, Factor As Long
    Dim LMant As Long, DecimalHasDisplayed As Boolean

    ' Special case for zero.
    If (Value = 0.0) Then
        Call PutByte(ASCIIzero)
        Call PutByte(ASCIIdecimal)
        Call PutByte(ASCIIzero)
        Exit Sub
    End If

    X = Abs(Value)

    ' Use scientific notation for values too big or too small.
    If (X < 0.1) Or (X > 999999.9) Then
        Call PutSci(Value)
        Exit Sub
    End If

    ' What follows is non-exponent displays for 0.1000000 < Value < 999999.9

    ' Sign.
    If (Value < 0.0) Then
        Call PutByte(ASCIIminus)
    End If

    If (X < 1.0) Then
        Call PutByte(ASCIIzero)    ' Leading zero.
        Call PutByte(ASCIIdecimal)
        DecimalHasDisplayed = True
        DecimalPlace = 0

        ' Convert number to a 7-digit integer.
        LMant = FixL((X * 10000000.0) + 0.5)
    Else
        Call SplitFloat(X, Mantissa, Exponent)
        DecimalPlace = Exponent + 2

        ' Convert mantissa to a 7-digit integer.
        LMant = FixL((Abs(Mantissa) * 1000000.0) + 0.5)
```

```
        ' Correct for roundoff error. Mantissa can't be > 9.999999
        If (LMant > 9999999) Then
            LMant = 9999999
        End If

        DecimalHasDisplayed = False
    End If

    Factor = 1000000

    For DigitPosition = 1 To 7

        If (DigitPosition = DecimalPlace) Then
            Call PutByte(ASCIIdecimal)
            DecimalHasDisplayed = True
        End If

        Call PutByte( CByte(LMant \ Factor) + ASCIIzero )

        LMant = LMant Mod Factor

        ' Stop trailing zeros, except for one immediately following the
        ' decimal place.
        If (LMant = 0) Then
            If (DecimalHasDisplayed) Then
                Exit Sub
            End If
        End If

        Factor = Factor \ 10
    Next

End Sub
'-------------------------------------------------------------------------------
Private Sub SplitFloat( _
    ByVal Value As Single, _
    ByRef Mantissa As Single, _
    ByRef Exponent As Integer)

' Splits a floating point number into mantissa and exponent. The mantissa
' range is such that 1.0 <= Abs(Mantissa) < 10.0 for nonzero numbers, and
' zero otherwise.

    Dim X As Single, Factor As Single

    ' Zero is a special case.
```

```
    If (Value = 0.0) Then
        Mantissa = 0.0
        Exponent = 0
        Exit Sub
    End If

    X = Abs(Value)

    Exponent = 0
    Factor = 1.0

    ' Multiply or divide by ten to transform number to value between 1 and 10.
    Do
        If (X >= 10.0) Then
            X = X / 10.0
            Factor = Factor * 10.0
            Exponent = Exponent + 1
        ElseIf (X < 1.0) Then
            X = X * 10.0
            Factor = Factor * 10.0
            Exponent = Exponent - 1
        Else
            ' When we reach this point, then 1.0 <= mantissa < 10.0.
            Exit Do
        End If
    Loop

    ' Determine mantissa.
    If (Exponent = 0) Then
        Mantissa = Value
    ElseIf (Exponent > 0) Then
        Mantissa = Value / Factor
    Else
        Mantissa = Value * Factor
    End If

End Sub
'----------------------------------------------------------------------------
```

Section 8.3 Matlab Source Code

```
%id_break
close

%set contants
seg_length = 200;
zero_factor = 2.2;
small_break_thresh_top = .135;
small_break_thresh_bot = .05;

color_var_A = [0 0 1];
color_var_B = [0 0 1];
color_var_C = [0 0 1];
color_var_D = [0 0 1];

acc_data = load('150rpm.txt');

data_set_length = size(acc_data,1);

acc_A = acc_data(:,1);
acc_B = acc_data(:,2);
acc_C = acc_data(:,3);
acc_D = acc_data(:,4);

for time_count = 1:seg_length:(data_set_length - 2*seg_length)

   % Accelerometer 1

   acc_last_A = acc_A(time_count:time_count+seg_length);
   acc_current_A = acc_A(time_count+seg_length:time_count+2*seg_length);
   norm_factor = max(max([acc_last_A acc_current_A]));

   norm_acc_last_A = acc_last_A/norm_factor;
   norm_acc_current_A = acc_current_A/norm_factor;

   max_y_axis = max(max([norm_acc_last_A norm_acc_current_A]));
   min_y_axis = min(min([norm_acc_last_A norm_acc_current_A]));

   %Find wave length

   high_temp_last_A = norm_acc_last_A-.65;
   low_temp_last_A = norm_acc_last_A-.35;

   high_zero_crossings_A = 0;
   low_zero_crossings_A = 0;
```

```
for count = 1:seg_length-1
   if (high_temp_last_A(count) < 0) & (high_temp_last_A(count+1) > 0)
       high_zero_crossings_A = high_zero_crossings_A + 1;
   end
   if (low_temp_last_A(count) < 0) & (low_temp_last_A(count+1) > 0)
       low_zero_crossings_A = low_zero_crossings_A + 1;
   end
end

zero_crossings_A = min([high_zero_crossings_A  low_zero_crossings_A]);
small_seg_length = floor(seg_length/(zero_crossings_A*0.9));

subplot(4,2,1)
H = plot(norm_acc_last_A);
set(H, 'color', color_var_A)
axis([0 seg_length min_y_axis max_y_axis])
title('Accelerometer 1 t-1')

short_peak_count = 0;
for small_seg_count = 1:small_seg_length:seg_length-small_seg_length
   small_last_A = norm_acc_last_A(small_seg_count:small_seg_count+small_seg_length);

 if abs(max(small_last_A) - max(norm_acc_current_A)) > small_break_thresh_top
       if abs(min(small_last_A) - min(norm_acc_current_A)) > small_break_thresh_bot
           short_peak_count = short_peak_count + 1;
       end
   end

  if short_peak_count >= zero_crossings_A/zero_factor
       color_var_A = [1 0 0];
       disp(['A - Break occured at ' num2str(time_count + seg_length)])
   end
end

subplot(4,2,2);
H1 = plot(norm_acc_current_A);
set(H1, 'color', color_var_A)
axis([0 seg_length min_y_axis max_y_axis])
title('Accelerometer 1 t')

% Accelerometer 2

acc_last_B = acc_B(time_count:time_count+seg_length);
acc_current_B = acc_B(time_count+seg_length:time_count+2*seg_length);
norm_factor = max(max([acc_last_B acc_current_B]));
```

```
norm_acc_last_B = acc_last_B/norm_factor;
norm_acc_current_B = acc_current_B/norm_factor;

max_y_axis = max(max([norm_acc_last_B norm_acc_current_B]));
min_y_axis = min(min([norm_acc_last_B norm_acc_current_B]));

%Find wave length

high_temp_last_B = norm_acc_last_B-.65;
low_temp_last_B = norm_acc_last_B-.35;

high_zero_crossings_B = 0;
low_zero_crossings_B = 0;

for count = 1:seg_length-1
   if (high_temp_last_B(count) < 0) & (high_temp_last_B(count+1) > 0)
       high_zero_crossings_B = high_zero_crossings_B + 1;
   end
   if (low_temp_last_B(count) < 0) & (low_temp_last_B(count+1) > 0)
       low_zero_crossings_B = low_zero_crossings_B + 1;
   end
end

zero_crossings_B = min([high_zero_crossings_B  low_zero_crossings_B]);
small_seg_length = floor(seg_length/(zero_crossings_B*0.9));

subplot(4,2,3)
H = plot(norm_acc_last_B);
set(H, 'color', color_var_B)
axis([0 seg_length min_y_axis max_y_axis])
title('Accelerometer 2 t-1')

short_peak_count = 0;
for small_seg_count = 1:small_seg_length:seg_length-small_seg_length
   small_last_B = norm_acc_last_B(small_seg_count:small_seg_count+small_seg_length);

  if abs(max(small_last_B) - max(norm_acc_current_B)) > small_break_thresh_top
       if abs(min(small_last_B) - min(norm_acc_current_B)) > small_break_thresh_bot
           short_peak_count = short_peak_count + 1;
       end
   end

  if short_peak_count >= zero_crossings_B/zero_factor
       color_var_B = [1 0 0];
       disp(['B - Break occured at ' num2str(time_count + seg_length)])
```

```
  end
end


subplot(4,2,4);
H1 = plot(norm_acc_current_B);
set(H1, 'color', color_var_B)
axis([0 seg_length min_y_axis max_y_axis])
title('Accelerometer 2 t')

% Accelerometer 3

acc_last_C = acc_C(time_count:time_count+seg_length);
acc_current_C = acc_C(time_count+seg_length:time_count+2*seg_length);
norm_factor = max(max([acc_last_C acc_current_C]));

norm_acc_last_C = acc_last_C/norm_factor;
norm_acc_current_C = acc_current_C/norm_factor;

max_y_axis = max(max([norm_acc_last_C norm_acc_current_C]));
min_y_axis = min(min([norm_acc_last_C norm_acc_current_C]));

%Find wave length

high_temp_last_C = norm_acc_last_C-.65;
low_temp_last_C = norm_acc_last_C-.35;

high_zero_crossings_C = 0;
low_zero_crossings_C = 0;

for count = 1:seg_length-1
  if (high_temp_last_C(count) < 0) & (high_temp_last_C(count+1) > 0)
     high_zero_crossings_C = high_zero_crossings_C + 1;
  end
  if (low_temp_last_C(count) < 0) & (low_temp_last_C(count+1) > 0)
     low_zero_crossings_C = low_zero_crossings_C + 1;
  end
end

zero_crossings_C = min([high_zero_crossings_C  low_zero_crossings_C]);
small_seg_length = floor(seg_length/(zero_crossings_C*0.9));

subplot(4,2,5)
H = plot(norm_acc_last_C);
set(H, 'color', color_var_C)
axis([0 seg_length min_y_axis max_y_axis])
```

```
title('Accelerometer 3 t-1')

short_peak_count = 0;
for small_seg_count = 1:small_seg_length:seg_length-small_seg_length
   small_last_C = norm_acc_last_C(small_seg_count:small_seg_count+small_seg_length);

 if abs(max(small_last_C) - max(norm_acc_current_C)) > small_break_thresh_top
      if abs(min(small_last_C) - min(norm_acc_current_C)) > small_break_thresh_bot
          short_peak_count = short_peak_count + 1;
      end
   end
   if short_peak_count >= zero_crossings_C/zero_factor
      color_var_C = [1 0 0];
      disp(['C - Break occured at ' num2str(time_count + seg_length)])
   end
end

subplot(4,2,6);
H1 = plot(norm_acc_current_C);
set(H1, 'color', color_var_C)
axis([0 seg_length min_y_axis max_y_axis])
title('Accelerometer 3 t')

% Accelerometer 4

acc_last_D = acc_D(time_count:time_count+seg_length);
acc_current_D = acc_D(time_count+seg_length:time_count+2*seg_length);
norm_factor = max(max([acc_last_D acc_current_D]));

norm_acc_last_D = acc_last_D/norm_factor;
norm_acc_current_D = acc_current_D/norm_factor;

max_y_axis = max(max([norm_acc_last_D norm_acc_current_D]));
min_y_axis = min(min([norm_acc_last_D norm_acc_current_D]));

%Find wave length

high_temp_last_D = norm_acc_last_D-.65;
low_temp_last_D = norm_acc_last_D-.35;

high_zero_crossings_D = 0;
low_zero_crossings_D = 0;

for count = 1:seg_length-1
   if (high_temp_last_D(count) < 0) & (high_temp_last_D(count+1) > 0)
      high_zero_crossings_D = high_zero_crossings_D + 1;
```

```
    end
    if (low_temp_last_D(count) < 0) & (low_temp_last_D(count+1) > 0)
        low_zero_crossings_D = low_zero_crossings_D + 1;
    end
  end

  zero_crossings_D = min([high_zero_crossings_D  low_zero_crossings_D]);
  small_seg_length = floor(seg_length/(zero_crossings_D*0.9));

  subplot(4,2,7)
  H = plot(norm_acc_last_D);
  set(H, 'color', color_var_D)
  axis([0 seg_length min_y_axis max_y_axis])
  title('Accelerometer 4 t-1')

  short_peak_count = 0;
  for small_seg_count = 1:small_seg_length:seg_length-small_seg_length
    small_last_D = norm_acc_last_D(small_seg_count:small_seg_count+small_seg_length);

  if abs(max(small_last_D) - max(norm_acc_current_D)) > small_break_thresh_top
        if abs(min(small_last_D) - min(norm_acc_current_D)) > small_break_thresh_bot
            short_peak_count = short_peak_count + 1;
        end
    end
    if short_peak_count >= zero_crossings_D/zero_factor
        color_var_D = [1 0 0];
        disp(['D - Break occured at ' num2str(time_count + seg_length)])
    end
  end

  subplot(4,2,8);
  H1 = plot(norm_acc_current_D);
  set(H1, 'color', color_var_D)
  axis([0 seg_length min_y_axis max_y_axis])
  title('Accelerometer 4 t')

  pause
end
```

Section 8.4 Accelerometer Data Sheet

This is the top page of the Analog Devices ADXL202AE Accelerometer, the entire data sheet can be viewed at:

<http://www.analog.com/UploadedFiles/Datasheets/567227477ADXL202E_a.pdf>

ANALOG DEVICES

Low-Cost ±2 *g* Dual-Axis Accelerometer with Duty Cycle Output

ADXL202E*

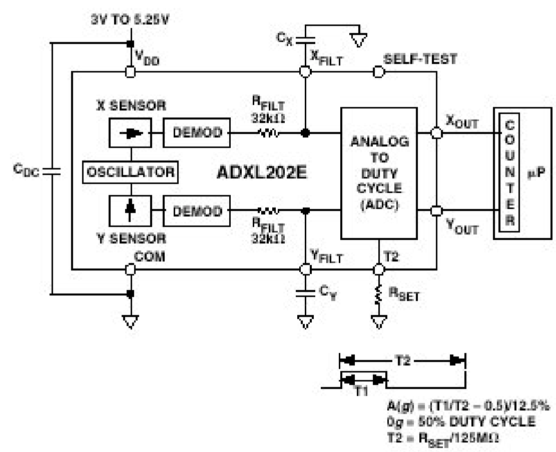
FEATURES

- 2-Axis Acceleration Sensor on a Single IC Chip
- 5 mm × 5 mm × 2 mm Ultrasmall Chip Scale Package
- 2 m*g* Resolution at 60 Hz
- Low-Power < 0.6 mA
- Direct Interface to Low-Cost Microcontrollers via Duty Cycle Output
- BW Adjustment with a Single Capacitor
- 3 V to 5.25 V Single Supply Operation
- 1000 *g* Shock Survival

APPLICATIONS

- 2-Axis Tilt Sensing with Faster Response than Electrolytic, Mercury, or Thermal Sensors
- Computer Peripherals
- Information Appliances
- Alarms and Motion Detectors
- Disk Drives
- Vehicle Security

FUNCTIONAL BLOCK DIAGRAM



GENERAL DESCRIPTION

The ADXL202E is a low-cost, low-power, complete 2-axis accelerometer with a digital output, all on a single monolithic IC. It is an improved version of the ADXL202AQC/JQC. The ADXL202E will measure accelerations with a full-scale range of ±2 *g*. The ADXL202E can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity).

The outputs are analog voltage or digital signals whose duty cycles (ratio of pulsewidth to period) are proportional to acceleration. The duty cycle outputs can be directly measured by a microprocessor counter, without an A/D converter or glue logic. The duty cycle period is adjustable from 0.5 ms to 10 ms via a single resistor (R_{SET}).

The typical noise floor is 200 $\mu g/\sqrt{Hz}$, allowing signals below 2 m*g* (at 60 Hz bandwidth) to be resolved.

The bandwidth of the accelerometer is set with capacitors C_X and C_Y at the X_{FILT} and Y_{FILT} pins. An analog output can be reconstructed by filtering the duty cycle output.

The ADXL202E is available in 5 mm × 5 mm × 2 mm 8-lead hermetic LCC package.

Figure 8.20 ADXL202AE Top page of Data sheet

Section 8.5 Motor Specifications

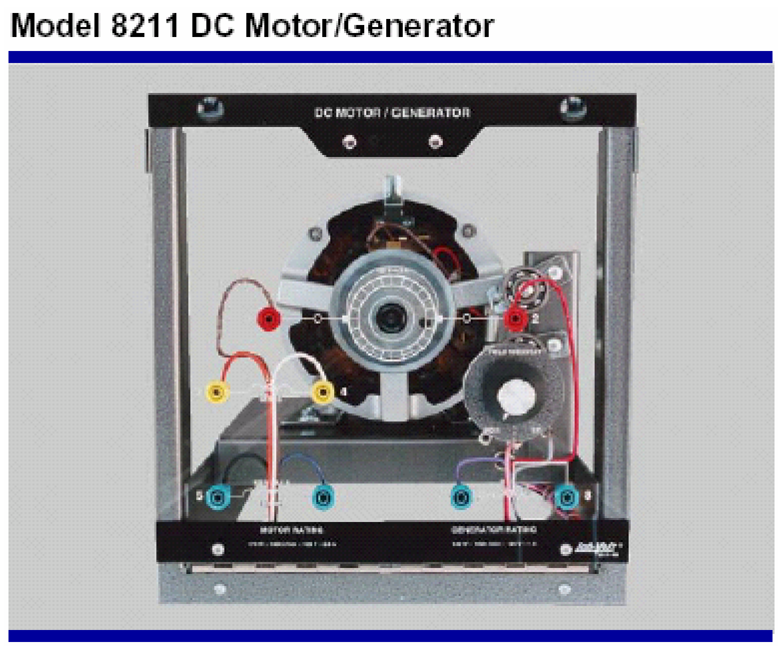


Figure 8.21 Motor picture from data sheet.

The description from the data sheet is as follows:

This machine can be run independently as a DC motor or a DC generator. The armature, shunt field, and series field windings are terminated separately on the faceplate to permit long and short shunt as well as cumulatively and differentially compounded motor and generator connections. This machine is fitted with exposed movable brushes to allow students to study the effect of armature reaction and commutation while the machine is operating under load. An independent, circuit- breaker protected, shunt-field rheostat is mounted on the faceplate for motor speed control or generator output voltage adjustment.

Table 8.1 Motor Characteristics from Data Sheet

Model 8211 DC Motor/Generator	**120/208 V ~ 60 Hz**	**220/380 V ~ 50 Hz**	**240/415 V ~ 50Hz**
Power Requirement	120/208 V	220/380 V	240/415 V
Rating			
Motor Output Power	175 W		
Generator Output Power	120 W	110 W	120 W
Armature Voltage	120 V ~ DC	220 V ~ DC	240 V ~ DC
Shunt Field Voltage	120 V ~ DC	220 V ~ DC	240 V ~ DC
Full Load Speed	1800 r/min	1500 r/min	1500 r/min
Full Load Motor Current	2.8 A	1.3 A	1.1 S
Full Load Generator Current	1 A	.5 A	.5 A
Physical Characteristics			
Dimensions (H x W x D)	308 x 291 x 440 mm (12.1 x 11.5 x 17.3 in)		
Net Weight	14.1 kg (31 lb)		

Section 8.6 Frame Dimensions

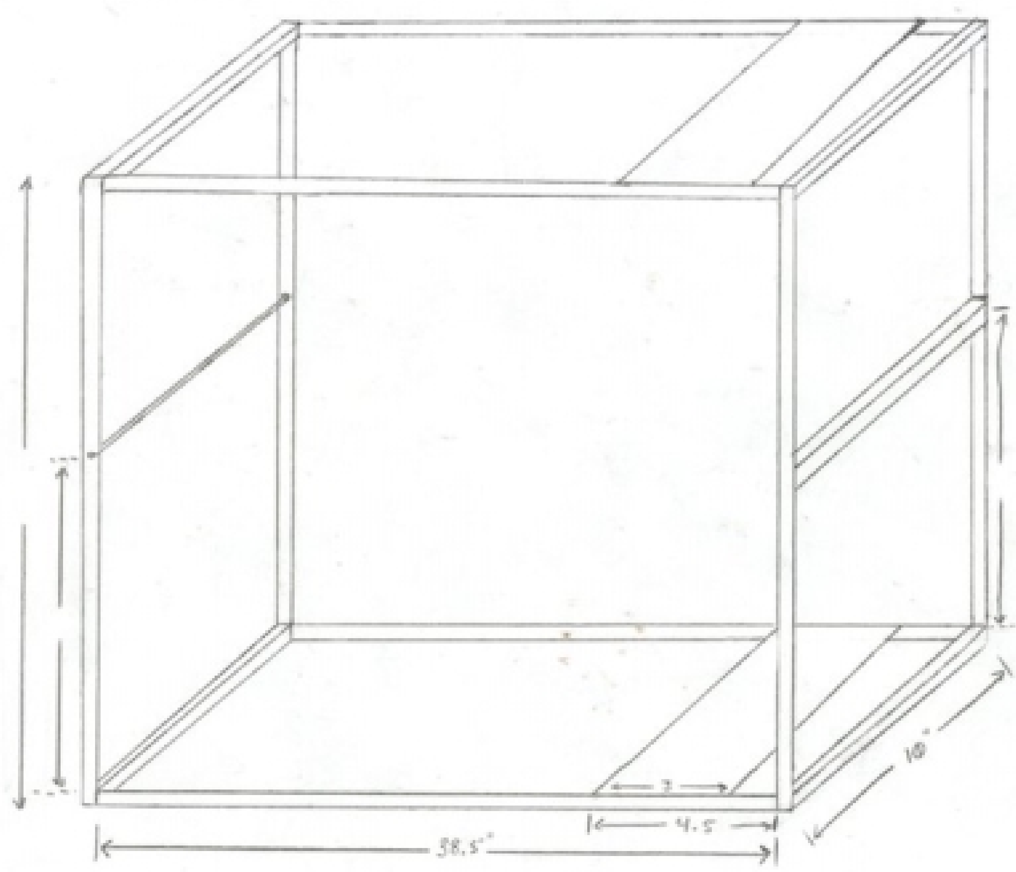


Figure 8.22 Original sketch of the frame

Table 8.2 Frame Dimensions

Length	38.5”
Width	10”
Height	30”
Height of “Comb”	20”
Height of “Warp Beam”	18.5”
Heddle platform width (top and bottom)	3”
Heddle platforms (top and bottom) distance from bottom right corner	1.5”

Section 8.7 Cam Dimensions

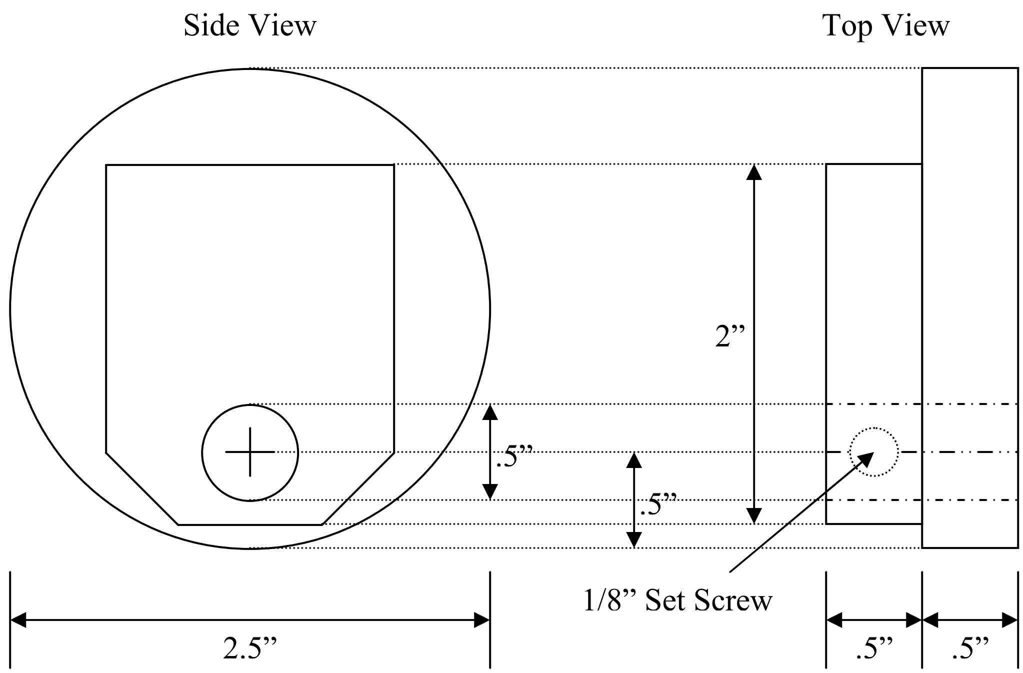


Figure 8.23 Cam Dimensions and diagram.

Section 8.8 Lever Dimensions

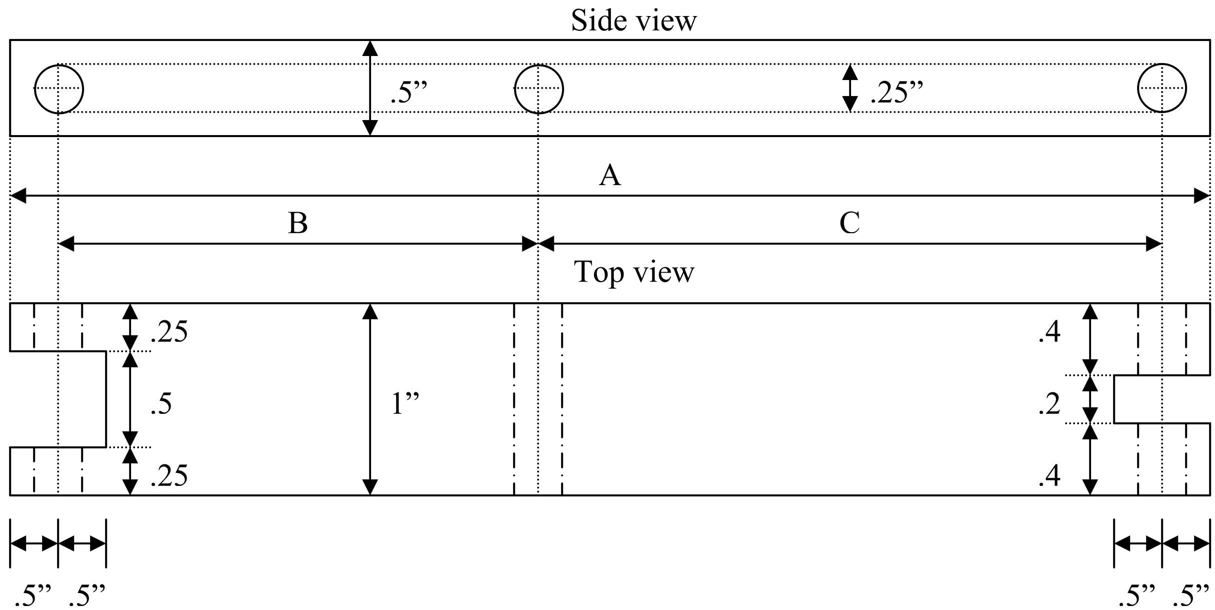


Figure 8.24 Lever Dimensions and diagram. Dimensions A, B, and C vary depending on lever position.

Table 8.3 Lever dimensions based on lever position.

Lever	Dimension A	Dimension B	Dimension C
1	9”	6.167”	2.833”
2	9.5”	6.500”	3.000”
3	10”	6.667”	3.167”
4	10.5”	7.167”	3.333”