Real-Time Health Monitoring System for Mast-Arm Sign Support Structures

Andrew David Smith
Marquette University

REAL-TIME HEALTH MONITORING SYSTEM
FOR MAST-ARM SIGN SUPPORT
STRUCTURES

by

Andrew D. Smith, B.S.

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

May 2010

ABSTRACT
REAL-TIME HEALTH MONITORING SYSTEM FOR MAST-ARM SIGN SUPPORT STRUCTURES

Andrew D. Smith, B.S.

Marquette University, 2010

There has been an ongoing study conducted by Marquette University and supported by the Wisconsin Highway Research Program that is seeking to understand the risk of fatigue-induced fracture in the connections of mast-arm sign support structures. This study has been brought on by recent problems encountered with the connections contained in, and the in-service performance of, several cantilevered mast-arm sign support structures. A resulting recommendation of phase one in this effort was to monitoring an in-service cantilevered mast-arm sign support structure.

The health monitoring system developed in this thesis effort includes a grouping of strain gages (full-bridge arrangements) positioned around the perimeter of the mast-arm tube to monitor wind-induced strain near the mast-arm connection weld toes. This data is automatically recorded (continuously) using the data acquisition system designed and the software developed. Wind speed and direction is simultaneously read through an anemometer, synched to the strain data being acquired, and stored within the DAQ system using the software developed in this thesis. Finally, the thesis includes development of algorithms for synthesizing the data for later use in the WHRP project. The health-monitoring system designed and developed in this thesis will be left in place for future use in the WHRP research effort.

ACKNOWLEDGEMENTS

Andrew D. Smith, B.S.

I would like to thank the Wisconsin Highway Research Program for funding provided to this research effort. It has allowed me to pursue an excellent educational experience. I would also like to thank the faculty at Marquette University for providing that excellent education experience.

I would especially like to thank Dr. Chris Foley, for his instruction and guidance in this research effort. Working with him has been enjoyable and rewarding. I would also like to thank Chris Nelson and the technical support staff at National Instruments, for their assistance in developing and deploying a data acquisition system. The help from Dave Newman in developing and implementing the field monitoring system is much appreciated.

To all my friends and family, thanks for your support and care. Special thanks to my parents for always being supportive, without you I would not have achieved the level of success I enjoy today. To the most important person in my life, Amanda, thank you for your love and support during this time. Lastly, I give thanks to the Lord above, for whom without, I am nothing.

TABLE OF CONTENTS

Chapter 1

Motivation for Present Work

1.1 Introduction

There has been an ongoing study conducted by Marquette University and supported by the Wisconsin Highway Research Program that is seeking to understand the risk of fatigue-induced fracture in the connections of mast-arm sign support structures. This study has been brought on by recent problems encountered with the connections contained in, and the in-service performance of, several cantilevered mast-arm sign support structures. Figure 1.1 represents a typical sign support structure found in the state of Wisconsin and Figure 1.2 shows a typical mast-arm-to-pole connection, which is the major area of concern in these structural systems. Current AASHTO design specifications (AASHTO 2001) entail provisions for fatigue design, which are intended to address the tendency for fatigue-induced cracking in these structures. However fatigue-induced fracture is still not well understood and there are currently many in-service sign structures that were designed before these provisions took effect.

The WHRP-sponsored research effort as a whole has been tasked with the following: (a) implement state-of-the-art fatigue reliability analysis and current knowledge regarding fatigue lives of connections in a systematic assessment of fatigue-induced fracture risk in WisDOT’s sign support structures; (b) recommend the most effective retrofit strategies in instances where fatigue induced fracture is likely; and (c) assign inspection cycle frequencies for these structures and their components.

To meet the tasks listed above, the research has been, and is being conducted in phases. Each phase contains its own set of tasks and goals. Each successive phase is directed by the findings and results of the previous phase. Phase 1 was completed in December of 2008 and the results were submitted in a report to the Wisconsin Highway Research Program - WHRP (Foley *et al.* 2008). A brief review of the Phase 1 report can be found in Chapter 2 of this thesis.

The first phase of the WHRP effort laid the groundwork for defining what it meant to "quantify" risk and what would be needed to accomplish this. It involved an extensive review of previous studies conducted in the realm of fatigue-induced fracture. It also involved gathering a large amount of existing wind data, a synthesis of existing fatigue test data for connections similar to those found in the structures under consideration, and included FE simulation of the behavior of typical 4-bolt mast-arm-to-pole connections with several conditions of bolt pretensioning. Phase 1 assembled statistical information that was already available, and determined what further information would be needed to fill in the gaps. This additional information would be obtained by fatigue testing under Phase 2 of the research effort and field monitoring of a sign support structure described in this thesis.

1.2 Motivation, Outline and Contributions of Thesis

Recommendations in Phase 1 included assembly of existing laboratory testing data for a number of mast arm specimens at given stress ranges, and recommendations regarding monitoring an in-service cantilevered mast-arm sign support structure based upon wind speed histories accumulated for Wisconsin and future supplemental fatigue testing. As of this writing, the laboratory fatigue testing recommended will be conducted at a later time. The objectives of this thesis effort are to establish a field monitoring station for an in-service sign support structure and establish protocols for gathering and synthesizing this data for use in phase 3 in the WHRP effort.

This includes procuring the equipment needed to conduct the instrumentation and long term monitoring of the mast-arm sign support structure selected. It includes discussion of the data acquisition (DAQ) software and hardware needed and the necessary programming required to acquire and process field data. The thesis also outlines laboratory testing required to verify proper functioning of devices and sensors utilized in the monitoring station. The field deployment of the system is also described including setting up equipment such as the anemometer tower, power supply, sensors, and conduit for wiring.

The health monitoring system developed in this thesis effort includes a grouping of strain gages (full-bridge arrangements) positioned around the perimeter of the mast-arm tube to monitor wind-induced strain near the mast-arm connection weld toes. This data is automatically recorded (continuously) using the data acquisition system designed and the software developed. Wind speed and direction are simultaneously read through an anemometer, synced to the strain data being acquired, and stored within the DAQ system using the software developed. Finally, the thesis includes development of algorithms for synthesizing the data for later use in the WHRP project. The health-monitoring system designed and developed will be left in place after for future use in the WHRP research effort.

Since there is currently insufficient understanding with regard to how mast-arm signal and sign support structures perform under long-term natural wind-induced loading, the data obtained from the health monitoring system and data synthesis algorithms developed through this thesis will make significant contributions towards this understanding. It will also make contributions to the overall research effort in seeking to better understand the risk of fatigue-induced fracture; better estimating the longevity; establishing appropriate and substantiated inspection frequencies; and better quantifying life-cycle cost of these types of structures. It is hoped that this thesis will also provide a template for others to use in future structural health monitoring projects.

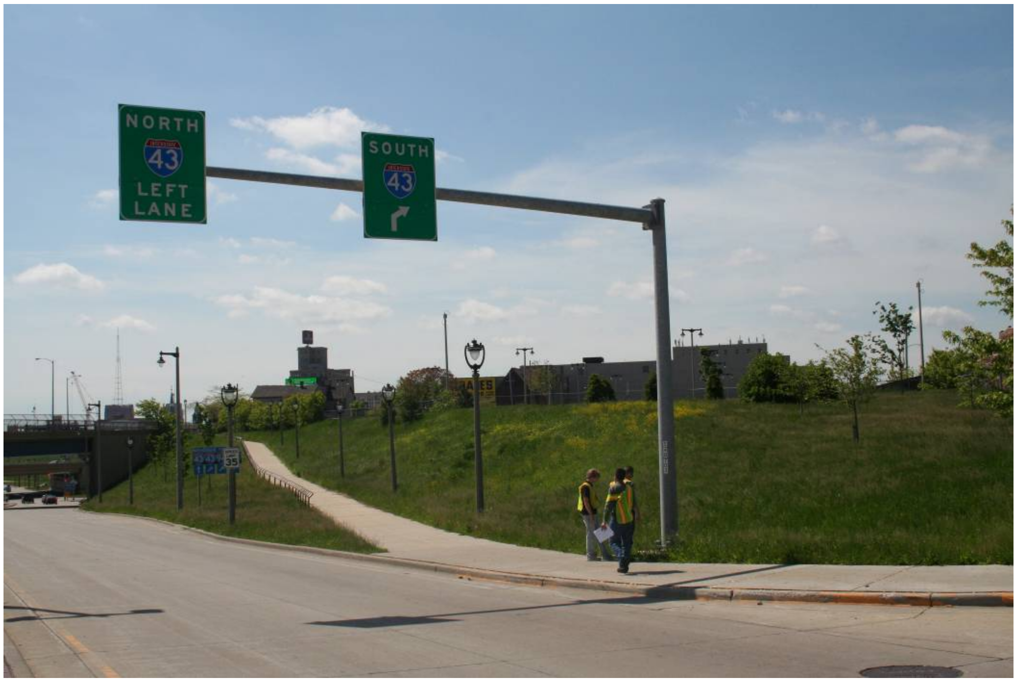


Figure 1.1 Typical Cantilevered Sign Support Structure

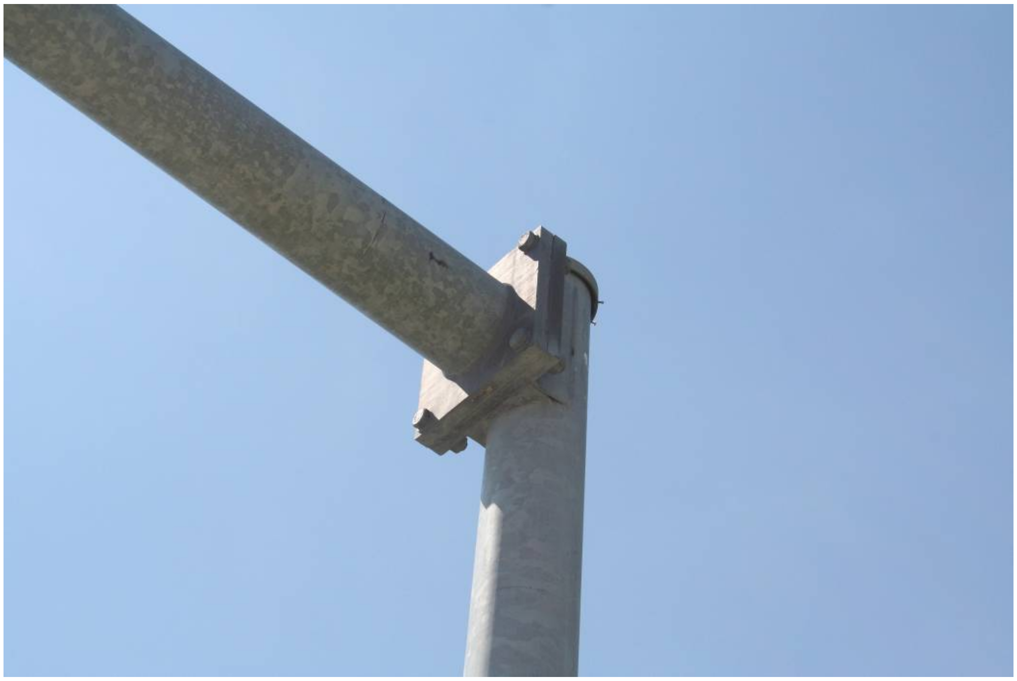


Figure 1.2 Typical Mast-Arm-to-Pole Connection

Chapter 2

Literature Review

2.1 Introduction

Support structures, including cantilevered sign and signal supports, while seemingly simple in design and purpose, have been a sustained area of research. There is much uncertainty in the fatigue life of these structures due the environment in which they exist, their manufacture, and the loading they are subjected to. These structures are often manufactured with fatigue-sensitive details and spend their in-service lives undergoing highly variable and relatively chaotic loading conditions. This literature review focuses on previous projects that involved monitoring of structures similar to that considered in this thesis for the reasons of understanding fatigue-induced fracture. It also serves to enlighten the reader as to the nature of fatigue and why it is such an active area of study and debate even though codes, such as the recent AASHTO specifications (AASHTO 2001), have provisions for addressing fatigue design.

The following sections in this chapter outline recent research efforts closely related to the objectives of this thesis. The review covered in section 2.4 is phase 1 of the WHRP research effort (Foley *et al.* 2008) for which the present thesis effort contributes. The report for phase 1 of the WHRP effort (Foley *et al.* 2008) has an extensive review of prior fatigue studies, some of which are discussed below in a more in-depth manner in this thesis to highlight efforts of field instrumentation and monitoring. It is recommended that if the reader wishes a more extensive review of topics related to fatigue-induced fracture he/she should refer to this report.

2.2 Mast-Arm Monitoring (Alderson 1999); (Chen *et al.* 2003)

The Missouri Department of Transportation (MDOT) discovered failures in over a dozen cantilevered traffic signal mast-arms in the six years previous to 1999. Failures occurred mostly in a particular manufacture's structures within less than three years of service. The failures predominantly occurred in the top of the welded mast-arm-to-pole connection.

The research team conducted testing of two in-service specimens to measure strain near the connection during various conditions. Selection of the specimens was based on: proximity to research team's base of operation; selection of mast-arms from two different fabricators to determine whether they exhibited similar behavior and whether the manufacturer's quality control might have contributed to the premature failures; selection of mast-arms of two different lengths to compare/contrast dynamic properties and the susceptibility of the mast-arms to "galloping," or presence of large vertical stresses; and the presence of a relatively open area without interference from trees and buildings while being exposed to high speed traffic, especially trucks.

The mast-arm structure selected was a 54-foot long cantilevered traffic signal support structure fabricated by Valmont Industries. The mast-arm had a round cross-section which supported four signals and three signs. This particular specimen had been in service since December of 1997 (within 2 years of the study), and existed along a route posted at 50mph with some truck traffic present. A second specimen selected was a 42-foot long cantilevered traffic signal support structure fabricated by JEM. This mast-arm had an octagonal cross-section which supported three signals and two signs. This particular specimen had been in service since 1986 (10+ years of service), and existed along a route posted at 40mph also with some truck traffic present.

The 54-foot structure was modeled using SAP2000 to determine the analytical modes of vibration and corresponding natural periods. The first mode shape had the mast-arm bending out-of-plane (horizontal motion) with twisting about the pole, which corresponded to a natural period of 1.34 seconds (0.75 Hz). The second mode shape was an in-plane “hatchet”, or vertical motion with a corresponding 1.28sec period (0.78 Hz).

The field monitoring included utilization of 12 strain gauges per specimen. Four transverse and four longitudinal gauges were placed four inches from the weld joint with an additional four gauges placed 18-feet and 15-feet (specimen 1 and specimen 2, respectively) from the welded joint. An accelerometer was placed on the tip of each mast-arm. Testing was conducted during special events; one being truck-induced loading and the second being on days with the wind out of a favorable direction (normal to the longitudinal direction of the mast-arm and perpendicular to the sign face).

Due to a lack of truck traffic and the constant interruption of traffic flow by the traffic signals, the “boom-truck” shown in Figure 2.1 was used by the researchers to mimic the effect of large trucks passing beneath the mast-arm at the posted speed limit. After conducting a number of tests, the researchers were unsuccessful in their attempts to create a large response in the system using the boom-truck. Peak stress values of 0.97 to 1.06 ksi were observed for a total of six tests with the boom-truck. The largest stresses at the connection recorded during these controlled tests were in the horizontal plane, or parallel with the road. Since, large stress cycles were not observed and the fact that fatigue cracking primarily occurs on the top along the weld connection, it was surmised that truck-induced loading is probably not the cause for premature failures.

The constant amplitude fatigue limit, or CAFL for this type of structure is 2.6 ksi (AASHTO 2001). Stress cycles exceeding this value would reduce the fatigue life of the structure. While monitoring the mast-arm during periods of relatively high winds, the stress ranges within the mast-arm exceeded the CAFL 35.1% of the time. A graphical summary of the maximum recorded stress ranges and their corresponding velocities for both sign structures is shown in Figure 2.2. Verification by the accelerometers showed the largest stresses to be in the horizontal plane. “Galloping” of the mast-arm was not observed during these periods of high wind. The researchers concluded that without continuous monitoring of the mast-arm behavior and its load history, it could not be determined whether the percentage of all stress cycles exceeding the CAFL is large enough to produce fatigue damage. Since the largest stresses occurred in the horizontal plane, and not in the vertical plane where fracture typically occurs, it was concluded that natural wind gusts do not appear to be the primary cause of fatigue damage.

2.3 High-Mast Lighting Towers Monitoring (Connor and Hodgson 2006)

During the month of November 2003 a 140-foot high-mast lighting pole (HMLP) collapsed in Sioux City, Iowa. The investigative study that followed discovered fatigue cracks and loose anchor bolts on many other HMLPs. The research resulting from the investigation was conducted in two phases. Both phases were designed to be similar to one another and therefore allow for comparison.

Phase 1 involved the long-term monitoring of two high-mast towers. The first tower was an “as-built” tower, un-cracked and not retrofitted. The second tower was a “retrofit” tower, which had a steel splice jacket at the pole base. As a second part to phase

1, ten specimens were chosen to conduct pluck-testing on, to determine the natural frequencies and damping characteristics by free decay vibration. Phase 2 of this research involved two more specimens. The first was identical to the failed tower and the second was another retrofit tower. These two specimens were also field tested to obtain the dynamic characteristics and stress magnitudes at critical locations to be compared with the static and dynamic behaviors of the original and retrofit towers of Phase 1.

Instrumentation was done using strain gauges at various locations around the base of the as-built tower. Anemometers were mounted at various heights on the poles. Data was streamed via a satellite internet connection. Stress range histograms were recorded every ten minutes using a rainflow cycle counting algorithm for a total of 347 days. The rainflow analysis tallied the number of stress cycles for a given stress range, while excluding any stress ranges below 0.5 ksi. The rainflow analysis was done for each individual strain gauge around the base perimeter. Each gauge corresponds to a channel, the location of which is described in Table 2.1. This was done so that fatigue life estimates could be calculated for specific areas of the base connection. The resulting “as-built” tower stress histogram for the various channels can be seen in Figure 2.3, which shows a variance in stress cycles and number of cycles at the individual gauges around the perimeter. Also recorded were stress-time histories when triggered by an event, such as large wind gusts. 3-minute average wind speed and direction were recorded continuously for a total of 275 days.

The long-term monitoring study discovered that stresses in the “as-built” towers were much larger that those found in the “retrofit” towers. HMLP socket connections fall into category E’ per AASHTO specifications (AASHTO 2001), which have a CAFL of

2.6 ksi. The variable amplitude stress ranges were represented by an equivalent constant-amplitude stress range using Miner's rule. The research team suggested that if 0.01% of stress cycles exceed the CAFL it is considered damaging to the fatigue life of the structure. Data obtained from the long-term study showed that at some gauge locations the cumulative stress ranges exceeded the CAFL 0.04% of the time. Stress ranges below one-quarter of the CAFL were removed from consideration of the fatigue life analysis.

2.4 WisDOT Sign Support Structures (Foley *et al.* 2008)

Quantifying risk depends heavily on having statistical information for modeling all pertinent random variables including those that effect demand (i.e. wind loading) and capacity (i.e. fatigue life of weld connection). Risk allows one to associate probability with failure which can be used to set appropriate inspection and retrofit cycles.

This recently completed research effort was primarily concerned with determining whether presently available data obtained from prior research efforts would be enough to "quantify" risk or whether data obtained from further testing would be necessary. Sign support structures are unique and have been the focus of so much research because significant levels of variability are introduced through fabrication and loading of these structures. Adequately assessing the risk of fatigue-induced failure of these structures requires a thorough understanding of both the fabrication procedures and in-service loading of these structures.

Historically, there have been six major suppliers of signal and sign support structures in the state of Wisconsin. All suppliers provide similar structures as they are based on Wisconsin standards, but the state specifications do allow for some flexibility in design and manufacture

The first part of the research included review of prior studies involving fatigue testing of structures similar to those found in Wisconsin. After review of these prior studies, several conclusions/recommendations were made. First, the variability in fatigue life seen in the fatigue testing results for characteristic connections completed to date could be modeled using lognormal cumulative distribution functions (CDFs). It was also emphasized that low stress ranges are particularly susceptible to high levels of variability in expected fatigue life. The experimental data reviewed clearly illustrated that as the stress range decreased, the variability in fatigue life significantly increases. Detailed statistical analysis was undertaken to make recommendations for additional testing of mast-arm specimens at two target stress ranges. These additional tests were shown to provide significantly enhanced understanding in the uncertainty in fatigue life of un-stiffened mast-arm connections.

A second portion of this research effort involved collecting and synthesizing large amounts of wind history data. Ten contiguous years of 2-minute averaged hourly wind speed and direction data were collected for eight major cities located in Wisconsin. This data was obtained from the National Oceanic and Atmospheric Administration (NOAA) Automated Surface Observation Station (ASOS) sites. The data was then synthesized to produce useful statistical representations for wind speed, wind direction, the probability of wind speeds conditioned on direction. Figure 2.4 illustrates an example of the synthesized data describing the probability of occurrence for a given wind speed in Milwaukee, WI. Figure 2.5 provides a wind rose illustrating the probability that wind will be out of a given direction. Figure 2.6 is a similar representation of wind direction probability binned using the traditional eight cardinal directions. The resolution change in the wind rose is apparent upon comparison of these two figures.

Of most importance in characterizing the variability in demand on sign support structures is the combined probabilities of speed and direction. The probability data illustrated in Figures 2.4 and 2.6 were further synthesized to produce conditional probability data. Table 2.2 provides the conditional probability of a defined wind speed coming from a specific direction. For example, there is a 2.98% chance that at any given time the wind speed will be 5 mph out of the North.

Evaluation of wind speeds and directions in the eight chosen cities resulted in the recommendation that Green Bay or Milwaukee, WI be used as locations for field instrumentation. It was found that these two cities exhibited a large 2-min (averaged) wind speed coming from many of the cardinal directions. As a result, the present thesis work selected a sign support structure within the city of Milwaukee as the target for instrumentation and monitoring. The wind data obtained from this current study will be synthesized and packaged in a similar manner to the wind speed and direction probability models just discussed.

2.5 Recommended Revisions to Current Design Specifications (Hosch and Fouad 2009)

The current U.S. design specifications (AASHTO 2001) contain provisions for fatigue-related design of sign support structures. These provisions do not directly take into account the individual dynamic properties of the structures, which can vary greatly from specimen to specimen. Site-specific wind speed variability is also not accounted for in the design specifications. As a result, it is not uncommon for the design life of these structures to be over- and/or under-estimated.

Those conducting this study proposed a method for calculating the fatigue design load for sign support structures, based on the structure's individual dynamic properties. A finite element analysis was done to validate the results of their method. The formulation of their procedure is not reviewed here as it does not pertain to the topic at hand and the reader should refer to the authors' published work for a full understanding of their proposed method. Pertinent to this discussion is the findings of the study.

Their conclusion, based on the results, was that the AASHTO specifications will underestimate the fatigue load of structures with damping ratios less than 2%, and conversely overestimate when greater than 2%. The specifications will also underestimate the fatigue load for cases in which the natural frequency is less than 2Hz. This effectively demonstrates the need to have individual dynamic properties considered in determining the fatigue life of sign support structures.

Certainly the structure being studied in this thesis has low-damping characteristics as is typical of these types of structures, but without additional testing the magnitude of damping cannot be determined. It will be shown however that the frequencies of vibration corresponding to mode 1 and 2 of the structure are less than 2 Hz.

2.6 Synthesis of the Literature and Concluding Remarks

Review of the selected prior research studies has shown that fatigue-induced fracture has been, and still is, an issue in the design of sign/signal support structures and high-mast lighting towers that contain fatigue-sensitive details. While provisions have been made to more effectively account for the presence of these types of details, the appropriateness of these design provisions (AASHTO 2001) is still being questioned.

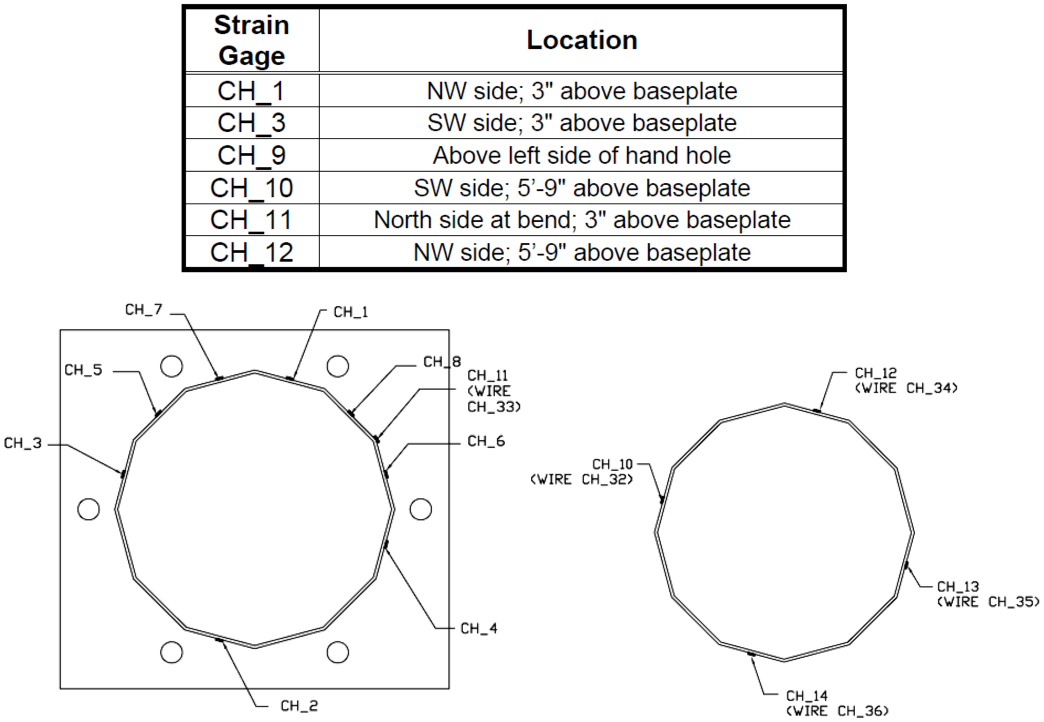
The research conducted by the University of Missouri-Columbia over ten years ago (Alderson 1999) is arguably the most similar to this current study in terms of what was being monitored. The review of that previous study gives good understanding of what this study can expect in terms of general dynamic properties of cantilevered mast-arm support structures, such as modal shapes and natural frequencies; and also an indication as to what range of strain readings might be seen in the sign structure monitored using the system developed in this thesis. This research effort also provides insight into the design of the instrumentation and data acquisition system used in the health monitoring system for the target sign structure in this study. This former study lacked recommendations related to synthesizing large amounts of long-term continuous wind and strain data for later statistical assessment for quantifying the fatigue-induced fracture risk associated with these types of structures. The present thesis hopes to make a contribution in this regard.

Previous research related to instrumentation and monitoring of high mast light poles (Connor and Hodgson 2006), while different in design and purpose than the structure of the current study, provides insights into how to conduct long term monitoring and what types of conclusions and recommendations can be made based on long-term data acquisition. Presentation of data obtained from this current long-term monitoring project will likely take a similar approach to that done in this former study.

The phase 1 report of the WHRP-sponsored effort (Foley *et al.* 2008) is the foundation for this thesis project and was reviewed to provide the reader with an understanding of how the site location and specimen selection were arrived at. It also provides a large overview of the ultimate tasks and goals of the research project, and how this thesis project is a subset of a larger study. Phase 1 included an in depth analysis of wind data for the Milwaukee area and it will be interesting to compare those wind patterns with those specific to the site and measured using the monitoring system designed and installed as a result of this thesis.

A recently completed effort (Hosch and Fouad 2009) provides questions regarding the validity of current AASHTO provisions (AASHTO 2001) for design of sign and signal support structures containing fatigue-sensitive details. Analysis of analytical models within this study showed that the current provisions can result in uneconomical designs in some cases, and put other designs at high risk for reduced fatigue lives. This review shows that some of the current assumptions within the provisions need to be validated and that there is a significant lack of understanding (and data) of how these structures behave long-term. This current study will make significant contributions to understanding the long-term behavior and whether these underlying assumptions are valid based on the data acquired during this study.

Ultimately, this thesis project is concerned with the development of an effective structural health monitoring system that will provide long-term data regarding the expected fatigue lives of mast-arm sign support structures and the risk of fatigue-induced fracture at critical details within these structural systems. The review of prior research will help in developing this system. This thesis will also provide tools for presentation and analysis of the data that can be used over the longer period of data acquisition which will go on after this thesis project is concluded. Some of the methods for presenting and analyzing data in prior research will serve as a starting point for developing such tools tailored to this current study.



Strain Gage	Location
CH_1	NW side; 3" above baseplate
CH_3	SW side; 3" above baseplate
CH_9	Above left side of hand hole
CH_10	SW side; 5'-9" above baseplate
CH_11	North side at bend; 3" above baseplate
CH_12	NW side; 5'-9" above baseplate

Table 2.1 Location of HMLP Strain Gages (Connor and Hodgson 2006)

		Wind Direction									
		0	North	Northeast	East	Southeast	South	Southwest	West	Northwest	SUM
2-Minute Averaged Wind Speed (mph)	0 mph	0.07077	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.07077
	5 mph	0.00000	0.02984	0.03008	0.02825	0.02971	0.05679	0.04610	0.05640	0.04995	0.32711
	10 mph	0.00000	0.04698	0.04363	0.02738	0.03821	0.04256	0.05817	0.05981	0.04996	0.36670
	15 mph	0.00000	0.02758	0.02299	0.01424	0.01791	0.01421	0.03177	0.02816	0.02600	0.18285
	20 mph	0.00000	0.00650	0.00538	0.00378	0.00328	0.00232	0.00844	0.00659	0.00549	0.04179
	25 mph	0.00000	0.00180	0.00103	0.00093	0.00078	0.00032	0.00168	0.00150	0.00085	0.00890
	30 mph	0.00000	0.00013	0.00007	0.00034	0.00015	0.00006	0.00052	0.00021	0.00015	0.00162
	35 mph	0.00000	0.00001	0.00000	0.00009	0.00000	0.00000	0.00012	0.00002	0.00001	0.00024
	40 mph	0.00000	0.00000	0.00000	0.00001	0.00000	0.00000	0.00000	0.00000	0.00001	0.00002
	45 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00001	0.00000	0.00001
	50 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	55 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	60 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	65 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	70 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	75 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	80 mph	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
	SUM	0.07077	0.11284	0.10317	0.07501	0.09003	0.11627	0.14680	0.15270	0.13241	1.00000

Table 2.2 Combined Probabilities, P[*Speed* | *Direction*], for Milwaukee, Wisconsin for Period January 1998 through December 2007 (Foley *et al.* 2008)

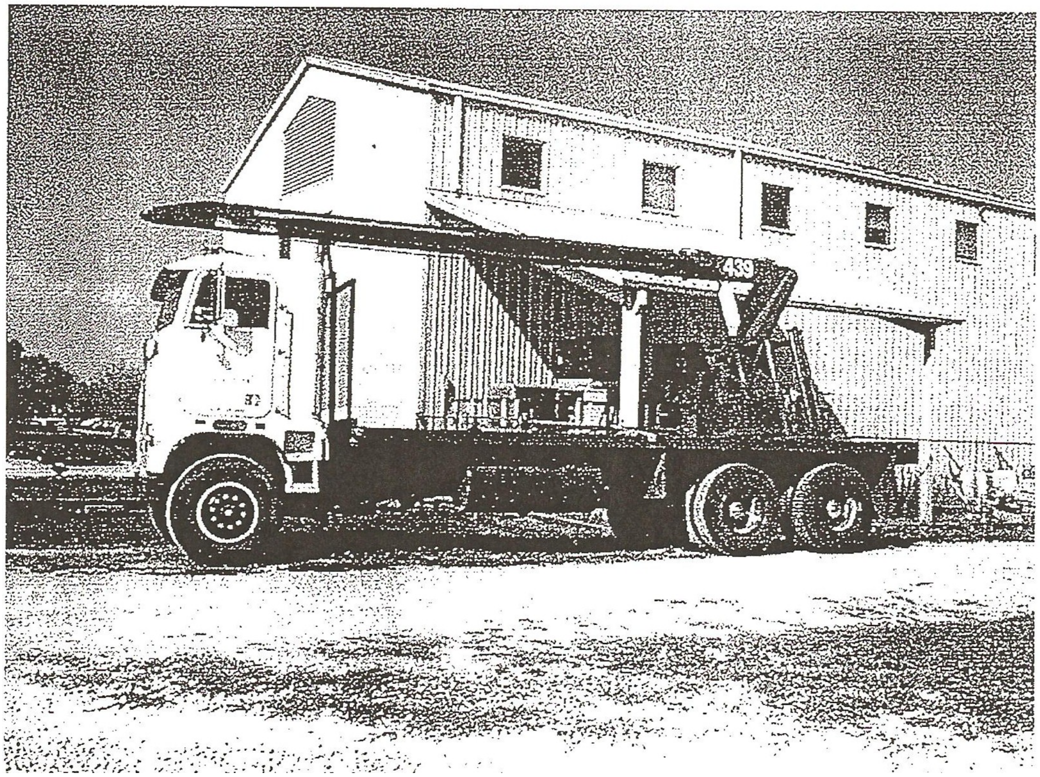


Figure 2.1 Boom Truck (Alderson 1999)

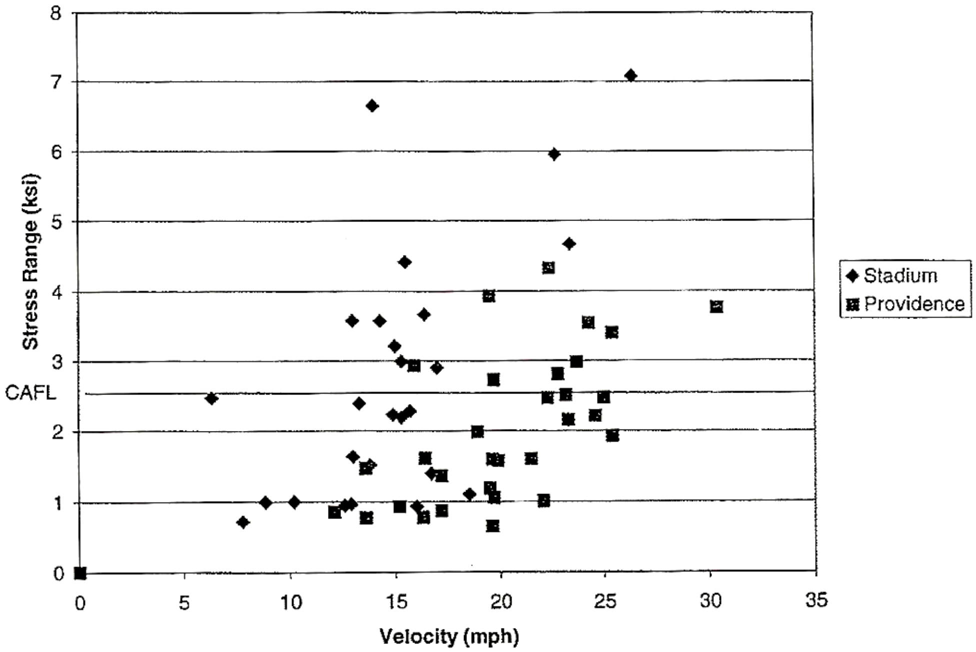


Figure 2.2 Summary Plot of Stress Range vs. Wind Velocity (Alderson 1999)

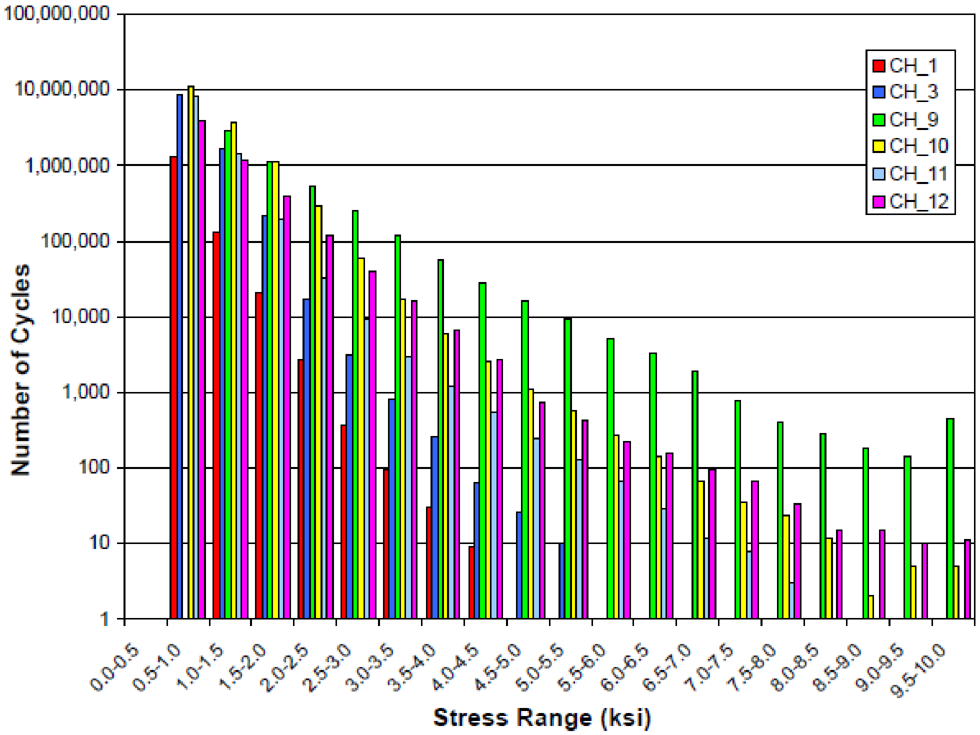


Figure 2.3 Stress-range histogram (Connor and Hodgson 2006)

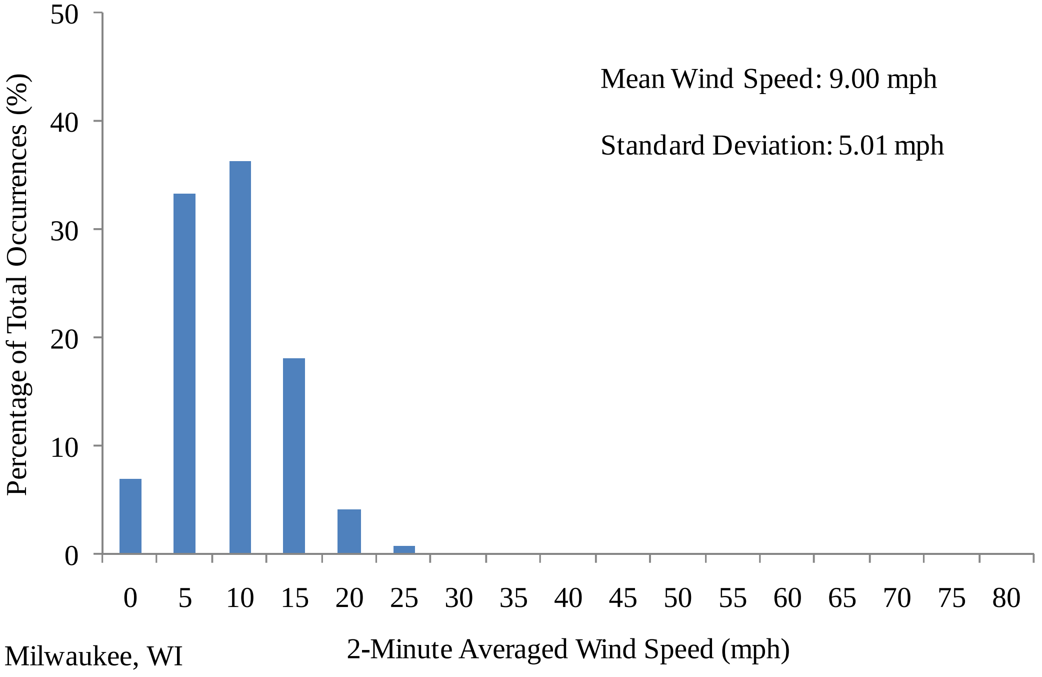


Figure 2.4 Probability Mass Function for Wind Speed Irrespective of Direction for Milwaukee, Wisconsin for Period January 1998 through December 2007 (Foley *et al.* 2008)

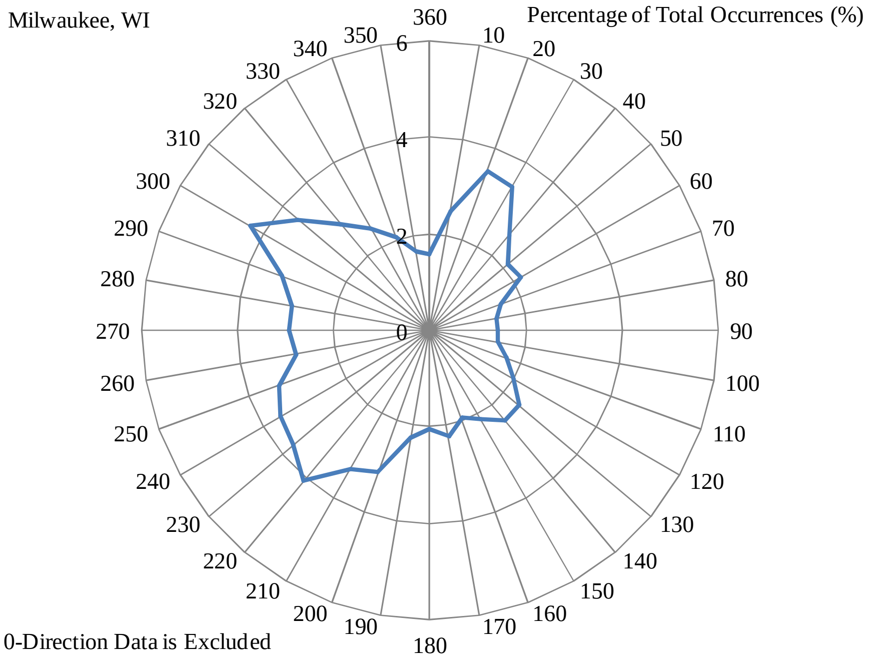


Figure 2.5 Probability Mass Functions for Wind Direction Irrespective of Speed for Milwaukee, Wisconsin for Period January 1998 through December 2007. (Foley *et al.* 2008)

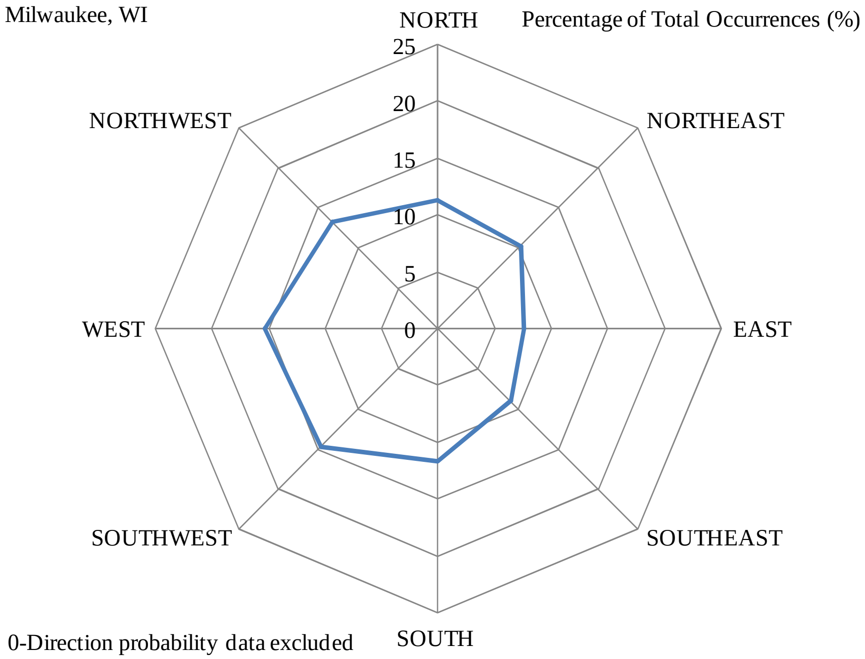


Figure 2.6 Probability Mass Functions for Wind Direction Irrespective of Speed for Milwaukee, Wisconsin for Period January 1998 through December 2007. (Foley *et al.* 2008)

Chapter 3

Field Monitoring System

3.1 Introduction

Phase 1 of an on-going WHRP research effort (Foley, *et al.* 2008) indicated that the cities of Green Bay, WI and Milwaukee, WI would be preferable locations to instrument a sign support structure based on the natural wind patterns of the area. Since the research is being conducted by Marquette University faculty, students, and staff, it was ideal to locate a sign support structure in the metro-Milwaukee area. Other considerations included a relatively obstruction free area in which the sign was located to avoid an excessively turbulent wind stream hitting the structure and the signs that it supports.

The sign chosen for monitoring is WisDOT S-40-703 (Figures 3.1 and 3.2). This sign structure is located in Milwaukee, Wisconsin just south of the intersection of Walnut Street and Fond du Lac Avenue (Figure 3.3). The sign is located in an area relatively free of wind-stream obstructions in the near vicinity of the sign. Fond du Lac Avenue runs northwest/southeast and Walnut Street runs east/west. Due to the odd directions in which the roads converge at the intersection there are large grassy areas and parking lots on virtually all sides of the sign support structure. There is, however, a city maintenance storage yard to the southwest of the sign, which is for the most part an open area with the exception of a single building. The size of the upwind obstructions and their locations relative to the sign support structure and the anemometer tower used to measure wind speed will be discussed in later sections of this chapter.

It was important to select a sign support structure that provided a good representation of those typically found in the state of Wisconsin. The Wisconsin DOT was contacted to obtain the design and shop drawings for S-40-703. Examples of these are given in Figures 3.4 and 3.5.

The chosen sign-support structure is a cantilevered monotube structure and was designed and manufactured by Valmont Industries of Valley, NE. It was installed as part of the Marquette Interchange project in 2006 and currently supports two overhead signs each measuring 4 feet 6 inches wide by 6 feet high. The mast-arm itself is a round hollow structural shape with an outer diameter of 11 inches at the connection and a nominal wall thickness of 3/16”. The mast-arm tapers to 6.38 inches at the free end. The material comprising the tube is ASTM A595 Gr. A steel. The mast-arm extends 33 feet from its connection to the vertical pole. The first sign is located 18 feet 6 inches from the vertical pole centerline. The second sign is located 12 feet from the centerline of the first sign.

The mast-arm-to-pole connection consists of the mast-arm welded to a field-bolted plate, which is bolted to the mounting plate on the vertical pole. Both plates are 1.75 inches thick. The mast-arm tube is socketed to a depth of 1-9/16 inches. The connection plate is composed of A36 steel, and measures 18.5” square with 1-5/16-inch diameter bolt holes at the four corners at an equal edge distance of approximately 1 inch. The mast-arm tube is welded to the connection plate along two perimeters. An asymmetrical fillet weld with a horizontal fillet leg dimension of 0.25 inches and an extra vertical leg weld pass to make the vertical leg of the weld 0.44 inches long connects the tube to the front face of the connection plate. A 3/16” fillet weld connects the tube to the plate along the backside where it rests in the socket. The connection plate is then bolted using four 1.25-inch diameter bolts to the equally thick mounting plate which is connected to the vertical pole with gussets.

This sign support structure selected represents the typical mast-arm-pole support structure found in Wisconsin and is ideal for instrumentation due to its proximity to the Marquette University campus and its relative openness to wind.

3.2 Monitoring System Overview

The monitoring system includes strain gauges, an anemometer, an aluminum weather station tower, solar panel for marine-battery charging, an enclosure for a battery and solar power regulator, and an enclosure for data acquisition software and hardware. The solar panel, battery, and solar power regulator were provided by the Wisconsin Department of Transportation. Sensors for obtaining data include a Gill Windsonic 2D sonic anemometer, and eight Vishay Micromeasurements 350-ohm weldable strain gauges. A National Instruments CompactRIO data acquisition hardware chassis and National Instruments 9237 full-bridge conditioning card, and LabVIEW constitute the data acquisition system.

The steps involved in instrumentation of the sign support structure include: setting up the support tower, running conduit, mounting, and wiring up equipment. The strain gauges are positioned around the mast arm perimeter in a manner that is suitable for measuring bending strains in a full bridge conditioning arrangement. The strain gauges are located near the mast-arm-to-plate connection weld toes. Wind speed, direction, and bi-axial bending will simultaneously be read. This monitoring system will allow for data to be automatically recorded (continuously). The system will be powered using a 12V marine battery which will be charged using a solar panel mounted on the weather tower.

3.3 Strain Gauges

Vishay Micromeasurements offers a type of strain gauge that is specifically designed for long term use in outdoor environments (Figure 3.6). These gauges are shipped from the manufacturer attached to a metal carrier and are then mounted to a steel specimen using a spot welder. The gauges are pre-connected with lead wires at the factory and are also factory coated with a protective layer to increase their durability and increase their reliability in harsh environments. These gauges have a nominal resistance of 350 ohms. These gauges are ideal for the field instrumentation.

In all previous incidents of sub-standard performance of these types of structures, WisDOT inspectors found fractures occurred in the mast-arm–to-plate connection along the toes of welds. Since strain normal to the connection is the basis for determining the fatigue life of these connections, the research team is interested in monitoring the normal strain due to *bending* at this location on the mast-arm due to naturally occurring wind loading.

Strain is a measure of deformation, specifically the change in length of fibers within an object relative to their original length due to an applied state of stress. Strain gauges measure this change via change in electrical resistance due to the lengthening or shortening of the resistive element within the gauge that is bonded to the component. As the wires within a gauge are elongated the resistance will increase, conversely as they shorten the resistance will lessen. This change in resistance is proportional to the strain within the member. Strain however, is often a very small value, typically measured in microstrain (strain x 10^6) and can be affected by such factors as temperature and lead wire resistance. Accounting for such small changes in resistance is most commonly done by placing strain gauges in a group called a Wheatstone bridge configuration which has four resistive legs and is wired to a voltage excitation source, V_{EX} (Figure 3.7). The Wheatstone bridge circuit is essentially two halves, left and right, with the two resistors comprising each half, acting in parallel. The measured voltage difference, V_O, between the left and right halves is given by the following equation:

$$V_O = \left(\frac{R3}{R3 + R4} - \frac{R2}{R1 + R2}\right) \cdot V_{EX} \qquad (3\text{-}1)$$

When R1/R2 = R4/R3, V_O will be equal to zero and the bridge circuit would be considered *balanced.* When strain is present it will deform the gauges, such that the bridge will no longer be balanced. The resulting voltage output, V_O will be taken as a ratio with respect to the excitation voltage, V_{EX}. This voltage ratio, Vr in units of V/V can then be used to calculate strain.

The Wheatstone bridge can be configured multiple ways and can consist of one, two, or four *active* resistors. The full bridge configuration has all four gauges in the Wheatstone bridge as active resistors and this results in increased sensitivity of the circuit by increasing the measured output. Increased sensitivity is ideal for the current setup because there is over 200 feet of leadwire connecting the gauges to the DAQ device which results in a voltage drop due to the wire resistance.

A four active gauge circuit (i.e. a full bridge) has several configuration possibilities that include or exclude certain factors in measurement. Since the research team is interested in measuring bending strain, a *Full Bridge Type II* configuration (www.ni.com) was chosen. A diagram of this configuration is shown in Figure 3.8. Gauge 3 and 4 lie longitudinally along the beam (or mast-arm) to measure bending strain, whereas gauges 1 and 2 lie transverse to the principle axis of bending and measure the Poisson's effect of the material. Gauges 1 and 4 are placed on the top face, while gauges 2 and 3 are placed on the bottom face. The advantages of this configuration are: 1) rejection of axial strain, 2) compensation for temperature effects, 3) compensation for lead-wire resistance, and 4) compensation for principle strain measurements through accounting for the effect of Poisson's ratio for the material. When a material is stretched in one direction there will be a tendency to shrink in the transverse direction, this is the Poisson effect. Since strain gauges rely on the expansion or contraction of the resistive area, this configuration compensates for the small changes of area in the transverse direction that will occur in the bending gauges. This is not expected to be very large for the present scenario.

Compensation of the temperature effect and lead-wire resistance is possible in this configuration by nature of having four active gauges in the bridge circuit. All four gauges are exposed to the environment, and have essentially the same length of wire attached to each individual gauge. Consequently, temperature affects each gauge equally, and lead wire resistance in each gauge will be nearly the same magnitude.

The relationship between voltage ratio, Vr, and strain for the full bridge configuration chosen is given by:

$$\text{Strain}, \varepsilon = \frac{-2 \cdot V_r}{GF \cdot (1 + \nu)} \tag{3-2}$$

where, GF is the gauge factor - nominally a manufacturer assigned value (approximately equal to 2.0) for the given set of gauges; and ν is Poisson’s ratio – which without material testing is taken as 0.3 for steel. Figure 3.8 illustrates use of equations (3-1) and (3-2) in a case of downward bending of a beam. Gauge 4 will experience tensile strain, which results in the gauge elongating thereby increasing the resistance of that gauge. On the reverse, side gauge 3 will be compressing and lessening its resistance. Ignoring any change in the Poisson gauges, this resistance imbalance of gauge 3 compared with gauge 4 will yield a negative value for equation (3-1) and consequently the voltage ratio, Vr. A negative value used in equation (3-2), will result in positive strain or tension in the top fibers.

Two sets of four gauges in full bridge type II configuration were placed on the mast-arm. A diagram of this placement can be seen in Figure 3.9, and is viewed along the mast-arm toward the free end. The first set of gauges, grouped and described as channel F1, were placed along the vertical axis of the mast-arm to measure strain due to vertical motion or bending about major (x) axis of the mast-arm. The second set of gauges (F2), were placed along the horizontal axis to measure strain resulting from horizontal motion or bending about the minor (y) axis of the mast-arm. The four active gauges in a bridge circuit (F1 or F2) do not result in four separate strain readings, but results in a single output, V_O, which is the basis for strain calculation. The strain readings from these two sets of gauges can then be used to calculate the resultant strain and its location along the mast-arm’s cross-section perimeter (this will be discussed later). A detailed discussion on the reference standard for interpretation of results (i.e what type of bending corresponds to positive strain values) can be found in section 5.4.

Strain Gauge Mounting

Welding gauges to the mast-arm was done using a spot welder rented from the strain gauge manufacturer (Vishay). The portable welding unit is shown in Figure 3.10. Placement of the gages is illustrated in Figure 3.9, as viewed along the mast-arm toward the free end. The F1 gauges are mounted on top and bottom of the mast-arm tube, with the F2 gauges mounted on the left and right extreme. To get a close approximation of true top, right, bottom and left a level and square were used. The square was held to the mast-arm and rotated until the level indicated the square was vertical or horizontal. These tangent points were then marked. A grinder was used to remove the galvanizing (zinc) coating and level out any imperfections on the surface. A cloth was then used to wipe down the area to remove any excess residue. The level and square were used again to mark the exact placement of the gauges. Electrical tape was used to hold the gauges in place while being spot welded to the mast-arm. The gauges intended to read bending strain were placed 2” from the mast-arm connection plate. The Poisson’s gauges were placed perpendicular to and 1” further from the bending gauges.

The metal carrier to which the actual gauges are mounted measures 0.4 inches by 0.8 inches and is composed of stainless steel. The manufacturer (Vishay) recommends spot welding patterns as shown in Figure 3.11 where there are two rows of spot welds on either side of the gauge offset by 1/32” with spot weld longitudinal spacing of 1/16”. After the gauges were welded in the field they were sealed with silicone for added protection. The wires were then run to the DAQ enclosure on the tower. Before leaving, the resistance of the gauges plus 200 feet of lead wire were measured. All gauges registered 361.5 +/- 0.4 ohms which was deemed satisfactory. The gauges themselves are rated for 350 ohms; the remainder is due to the resistance of the wire runs which will be taken into account in the DAQ hardware and software. The sign support structure in its instrumented state can be seen in Figure 3.12.

National Instruments Definable Bridge Module (NI 9237)

The National Instruments (NI) module 9237 (Figure 3.13) conditioning module is a full bridge completion module designed specifically for Wheatstone bridge circuit configurations. The module provides signal conditioning and power to the gauges, as well as providing the ability to shunt calibrate. The NI 9237 directly measures the voltage across the bridge, which then converts this to volts per volt of excitation, by dividing both sides of equation 3-1 by the excitation voltage supplied to the bridge (currently set to 3.3V). The 9237 conditioning module directly outputs the voltage ratio, Vr, which the developed DAQ program will read in for calculation of strain using equation 3-2.

The gauges welded to the mast-arm are individually wired and run back to the DAQ enclosure using CAT5 cable. The wires enter the enclosure and are wired directly to a terminal strip. This is physically where the gauges are wired into the Wheatstone bridge configuration with four nodes (1-4) as shown in Figure 3.14. The top end of the terminal strip is wired to an RJ50 cable, which then plugs directly into the NI 9237 module. The RJ50 cable is physically responsible for connecting the four nodes of the Wheatstone bridge circuit to the correct pin assignments. These pin assignments correspond to the excitation, voltage sense, remote sense, and shunt calibration leads. Figure 3.14 shows a simplified conceptual wiring diagram, where Figure 3.15 is the actual field wiring diagram. The diagrams of Figures 3.14 and 3.15 result in the same wiring, just presented differently. The *excitation* (EX) pins provide the power to the bridge. The *voltage sense* (AI) pins read the voltage difference in the bridge. Wiring to the *remote sense* (RS) and *shunt calibration* (SC) pins provide a means to improve the accuracy of strain readings.

Wire can introduce significant resistance to the system which causes a voltage drop through the bridge; this is a source of *gain error*. Remote sense is a feature built into the NI 9237 which continuously and automatically corrects for gain errors in excitation leads. Shunt calibration is a programmatically set feature that can correct for these gain errors and error within the arms of the bridge.

3.4 Anemometer

The anemometer selected for this project was a Gill Windsonic 2-D Sonic anemometer shown in Figure 3.16. This anemometer has a unique feature in that is has no moving parts, unlike the combined 3-cup anemometer with wind-vane one might typically visualize. It uses two orthogonally oriented transducers to sense horizontal wind speed and its corresponding direction (Gill 2010). This type of setup was chosen for the present research effort as it seemed better suited to resist malfunctioning resulting from icing and snow accumulation during the winter months.

Configuration

The Gill anemometer used for the present research effort implements an RS-232 interface and can be configured to a number of different settings. Currently the anemometer is configured to a sampling rate of 4Hz. Wind speed is configured to output data in units of miles per hour. The manufacturer rates the accuracy of velocity readings as +/- 2% of reading. Wind direction is represented using polar coordinates with a resolution of 1 degree. Accuracy is listed as +/- 3 degrees. A reading of 0 degrees indicates a wind out of the north, 90 degrees is out of the east, and so forth.

Anemometer configurations can be set using a PC that supports an RS-232 port. The Windows program *HyperTerminal* provides the easiest means of communication and configuring the anemometer. The proper port settings for the anemometer are as follows: Baud Rate: 38400, Data Bits: 8, Parity: None, Stop Bits: 1, Flow Control: None. Once the port settings are configured, data should begin scrolling on the HyperTerminal white screen at the set output rate. Configuration mode is entered by typing “ * ” Once configuration is complete, one can return to measuring mode by typing “ Q ” and pressing “ Enter ”. While in configuration mode, commands can be entered to change settings such as the output rate and units. A list of these commands can be found at the manufacture's website: *http://www.gill.co.uk/products/anemometer/windsonic.htm* The commands are found in the product documentation pages within the website.

Location

A 20-foot aluminum weather station tower was purchased from Campbell Scientific for the main purpose of providing an elevated platform to mount the anemometer, which would give it more accurate wind readings away from trees and other obstructions. Figure 3.17 provides a good picture of the tower. The small black object mounted at the top of the tower is the anemometer.

The tower, and ultimately the anemometer mounted on top of the tower, needed to be placed in a location where the surrounding buildings, trees, etc…, would minimally disturb the natural wind stream. A survey was done of the site and is shown in Figure 3.18. The triangle on the survey indicates the location of the tower which sits roughly 160-feet south of the sign support structure. The tower sits on a gradual hill elevated about 10-feet above the sign. The specific location for the tower was chosen because it had good line-of-sight to the sign support structure. The trees present on the site made it difficult to place the tower further east which would make trenching difficult when it came time to put conduit in the ground between the tower and sign. The anemometer height is 33 feet (10 meters) above the ground with respect to the sign support structure's base elevation.

The anemometer mounted on the tower sits roughly 22 feet above the tower base. The manual (Campbell Scientific 2010) recommends that all obstructions be at a minimum distance equal to 10 times the obstruction height from the anemometer. This guideline unfortunately is not entirely feasible in urban location chosen for monitoring. Several trees are planted on the site in the near vicinity of tower, but are not tall enough to significantly interfere (if at all) with the anemometer's function of measuring free-stream wind speed and direction.

The site is relatively free of obstructions. Open spaces exist in all directions except the westerly direction, as shown in Figures 3.19-3.21. A public works maintenance yard with a single building is located immediately west of the tower across 13th Street as shown in Figure 3.22. The nearest building in this complex is hexagonal in shape and has a peaked roof much like a steeple. The building sits 90 feet from tower and covers a front of 36 degrees measured from one edge of the building to the other. The roof edge sits nearly level with the anemometer, and from there slopes to a peak. Fortunately, this building does not present a large cross-sectional area to block wind. It is anticipated that this building will not affect westerly wind speed readings significantly. There is another building to the southwest of the tower (upper left corner of Figure 3.18). This building is much less obstructive as it is 172 feet from the tower. The area projected toward the tower covers approximately 15 degrees of potential wind directions

Alignment

The anemometer contains a 0-degree indicating mark (i.e. north indicator) installed by the manufacturer and this marking is calibrated in such a manner that the sensor reads zero degrees when detecting wind applied normal to the marker face. This marker needs to be aligned with true north. This poses a challenge as the anemometer is mounted 20 feet in the air. Ideally, if a lift was available this could be used to hoist a person within range of the anemometer for alignment with the aid of a compass. A lift was not available, so an alternative method was conceived and implemented.

True north is the geographical north direction. Magnetic north, as the needle on a compass indicates, approximates true north. The degree of accuracy in magnetic north relative to true north varies with location. In fact, the deviation of magnetic north from true north is the magnetic declination, or the angle between the two directions. Magnetic declination for any location can be found at the following U.S. government website:

http://www.ngdc.noaa.gov/geomagmodels/struts/calcDeclination.

Magnetic declination for Milwaukee, WI is 3° 25' W (west) changing by 0° 4' W/year. In other words, magnetic north is 3° 25' west of true north.

The tower upon which the anemometer is mounted consists of three legs in the form of an equal-side triangle. If it is assumed that the center point of this triangle is along the centerline of the tower and anemometer mounted above, the base could be used to align the anemometer. This is, in fact, the assumption made to align the anemometer in the present study. The tower base is bolted to supports imbedded in concrete and can be lowered down on the back two legs by unbolting the front leg from the support (Figure 3.23).

When the tower is laid on the ground it creates a "line" oriented in some direction (which happens to be an approximate north-south line). A total station is set up over the center point of the triangle and turned toward magnetic north using a compass; the angle is set to zero on the instrument. The instrument is then turned to the anemometer mounted at the far end of the tower. Two things are check at this point. First, when sighting the anemometer north marker, it should be centered on the sight's vertical cross hair, so that when the tower is raised back up, it will be aligned with the triangular base's front leg and not turned slightly off to one side. Second, the angle transited from magnetic north to the tower “line” is recorded. This was found to be 189°50’ away from magnetic north (Figure 3.24). Consequently the angle between magnetic north and the front leg of the tower is that angle minus 180° or 9°50’. Finally, since we are interested in true north we subtract the declination of 3°25’ to get an angle of 6°25’ between the direction the anemometer faces and true north. Instead of trying to rotate the anemometer to true north, 6°25’ will be added to the acquired data during post-processing. If the data indicates a wind direction out of the north (zero degrees) it will really be 6°25’.

3.5 Data Acquisition Hardware

The data acquisition (DAQ) hardware device was purchased from National Instruments (NI). Their equipment has been, and is being, used in various capacities within the College of Engineering at Marquette University. National Instruments uses LabVIEW programming software as the foundation for their equipment. NI offers a data acquisition system called the CompactRIO (cRIO), which is a chassis for data signal conditioning modules and is shown in Figure 3.13. The cRIO, is a rugged, versatile device, ideal for remote applications in harsh environments (NI 2010). A unique aspect to the cRIO is its ability to accept various modules used for processing different types of signals, like the NI 9237 conditioning module described earlier. These modules are swappable and can be removed or added at anytime. The cRIO also has a controller which acts as a processor, much like a computer, for running the DAQ system. The cRIO also incorporates an embedded field-programmable gate array, or FPGA which allows for advanced programming features.

Data Retrieval

Data is stored locally on the cRIO's 4GB hard drive in an ASCII txt file format. These files can be retrieved using an ethernet cable by connecting a laptop computer to the ethernet port "1" on the front panel of the cRIO as shown in Figure 3.25. Once the connection is made, open "My Computer" by either double clicking the icon on the laptop desktop or by opening it via the windows menu. Once open, locate the file path menu along the top of the open window and type the IP address of the cRIO hardware device: *ftp://192.168.0.2/data/.*

Any data files written to the cRIO's hard drive can be found here, which can then be copied over to the laptop's hard drive or deleted from the cRIO altogether. It should be noted that the current data acquisition program written for this thesis and loaded on the cRIO does *NOT* have a file management feature. This means that someone must manually delete old files on the cRIO to make room for new ones. Under current conditions, data space required is approximately 47MB per day or 329MB per week of continuous data acquisition.

3.6 Supporting Elements

The tower, as described in Section 3.4 of this chapter and shown in Figure 3.17, serves as the central location for all of the supporting elements necessary to carry out the research effort and field monitoring. A solar panel and two enclosures – one for the data logger, the other for a battery and solar regulator (Figures 3.26 and 3.27) are mounted on the anemometer tower. Both enclosures were mounted roughly 6 feet above ground level.

This was done to avoid any tampering by outsiders, yet renders the enclosures easily accessible from the back of a pickup truck.

Flexible conduit was used to connect all of the various elements and enclosures mounted on the tower together. The strain gauge wires had to be run 160 feet from the sign support structure to the tower. It was determined that the best way to protect these wires was to bury them inside PVC conduit. A trencher was rented to run a shallow trench from the tower to the sign structure. PVC was used for the conduit, which came in 10’ sections and was glued together as they were placed into the trench (Figure 3.28). A string was fed through each section as it was put into the trench which would allow the team to pull wires through the conduit. At the sign structure end, the conduit comes up out of the ground and runs to an access box that the team mounted onto the sign structure (Figure 3.29). On the opposite end, conduit comes out of the ground and connects to the bottom of the DAQ enclosure (Figure 3.26)

All the power equipment (e.g. solar panel and 12-volt marine battery) was obtained from the Wisconsin DOT and it had been used in a prior WisDOT deck-truss monitoring project. Power is supplied via a 12V marine battery, which is housed in one of the enclosures. The battery is charged using a solar panel which is regulated by a small *Sunsaver* solar regulator. The solar panel required mounting that could be tilted to allow for an optimum positioning for sunlight (Figure 3.30). This was accommodated using hinge pieces. A guideline to follow for determining tilt of the panel during the winter months, is to take the latitude of the site, multiply by 0.9 and add 30 degrees to that number (The Energy Grid 2010). Milwaukee sits at approximately latitude 43 degrees, resulting in an optimum tilt of 69 degrees, referenced to the horizontal plane. In addition to the tilt angle, the solar panel should be positioned facing south. Based on the recommendations, the solar panel was tilted to 69° and mounted on the south face of the tower.

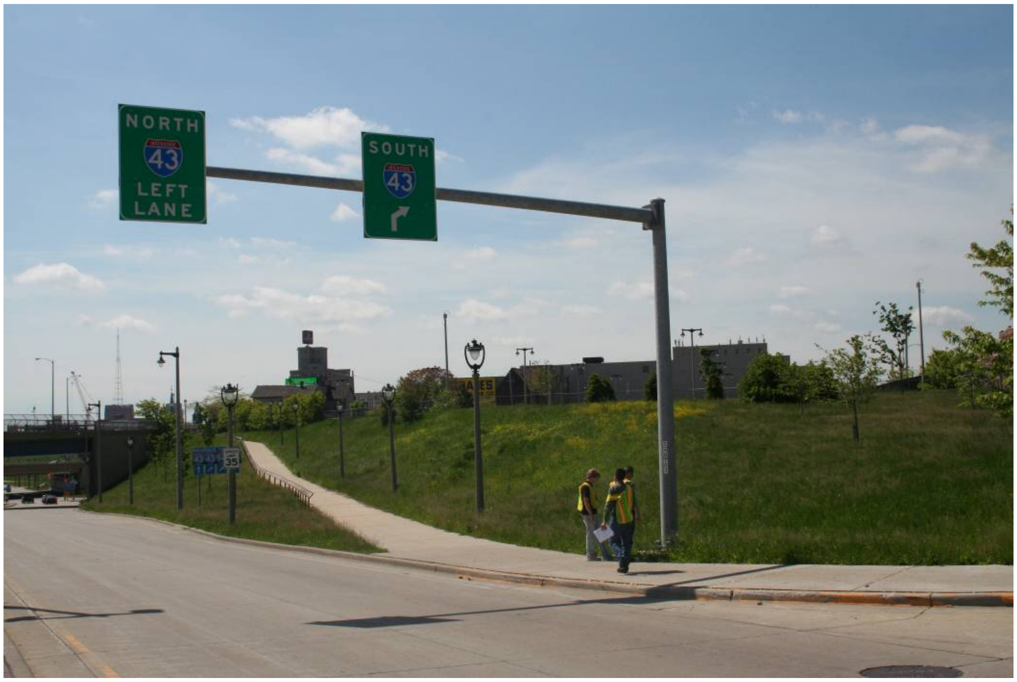


Figure 3.1 Sign S-40-703 in Milwaukee, Wisconsin.

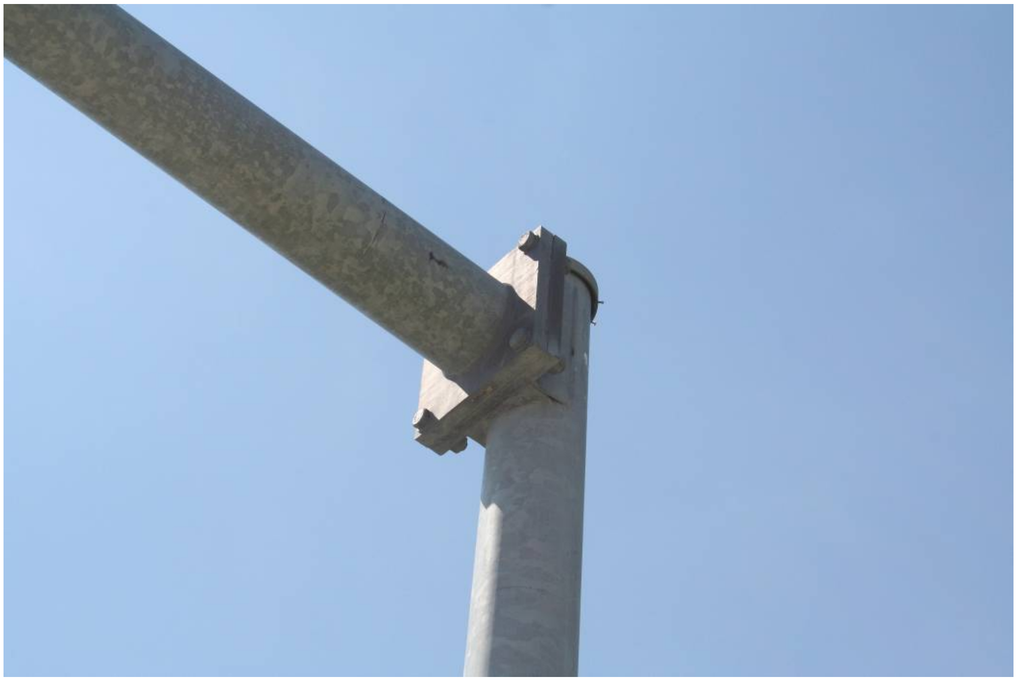


Figure 3.2 Mast-Arm-to-Pole Connection Detail

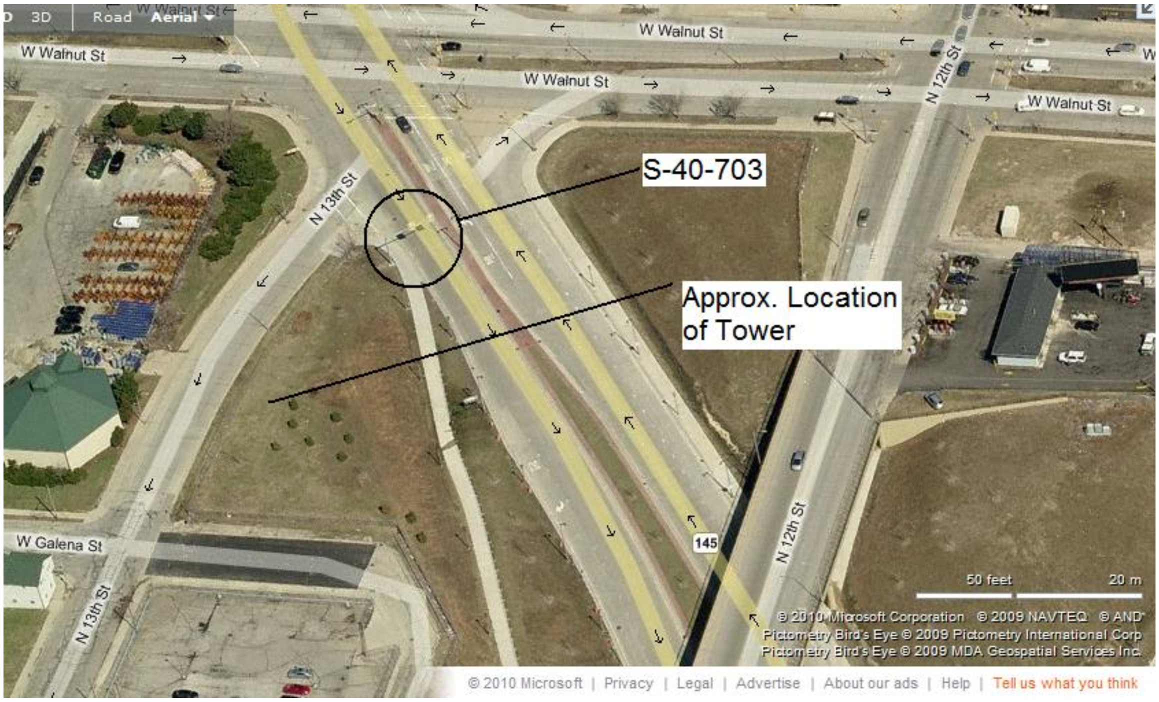

3D
Road
Aerial
W Walnut St
W Walnut St
W Walnut St
W Walnut St
N 12th St
N 13th St
S-40-703
Approx. Location
of Tower
W Galena St
N 13th St
145
N 12th St
50 feet
20 m
© 2010 Microsoft Corporation © 2009 NAVTEQ © AND
Pictometry Bird's Eye © 2009 Pictometry International Corp
Pictometry Bird's Eye © 2009 MDA Geospatial Services Inc.
© 2010 Microsoft | Privacy | Legal | Advertise | About our ads | Help | Tell us what you think

Figure 3.3 Sign Support Structure Location at 13th Street and Galena in Milwaukee, Wisconsin (www.bing.com/maps)

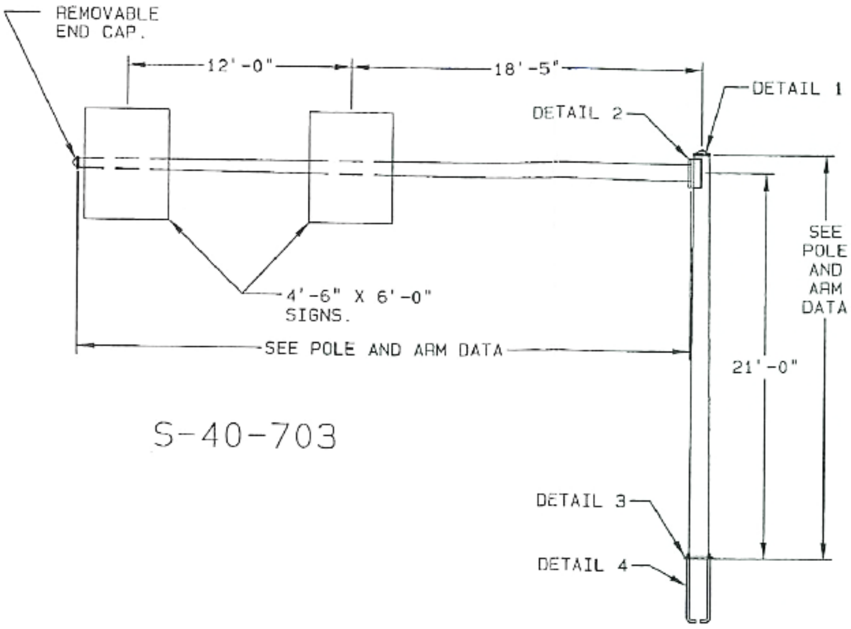
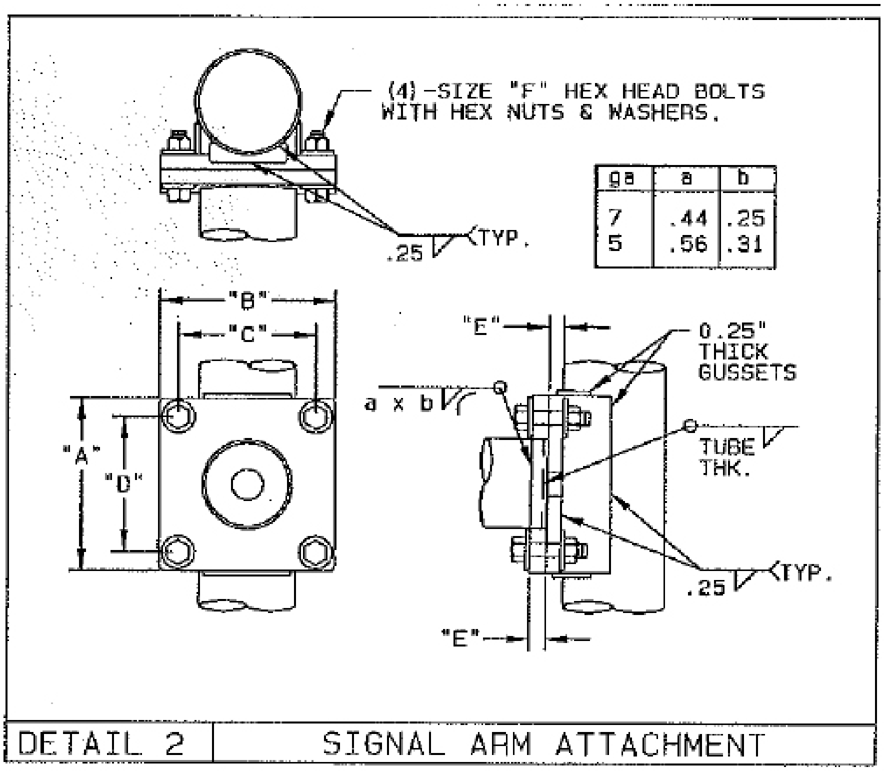


Figure 3.4 Elevation View of Sign Support S-40-703 and Attached Signs (Valmont 2004)



ga	a	b
7	.44	.25
5	.56	.31

Figure 3.5 Mast-Arm-to-Pole Connection Detail Schematic (Valmont 2004)

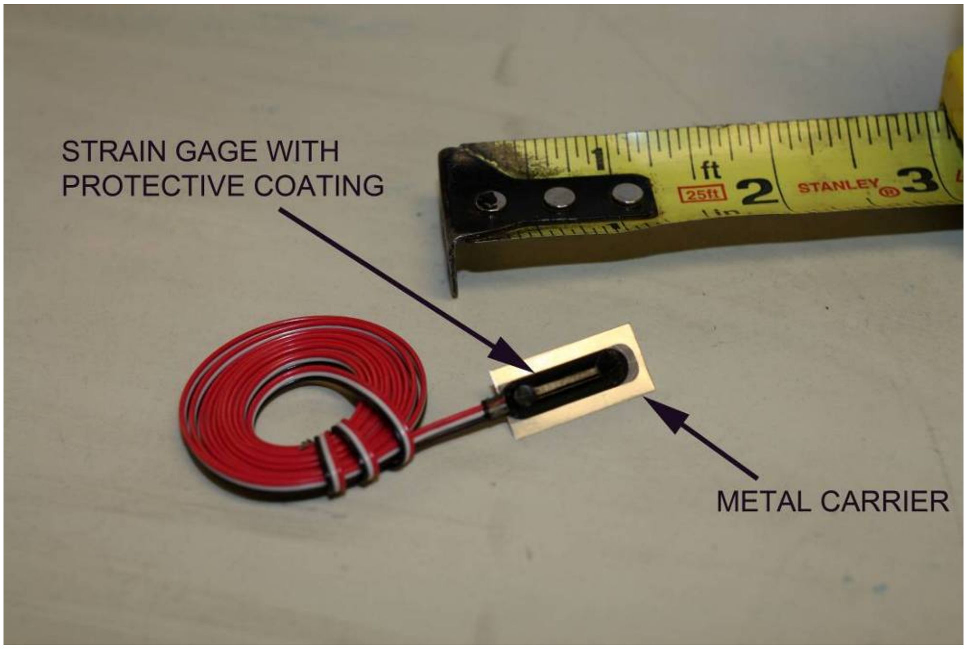


Figure 3.6 Vishay Weldable Strain Gauge (LEA-06-W125E-350/3R)

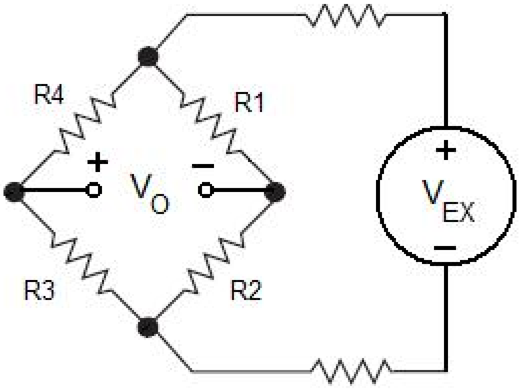


Figure 3.7 Wheatstone Bridge Circuit

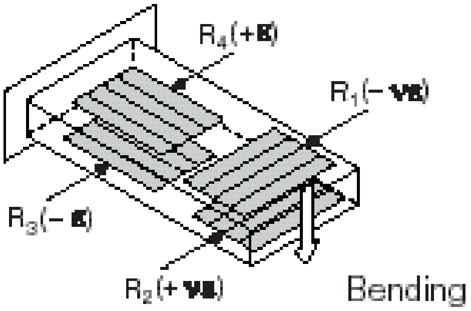


Figure 3.8 Full Bridge Type II Configuration Suitable for Measuring Normal Strains Due to Bending Strains (NI 2010).

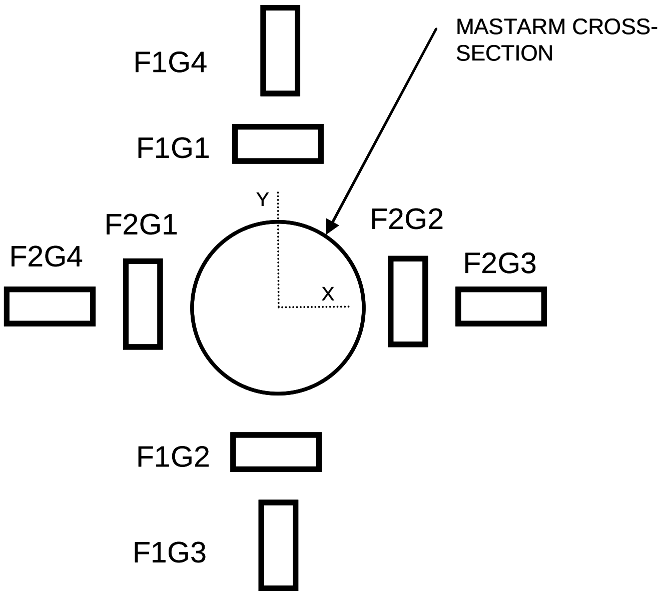


Figure 3.9 Mast-arm Strain Gauge Placement Diagram (looking along mast-arm toward the free-end)

Figure 3.10 Vishay Spot Welder for Weldable Strain Gauges

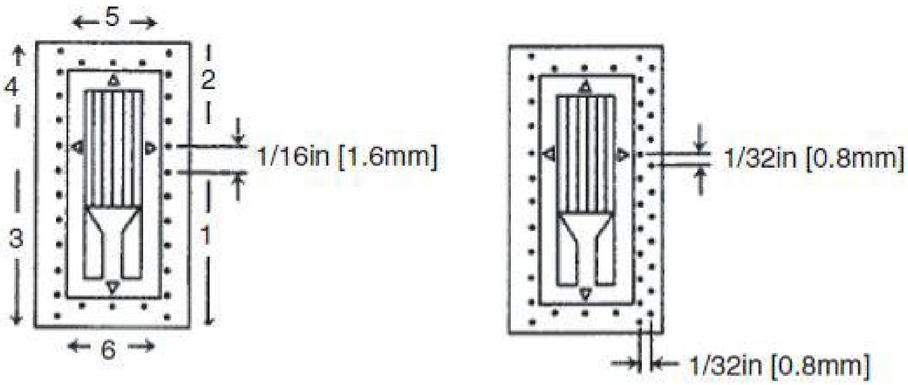


Figure 3.11 Spot Weld Pattern Recommended for Weldable Strain Gauges (Vishay 2010).

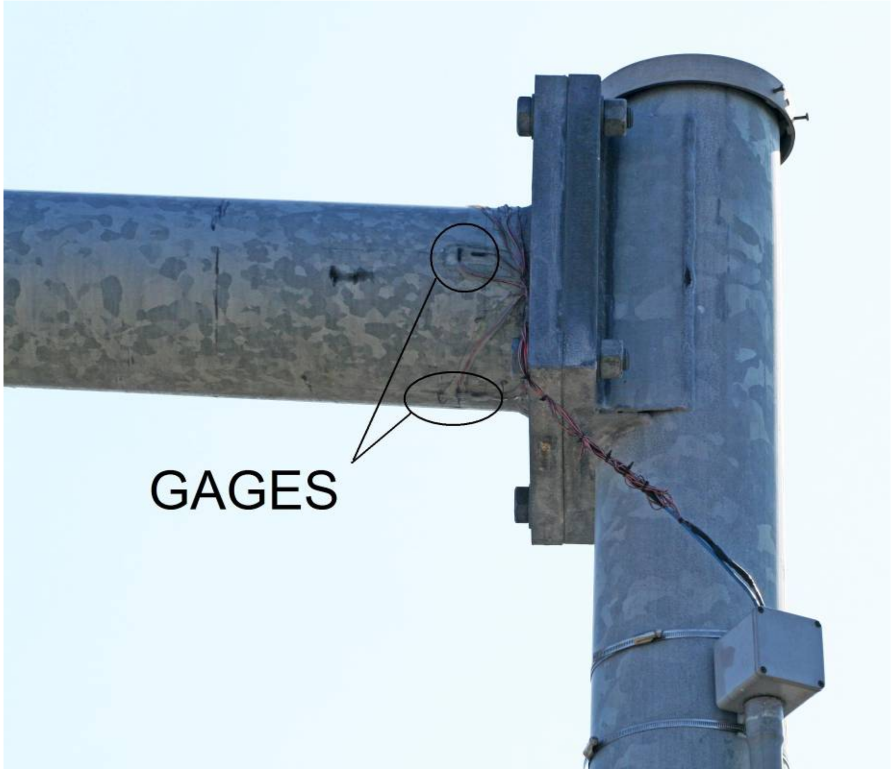


Figure 3.12 Strain Gauge Locations as Installed on Instrumented Sign Support Structure

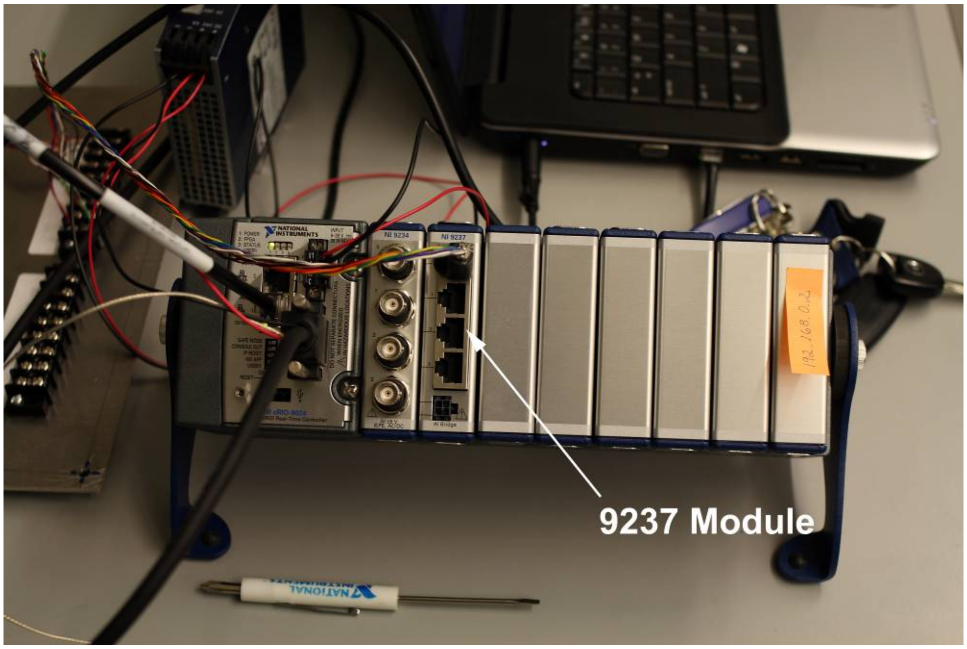


Figure 3.13 National Instruments NI 9237 Strain Gauge Module in Compact RIO.

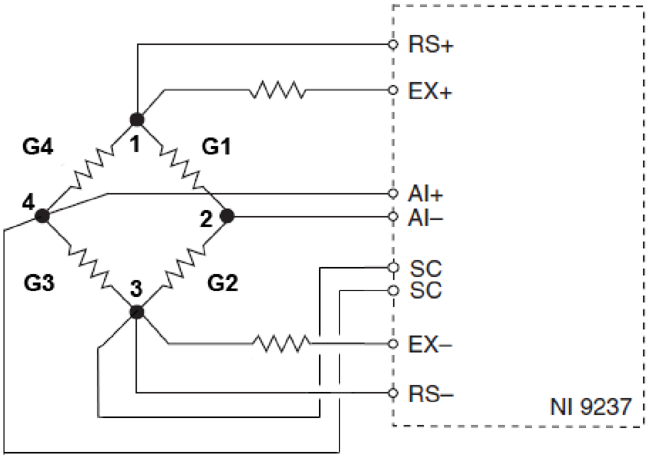


Figure 3.14 Wiring Schematic for Full Wheatstone Bridge

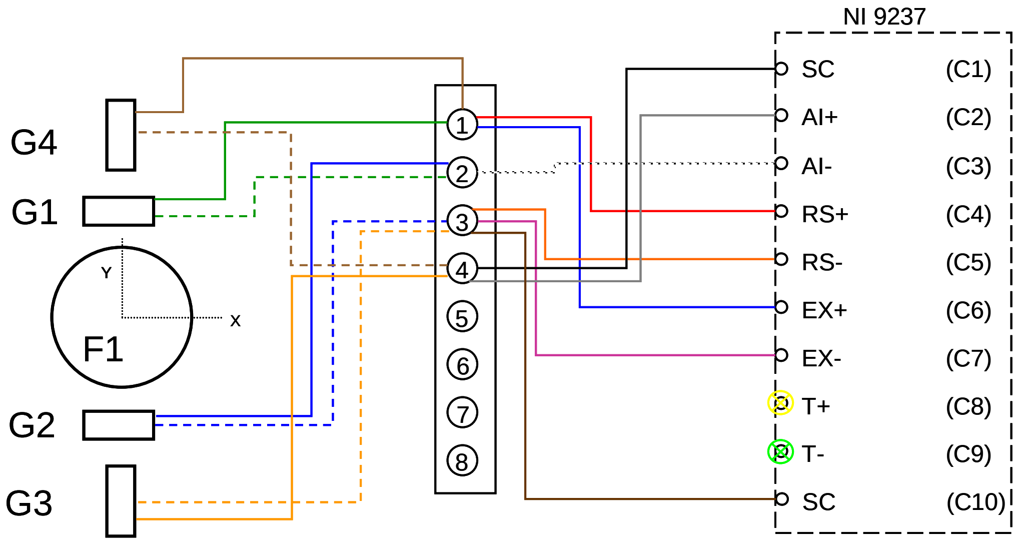


Figure 3.15 Field Wiring Diagram Relating Strain Gauge Wiring to NI 9237 Conditioning Cards.

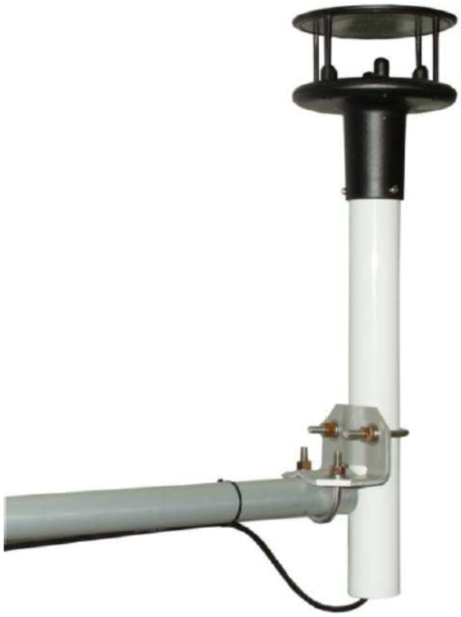


Figure 3.16 Gill Windsonic 2-D Sonic Anemometer (Gill 2010)

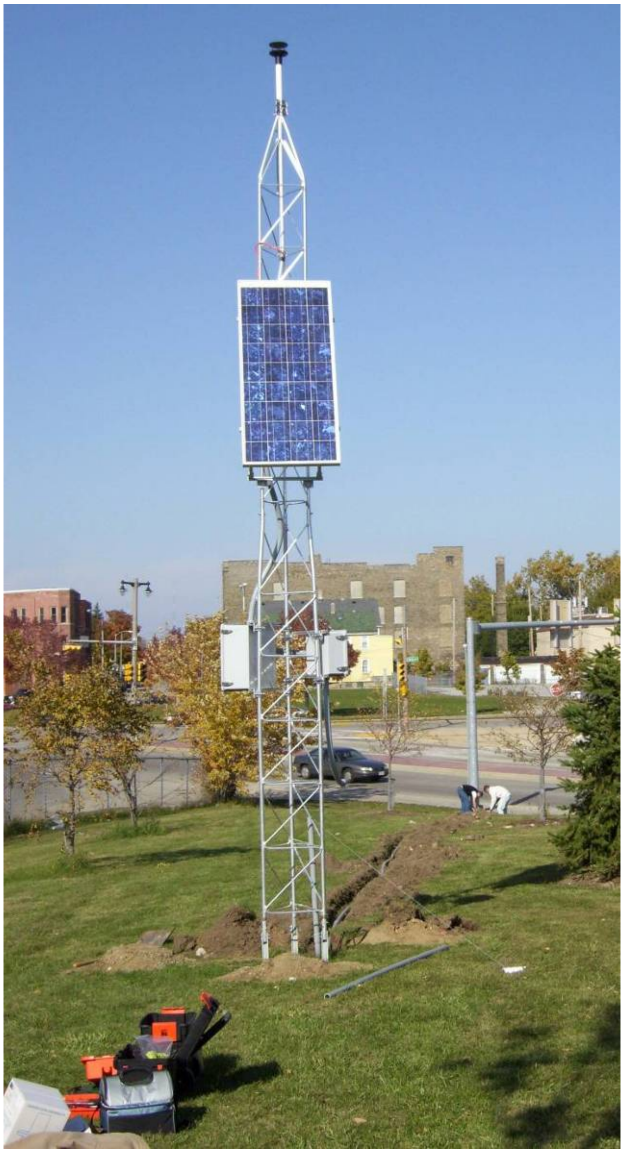


Figure 3.17 Twenty Foot Tall Aluminum Weather Station Tower Installation.

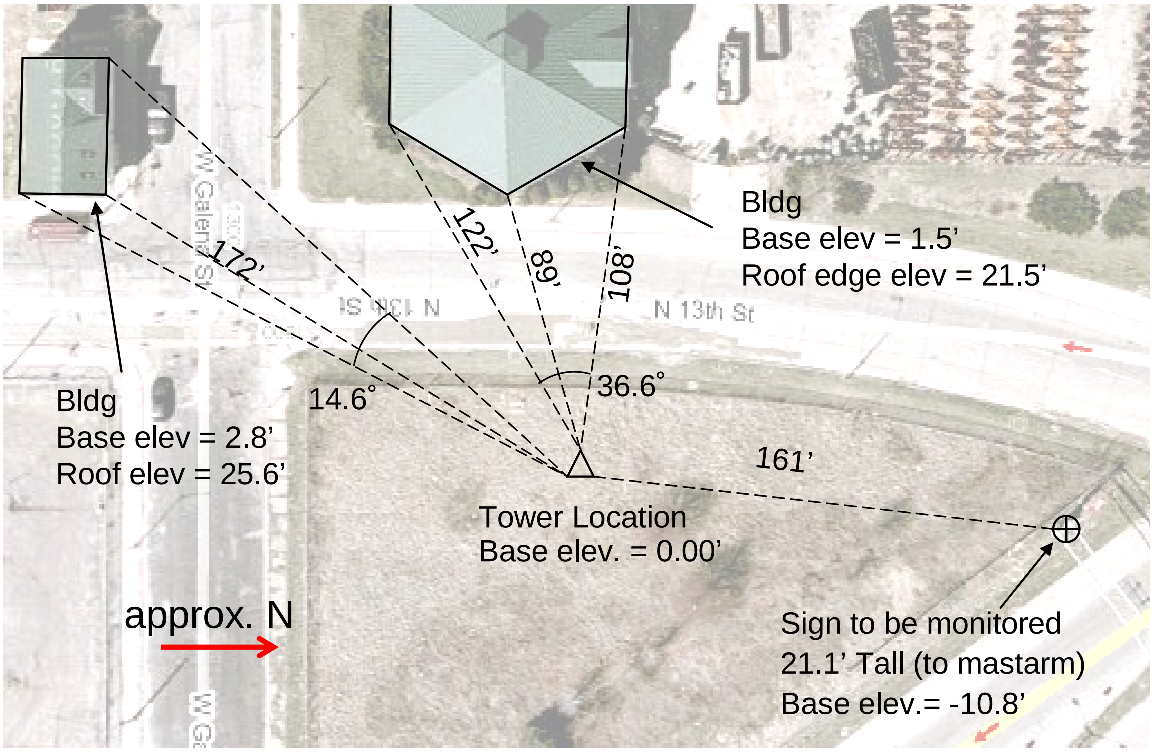


Figure 3.18 Site Survey with Identification of Potential Wind Stream Obstructions

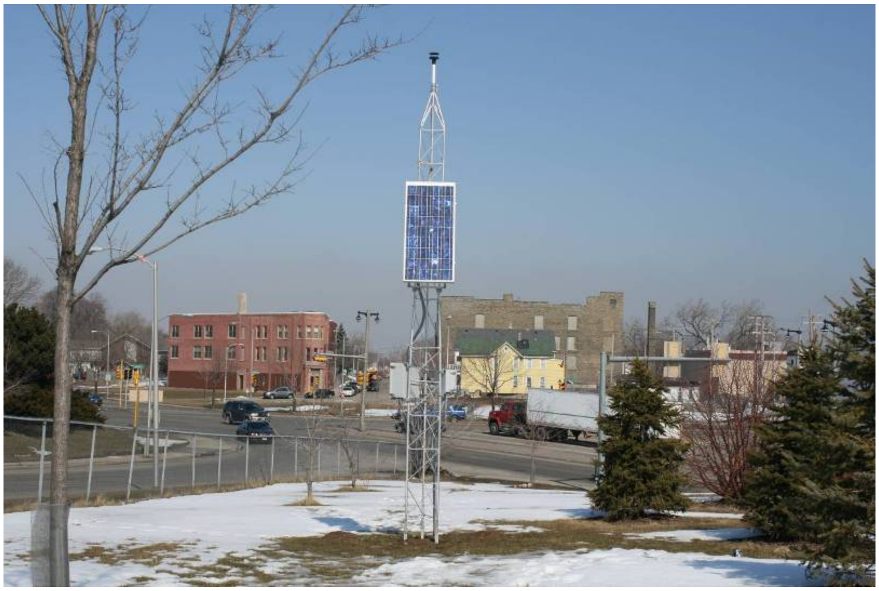


Figure 3.19 Tower Looking North

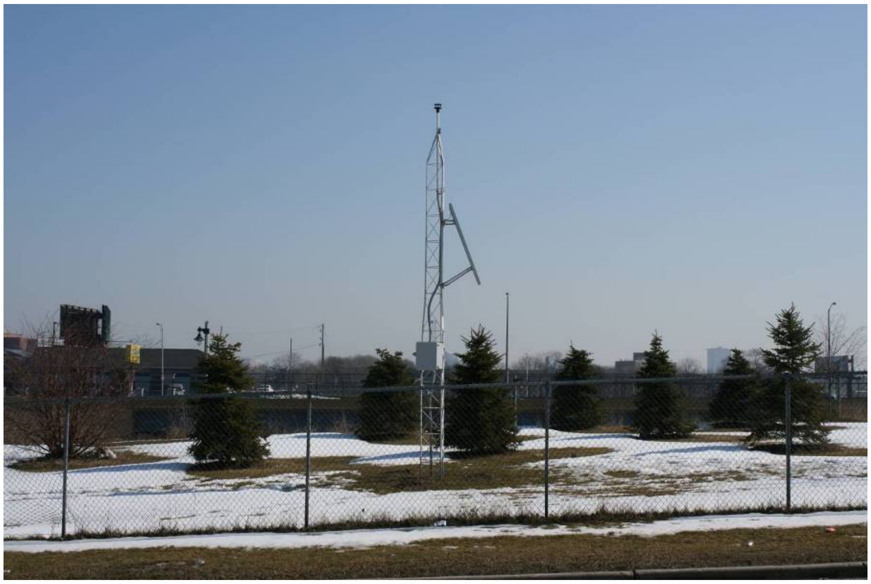


Figure 3.20 Tower Looking East

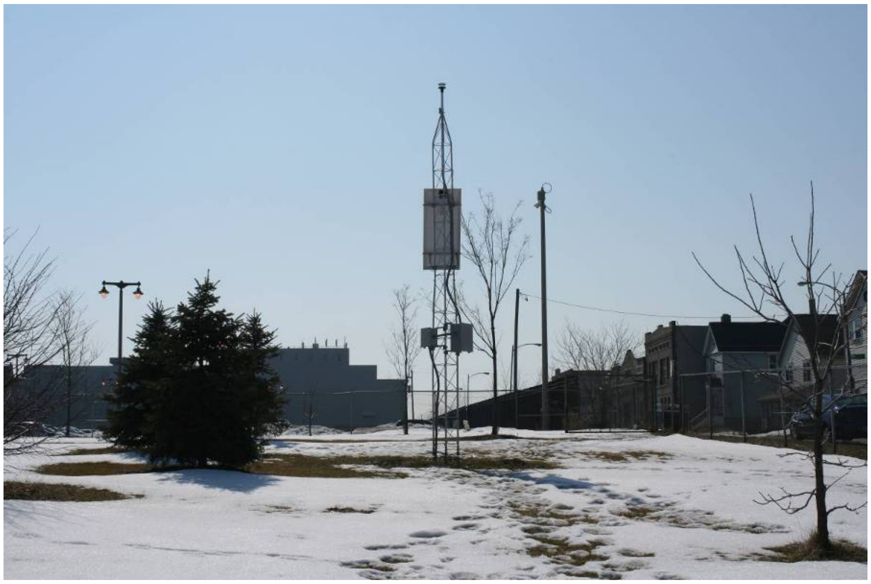


Figure 3.21 Tower Looking South

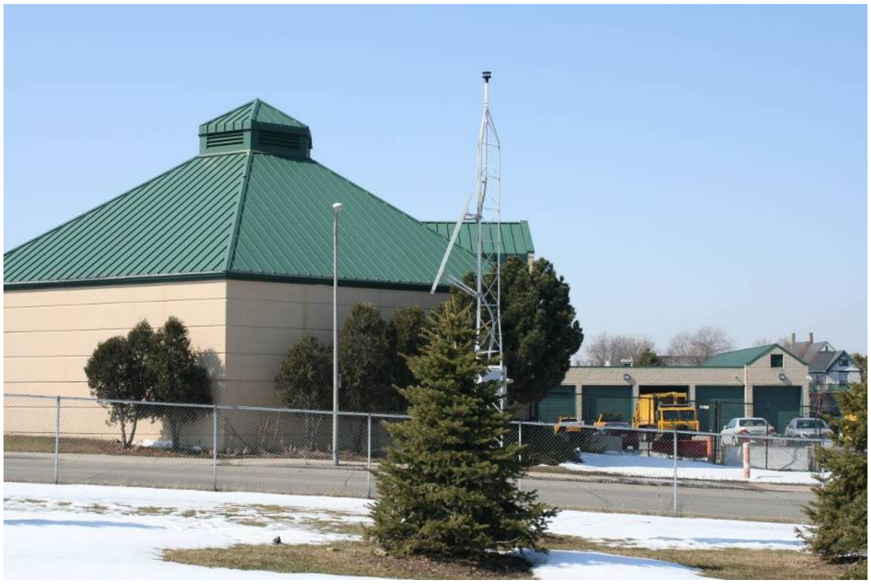


Figure 3.22 Tower Looking West

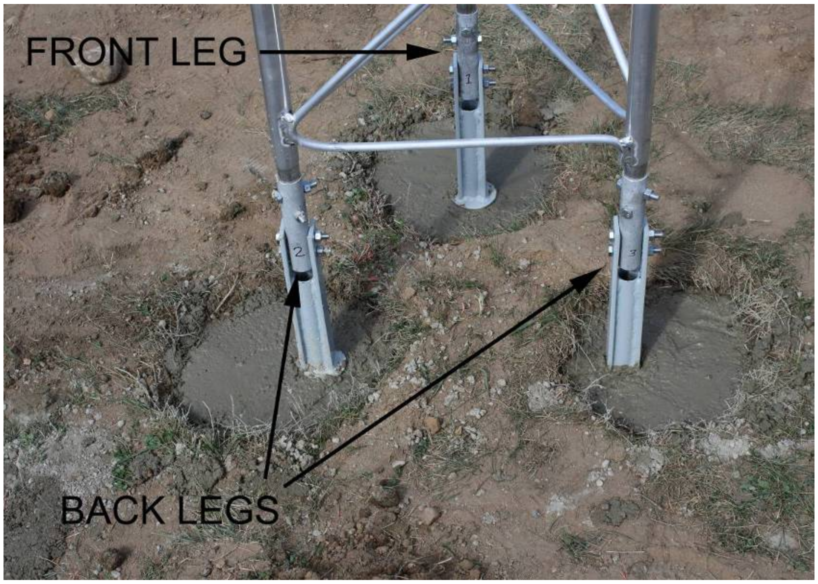


Figure 3.23 Foundation and Tower Base

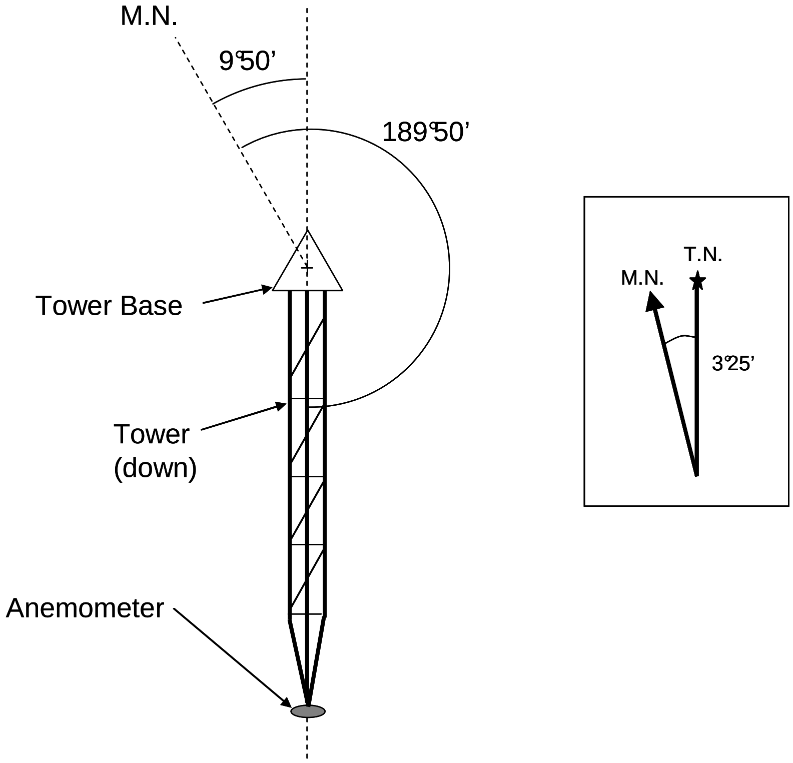


Figure 3.24 Plan-View Schematic of Procedure Used to Define Angle of Anemometer North.

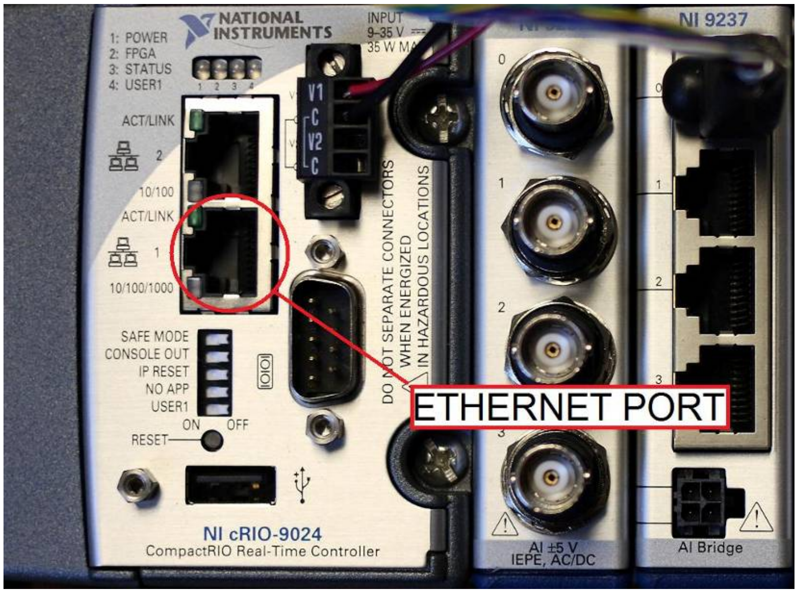


Figure 3.25 Ethernet Port on National Instruments Compact RIO (cRIO)

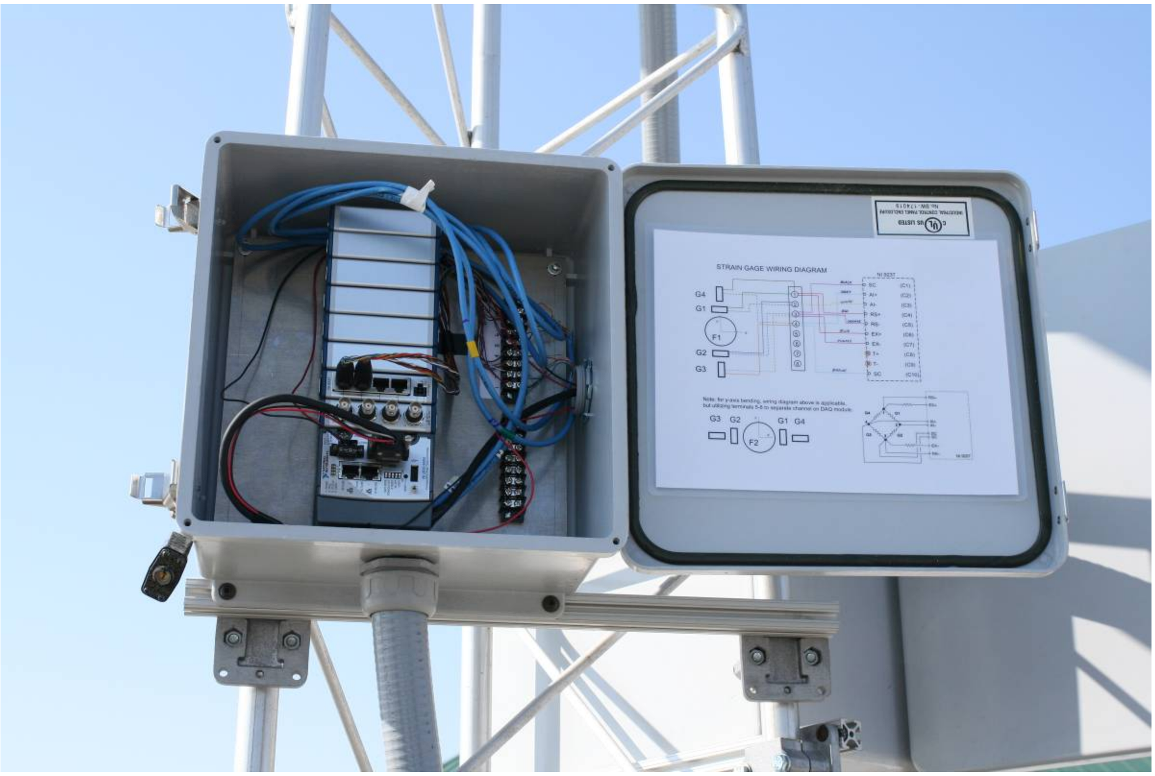


Figure 3.26 Data Acquisition (DAQ) System Enclosure

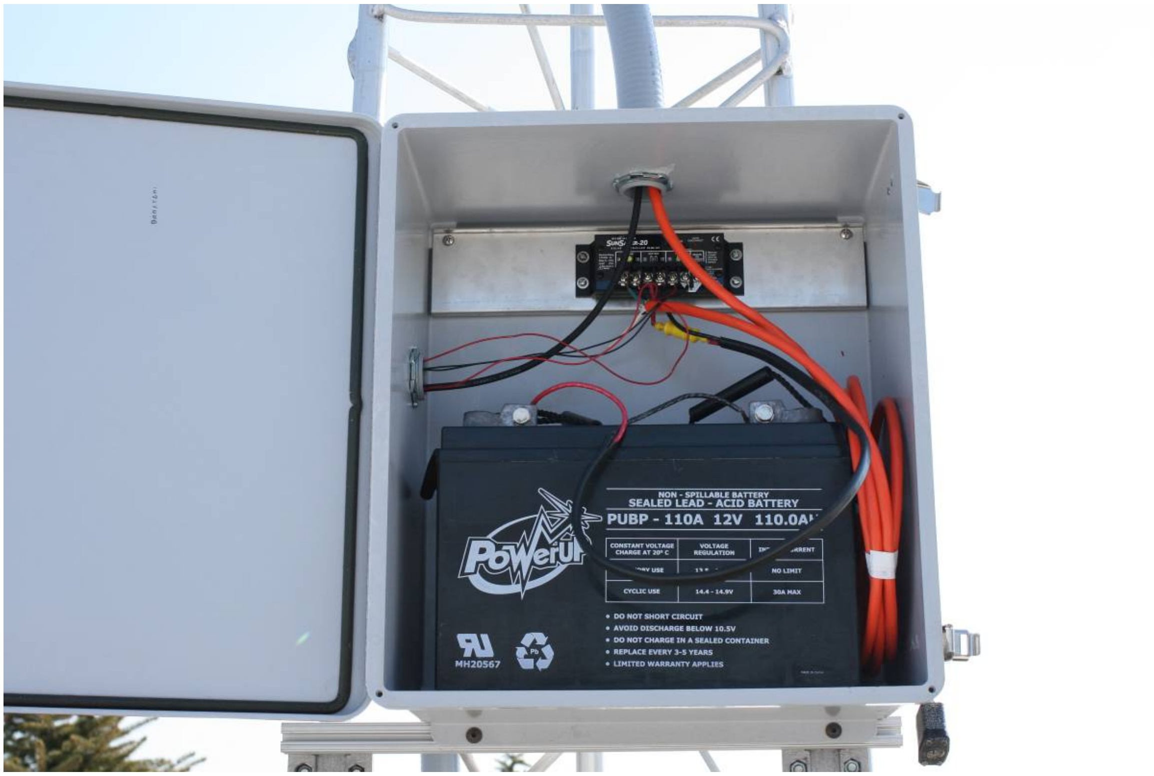

PowerUP
NON - SPILLABLE BATTERY
SEALED LEAD - ACID BATTERY
PUBP - 110A 12V 110.0AH
CONSTANT VOLTAGE CHARGE AT 20° C
VOLTAGE REGULATION
CYCLIC USE
14.4 - 14.9V
NO LIMIT
30A MAX
• DO NOT SHORT CIRCUIT
• AVOID DISCHARGE BELOW 10.5V
• DO NOT CHARGE IN A SEALED CONTAINER
• REPLACE EVERY 3-5 YEARS
• LIMITED WARRANTY APPLIES
MH20567
Pb

Figure 3.27 Battery Enclosure

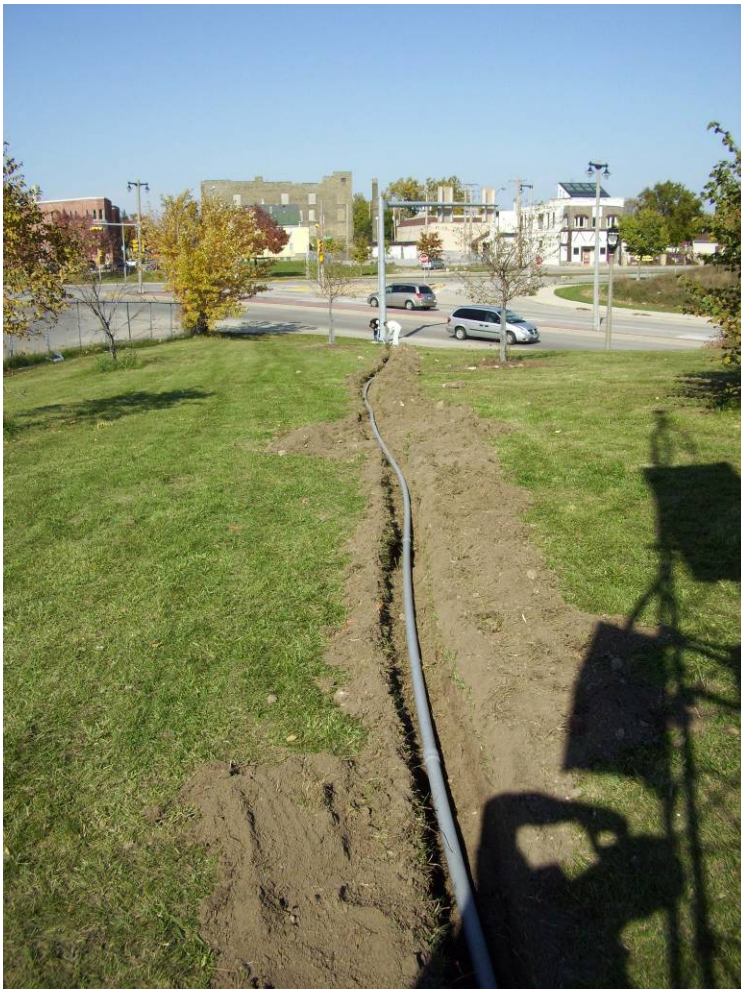


Figure 3.28 Conduit Run

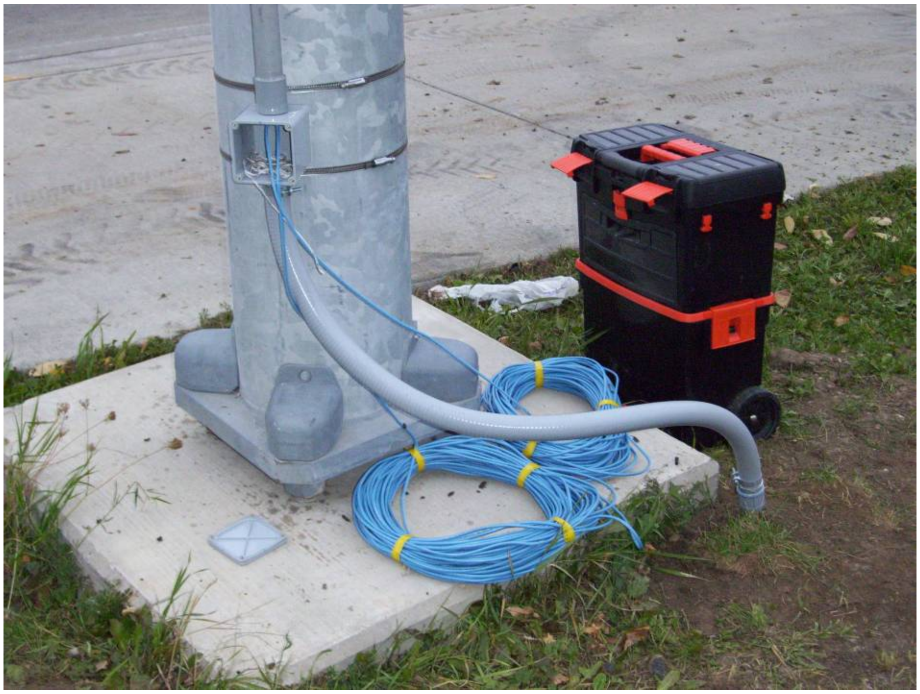


Figure 3.29 Conduit Connection to Sign Support Structure

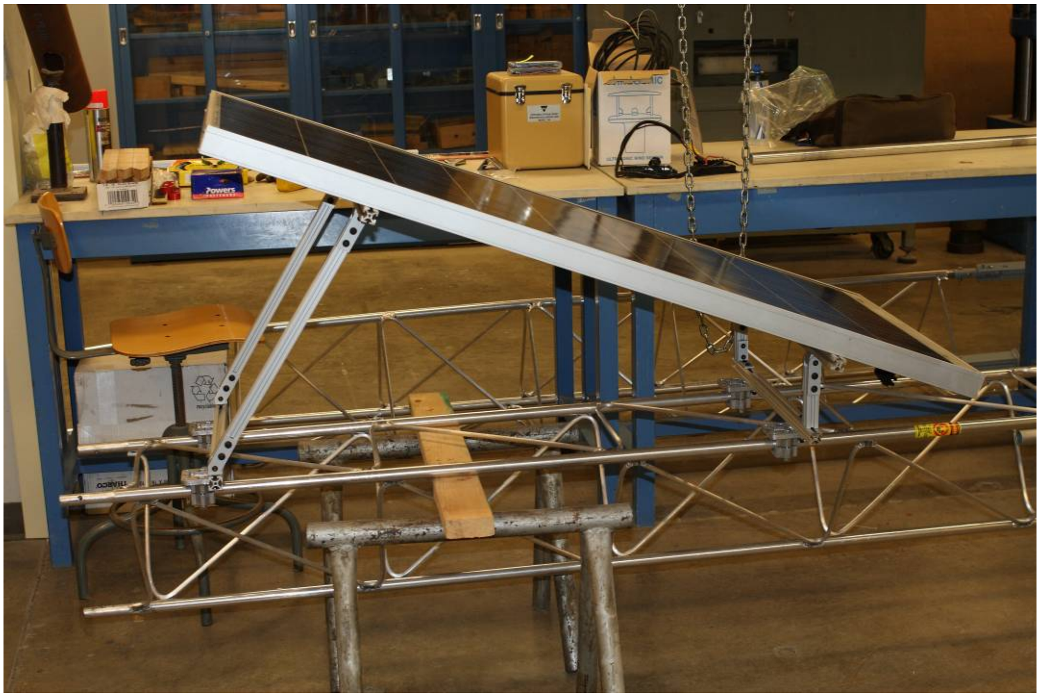


Figure 3.30 Solar Panel Mounting

Chapter 4

Data Acquisition System Software

4.1 Overview

This chapter serves as a detailed description of the data acquisition software program developed for continuous data acquisition at the field monitoring site. Appendix A provides a step-by-step procedure on how to run this program, how it can be deployed to the cRIO, as well as helpful hints and a trouble shooting guide.

LabVIEW Overview

LabVIEW is a graphical programming tool that is used in many capacities across many industries, but specific to this current project it is the means by which to program the data acquisition hardware to process and log data.

LabVIEW uses a flow chart approach to programming, complete with graphical icons and wires which are saved into *Virtual Instrument* files or *VIs* for short. Each VI contains a *Front Panel* and a *Block Diagram.* The VI Front Panel provides a customizable user interface for displaying data and controlling the program inputs. The VI Block Diagram is the meat of the program where the graphical coding takes place. Compare this to typical programming languages which use a succession of text that follows the order in which it was coded, i.e. top to bottom or left to right, LabVIEW executes a function or series of functions after receiving all the required inputs. Multiple VIs might be used, especially if coding becomes messy and complex, and can be placed inside the block diagram of another VI. All the used VIs are then referenced into a *project*, which is in essence a library where all relevant VIs are placed so they can communicate with one another.

VIs are created using a PC with LabVIEW software installed on it. These VIs can then be deployed onto a remote target, such as the cRIO data acquisition device. Once deployed, the cRIO will run this program until either a program feature tells it to stop, or a PC is directly connected via an ethernet cable and it is manually stopped via the interface.

Graphical programming, while visually appealing does not make programming a significantly easier task, especially for those unfamiliar with programming in general. A large amount of time was spent developing the data acquisition program. However the developers of LabVIEW, National Instruments, have a wide array of resources available on their website (*http://www.ni.com)* that aid in programming. These include example VI programs that can be used and customized for a particular application.

Virtual Instrument (VI) Overview

Under the current scenario, three channels of data exist. This data must be read, processed, and written to a file that will be saved on the local hard drive for periodic removal. One channel belongs to the anemometer which outputs data in a text string format at 4Hz via the RS-232 port on the front of the cRIO. This text string has the form,

Q, 229, 002.74, M, 00, 16

The second and third items in the string of values (comma separated), indicate wind direction and wind speed, respectively. The program developed for this thesis takes this text string and parses it to obtain wind speed and direction, which are then converted to numeric form.

The two additional channels are for the 2 sets of strain gauges in the two full-bridge configurations: F1 (major axis bending) and F2 (minor axis bending). The strain gauge conditioning module, NI 9237, is (by default) reading voltage across the bridge at 50 kHz. This rate of data acquisition is significantly slowed down later in the programming to 20 Hz. The anemometer's maximum sampling rate is 4 Hz, whereas the sampling rate for the strain readings was established using a study of the natural frequencies of vibration for the mast-arm sign support structure (later described in section 4.4). The voltage across the full-bridge is then converted into values of microstrain (strain x 10^6) using equation (3-2).

After the data is processed, it is assembled, time stamped, and written to a text file. File names have the prefix *cRIO* followed by the date and time the file was created:

cRIO_100204_1127.txt

was created February 4th, 2010 at 11:27 AM

A sample text file has the form shown in Figure 4.1. There are five strain readings per strain channel in the time it takes to receive one anemometer reading. Each line of data is written at 4 Hz, as indicated by the timestamp – 4 readings with the timestamp 12:25:13 PM (Figure 4.1), while still accounting for 20Hz of strain data. This data can be retrieved off of the cRIO's hard drive as described in section 3.5.

4.2 Program Controls

Two VIs are deployed onto the cRIO, *Combined.vi* and *FPGA.vi*. Both work together, however the FPGA VI works behind the scenes and does not need to be accessed by the user, unless a code modification must be made. *Combined.vi* is the main or host VI, responsible for most of the program features and operations. It also provides the main user interface for control purposes. Figure 4.2 illustrates the VI's front panel that the user is presented with upon running the virtual instrument program in LabVIEW.

The left half of front panel in Figure 4.2 contains all of the control features, subdivided into controls for the NI 9237 module (strain gauges), serial port (anemometer), and data logging. The right half of the front panel displays real-time data coming in from the strain gauges and the anemometer.

As a side note, the DAQ system was designed to run on its own and since a human will not interact with the program once deployed, a front panel user interface has no real purpose on a remote system. This front panel was developed during programming as a tool for development and still can be used as a visual lesson for illustrating what is happening real-time in the monitoring system. An executable file, however, can be created to take front panel default input values (which can be user specified before hand) and apply them to the program when it becomes deployed. Creating an executable file and proper deployment procedures can be found in Appendix A. However, as mentioned, nothing needs to be done in terms of this unless it is found that a code modification is needed.

Bridge Module Controls (NI 9237)

The NI 9237 controls, begin with a box titled *Resource Parameters* (refer to Figure 4.2). This box simply identifies the physical target the program is running on, i.e. the cRIO. Below that box is the shunt calibration feature which is controlled by a button. Described in section 3.3, shunt calibration is a method for increasing the accuracy of strain readings as it can account for gain error due to circuitry wiring. The shunt calibration procedure was applied the day the DAQ system was deployed to the field. The shunt calibration is activated programmically and the calibration takes place internally with hardware. There was no observed affect on the real-time strain readings, which is expected as the configuration of strain gauges, described in section 3.3, cancels out the effect of lead-wire resistance. Improvements in the accuracy of the strain readings would be minimal.

The offset controls follow. In theory, the strain gauge bridge circuits F1 and F2, should read zero in an unstrained state. In reality, this will never be the case, and the system will likely read some amount of strain in the unstrained state. Offset is a way to remove the initial imbalance from the bridge circuits by entering numerical values into the boxes of “offset e1” and “offset e2” for F1 and F2 respectively. When the system was deployed to the field, channel F1 and F2 were reading relatively stable strain values around 114 and -93 µɛ, respectively. Light winds were present and consequently the motion of the mast-arm was minor. Due to the stability of the mast-arm at that specific time, it was assumed that 114 and -93 effectively approximated the initial imbalance of the bridge circuits and were taken as the offset values. This offset resulted in strain readings near zero for both channels. A more systematic offset procedure is laid out in section 5.4, which is done post-data process for reasons discussed in that section.

Below the offset controls are the *Timing Parameters*. The FPGA VI will be discussed in more detail later, but the timing parameter controls are really controlling the FPGA VI which is linked through the main VI. Basically, the FPGA VI is gathering raw data from channels F1 and F2 on the NI 9237 and then storing it until the main VI calls for the stored data. The timing parameters control how much data is gathered and stored by the FPGA. The *samples per channel,* currently set at 5, is how many data points per channel the main VI pulls from the FPGA VI memory at one time. The *data rate* sets how fast the NI 9237 module is reading raw data. It has a default read rate of 50 kHz and can be changed to slower preset rates using the data rate control, but this value has been left at 50 kHz because the 20 Hz desired in our acquisition is not a selectable preset value, and customizable control over the data speed was desired. As a consequence of this desire and decision, the *Count(msec)* control was included in the program which tells the FPGA VI to read data from the NI 9237, in this case, every 50 msec. The NI 9237 module is providing data at 50 kHz and the FPGA VI is reading a single data point every 50 msec. The balance of readings is discarded. It is highly recommended that the timing parameters are not changed because they are interrelated. Changing these values can have an adverse effect on the program if they are not changed correctly and therefore are not in agreement with one another.

Below the timing parameters are controls for the gauge factor and Poisson’s ratio. Both values are used in the voltage to strain equation (3-2) in section 3.3. The gauge factor is specified by the manufacturer and a value of 2.0 is appropriate for the gauges used in this project. Poisson’s ratio was taken to be 0.3.

Serial Communication Controls

The program was not designed to change the configurable settings on the anemometer. It simply establishes RS-232 communication parameters so that meaningful data can be obtained. Configuration of the anemometer is best done using Windows Hyperterminal as described in section 3.4. The values in these boxes have been set as the default values specific to the anemometer and should not be altered unless a different device is connected via the RS-232 port and require different settings.

Data Logging Controls

Underneath both the NI 9237 and Serial controls, is the Data Logging control. There is really only one control available to the user on the front panel, the *Divisor* (bottom of Figure 4.2). Data Logging is done automatically. The user can change how often these files are saved and how often a new one is opened through defining this parameter or variable. Unfortunately, it was more efficient to have this controlled based on the number of loop iterations within the program, instead of something more tangible like every *x* number of minutes, hours or days. In most programming cases, a loop can be used to execute a block of code over and over again. The present LabVIEW program iterates every 4 Hz. This iteration speed is assigned as a result of the anemometer sampling rate being limited to this speed. Every iteration takes 250 msec or 0.25sec at 4 Hz. The numeric value place within the *Divisor* box corresponds to the number of iterations that should take place before the current data file is saved, closed, and a new one opened in its place. If each iteration takes 0.25 seconds then values 240, 14400, 345600 correspond to

1 minute, 1 hour, and 1 day; respectively. The current deployed setting is to have a new file write every day.

The other boxes in Figure 4.2 are indicators used to illustrate the parameters or variables that are set and being used by the LabVIEW program. The *Current File Path* provides the path of the current file being accessed and written to. The *State* indicates the status of data logging; 0-state, file open; 1-state, file write; 2-state: file close. The *Iteration* and *Remainder* boxes are the result of a feature called *quotient and remainder* (Figure 4.3), which keeps track of the loop iterations and compares this with the divisor to find multiples. When not an integer multiple there will be a remainder, which will cycle between 0 and the divisor minus 1, continuously. The code in Figure 4.3 is part of the Block Diagram which is checking the remainder value. When the remainder is a non zero value it allows the program to continue logging data to a file. When the remainder becomes a value of zero it forces the program to close the current file and proceed to open a new file. This is governed by a True/False condition: if remainder equals zero (true) then proceed to 2-state (close file), if non-zero (false) continue with 1-state (write to file).

Front Panel Display

Four charts are on the front panel in the upper-right side of Figure 4.2. Two along the top display the strain readings from channel F1 and F2 (Figure 3.9). The two along the bottom display wind speed (mph) and wind direction (polar coordinates). Immediately below the F1 and F2 charts are two boxes that provide a numerical display of the mean strain values. The main VI obtains five strain readings from the FPGA VI at each iteration in the programmatic loop. The value being displayed is the mean of these five values. Immediately below the wind speed and direction charts are numerical displays of the data being plotted.

Along the bottom of the display in the front panel in Figure 4.2 are four boxes. The first box, titled *read string,* is displaying the text string being directly read from the anemometer. The middle two boxes are controls used to parse the text string into the useful data, which should not be altered as it is specific to the current output of the device. The remaining string is displayed in the last box which is then converted into numerical data.

4.3 Block Diagramming Program Code

The block diagram area within LabVIEW provides the workspace area for programming. Since LabVIEW takes a flowchart approach to programming, it will be easy to discuss the program in that manner. Typically a program requires a set of inputs which are defined on the front panel. These values are linked to the block diagram and can be seen in Figures 4.4(a) and 4.4(b). It should be noted that objects referenced in later discussion are numeric values in boxes in Figure 4.4(b).

Figure 4.4(a) shows the code that initializes the serial port on the cRIO by reading the baud rate, data bits, etc… from the front panel and then passing them to the block diagram. The termination of all these “wires” is at a serial port configuration VI, which takes all of these values and initializes the serial port on the cRIO. Certain features have been developed by LabVIEW developers and come shipped with the software, which are placed in a library of tools, which the programmer can use for his or her program. This serial VI is an example of such features. This VI was taken out of a set of tools and simply placed onto the block diagram of the main VI and connected using the programmatic "wires".

The FPGA VI is ultimately responsible for communicating with channels F1 and F2 (Figure 3.9) on the NI 9237 module and storing this data. It runs separately from the main VI. It is, however, ideal to control both VIs from one place. The line of code shown in Figure 4.4(b) is responsible for initializing the FPGA VI, specifying some given values from which to operate, and then tells it to run. Object number 1 opens the FPGA reference to the physical FPGA target on the cRIO. Number 2 is a reset feature which clears the FPGA VI memory of any data stored from any previous runs. Number 3 takes the data rate and count values from the main VI front panel, and writes these values to the FPGA front panel, shown in Figure 4.5. Object 4, tells the FPGA VI to begin running on the FPGA target. At this point the FPGA VI begins recording strain data and storing it for retrieval by the main VI. Object 5, checks the shunt calibration status on main front panel and relays the status to the FPGA VI, which can be turned on and off while the program is running.

Returning to the discussion of the main VI, once the program has proceeded through all of the items shown in Figure 4.4, it proceeds into the acquisition loop. The acquisition loop is a *while loop* structure, that executes the code contained within that structure over and over until stopped.

The serial port is the point of communication between the DAQ system and the anemometer. It is configured outside the main acquisition loop. The anemometer is then able to provide data to the cRIO. The anemometer is configured to take readings at 4 Hz, which sends this reading to the serial port. The acquisition loop now needs to scan the serial port every 0.25 seconds for the data. Figure 4.6 illustrates the serial port block diagram coding. The figure is broken into two halves; the wires running to the right of Figure 4.6(a) connect with the left of Figure 4.6(b).

The first piece of code inside the acquisition loop is the *serial read VI*, which is another pre-written VI that can be taken from the tools menu and placed onto the block diagram. The serial read VI is specified to read 24 bytes (the size of a single anemometer output string) from the serial port each time. The output is displayed on the front panel using the *read string* indicator. Note that thick pink wires indicate that the data passing through them is text string type data. The remaining code in Figure 4.6(a) is parsing the output string. The full string is first passed into the *String Subset* function which is used to return a substring of the input beginning at the specified *offset* and containing *length* number of characters. The substring is then passed to the *Scan From String* function where the wind direction and speed are pulled out of the string and converted into a numeric data type (thin orange wires). The values pass into Figure 4.6(b) where they are displayed on the front panel charts. The values are then converted back into individual strings for eventual logging.

The second block diagramming code to be overviewed is the coding related to acquisition of strain (Figure 4.7). Once inside the acquisition loop, the strain data must be retrieved from the FPGA VI. Since the data being retrieved is bridge voltage it needs to then be converted to strain, displayed, and then logged. Figure 4.7(a) includes objects (in boxes) that are used as later references for discussion related to the block diagrammatic coding for strain acquistion.

Object number 7 in Figure 4.7(a) is responsible for pulling the data out of the FPGA VI. It takes out 10 elements at a time (5 per channel) which was specified in the front panel, whose value is wired to the *number of elements* box of object 7. Data then enters a *for-loop* structure which encloses a *formula structure* (object 9). When the 5 data points per channel are pulled out of the FPGA VI they are passed consecutively to the for-loop. The formula structure then converts the voltages across the full bridges into strain. Before the data leaves the for-loop, the offset is subtracted from each value. As the data passes out of the loop, each channel is diverted to three features. First it is passed to a function that calculates the mean value of the five data points. Second, it passes data to a chart that is displayed on the front panel. Lastly, the data is converted from numeric values to a text string for logging.

Directly inside the main acquisition loop is a *case structure* in which three cases exist: 1.) a file is created and opened, 2.) data is written to this file, and 3.) the file is closed. Each case is read and executed in succession. As described earlier and shown in Figure 4.3, this is controlled by a feature that monitors the number of loop iterations. The code shown in Figure 4.8 is case 1. It compiles a file path with a time stamped file name .txt, which is then passed to the *open file* function. The open file function has multiple settings, and specific to this program it is set to replace or create a file given the file path and name fed to it. As soon as a file is created the acquisition loop immediately moves to case 2.

The file write block diagrammatic coding is shown in Figure 4.9. This figure illustrates case 2 along with the bulk of the acquisition code responsible for obtaining wind and strain data. The acquisition code has taken all of the strain and wind data and converted it to individual text strings, which are then passed into a *build array* function. Also coming into this build array function is a time stamp with a resolution of seconds. Once all the individual text strings have been assembled into one text string it is formatted to a spreadsheet string (tab delimited) for importing into Microsoft Excel or similar spreadsheet programs. This formatted string is then passed to the *Write to TEXT file* function where it is appended to the current text file. After a specified number of loop iterations, case 2 moves to case 3 and the file is closed. Immediately upon closing the file, case 3 moves back to case 1 and the sequence is started all over again.

4.4 Sensor Validation and Sampling Rate Definition

Before the data acquisition device was deployed in the field it was important to verify the program ran properly, was returning values that made sense, as was recording data at a sufficient sampling rate. A series of tests were run to verify that the anemometer was reading wind speed and direction accurately using the Marquette University College of Engineering wind tunnel facility. A full-bridge strain gauge test specimen was also built to validate strain readings. Lastly, an analytical model of the mast-arm was created and analyzed using Mastan to obtain the modes of vibration and their respective natural periods.   These frequencies associated with these fundamental modes are used to define the sampling rates for strain.

Anemometer Validation

The Marquette University College of Engineering has a wind tunnel suitable for studying model structure behavior (Figure 4.10). This wind tunnel was utilized to verify the LabVIEW program written and the anemometer readings for wind speed and direction. A fixture for the Gill Windsonic anemometer was designed and fabricated so that the anemometer could be placed inside the wind tunnel horizontally and rotated to mimic wind coming out of different directions. This fixturing and installation of the anemometer in the wind tunnel is shown in Figure 4.11. The anemometer has a north marker which was transferred down to the cylindrical mount and aligned to prescribed locations using the angle markers on the wind tunnel wall. These marks can be seen in Figure 4.11. The wind readings were compared with those of a pitot tube, seen in Figure 4.10, which is calibrated to measure velocity in ft/min.

The accuracy of the wind direction being read from the anemometer was the first to be checked. The manufacturer-located north marker on the anemometer was oriented to 0, 90, 180, and 270 degrees from the upstream direction. The orientation was then checked with those readings obtained from the anemometer and DAQ software written. The data seen in Table 4.1 illustrate strong correlation between the pitot-tube based measurements and those obtained via the DAQ software and anemometer.

The wind speed values were the second reading to be validated. Wind speed in the tunnel was then varied over a range from approximately 12 to 44 mph. The velocity measured by the pitot tube and DAQ system were then compared. The data contained in Table 4.1 illustrates close correlation of the data. The pitot-tube readings were close to those of the DAQ system, but were consistently slightly less than the readings from the anemometer. The difference ranged from 0.5 mph at lower speeds to 1mph less at higher speeds. A possible reasons for this is that the anemometer is nearly in line with and upstream of the pitot tube. It seems a likely cause for the difference since the pitot-tube measurements are taken downstream of the anemometer and the wind speed may be slowed down slightly by the anemometer body.

In conclusion the anemometer appears to be calibrated correctly, although this comes as no surprise as the manufacturer shipped it calibrated. More importantly, it verifies that the program and equipment are communicating with the anemometer correctly and effectively.

Full-Bridge Strain Gauge Validation

Verifying the strain gauge instruments and data acquisition program through correlation with fundamental mechanics is crucial because it will be very difficult (if not impossible) to validate strain readings once the system is deployed. A small cantilever beam test specimen was created in the laboratory using four 120 ohm gauges arranged in a full bridge type II configuration as discussed earlier and a flat steel bar to which the gauges were attached (Figure 4.12). This cantilever beam test is a small scaled replica of the sign support structure S-40-703. The difference is that the laboratory validation includes only one axis of bending. The strain gauge full bridge configuration was discussed in detail in Chapter 3 of this thesis.

The strain gauges were mounted at the supported end of the bar and a hole was drilled at the cantilever tip to facilitate hanging of controlled weights at the tip as shown in Figure 4.13. The cantilever setup depicted in Figure 4.13 becomes a very simple mechanics problem and a very effective means to validate strain readings and the DAQ software. Ignoring the self-weight of the bar, the load, P, at the free end multiplied by the distance from P to the location of the strain gauges, will yield the moment at the point where the gauges are mounted to the beam. This moment can then be used to calculate the stress at that location using the flexure formula;

$$\sigma = \frac{My}{I}$$

The strain can be calculated using Hooke's Law by taking the stress and dividing by Young's modulus, E.

The bar was found to be 1-1/4 inches wide and 7/32 inches thick with the load being applied 9.9375 inches from the center of the top bending gauge. E was taken as 29,000ksi, and the applied load was 2.2515 lbs for the first trial and 4.503 lbs for the second. Before the bar was loaded, initial strain offsets were applied to remove the effects of bar self weight and the weight of the hanger attached to the end. Zero strain was recorded before the weights were applied. Based on these values there should theoretically be 77.4 and 154.8 microstrain, respectively for the two loads, at the center location of the top bending gauge. Actual results were 80.1 and 160.5 microstrain (an error of approximately 3.6%). Results of this comparison are summarized in Table 4.2.

Sampling Rate Definition

A model of sign support structure S-40-703 was created using Mastan (Ziemian and McGuire 2008). The model was created based on the actual specimen drawings shown in Figures 3.4 and 3.5. Dimensional information of the mast-arm is provided in section 3.1.

The pole is 21 feet in height the mast-arm is 33’ in length, The pole base diameter is 13 inches and tapers to 9.92 inches. The mast-arm has a diameter of 11 inches at the mast-arm-to-pole connection and tapers to 6.38 inches. Since 3 is a multiple of both the height and length dimensions of the members, the model includes 3-ft long elements. Figure 4.14 illustrates the FE model used. The pole is composed of 7 elements (E1-E7), and 11 elements (E8-E11) make up the mast-arm. Section properties for this element discretization were calculated using a spreadsheet and are summarized in Table 4.3. The following material properties were defined: a Young’s modulus of 29,000ksi, Poisson’s Ratio of 0.30, yield stress of 55 ksi, and a weight density of 490 lbs/ft^3. The base (node 3) was given a fixed condition.

The FE model was used to compute an estimate of the natural period for the model. A space-frame analysis was run using a gravitation acceleration of 386 in/s^2 (32.17 ft/s^2). Figures 4.15 through Figure 4.18 show the first four modes of vibration for the model. Their natural frequencies for modes 1 through 4 are 1.61, 1.75, 4.76, and 5.26 Hz, respectively.

Mode 1 is an out-of-plane or horizontal vibration with twisting about the pole. This vibration mode is most likely to be excited by naturally-occurring wind speed variability as this vibration mode is consistent in direction with wind hitting the projected area of the signs on the mast arm. Mode 2 is a “hatchet” vibration in the vertical plane and is likely correlated with vortex-shedding induced vibrations resulting from wind flowing over the mast arm. These mode shapes agree with the results of University Missouri – Columbia research (Alderson 1999), which obtained 0.75 and 0.78 Hz for the corresponding mode shape natural frequencies. These comparably slower frequencies seem reasonable as they modeled a 54-ft long mast-arm which should have a longer natural period and correspondingly lower natural frequency.

To ensure that the response of the structural system is accurately captured, the data acquisition rate should be set sufficiently faster than the approximate natural frequencies obtained by analysis. Unfortunately, wind speed and direction data sampling is limited to a rate of 4 Hz. However, the strain data sampling rate can be set to levels much faster than the wind speed sampling rate. The Nyquist theorem would suggest that strain sampling should occur at rates of at least two times the highest frequency for mode that is expected to be present in the response of the structure to wind loading (Wikipedia 2010). As a result, a sampling rate of 10 Hz would be conservative because the fourth mode of vibration has a frequency of 5.3 Hz. It was decided to sample stain data at 20 Hz because this resulted in a strain sampling rate that was a convenient multiple of the sampling rate for the wind speed and wind direction data (five strain readings per one wind reading).

	WIND TUNNEL SETUP		Pitot Tube Meas.		Anemometer Meas.	
	orientation	orientation	Velocity	Velocity	Velocity	Orientation
	degrees	description	ft/min	mph	mph	degrees
Directional Testing	0	N faces upstream	1500	17.0	17.3	0
	90	N faces downward	1500	17.0	17.4	90
	180	N faces downstream	1500	17.0	17.3	181
	270	N faces upward	1500	17.0	17.5	271
Wind Speed Testing	0	---	1100	12.5	13.1	0
	0	---	1850	21.0	21.4	0
	0	---	2200	25.0	25.4	0
	0	---	3200	36.4	37.6	0
	0	---	3800	43.2	44.7	0

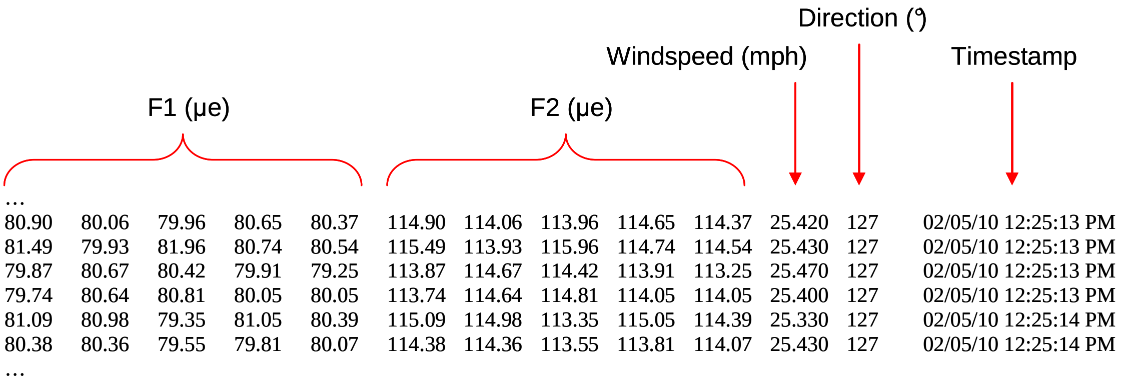
Table 4.1 Wind Tunnel Testing Results

Load, P (lbs)	Moment, M' (lb-in)	Stress, s (psi)	Theoretical, µe (million in/in)	Actual, µe (million in/in)	error %
2.252	22.37	2244	**77.4**	**80.1**	3.5
4.503	44.75	4489	**154.8**	**160.5**	3.7

Table 4.2 Strain Gauge Test Results

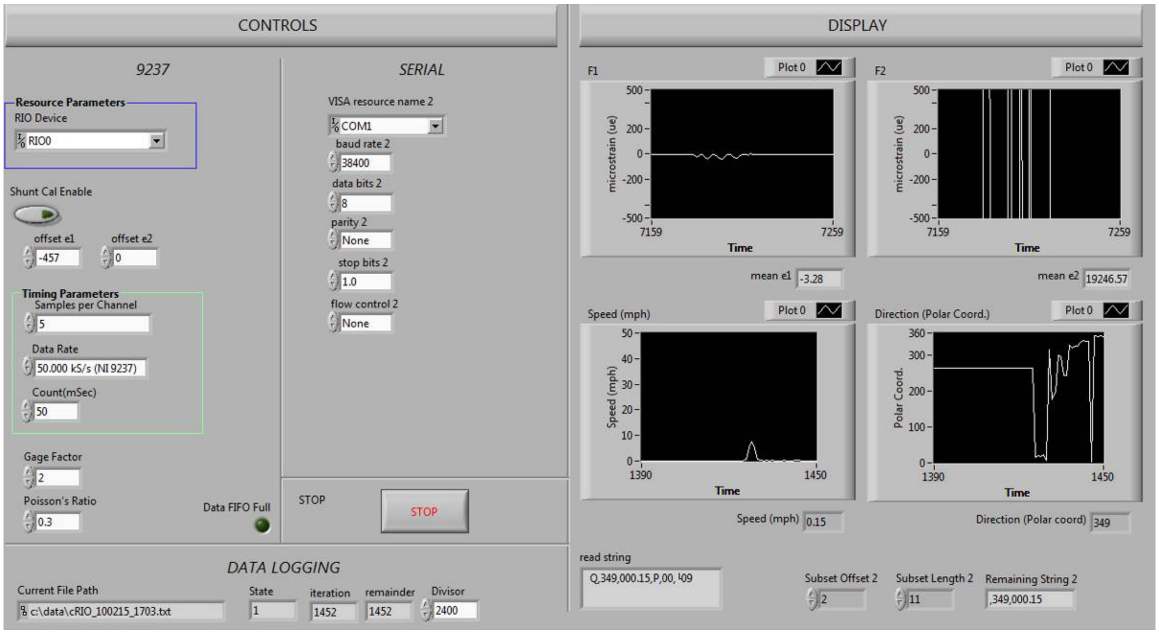
MAST-ARM					POLE				
Elem.	Dia,	Area,	I,	J,	Elem.	Dia,	Area,	I,	J,
	(in)	(in^2)	(in^4)	(in^4)		(in)	(in^2)	(in^4)	(in^4)
---	11.0	6.4	93.1	186.2	---	13.0	10.0	203.6	407.1
18	10.8	6.2	87.8	175.6	7	12.8	9.8	193.2	386.4
17	10.4	6.0	77.8	155.5	6	12.3	9.5	173.6	347.1
16	10.0	5.8	68.5	137.1	5	11.9	9.1	155.3	310.6
15	9.5	5.5	60.1	120.1	4	11.5	8.8	138.4	276.7
14	9.1	5.3	52.3	104.6	3	11.0	8.5	122.7	245.4
13	8.7	5.0	45.3	90.6	2	10.6	8.1	108.3	216.6
12	8.3	4.8	38.9	77.8	1	10.1	7.8	95.0	190.1
11	7.9	4.5	33.1	66.3	---	9.9	7.6	88.8	177.7
10	7.4	4.3	28.0	56.0					
9	7.0	4.0	23.4	46.8					
8	6.6	3.8	19.3	38.7					
---	6.4	3.6	17.5	35.0					

Table 4.3 Section Properties



```
…
80.90   80.06   79.96   80.65   80.37   114.90  114.06  113.96  114.65  114.37  25.420  127     02/05/10 12:25:13 PM
81.49   79.93   81.96   80.74   80.54   115.49  113.93  115.96  114.74  114.54  25.430  127     02/05/10 12:25:13 PM
79.87   80.67   80.42   79.91   79.25   113.87  114.67  114.42  113.91  113.25  25.470  127     02/05/10 12:25:13 PM
79.74   80.64   80.81   80.05   80.05   113.74  114.64  114.81  114.05  114.05  25.400  127     02/05/10 12:25:13 PM
81.09   80.98   79.35   81.05   80.39   115.09  114.98  113.35  115.05  114.39  25.330  127     02/05/10 12:25:14 PM
80.38   80.36   79.55   79.81   80.07   114.38  114.36  113.55  113.81  114.07  25.430  127     02/05/10 12:25:14 PM
…
```

Figure 4.1 Portion of ASCII Text File



Resource Parameters
RIO Device
RIO0

Shunt Cal Enable

offset e1
-457

offset e2
0

Timing Parameters
Samples per Channel
5

Data Rate
50.000 kS/s (NI 9237)

Count(mSec)
50

Gage Factor
2

Poisson's Ratio
0.3

Data FIFO Full

VISA resource name 2
COM1

baud rate 2
38400

data bits 2
8

parity 2
None

stop bits 2
1.0

flow control 2
None

Current File Path
c:\data\cRIO_100215_1703.txt

State
1

iteration
1452

remainder
1452

Divisor
2400

mean e1
-3.28

mean e2
19246.57

Speed (mph)
0.15

Direction (Polar coord)
349

read string
Q,349,000.15,P,00, ␉09

Subset Offset 2
2

Subset Length 2
11

Remaining String 2
,349,000.15

Figure 4.2 Main VI Front Panel

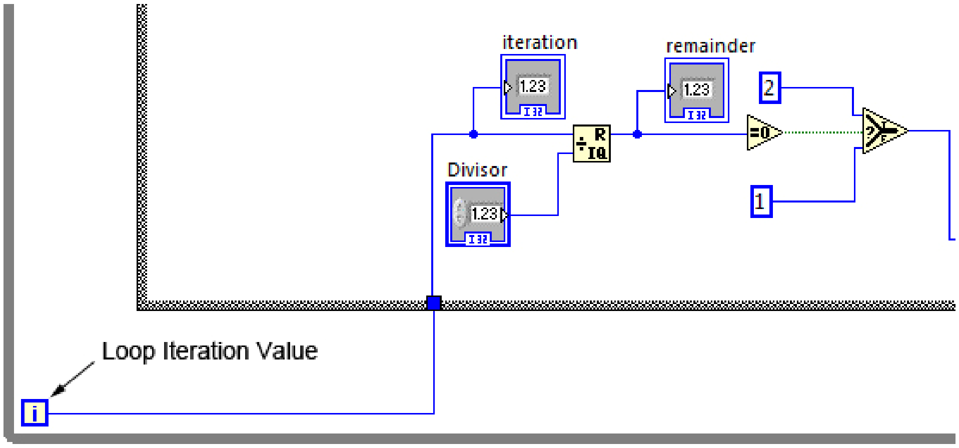
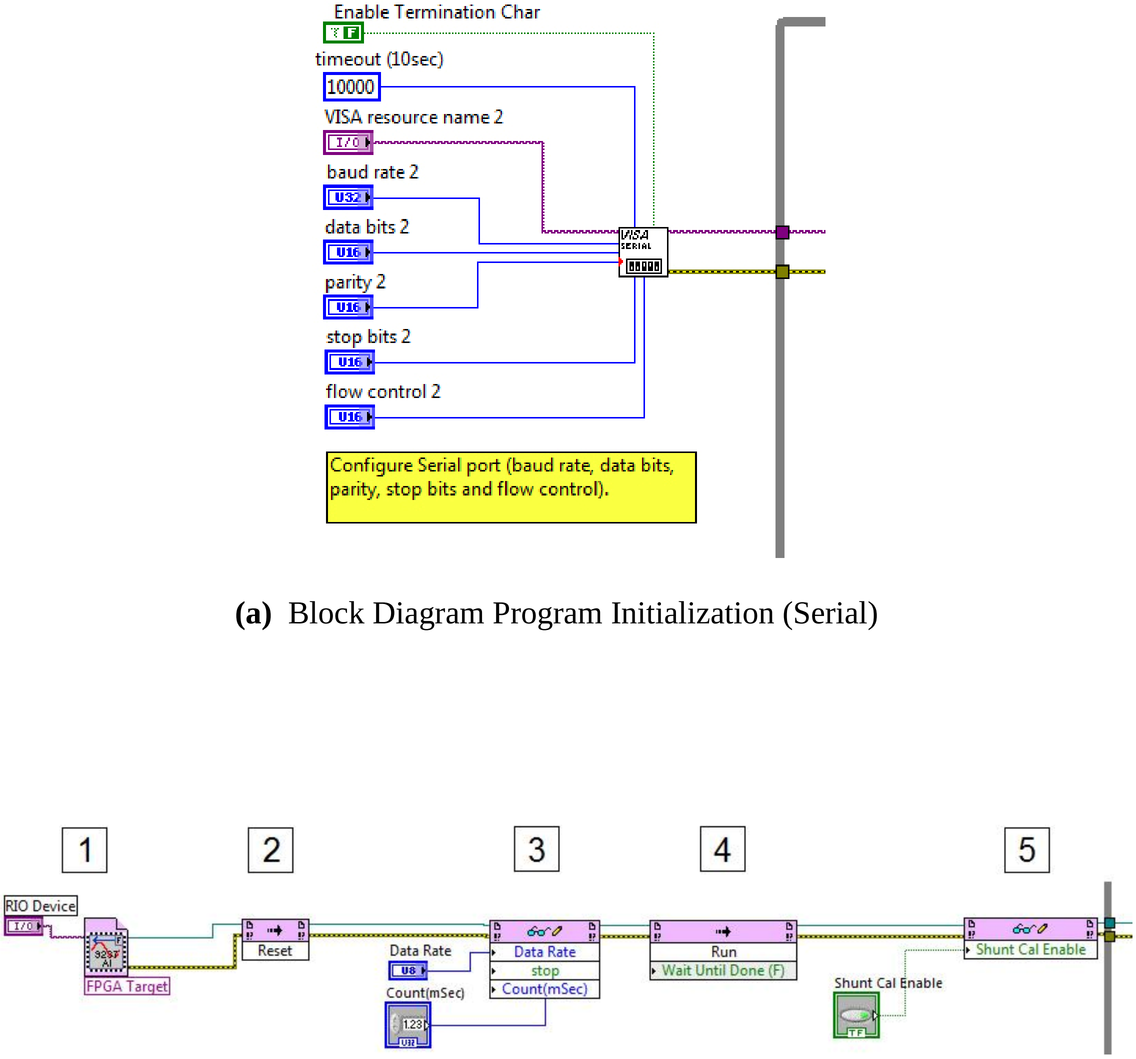


Figure 4.3 Quotient and Remainder Code

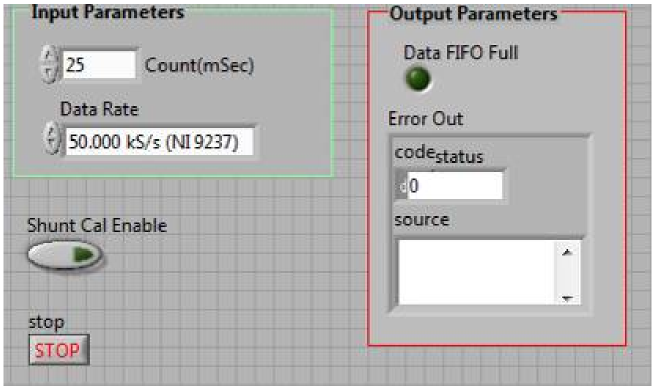


Configure Serial port (baud rate, data bits, parity, stop bits and flow control).

(a) Block Diagram Program Initialization (Serial)

(b) Block Diagram Program Initialization (FPGA)

Figure 4.4 Block Diagrams for Program Segments.

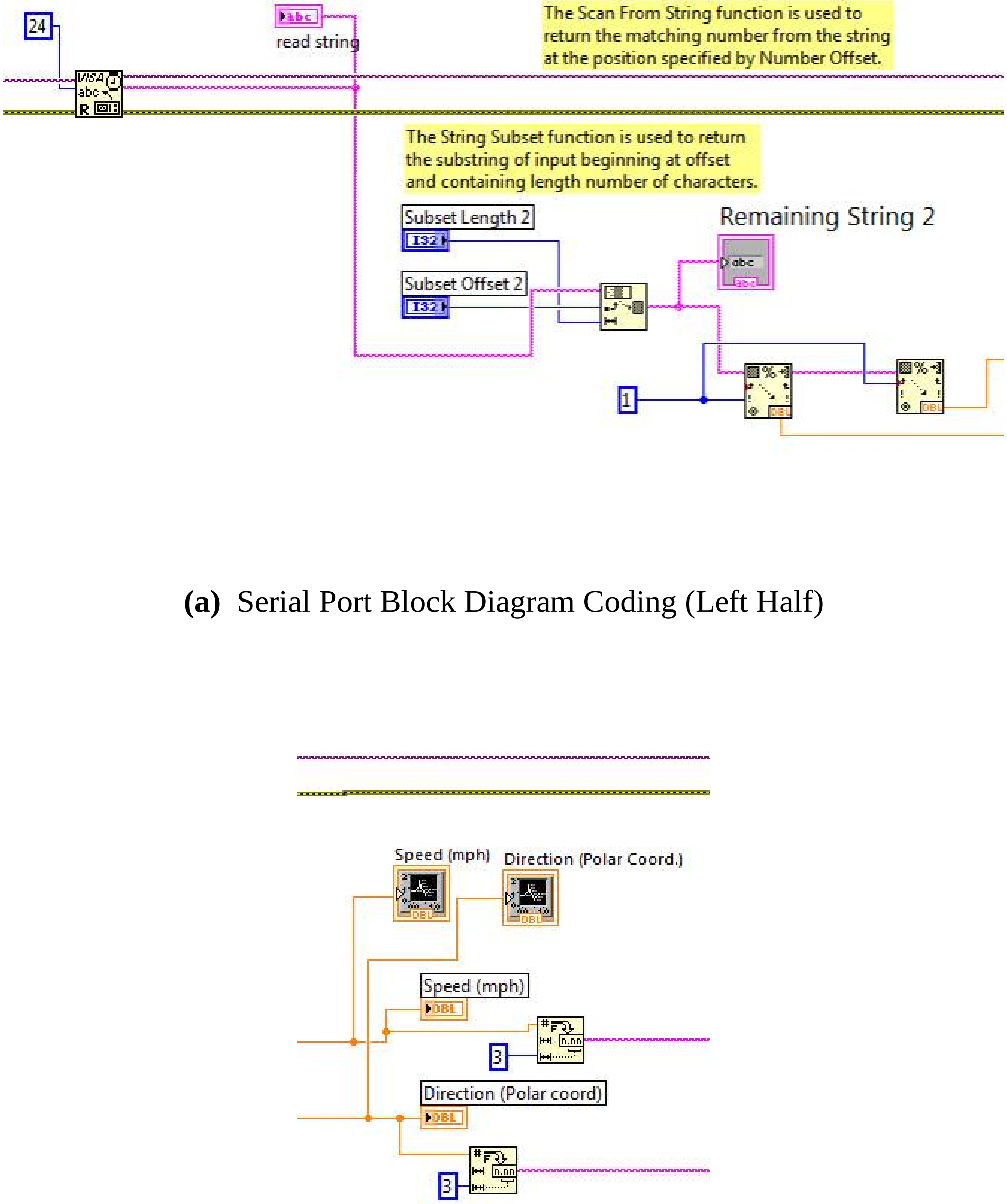


Count(mSec): 25

Data Rate: 50.000 kS/s (NI 9237)

Error Out
status
code: 0
source

Figure 4.5 FGPA VI Front Panel



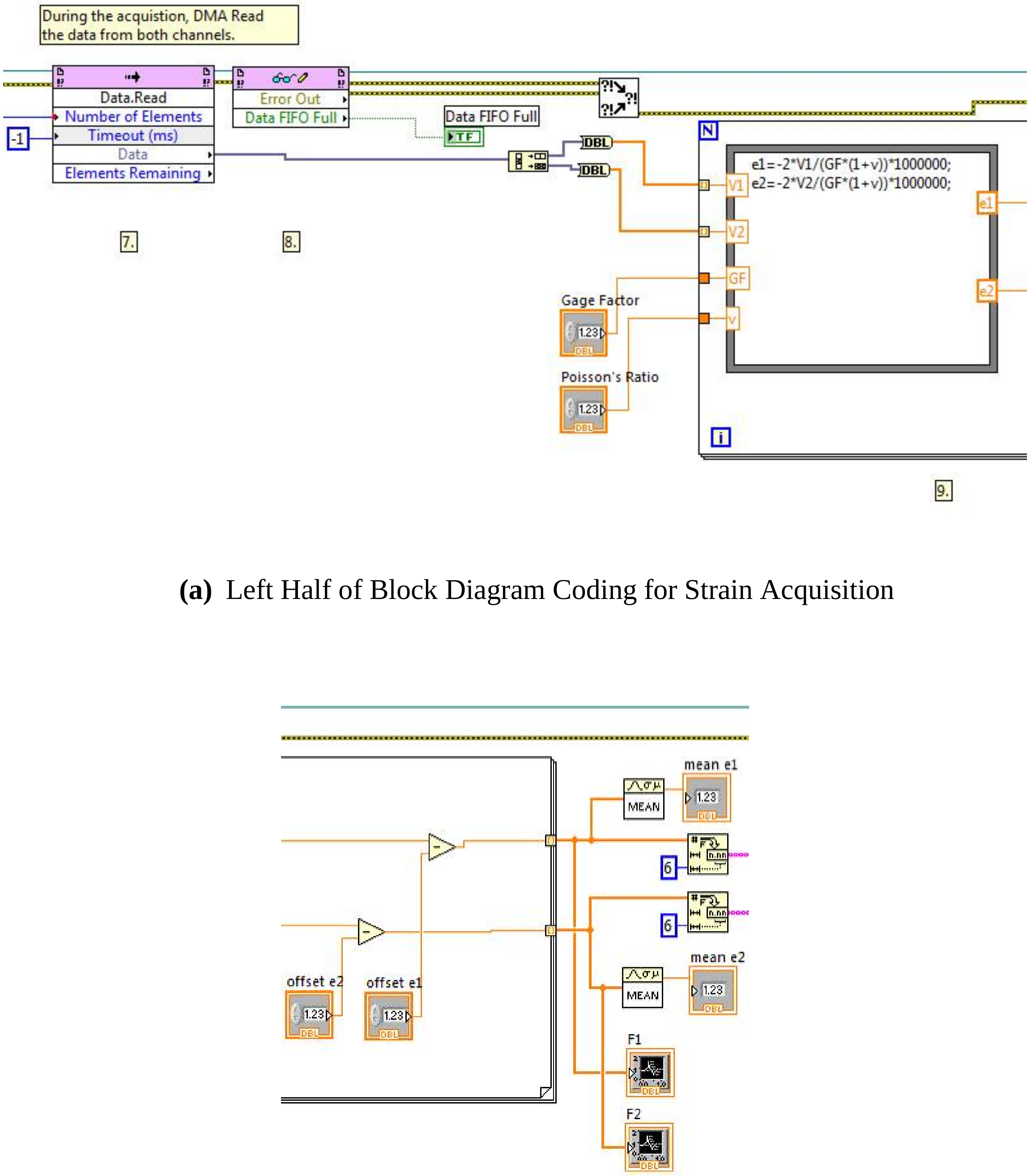
The Scan From String function is used to return the matching number from the string at the position specified by Number Offset.

The String Subset function is used to return the substring of input beginning at offset and containing length number of characters.

(a) Serial Port Block Diagram Coding (Left Half)

(b) Serial Port Block Diagram Coding (Right Half)

Figure 4.6 Serial Code Block Diagram Code for Anemometer.



During the acquistion, DMA Read the data from both channels.

```
e1=-2*V1/(GF*(1+v))*1000000;
e2=-2*V2/(GF*(1+v))*1000000;
```

(a) Left Half of Block Diagram Coding for Strain Acquisition

(b) Right Half of Block Diagram Coding for Strain Acquisition

Figure 4.7 Block Diagram Coding for Bending Strain Acquisition.

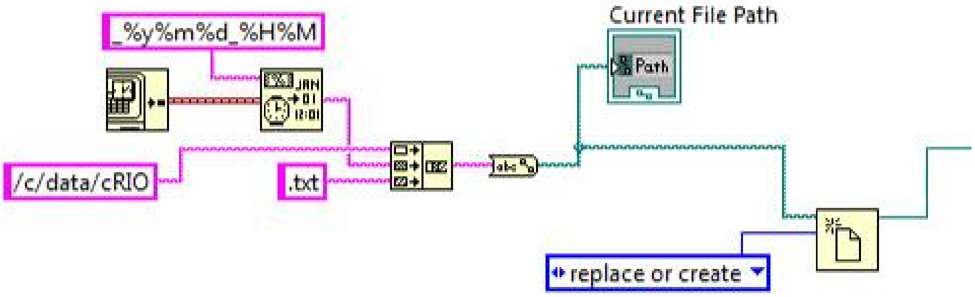


Figure 4.8 Block Diagrammatic Coding for File Creation

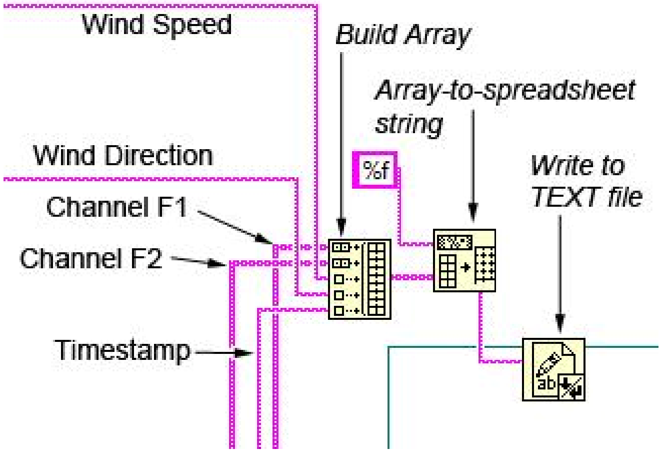


Figure 4.9 Block Diagrammatic Coding for File Writing

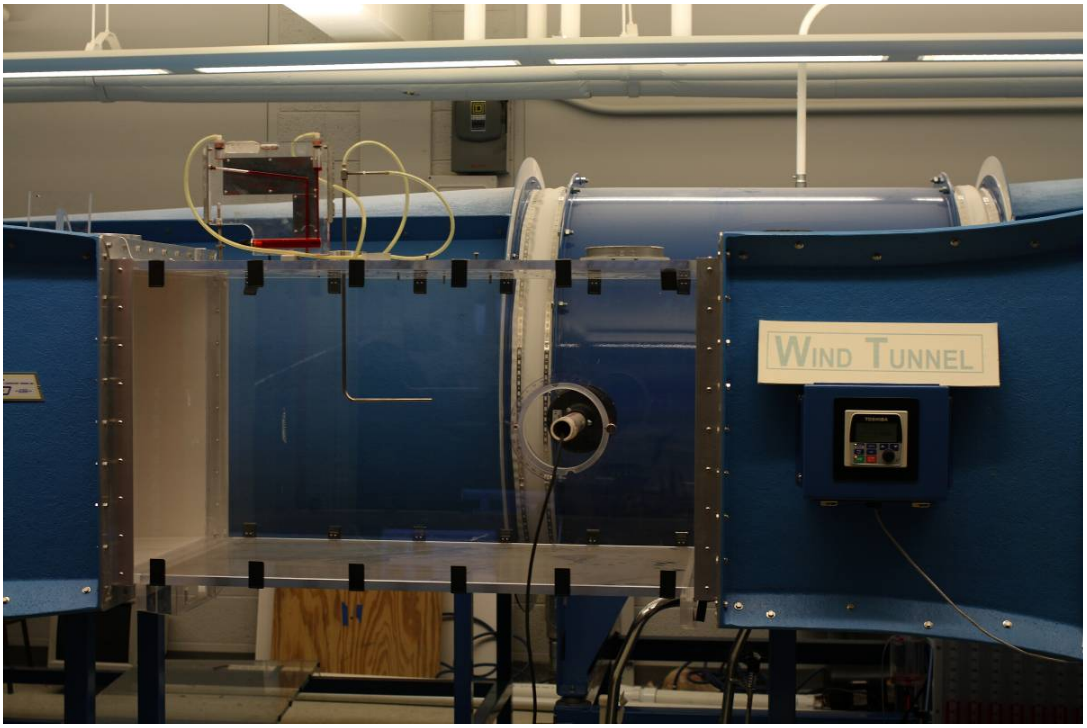


Figure 4.10 Marquette University College of Engineering Wind Tunnel

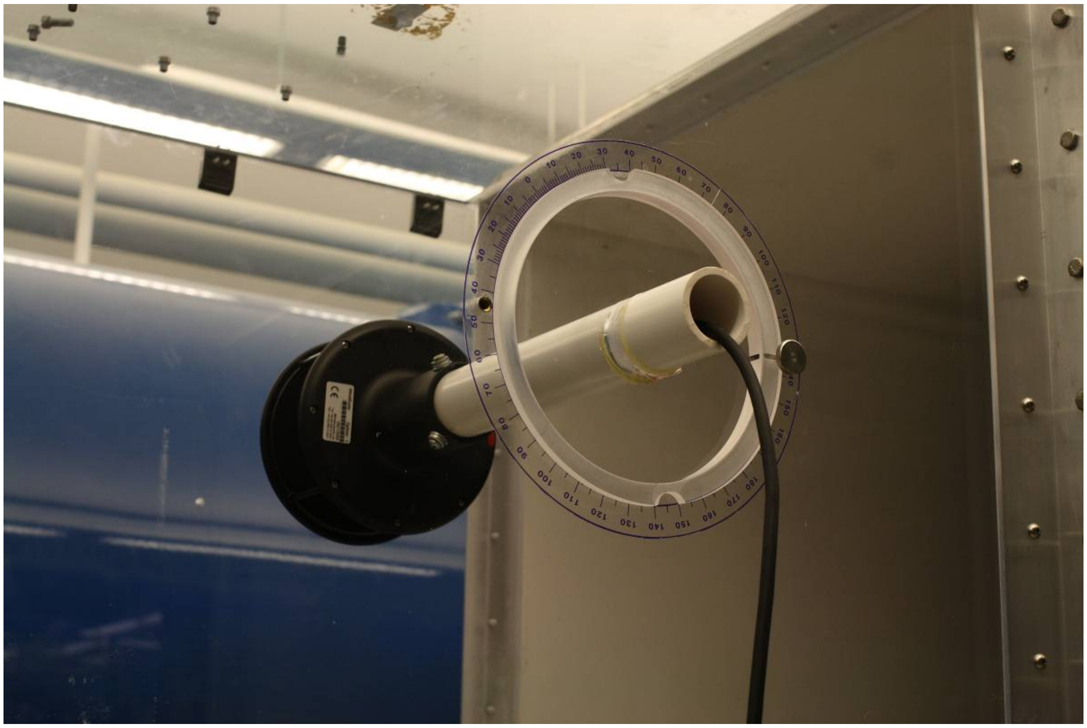


Figure 4.11 Anemometer Mount

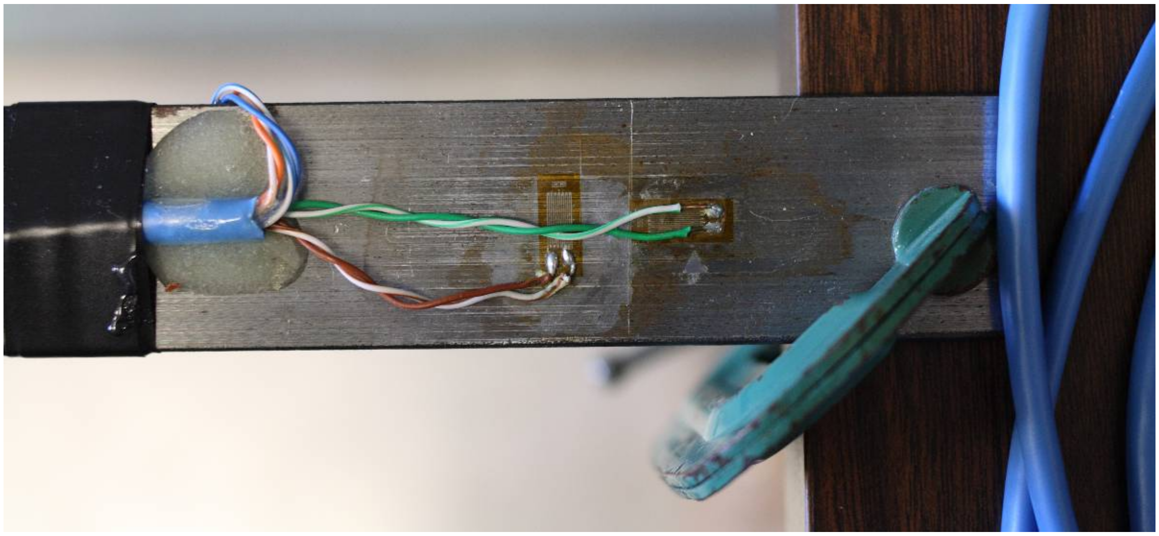


Figure 4.12 Full Bridge Strain Gauge Test Specimen (2 gauges are also mounted to the underside of the beam – not shown)

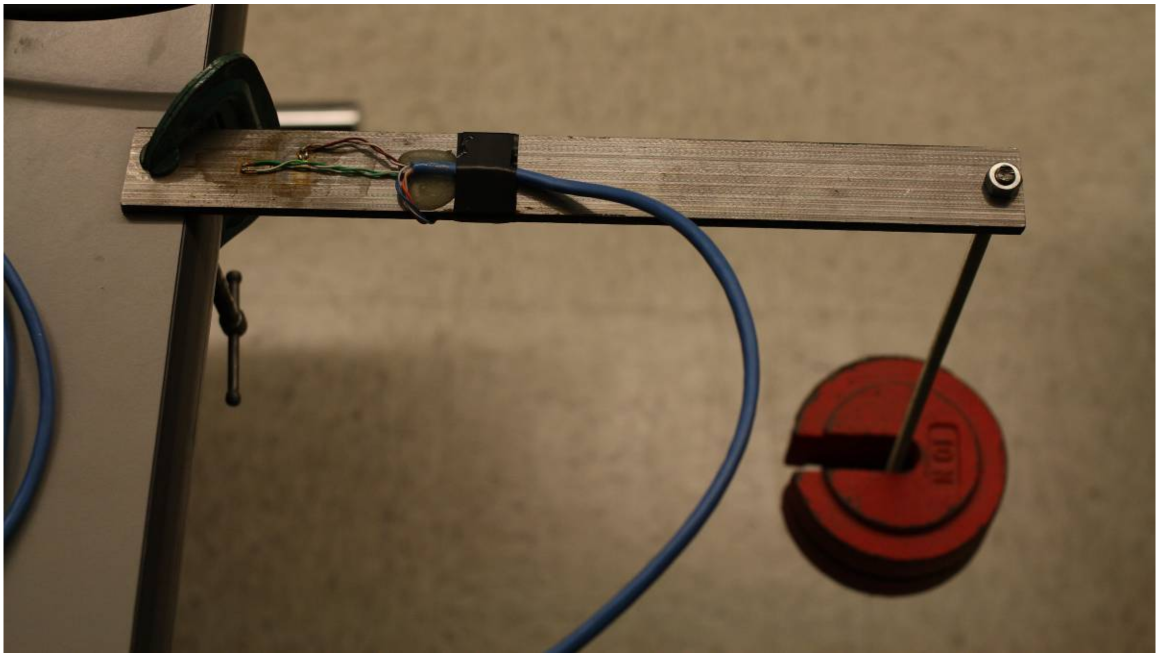


Figure 4.13 Cantilevered Metal Bar and Weighted on One End

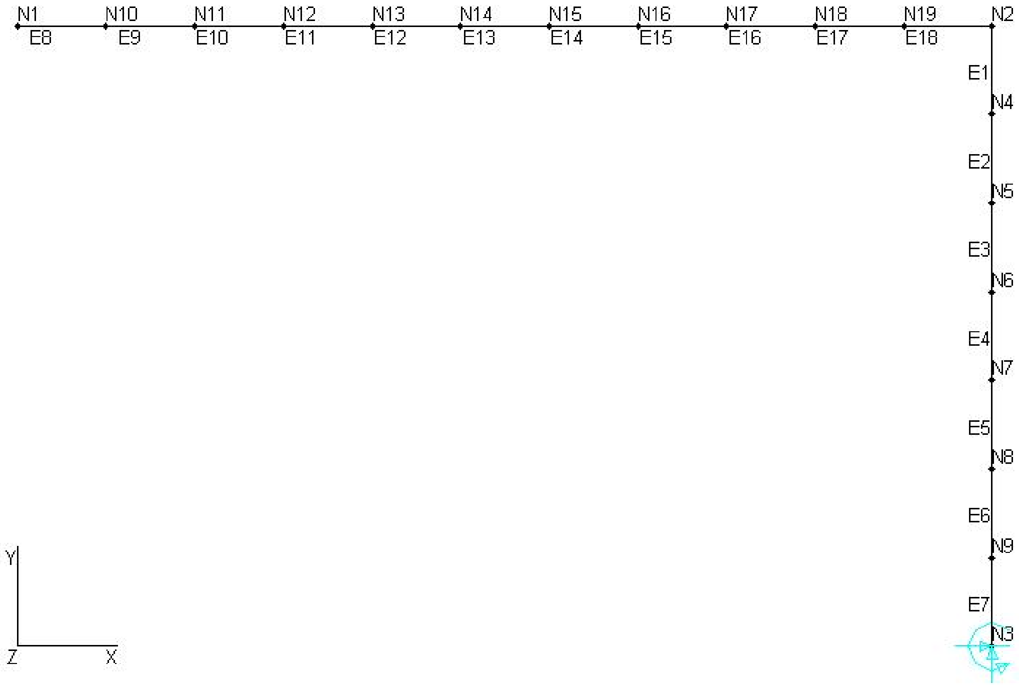


Figure 4.14 Mastan Analytical Model

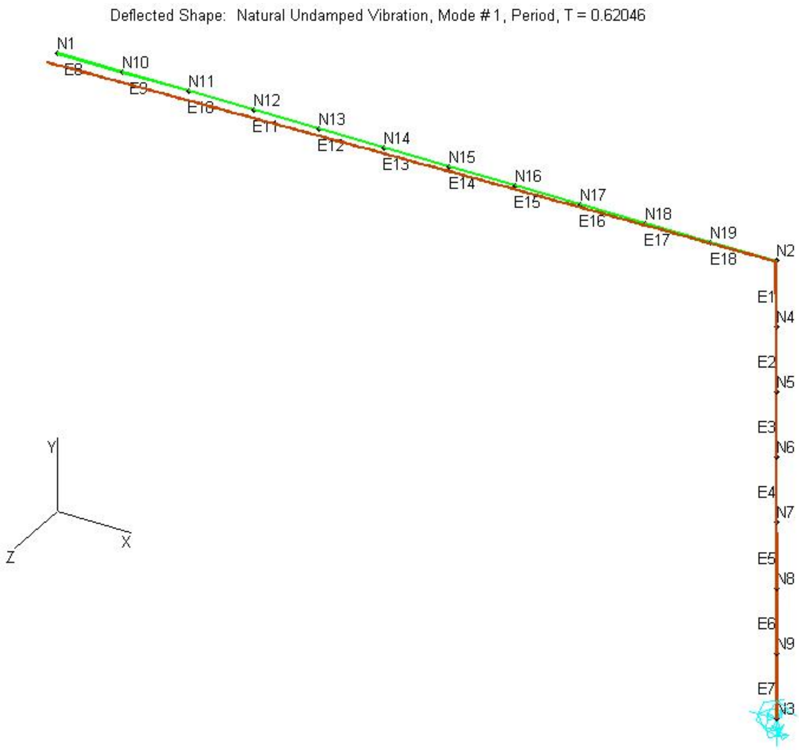


Figure 4.15 Mode Shape 1 – Horizontal Motion with Twisting About the Pole

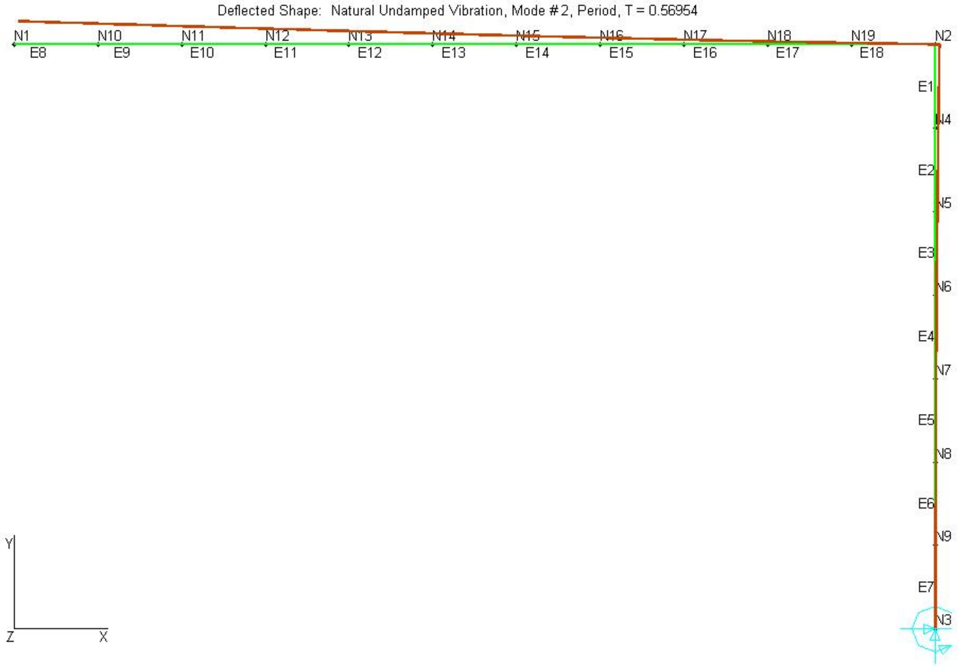


Figure 4.16 Mode Shape 2 – Vertical "Hatchet" Motion

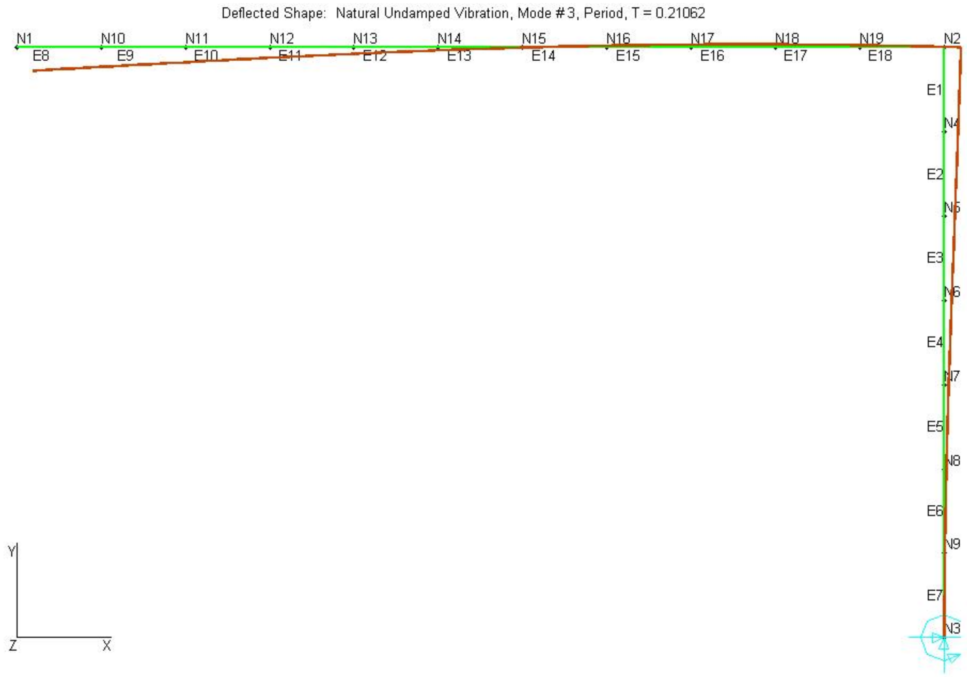


Figure 4.17 Mode Shape 3 – Vertical Motion with Pole Sway

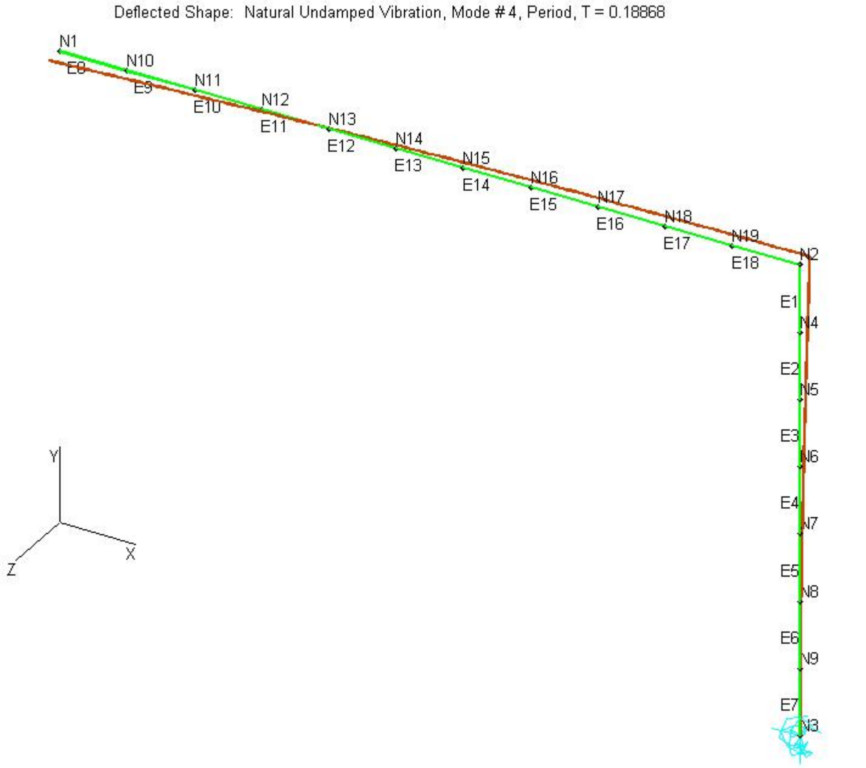


Figure 4.18 Mode Shape 4 – Horizontal Motion with Pole Side-sway

Chapter 5

Field Data Synthesis

There is an enormous amount of data recorded by the data acquisition system developed. If one includes the strain readings, wind speed and direction, and a timestamp feature, a single day worth of data is nearly four and half million data points (for the data sampling rate needed to capture data variability). Since the data acquisition system acquires data continuously, algorithms were developed to process and synthesize data using the program Matlab (Matlab 2008). This thesis will describe synthesizing one week of raw data. The overall objective of this research project from the Wisconsin Highway Research Program perspective is to obtain 6 months to a year of data. The objective of this thesis, as it relates to the data synthesis, is to develop the algorithms to facilitate synthesizing this data for later use in the WHRP effort. The present chapter provides discussion of the data synthesis algorithms developed and outlines their application to six days of raw data including the generation of a statistical analysis of the data. A strain prediction analysis based on AASHTO specifications (AASHTO 2001) is also discussed as it provides the basis for validating that the raw data acquired and subsequently synthesized is reasonable and correct.

5.1 Mast-Arm Strain Prediction

The laboratory testing, described in section 4.4, was conducted to verify that the program developed for the data acquisition system was obtaining accurate results and validate that the data acquisition instruments were operating properly. This provided confidence that the DAQ system could successfully deploy the system without error. The laboratory is a highly controlled environment and the field is not. Therefore, it was desirable to verify that the field data being recorded by the deployed system seemed reasonable. Basic mechanics coupled with AASHTO load specifications (AASHTO 2001) were used to develop a method to anticipate approximate strain readings at the gauges mounted on the mast-arm using an equivalent static wind pressure.

The model developed for the analysis is illustrated in Figure 5.1. The mast-arm-to-pole connection (right side of figure) is modeled as a fixed-end. The mast-arm is a tapered section, with a fixed end diameter of 11 inches. The tip of the mast-arm at the free end has a diameter of 6.4 inches. The strain gauges were mounted 2 inches from the connection plate. This distance was assumed to be negligible and consequently the diameter at the gauge location was assumed to be 11 inches. The plan drawings in Figure 3.4 show a centerline-to-centerline distance of 18.5 feet between the pole and the sign closest, and a distance of 12 feet between the two signs. The distance from the first sign centerline to the connecting plate is about 17.75 inches; this is after half the pole diameter, 5 inches, and two 1.75 inch connecting plates are subtracted from the 18.5 feet. The mast-arm was broken into 5 segments, two of which are 4.5 feet, or the width of the signs mounted to the mast-arm. The mast-arm segment between the two signs is 7.5 feet. To keep the arm segments similar in length, the segment to the right of the first sign was broken into two parts, one 7.5 and the other 8 feet. From left to right the arm segments are 7.7, 9.4, and 10.4 inches in diameter, taken as the average over their length.

Procedures to determine wind loading on cantilever mast-arm sign-support structures are available (AASHTO 2001). These procedures have been developed for design, but they can also be used for analysis of an existing structure. Equation 5-1 provides an estimate for the wind pressure acting on the projected area of a sign and support structure (AASHTO 2001):

$$P_z = 0.00256 K_z G V^2 I_r C_d \quad \text{(psf)} \tag{5-1}$$

where,
P_z – design wind pressure in pounds per square foot
K_z – height and exposure factor
G – gust effect factor
V – basic wind speed, expressed as 3-sec gust wind speed (mph)
I_r – importance factor
C_d – drag coefficient

The height and exposure factor is taken as 1.0 since the height of the mast-arm above the ground is less than 32.8 feet (AASHTO 2001). Since the procedure is intended for design based on statistical wind data whereas our purpose is analysis using actual wind velocities, the importance factor will also be taken as 1.0. AAHSTO recommends a gust effect factor of 1.14 (AASHTO 2001). This equation is based however, on a 3 second wind gust speed. The gust factor is intended to account for the dynamic magnification of loading on the structure due to wind gust. This analysis will be conducted for a 2 minute wind speed average. The objective of this analysis is to obtain ballpark estimates of strain for a given wind speed, and not venture into a detailed analysis, therefore, a gust factor of 1.0 will be used in analysis. The equation 5-1 thereby reduces to:

$$P_z = 0.00256 V^2 C_d \tag{5-2}$$

Wind speed information will be taken from a window of field data and used in Equation 5-2, the results of which will be used to calculate a predicted horizontal static normal strain on the structure. The actual strain data from this same window will be used to compare with the predicted value. The model should be used with data from a time period in which the wind direction is favorable for applying a pressure normal to the signs mounted on the mast-arm.

The orientation of the sign support structure was determined in an approximate manner using two lines drawn on an aerial map as indicated in Figure 5.2. One line was drawn pointing toward approximate north (top of map was assumed to be north), and the second along the orientation of the mast-arm. The angle between these two lines is approximately 50 degrees. The arm of the sign structure points in a northeasterly direction. This means that winds out of the northwest (~320° - pressure on front of sign and front of mast arm) and southeast (~140° - pressure on back of sign and mast arm) will be the most critical in terms of applying a horizontal wind force on the mast-arm.

5.2 Field Measurement Validation

The window of data chosen for this validation is 2-minutes in length, beginning at 6:45 PM on 3/12/2010 (Figure 5.3). The two minute window was chosen because it can be used to obtain 2-minute averaged wind data which is a standard measure used by entities such as the Automated Surface Observing System of the National Weather Service (NWS 2010). This 2 minute window lands approximately on the 8 hour location of Figure 5.6(a) (this figure is developed later in the discussion), which is a three-day channel history plot from March 12 – 14, 2010. The window of data was selected based on the wind direction recommendation of 320 degrees; wind direction was approximately 300-360 degrees for this window. This data is wind direction corrected (based on true north and the anemometer's alignment) but does not have a post-process offset applied to the strain readings, as will be discussed in section 5.4.

The most frequent wind direction of this window was 350 degrees, or 30 degrees from being applied normal to the mast-arm orientation. The averaged wind speed over the entire window, or the 2-min averaged wind speed, was 10.98 mph. Since wind was applied most frequently 30 degrees from normal, a correction factor of 0.866, or the sine of 60° was multiplied by the averaged wind speed to create an effective wind speed acting normal to the mast-arm. This resulted is an adjusted 2-min averaged wind speed of 9.51 mph. The average strain over this period was 8.53 microstrain. Peak wind speed (adjusted) and strain were 15.8 mph and 35.10 $\mu\varepsilon$, respectively for that period.

Now that the averaged wind speed, corresponding to V in equation 5-2, is known for this period, the analysis can proceed with the calculation of the drag coefficient, C_d. The drag coefficient for the signs mounted on the mast-arm is a function of the length to width ratio of the supported sign. The signs mounted on the mast-arm are 4.5’ x 6’ and C_d was taken as 1.15 by interpolating between values (AASHTO 2001). Since the mast-arm itself is tapered and cylindrical, the calculation of C_d is a function of the diameter and wind velocity and therefore it is a more involved calculation. The mast-arm was first broken into segments with average diameters, shown in Figure 5.1, so that the variance of C_d could be considered. The strain gauges mounted to the mast arm in the field are 2-inches from the connecting plate, therefore the segment to the right of the strain gauges was ignored. Values for C_d were 1.10 for all the mast-arm segments (AASHTO 2001).

Once the values of C_d are known, equation 5-2 can be used to find the effective static wind pressure, P_z which will vary for the signs and mast-arm segments. This pressure was converted into linear loads along the length of the mast-arm using projected heights of sign and mast arm to the wind stream. Summing moments about the right end of the mast-arm, the static normal stress at the strain gauge locations was calculated. Static strain at the horizontal gauge location was then calculated using Hooke's Law. Having a 2-minute averaged wind speed of 9.51 mph, the above procedure resulted in a predicted strain reading of 10.08 με.

The predicted horizontal strain value of 10.08 is close to the actual averaged strain value of 8.53. This analysis is a good approximation for low wind speeds, averaged over a 2-minute period, which tends to wash out the dynamic effects better captured by smaller wind averagings (i.e. 3-second wind gust average). It is expected that higher speed winds will be more turbulent and will make it harder to make loading predictions since the dynamic effects will be much greater. The gust factor is defined as the peak wind velocity over the averaged wind velocity (Simiu and Scanlan 1986). That means for this 2-minute window the gust factor is 15.8 / 9.51, or 1.66.

5.3 Natural Frequency of the System

By monitoring the response of the system, it should be possible to ascertain the natural frequency of the system. Figure 5.4 includes a vertical (F1) and horizontal (F2) bending strain history over a 6 second window. This window occurred at about 8 am on March 10, 2010. The average wind speed over that time period was 14.9 mph with a wind direction reading of 135° (Southeasterly) occurring most frequently over that interval. The wind direction is ideal because wind pressure was being applied nearly perpendicular to the backside of the signs mounted to the mast-arm and should be conducive for observing the 1st mode of vibration. This type of loading causes a negative bending moment about the vertical axis resulting in the negative F2 strain readings, seen in Figure 5.4.

There exists a well defined sinusoidal response in channel F2 with an amplitude of about 15 microstrain. Since wind is normal to the mast-arm's orientation, it seems likely that the horizontal mode of vibration would dominate the response of the structure. This is apparent as the vertical response does not have a clearly defined cycle, and for at least the first few seconds does not fluctuate more than 5 microstrain. The time between peak-to-peak or trough-to-trough of the horizontal response is 0.9 seconds (averaged period over the 6 second window) which corresponds to the natural period of the lightly damped system under the dynamically applied loading conditions. The natural frequency is therefore 1.11 Hz. The modal analysis described in Chapter 4, resulted in an approximation of the frequency for the first mode of vibration as 1.61 Hz. The actual response of the system appears to be a bit slower than predicted, but falls within reason to the analytical prediction made in the previous chapter.

Previous research involving a similar structure (Alderson 1999) included a pluck test. The natural frequency for the mast-arm structure in this study was measured to be 0.75 Hz for the first mode of vibration. The measured response and the analytical prediction made using modal analysis are all in reasonable agreement with one another.

5.4 Data Synthesis Algorithms and Statistical Results

Matlab (Matlab 2008) was used to develop algorithms that would complete certain tasks in synthesizing the data. M-files (".*m*" extension) are program files that are developed to complete a task. Mat-files (".*mat* " extension) are output files that can be used to save data in a Matlab-friendly file type. The first task that needs to be done when processing data is to bring in the raw field data from the text files compiled by the data acquisition system, and assemble them into a mat-file for later statistical analysis and other processing.

Text Read-in Algorithm

The data acquisition system records 346,000 lines of text every day, each line recorded every 0.25 seconds which contains a set of numerical data. The m-file `readtext.m` reads these 346,000 lines of text and parses out the data contained within each line: 5 strain readings from channel F1, 5 from F2, a single wind speed and direction reading, and a timestamp (with AM / PM format). This m-file parses this data into an array, *data*, for eventual use by other m-files. This type of hierarchy is necessary as three days of data takes `readtext.m`, under current computing power, 18 hours to process. Once this data is converted and saved to a mat-file, it allows much faster processing of the data by other m-files. `readtext.m` can be configured to read multiple text files corresponding to multiple days in a single session, but will take proportionally longer depending on how many files are being processed.

The five strain readings per channel within each line of text (Figure 4.1) are taken as an average, primarily as a way to reduce the amount of data in post-processing. Since the response of the system appears to be slower than the modal analysis predicted, averaging the strain readings should not result in a loss of any response characteristics. The effective rate of strain data acquisition is 4 Hz after averaging, and maintains a sampling rate at least twice the frequency of response (Wikipedia 2010) for mode 1 and 2 which contribute the most to the structure’s response. This results in an output array which contains an averaged F1 value, averaged F2 value, wind speed, wind direction, and a timestamp converted into numerical form with year, month, day, hour (24 hr format), minute, and second, each in its own column.

The six days of data processed were taken from the period March 12 at 11:24 am to March 18 at 11:32 am. The corresponding text files taken from the Compact RIO system are:

cRIO_100312.txt

cRIO_100313.txt

cRIO_100314.txt

cRIO_100315.txt

cRIO_100316.txt

cRIO_100317.txt.

Sustained high winds were prevalent during the first three days, followed by three days of relative calm. The time required to process the text files required that the first three days be processed into one mat-file followed by a second mat-file containing the last three days. The mat files are entitled:

data_100326.mat

data_100329.mat

`readtext.m` automatically creates a mat-file with the prefix "*data_*" followed by a timestamp corresponding to the day processed, double-digit year, month, and day. It is possible to overwrite an existing file if multiple segments of data are processed in a single day as they will have identical timestamps.

Reference for Interpreting Results

It is very important to provide a reference standard to interpret the results. Wind directional data is recorded in degrees, corresponding to that of a compass so that zero represents North; 45 degrees represents Northeast; 90 degrees represents East; and so forth.

Reference for strain data is made using an angle defined through viewing along the mast-arm toward the free-end. Figure 5.5 illustrates the viewing reference defining this angle. Zero degrees is the top face of the mast-arm, 90 is the left extreme location, 180 the bottom, and 270 the right extreme location.

When referencing data as it pertains to tensile (positive) strain, bending of the mast-arm downward corresponds to positive bending, with the moment vector, M_{F1} to the left (Figure 5.5). The strain gauges located at 0 degrees and 180 degrees (F1 gauges) are the instruments that measure this component of bending termed here as major axis bending.

Conversely, the strain gauges located at 90 and 270 degrees (F2 gauges) are considered the minor axis bending gauges where positive bending is that which causes positive (or tension) readings in channel F2, with the moment vector, M_{F2} pointing downward (Figure 5.5). These minor axis bending gauges provide capability of monitoring horizontal motion (bending) of the mast arm.

Figure 5.5 also illustrates computation of a resultant bending moment along with a reference angle. This resultant moment can be used to determine the angle at which peak strains occur around the perimeter of the mast arm. It can also be used to evaluate strain histories that occur at user-defined points around the mast arm perimeter.

Strain Offset and True North Correction

The data acquisition program described in Chapter 4 contains an offset feature that can be used to remove the initial non-zero readings from the strain gauges. In other words, when no wind is blowing, the strain gauges should read exactly zero. The strain gauges are instruments that utilize electrical signals and readings of zero are not probable nor expected in these circumstances. As a result, offsets are used to correct for these often minor variations from theoretically zero strain.

The DAQ system was deployed in early March on a day with light winds around 5-mph. Strain reading were relatively stable with strain readings of 114 and -93 microstrain for channels F1 and F2, respectively. The values of 114 and -93 were then entered into the system as initial offsets, adjusting both channels to read near zero strain. Over time, strain gauge signals have a tendency to drift, or a deviate from true readings of zero strain in unloaded states.

Rather than providing dynamic computation of offsets to periodically apply to the data acquired real time, it was decided to remove strain drift during the data post-processing. `DataAdj.m` was developed to quantify signal drift and apply an offset to compensate for this as a processing feature of the raw data. This m-file takes the mat-file written by `readtext.m` and pulls in the data. It runs through each data point looking for wind speeds less than or equal to 1 mph. When a wind speed meeting that criterion is found, the corresponding strain values from F1 and F2 are binned into separate arrays. After the m-file runs through the data in its entirety it takes an average of all the binned values for each of these channels. This value then becomes the offset value. The calculated offset is then applied to each value within the original strain data arrays.

The main assumption is that the mast-arm will be generally stable at low wind speeds and should read near zero strain whenever that occurs. It is also assumed that wind speeds less than 1-mph will most likely occur during periods of relative calm. Review of numerous wind speed history plots confirms the above listed assumptions. When `DataAdj.m` completes execution, it displays the offset values, and the max and min values of strain within the offset bin, as well as the number of readings included in the averaging. This display can be used to quickly verify the above assumptions seem appropriate, by having extreme values close to the averaged value.

Earlier discussion outlined the fact that the anemometer reading of North (zero degrees) was 6°25’ away from true north. Therefore, this value is added to the wind direction angle readings. `DataAdj.m` performs this adjustment through application of a 6° angle increase as correction. Direction readings from the anemometer are integer values and therefore, the 25-minute fraction is omitted.

`DataAdj.m` replaces the old arrays of data with the new adjusted arrays, which are then saved into the mat-file “*dataAdj_*(timestamp)*.mat* ”, with the same format as, but not replacing, the mat-file created by `readtext.m`, “*data_*(timestamp)*.mat* ”. At this point, someone wishing to synthesize the data would have two variations of the same data from which to work: adjusted with strain offset for the period synthesized or unadjusted (without strain offset). The adjusted mat-file contains all necessary information related to the offsets applied to the data and can be viewed by loading the mat-file and calling for the variables “offsetF1” and “offsetF2”. `DataAdj.m` can also be used to apply correction to the wind direction, and *not* perform an offset to the strain data if desired.

DataAdj.m was used to process 2 sets of three days (March 12-14 and March 15-17) of raw data. The first three days had only 1,302 strain readings (out of 1,036,800 total) where the wind speed was less than or equal to 1-mph. The offset calculated was 3.74 microstrain for F1 and 1.53 microstrain for F2. Since there were a small number of events that resulted in a relatively insignificant offset, it was decided an offset did not need to be applied to the results. The second set of three days however was a good candidate for applying an adjustment. There were 36,344 events, or 3.5% total occurrence, of wind speeds 1-mph or less. The offset for F1 and F2 were calculated to be -6.02 microstrain and -9.76 microstrain, respectively. The max and min values were 16.86 / -15.39 for F1 and 28.50 / -17.19 for F2. These min and max values are not large, but they are large enough that one might question whether the offset value determined is a good reflection of drift. The area in Figure 5.6(b) between 60 and 72 hours had some large wind gusts which appeared to contain some wind speeds below 1-mph. This is probably where the larger positive strain values originated (especially F2). Since the actual average is much closer to the minimum value, they must not have affected the results significantly. Therefore the last three days were synthesized using adjusted data.

Combined Plot

The most basic function developed was an m-file called Combined.m. This file simply takes the data generated through DataAdj.m and plots the overall history of wind speed, wind direction, strain at sensors F1 and strain at sensors F2. Combined.m created the plots shown in Figure 5.6, all with corrected wind directions. Figure 5.6 (a) is for the period March 12-14, 2010 (unadjusted with strain offset) and Figure 5.6(b) is for the period March 15-17, 2010 (adjusted with strain offset).

The plots in Figure 5.6 are useful in getting an overall picture of what was occurring during the given time period and could be used to identify an area in time that has interesting conditions for which further analysis might be conducted. This tool is also useful in quickly verifying that the data acquisition system appears to be working properly. In other words, one can use these plots to make sure the values being recorded make sense and ensure that large drift in the system is not occurring. Figure 5.6 includes periods of large magnitude strain during large wind gusts and conversely areas of relative inactivity occur during low wind speeds.

Vertical lines in Figure 5.6 define 12-hour intervals for the monitoring period. The starting time and date are listed in the lower-left corner. The algorithms to be discussed in the next section are able to combine old data with newer data in a concatenated manner. The present m-file being discussed does not do this. Combining long periods in a single plot becomes difficult to view. Instead it plots data files with user-defined boundaries (e.g. March 12-14) one at a time.

The period from March 12-14, 2010 exhibited periods with large wind speeds including some gusts on the order of 40mph. The wind direction was incredibly consistent, as winds were out of the north-northeast for two days straight. It is expected that wind out of that direction would cause positive bending of the mast-arm in the horizontal plane. This is seen in the data recorded. Horizontal motion (bending about the minor axis) is very active during this period with several strain readings of 120 microstrain or more.

Strain resulting from bending about the major axis (recorded by the vertical gauges F1) appears to be mirrored about the x-axis in Figure 5.6(a). The activity of this period made it extremely difficult to determine if drift was present in the system, which is partly why an offset was not applied to this set of data. It simply appears that vertical oscillations of the mast arm were occurring. This would indicate that aeroelastic phenomena (e.g. vortex shedding) may have been occurring nearly constantly during the time frame from March 12-14, 2010.

The active period from March 12-14, 2010 was followed by the relatively inactive period of March 15-17, 2010. Figure 5.6(b) includes the graphs of wind speed, wind direction and strains measured during this period. During this time period, the average wind speed was typically no more than 10-mph and the direction from which it came varied greatly. There were periods of relatively large average wind speeds at the start and end of this interval (e.g. 0 and 72 hours). However, the relative lack of wind activity in this period allows us to view different non-wind related phenomena that might affect the mast-arm (e.g. temperature-induced strain).

The strain histories plotted in Figure 5.6(b) have a cyclical nature during low-speed winds, which appear to have a period of one day. The plot begins rather actively at 11:27 AM on March 15. Twelve hours later the wind is light and both channels F1 and F2 appear to be drifting to lower values of strain. A sharp increase in strain is recorded by both channels at what is approximately 7 AM in the morning and the cycle repeats itself. It is theorized that sunlight is heating up the mast-arm and causing temperature induced strain resulting in vertical movement of the mast arm. Sunlight would affect the top fibers of the mast-arm more than the bottom fibers, as the bottom fibers would rarely see sunlight. Thermal expansion of the top fibers would cause a downward bending of the mast-arm resulting in positive values of strain for F1. Cooling during the evening would alleviate these strains and the mast-arm would move back to its normal state. The increase in horizontal strain (F2) around 7 AM is likely due to the increasing wind speed at that time out of a northerly direction which would also result in positive bending of the mast-arm.

The strain data plotted in Figure 5.6(b) had offsets applied that increased each F1 and F2 value by 6.02 and 9.76 microstrain, respectively. The original plot of *un*adjusted data recorded daytime strain values around zero, with nighttime values around -10 microstrain for both channels. The vertical offset calculation for this period of three days seems appropriate because the nights typically had very light wind and obviously no sunlight to cause metal expansion, therefore it can be argued that strain readings at night should be around zero, not -10 microstrain. The offset adjustment brought the nighttime values of F1 closer to zero. During the day it would then make sense that positive values of strain were occuring as the top of the mast-arm heated up and expanded.

The response of F2, over the same period shown in Figure 5.6(b), has 4 distinct active periods (0 and 72 hours being most active) separated by 3 inactive periods. At time zero, F2 is already active, large winds out of the north are causing positive values of strain, as expected. The following inactive period, beginning at about hour 8, appears at first to have been overcompensated by the applied offset of 9.76με, but by about 18 hours it appears that channel F2 has drifted down to around zero strain, which seems reasonable given the light winds during that specific period. Slightly after 18 hours, about 7 AM, around the same time F1 begins to get active, there is a noticeable uptick in F2 strain readings. Winds begin to pick up around this time, out of a northwesterly direction and consequently an increase in positive strain seems logical. As the plot passes the 24 hour mark the readings begin to drift more negative. This can be accounted by the fact that during this time the wind had been shifting from a northwesterly to easterly direction. Winds from the east will result in negative bending in the horizontal plane. This 24 hours cycle is repeated, attributed to the repeatability of wind patterns from day to day.

Wind Synthesis

Synthesizing the wind data is a two-step procedure to keep tasks separate and allow for quicker processing of data. The first step takes the form of `Wind.m`. This m-file pulls the wind data out of either mat-file "*data_*" or "*dataAdj_*". `Wind.m` then does a couple of things. First, the m-file creates information suitable for generating a histogram of wind speed and direction. A histogram analysis of wind speeds is performed using bins centered on multiples of 5 (i.e. 0, 5, 10,… mph bins). Then a wind directional histogram analysis is performed using the 8-cardinal directions. Lastly, the m-file creates individual arrays of wind speed based upon their wind direction counterpart reading. For example, a wind speed with a recorded wind direction of 45° would be placed in the northeast bin, this will allow for analysis of wind speeds for a given direction. Once these processes are complete, the synthesized speed and direction data is written to yet another mat-file "*wind_*(timestamp).*mat*"

`Wind.m` also allows the user to combine the recently processed data with data that was processed at an earlier time. The newest data is first binned and then appended to the old binned data and then new counts are added to the histograms. A new file is then output, which can again be brought in next time. The flow of adding data to existing data can be described as the “snowball” effect. The file of synthesized data will continue to get bigger and bigger, resulting in a continuous set of wind speed and direction data. Successive runs with `wind.m` will allow the user to synthesize as many days of data as computing power and time will allow.

`WindPlot.m` was created to take the output file from `Wind.m` and present the data graphically. The first graph, Figure 5.7, is a wind speed histogram with 5-mph bins beginning with zero and ending with 50 mph. For the six given days, wind speeds were predominantly in the 5-mph bin (40% of all wind speeds that occurred), and winds of 10-mph occurred 23 percent of the time. This histogram can eventually be used to evaluate wind speed probability models. Ideally, when more data is processed, Figure 5.7 will begin to look more like the histograms for the 10-year periods evaluated in phase one of the WHRP research effort (Foley *et al.* 2008).

The second graphical output is a wind direction probability polar histogram shown in Figure 5.8. This figure is modeled after the previous work of Foley *et al.* (2008) and can eventually be used to evaluate wind direction probability models. The data is binned according to the 8-cardinal directions of the compass (North, Northeast, East,…) in terms of a percentage of time wind was out of one of those directions.

In this specific time period, wind was predominantly out of the northwest, north, and northeast. These three directions make up nearly 80 percent of the prevailing wind direction for this six-day period. The mast-arm is oriented along the 50-230 degree line, or a line extending along the northeast-southwest directions. Therefore, the wind directions prevalent during the six days are favorable for applying wind pressures that cause positive bending about the vertical axis in the mast arm.

The average wind speed for each cardinal direction is shown in Figure 5.9. The first three days of sustained high winds were predominantly out of the north-northeast. This figure shows clearly that trend, as the average wind speeds for those two directions are over 10-mph, with the remaining direction bins not exceeding 5 mph wind speeds.

Strain Synthesis

The algorithm for processing the strain data is also a two-step process. `Strain.m` is responsible for breaking down the data mat-file into parts for use in statistical analysis of strain readings.

In trying to better understand fatigue induced fracture within mast-arm connections, it would be important to have a complete strain history that corresponds to certain locations around the mast-arm perimeter. It is expected that not all stress cycles will occur in the same location on the mast arm's circumference throughout the life of the mast-arm. The reason for this is that the motion of the mast arm is not strictly limited to a vertical or horizontal plane (as will be shown). The impact of this is that peak tension strains will tend to migrate around the perimeter of the mast arm in accordance with the major- and minor-axis bending moments.

The review and synthesis of the literature completed in Chapter 2 illustrated that the way previous researchers monitored strain and stress ranges was not clear. The previous research efforts seem to indicate that strain histories from individual gages were all that was used to assess fatigue life and fatigue damage. This methodology for assessing fatigue damage and stress ranges does not account for the fact that damaging stress ranges can tend to move around the perimeter of the cross-section and an individual gauge location may not be indicative of peak strain and therefore, stress.

The algorithm described in this section of the chapter was designed to calculate the resultant strain acting at specific locations around the circumference of the mast arm. This resultant strain is essentially the peak tensile strain resulting from the resultant bending moment acting on the cross-section of the mast-arm. These resultant tension strains are computed at each instant in time (4 times per second). The algorithm allows the location of peak tensile strain on the mast-arm circumference to be identified and tracked with time. Stress ranges around the perimeter of the mast arm can also be counted and regions more likely to suffer first cracking resulting from fatigue can be identified.

The following discussion is laid out to provide the reader with an understanding of how the strain and moment resultant are calculated and how they relate to one another. The resultant moment applied to the mast-arm connection was calculated using the flexure formula, shown in equation 5-3. M_{major} and M_{minor} correspond to M_{F1} and M_{F2}, respectively, according to Figure 5.5.

$$\sigma = \frac{M_R \cdot y}{I} = \frac{\sqrt{M_{\text{major}}^2 + M_{\text{minor}}^2} \cdot y}{I} \tag{5-3}$$

Again, referring to figure 5.5, the horizontal bending moment component, M_{F1}, causes a tensile strain ε_{F1}, applied 90 degrees clockwise. The relationship between M_{F1} and ε_{F1} is given by:

$$\varepsilon_{F1} = \frac{M_{F1} \cdot y}{E \cdot I} \qquad M_{F1} = \frac{\varepsilon_{F1} \cdot E \cdot I}{y}$$

Likewise, the vertical bending moment component, M_{F2}, causes a tensile strain ε_{F2}, applied 90 degrees clockwise. The relationship between M_{F2} and ε_{F2} is given by:

$$\varepsilon_{F2} = \frac{M_{F2} \cdot y}{E \cdot I} \qquad M_{F2} = \frac{\varepsilon_{F2} \cdot E \cdot I}{y}$$

The resultant moment is therefore,

$$M_R = \sqrt{M_{F1}^2 + M_{F2}^2} = \sqrt{\left(\frac{\varepsilon_{F1} \cdot E \cdot I}{y}\right)^2 + \left(\frac{\varepsilon_{F2} \cdot E \cdot I}{y}\right)^2} \qquad (5\text{-}4)$$

Pulling out the common terms within the radical,

$$M_R = \frac{E \cdot I}{y}\sqrt{\left(\varepsilon_{F1}\right)^2 + \left(\varepsilon_{F2}\right)^2} \qquad (5\text{-}5)$$

Moving the term outside the radical over to the left of equation 5-5 yields,

$$\varepsilon_R = \sqrt{\left(\varepsilon_{F1}\right)^2 + \left(\varepsilon_{F2}\right)^2} \qquad (5\text{-}6)$$

The angle defining the orientation of the resultant moment is found by geometric computation based on quadrants and the signs of F1 and F2. Figure 5.5 illustrates the component bending moments corresponding to positive F1 and F2 strains, and the resultant bending moment and the angle of orientation of this moment. As the m-file completes its execution it saves all the resultant strain and bending moment data and reference angles to the mat-file "*strain_*(timestamp).*mat*". As with `wind.m`, older data can be added to the data just processed so a contiguous record of data can be maintained.

`StrainPlot.m` accesses the mat-file saved by the aforementioned m-file, which then bins resultant strain based on its reference angle. The circumference of the mast-arm is broken into 36 bins (each being 10 degrees, and centered on multiples of 10). This creates a history of events that occur within these 10-degree slices of "pie". This is not a complete history however, only a history of maximum tensile strain that occurred within the boundaries of a given bin. In otherwords, if the resultant tensile strain and angle fall within a certain bin, the strain magnitude is recorded in the bin. When resultant tensile strain migrates along the circumference to a different bin, nothing is recorded in the original bin.

If the mast-arm was moving strictly in the horizontal plane, the maximum tensile strain would occur at either 90° or 270° (refer to Figure 5.5). If the mast-arm motion is not isolated to either the horizontal or vertical planes, the maximum tensile strain is going to move to different location around the mast arm circumference. Complete strain histories could be developed for each of the 36 bins. As a result, histograms of stress range data could be generated for the centroid location for each of the 36 bins. However, this was not done in the present study.

The first plot output by this m-file is a resultant strain location history in polar form, as shown in Figure 5.10. This figure plots the location of resultant tensile strain on the mast-arm circumference for each instant in time, presented as a percentage of occurrence. Two well defined components make up the plot, a horizontal band and a band more or less 45 degrees from horizontal. The horizontal band is likely a result of the first three days with strong winds creating large tensile strain in the 90° region. The angled band indicates that strain readings from F1 and F2 were close in magnitude for an extended period of time since resultant strain occurred often at a location nearly equidistant between the two sets of gauges. It would make sense that this is a representation of the last three days, which were calm, and did not give rise to a dominating motion in the horizontal plane.

The second graphical output, shown in Figure 5.11, is a polar chart that plots the maximum value of tension strain within each bin. The angled band seen in the previous figure all but disappeared. This implies that the portion of the mast-arm with bins centered on 75 to 105 degrees saw the largest strain, and that the large number of peak strain events that occurred 45 degrees from horizontal (Figure 5.10) were not large in magnitude. This chart (Figure 5.11) coupled with the previous chart (Figure 5.10) is a useful tool in identifying possible trouble spots on the mast-arm perimeter. Peak tension strains occurred in the 75-105 degree band frequently (Figure 5.10) in which some were large in magnitude (150με, Figure 5.11).

When contiguous months of data are eventually combined, Figure 5.11 may lose some of its effectiveness as a tool to identify trouble spots. It only displays a single value out of what might amount to hundreds of thousands within a single bin. Figure 5.12 presents the averaged value of strain for each bin and begins the process of migrating information in these bins toward statistical information within each. Except for the two deviations at 45° and 225°, the plot appears mirrored about the 90-270 degree (horizontal) axis. This is indication that vertical motion of the mast-arm is cyclical because the average strain for both top and bottom of the mast arm appear nearly the same. This is confirmed by the history of strain instrument F1 from Figure 5.6(a), which showed nearly every positive strain was mirrored. The horizontal motion, however, appears to be governed primarily by wind, as might be expected, since the overall trend is significantly offset toward 90° by wind that applies a pressure in the horizontal direction normal to the mast-arm. Offsets such as these are important to note because it implies that one face of the mast-arm is likely undergoing stress cycles primarily in the tension region, with little compression strain occurring.

When `StrainPlot.m` finishes it saves all the binned data to another mat-file, in the event that analysis of the data pertaining to a certain location on the mast-arm is desired. The m-file can also be used as the basis for developing statistical information in each of the 36 bins around the mast arm circumference.

Summary

Figure 5.13 illustrates the m-file programmatic flow and the data processing flow. The m-file, `readtext.m`, takes the raw field data and converts it into a Matlab friendly file format (.mat). The m-file, `DataAdj.m`, calculates and applies an offset to the strain data (if the user specifies), and also corrects the wind direction data to reflect true headings. The m-file, `CombinedPlot.m`, plots adjusted or unadjusted strain (according to user definition) and wind data history. The m-file, `Wind.m`, uses the data generated by the `DataAdj.m` and bins wind speed based on the corresponding wind direction. This m-file also creates histograms of wind speed and direction, which are then output to a mat-file. The m-file, `WindPlot.m`, imports data from `Wind.m` and plots it. The m-file, `Strain.m`, uses data generated by `DataAdj.m` and calculates resultant strain, resultant moment, and their respective locations around the circumference of the mast-arm. This data is then written to a mat-file. The m-file, `StrainPlot.m`, then accesses the data generated by `Strain.m` and bins resultant strain based on circumference location and generates several graphical representations of the data.

All Matlab m-files written and used for the synthesis described in this chapter are contained in Appendix B.

5.5 Concluding Remarks

An analysis of the raw field data was completed in this chapter. Six days of raw field data (March 12-17, 2010) was synthesized and mast-arm behavior was discussed. A review of 6 seconds of strain history during a significant wind event in an ideal orientation indicated that the mast-arm support structure appears to have a natural frequency of 1.11Hz with regard to mast-arm horizontal motion (twisting motion in the vertical pole), and a natural frequency of 1.23 Hz for vertical motion. These value were slightly less than the values obtained via modal analysis (1.61 and 1.75 Hz), but is consistent with fundamental natural frequencies found for similar structures in previous studies.

Data can be processed in many ways and from many points of view. It is not completely known yet what types of analyses will be done with the field data. Some of the future synthesis done will certainly follow suit with what has already been developed, such as the wind-speed and wind-direction statistical information. With that in mind, the algorithms discussed were developed as a means for further data analysis by allowing a user to take what was already processed and be able to manipulate to achieve a different end result.

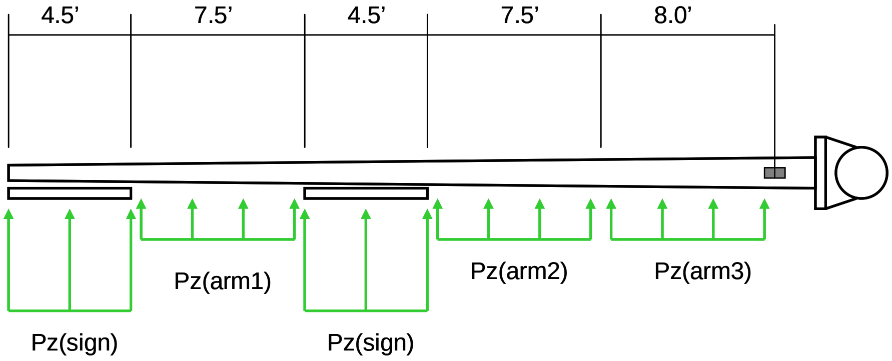


Figure 5.1 Mast-Arm Segment Modeling for Wind Pressure

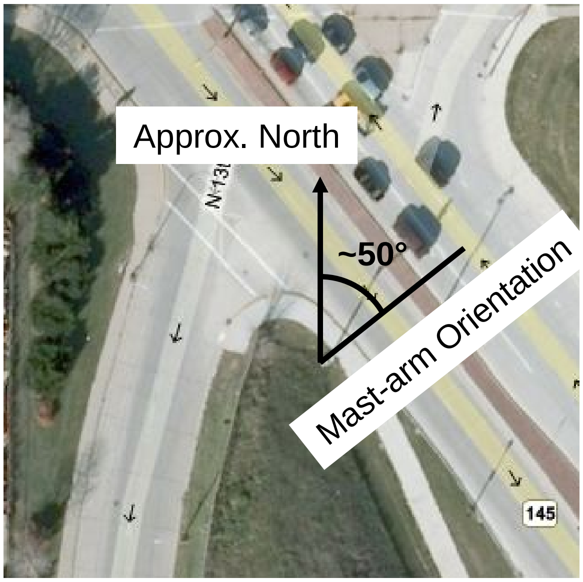


Figure 5.2 Approximate Mast-arm Orientation (http://www.bing.com/maps/)

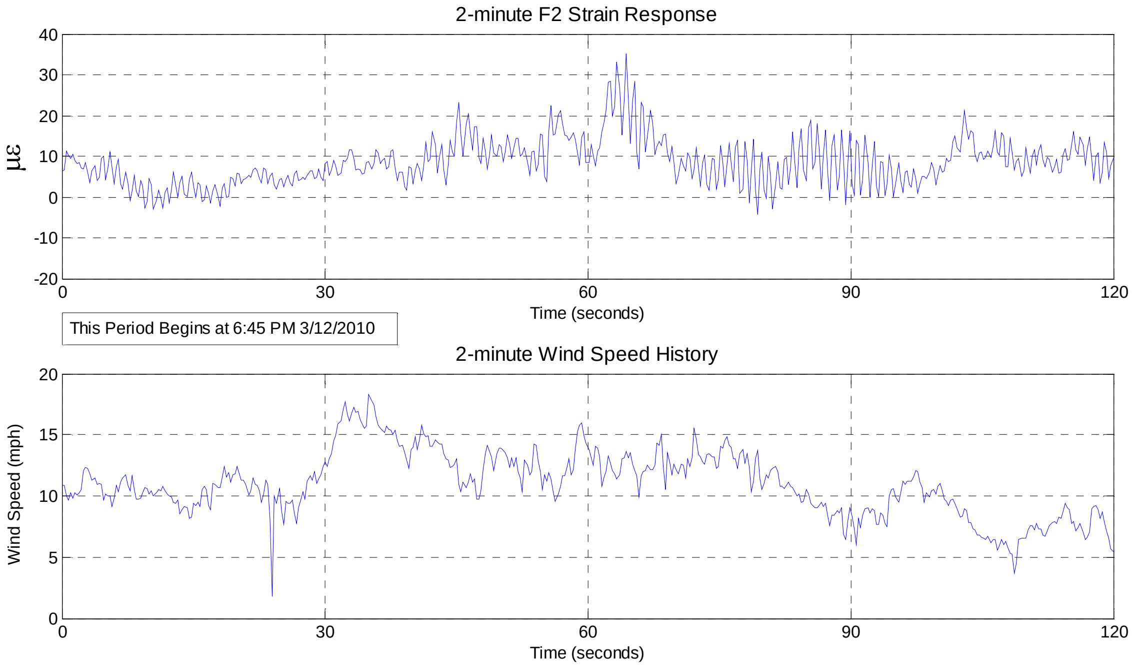


Figure 5.3 2-minute F2 and Wind Speed History Beginning At 6:45 PM 3/12/2010

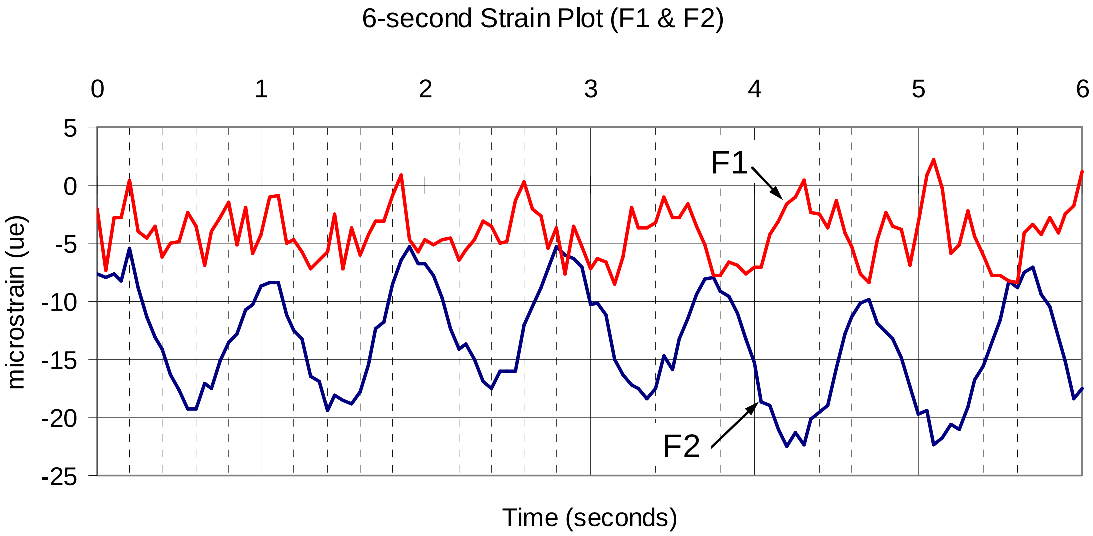


Figure 5.4 6 – Second Strain History Approx. 8 AM, March 10, 2010.

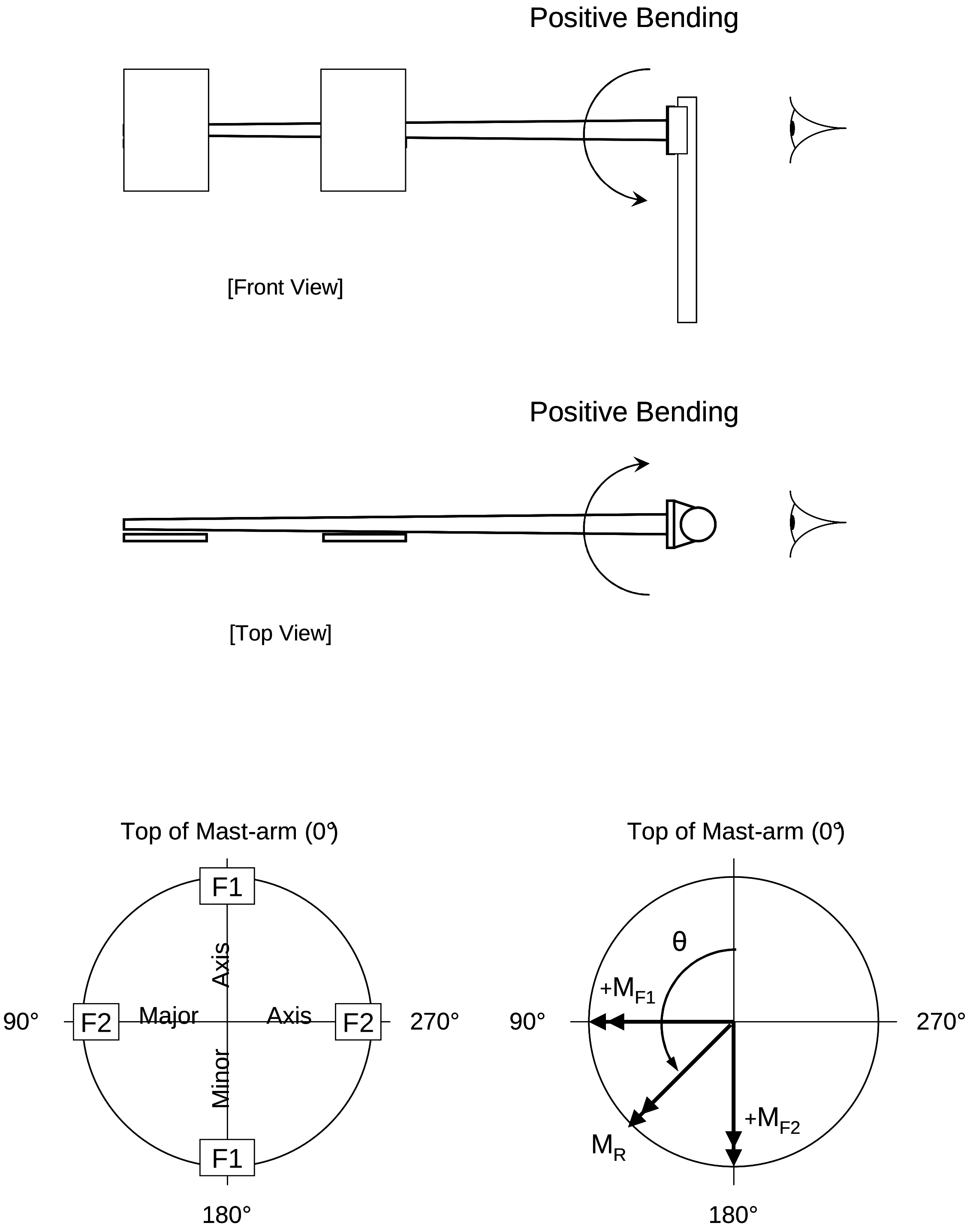
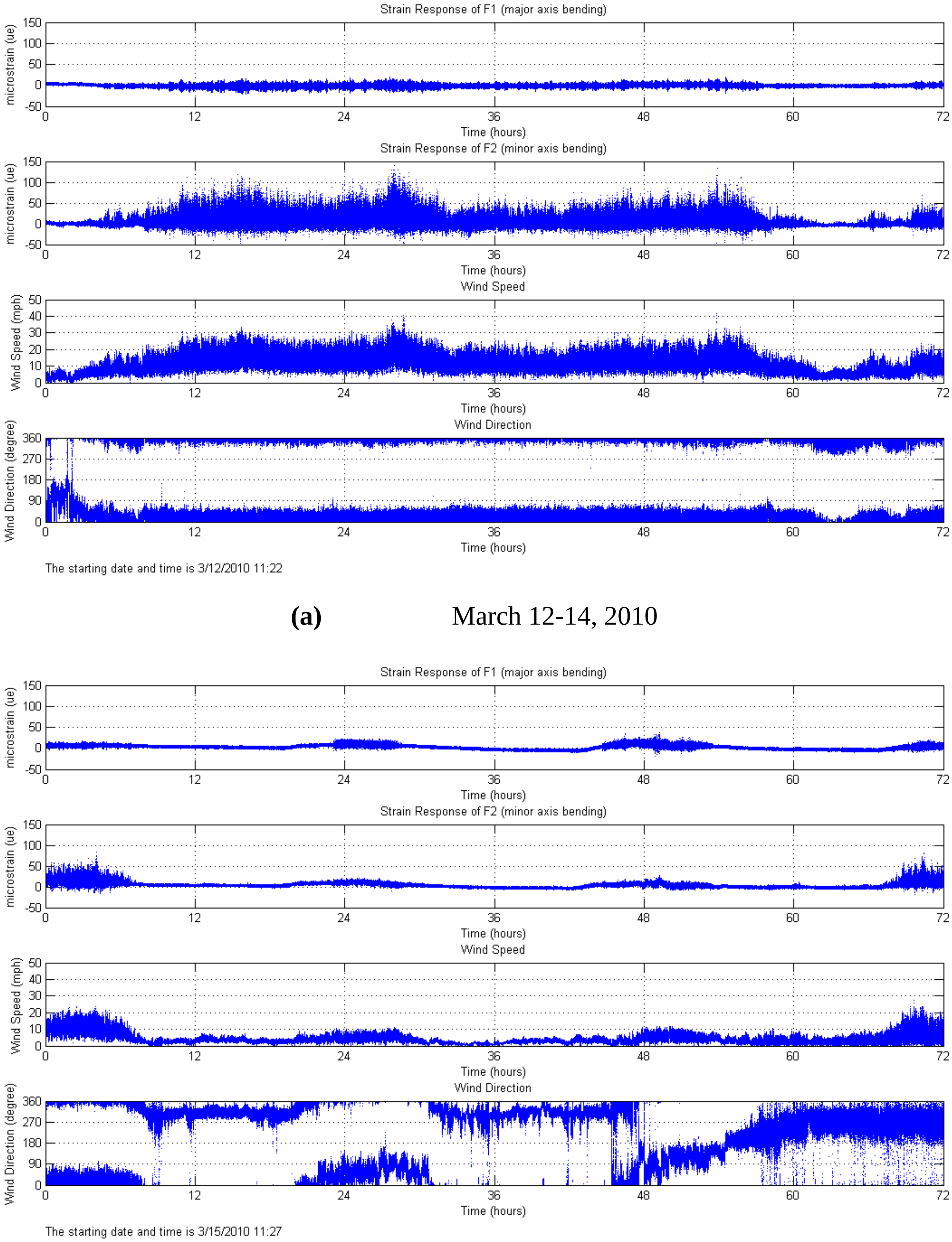


Figure 5.5 Reference Standard



(a) March 12-14, 2010

(b) March 15-17, 2010

Figure 5.6 Strain, Wind Speed and Wind Directions for March 12-17, 2010 Monitoring Period.

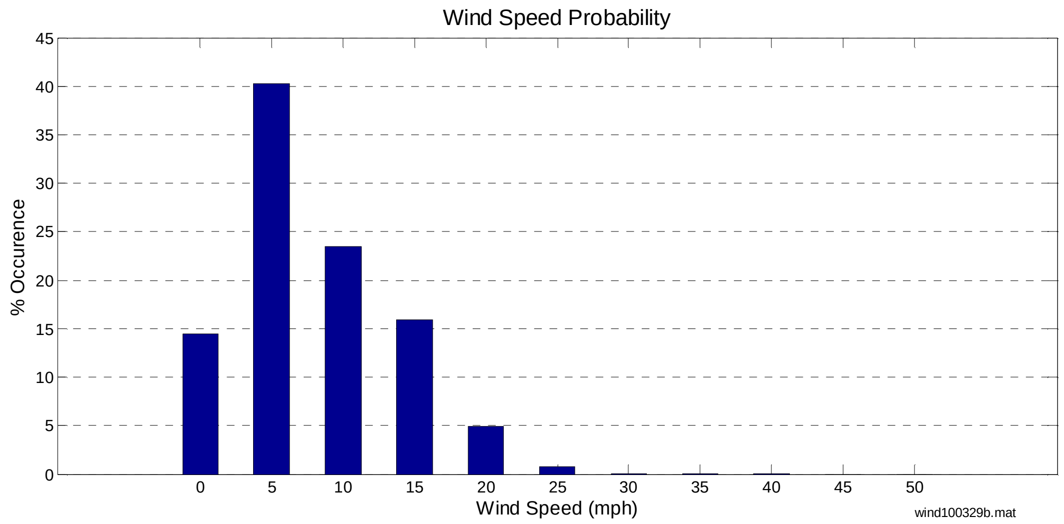


Figure 5.7 Wind Speed Histogram for Period March 12-17, 2010

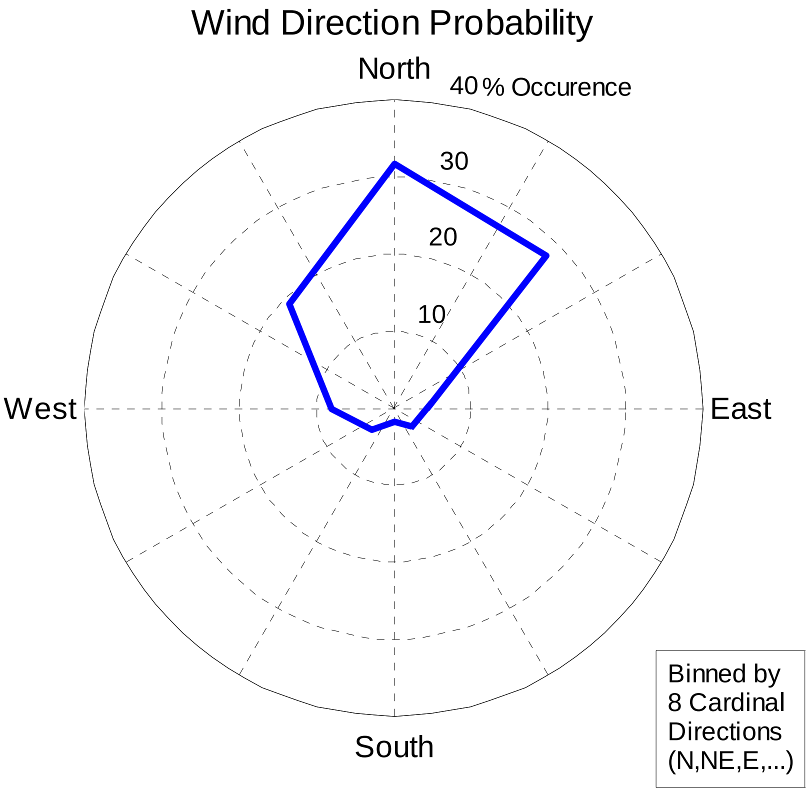


Figure 5.8 Wind Direction Histogram for Period March 12-17, 2010.

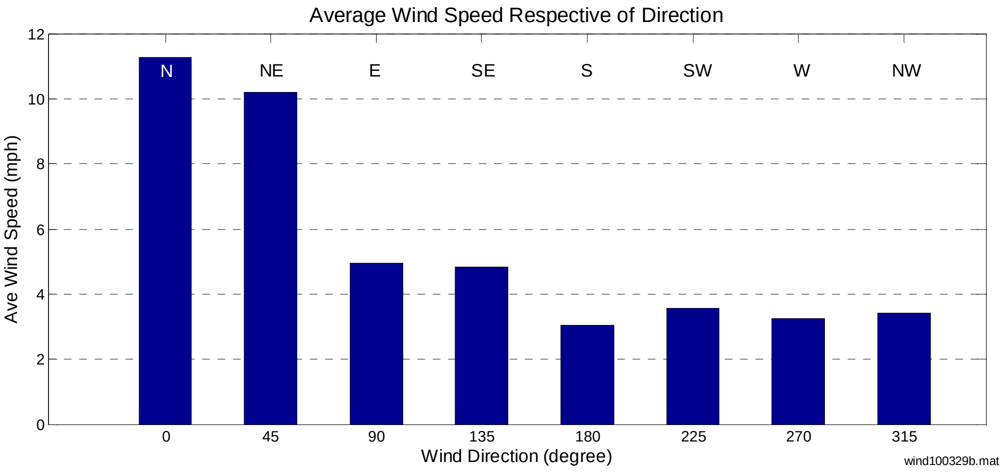


Figure 5.9 Average Wind Speed Respective of Direction

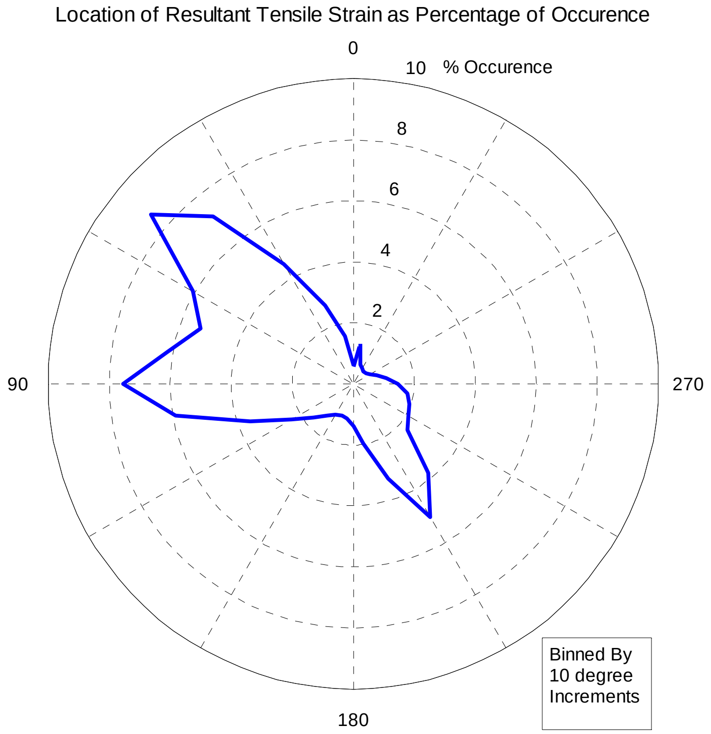


Figure 5.10 Resultant Strain History

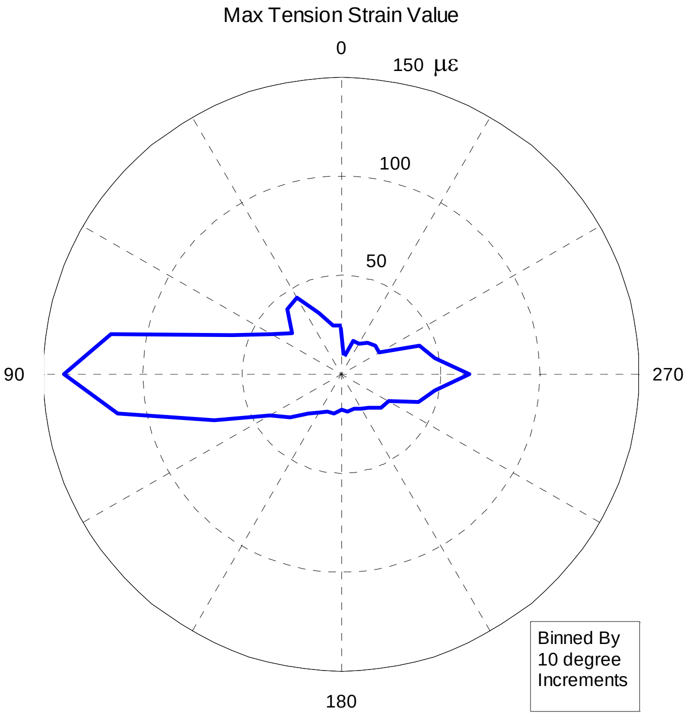


Figure 5.11 Maximum Tensile Strain

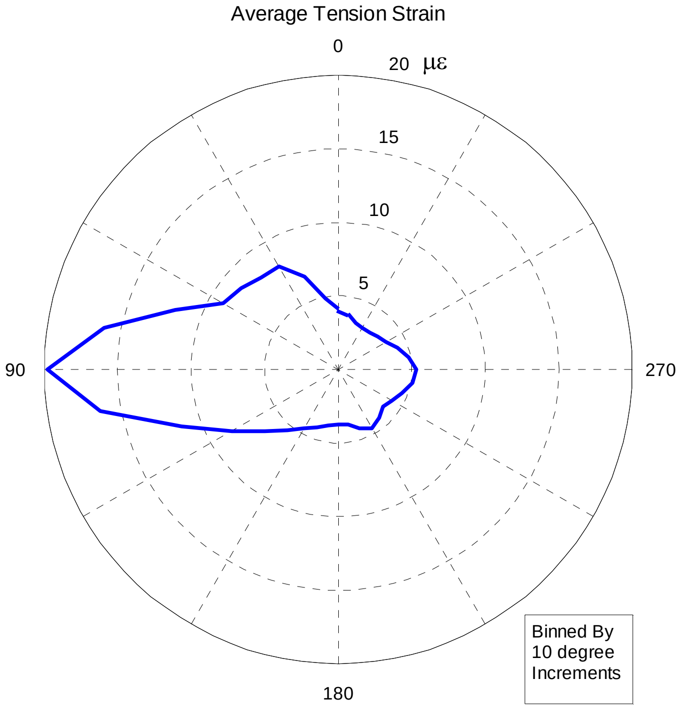


Figure 5.12 Average Tensile Strain

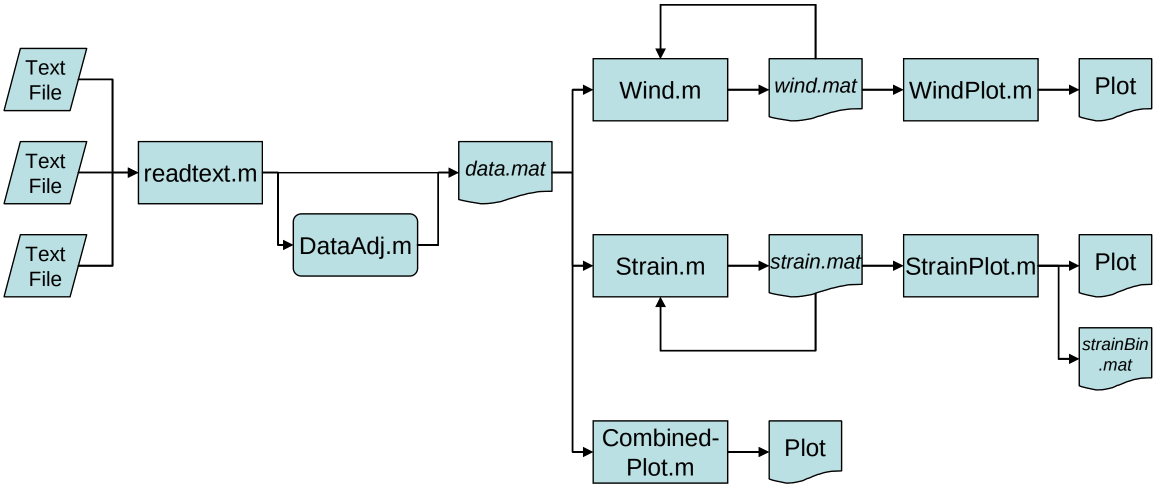


Figure 5.13 Data Process Flow

Chapter 6

Concluding Remarks and Recommendations for Future Work

6.1 Project Summary

Wisconsin sign support structure S-40-703 in Milwaukee, Wisconsin was identified late summer of 2009 as an ideal specimen for which to monitor the effects of wind-induced strain on a mast-arm-to-plate connection. The equipment necessary to carry out the structural health monitoring project was acquired during the months of August and September. The monitoring system infrastructure was assembled and put into place during the fall of 2009, which included instrumentation of the mast-arm connection with strain gages, running conduit and wiring, constructing the weather station tower foundation, installing the solar panel and equipment enclosures on the tower. Development of the LabVIEW data acquisition (DAQ) software occurred in early winter. The main hardware components of the DAQ system include a National Instruments CompactRIO DAQ hardware device, a Gill Windsonic anemometer, a National Instruments 9237 full-bridge conditioning module, and arrays of 350 ohm Vishay weldable strain gauges.

Once the DAQ system hardware components were assembled and software written, a small-scale cantilevered metal bar instrumented with a full bridge bending strain gages and the Marquette University’s college of engineering wind tunnel were used to validate the DAQ system hardware and software.

The health monitoring system was deployed on March 4, 2010 and began its automatic and continuous acquisition of strain and wind data. This data acquisition period can extend indefinitely. Algorithms to synthesize the field data were developed as part of this thesis. These algorithms are intended to be used by researchers to synthesize all data obtained by the health monitoring system as the research program continues.

The structural health monitoring system exists as a stand-alone, self-sufficient system. It will continue to actively monitor wind and strain readings indefinitely as long as it draws power from the battery that is charged by the solar panel. If power is temporarily lost, the DAQ system will restart as soon as power is restored. The only maintenance required is the periodic removal of data from the Compact RIO DAQ hard-drive. The CompactRIO internal memory is sufficient to allow a periods of up to two months in between data download.

6.2 Concluding Remarks and Recommendations

There have been a considerable number of studies in the area of fatigue-induced fracture of infrastructure components. The specific objectives of each study vary. It is hoped that the long-term data acquired from the hardware and software developed through the course of this study will be the foundation for future research efforts. The health monitoring system will facilitate obtaining more detailed information regarding the response characteristics of these types of structures and a better understanding of the loading histories these mast-arm structures experience during their service lives.

The algorithms developed through this thesis will assist the project members, both current and future, in synthesizing the raw data being collected. An algorithm was developed to bring in the raw data from the text files and parse the data. A second algorithm takes this data and applies an offset to compensate for any drift in the strain signal. It also adjusts the wind direction to compensate for the difference in the anemometer’s alignment with respect to true north. The remaining algorithms developed take the data and synthesize the raw data into useful statistical information and graphical displays of the data.

The six days of data (March 12-17, 2010) has already revealed very useful information with regard to the response of the mast-arm structure being monitored. It has shown that accepted methods of predicting strain and stress levels in mast-arm structures subjected to wind loading provide reasonably accurate predictions of horizontal-plane bending strains in the mast-arm for a given wind speed.

Detailed study of the response allowed the frequency corresponding to horizontal and vertical motions of the mast arm to be determined. The values seen for this frequency and mode of vibration in the structure monitored are consistent with analytical predictions made using modal analysis and those seen in previous research efforts.

Synthesis of the data for the six-day period included in this thesis discussion reveals that mast-arm structures are being subjected to deformations in both the vertical and horizontal planes. As a result, the mast arm structure is likely being subjected dynamic along-wind response resulting from natural wind buffeting as well as across-wind response resulting from aeroelastic phenomena such as vortex shedding. The data synthesis also indicates that tensile strains migrate around the circumference of the mast arm. The algorithms developed will allow future researchers to identify locations on the mast-arm circumference where fatigue cracks are most likely to form.

It is recommended that when data synthesis is carried out for many consecutive months (even years) advanced computing technology be employed. The typical desktop PC takes 18 hours just to read in 3 days of data and synthesize it into a single array. If this three-day period is extended to months, the computing time required will be proportionally longer.

It is also recommended that a rainflow (stress-range) counting algorithm be developed for synthesizing the data. Rainflow counting provides an effective way to count the magnitude and number of stress-ranges. The number of stress-ranges is key to developing fatigue life estimates. The mast-arm-to-pole connections in cantilevered sign support structures fall under detail category E' in the AASHTO specifications (AASHTO 2001). This detail category has a constant amplitude fatigue limit (CAFL) of 2.6-ksi (AASHTO 2001). The CAFL establishes the threshold between damaging and non-damaging stress ranges. A simple calculation using Hooke's Law shows that 100με corresponds to a normal stress of 2.9 ksi. A recorded value of 100 microstrain at a location does not provide enough information to determine the amplitude of the stress range experienced at that location. It does however put up a red flag, indicating that there are large stress cycles occurring at locations around the mast-arm circumference, and should be investigated. Rainflow counting of stress ranges at locations around the circumference of the mast-arm will provide researchers with indication of fatigue-sensitive locations.

It would be extremely enlightening to analyze stress-ranges, their magnitudes and frequency of occurrence at specific locations around the perimeter of the mast-arm. The data within each 10° bin provided by `StrainPlot.m` does not make a contiguous strain history in the bin. It records only the resultant tensile strain values that occur within boundaries of the bin. However, data synthesized in `Strain.m` provides the information that would be necessary to create contiguous stress-histories within specific 10-degree segments of mast-arm circumference. This would require additional coding to accomplish, but it is certainly feasible given the m-file architecture developed. The algorithms developed in this thesis can be used to identify trouble areas, or “hot-spots” along the mast-arm circumference. These areas will likely have large tension stress-ranges as well as large tension stress peaks. These hot spots are very useful for inspection guidance when looking for fatigue-induced cracking.

Identifying “hot-spots” might be done using the resultant tensile strain history (Figure 5.10) coupled with the average strain values (Figure 5.12) to identify an area that has frequent occurrences of large magnitude tension strain. Use of the output file from `StrainPlot.m` that contains all the individual bins of strain data will also facilitate this identification. An algorithm could easily be developed to sift through a single bin and return the number of occurrences a certain strain value is exceeded (i.e. the number of times magnitudes of 100 microstrain or more occur). If a bin is found to have an excessive amount of high strain values, this might be indicative of a “hot-spot”.

In conclusion, an efficient structural health monitoring system is in place and providing continuous long-term data for use by future researchers. Irrespective of a user's eventual use for the raw data being recorded by the monitoring system the foundational tools have been developed to aid in his or her objectives.

BIBLIOGRAPHY

Alderson, Joseph L. *Fatigue Study of Cantilevered Traffic Signal Mast Arms*. Thesis. University of Missouri – Columbia. 1999.

American Association of State Highway and Transportation Officials. *Standard Specifications For Structural Supports For Highway Signs, Luminaries, and Traffic Signals*. 2006 Interim Revisions. AASHTO. 2001.

Campbell Scientific: dataloggers, data acquisition systems, weather stations. Campbell Scientific Inc, 2010. Web. 15 Sep. 2009. *http://www.campbellsci.com.*

Chen, Genda, M. Barker, L. R. Dharani, and C. Ramsay. *Signal Mast Arm Fatigue Failure Investigation*. RDT 03-010. Missouri Department of Transportation. Research, Development and Technology. Jefferson City, Missouri. 2003.

Connor, Robert J., and Ian C. Hodgson. *Field Instrumentation and Testing of High-Mast Lighting Towers in the State of Iowa*. 2006.

Foley, *et al*. *Fatigue Risks in the Connections of Sign Support Structures – Phase 1.* WHRP 08-05. Wisconsin Highway Research Program. Madison, WI. 2008.

Gill Instruments – Wind Speed Measurement Anemometers, Engine Controls. Gill Instruments Ltd, 2010. Web. 23 Jan. 2010. *http://www.gill.co.uk*.

Hosch, Ian E., and Fouad H. Fouad. *Fatigue Design of Sign Support Structures for Loading Caused by Natural Wind Loads.* Transportation Research Record: Journal of the Transportation Board, No.2131 (2009): pp. 15-22.

Matlab. Vers. 7.6.0.324. MathWorks, Inc, 2008. Computer Software

Milwaukee, Wisconsin. [map]. Bing Maps, 2010. Web. 29 March 2010. *http://www.bing.com/maps.*

National Instruments - Test and Measurement. National Instruments, 2010. Web. 23 Jan. 2010. *http://www.ni.com*.

NWS ASOS Program. NOAA National Weather Service – Automated Surface Observing System. Web. 8 Apr. 2010. *http://www.weather.gov/asos.*

NOAA's Geophysical Data Center – Geomagnetic Data. National Geophysical Data Center, 2010. Web. 15 Feb 2010. *http://www.ngdc.noaa.gov/geomagmodels/Declination.jsp*

Nyquist Rate. Wikipedia, The Free Encyclopedia. 8 Jan 2010. Web. 7 April 2010. *http://en.wikipedia.org/wiki/Nyquist_rate*

Simiu, Emil, and Robert H. Scanlan. *Wind Effects on Structures: An Introduction to Wind Engineering*. New York: Wiley, 1986.

Tilt and Angle Orientation of Solar Panels. The Energy Grid. Web. 15 Sep. 2009. *http://www.theenergygrid.com/grid/articles/paneltilt.html*

Valmont Industries, Inc. *Overhead Sign Structures*. Marquette Exchange IH 43 Project No: 1060-05-71. Drawing Number: WI4C57111. Valley, NE. 2004.

Vishay – Manufacturers of Discrete Semiconductors and Passive Components. Vishay Intertechnology, Inc. Web. 15 Sep. 2009. *http://www.vishay.com*.

Ziemian, Ronald D., and William McGuire. *Mastan2*. Computer software. Vers. 3.2. 2008. Web.

APPENDICIES

Appendix A: cRIO Program Guide

A.1 How to Open and Run the Program

For ease in following the discussion, note the following definitions: 1) *LabVIEW* is the programming software used to create VIs, 2) a *VI* is an actual program developed for the data acquisition device using LabVIEW software, 3) a *Project* is a file in which all the VIs used for a specific task or purpose are compiled.

To begin one should open Labview. A welcome page will appear as shown below in Figure A.1.

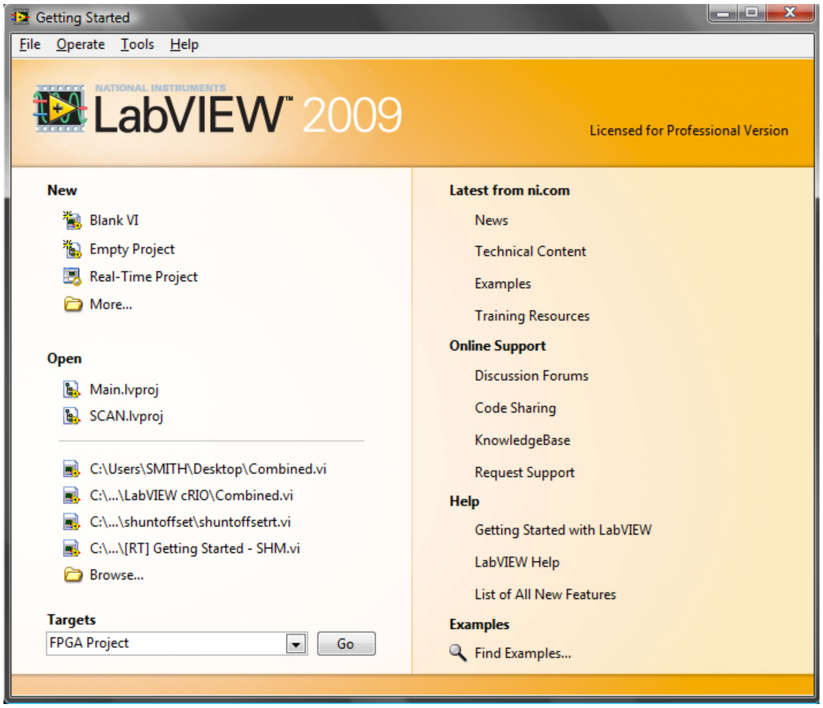
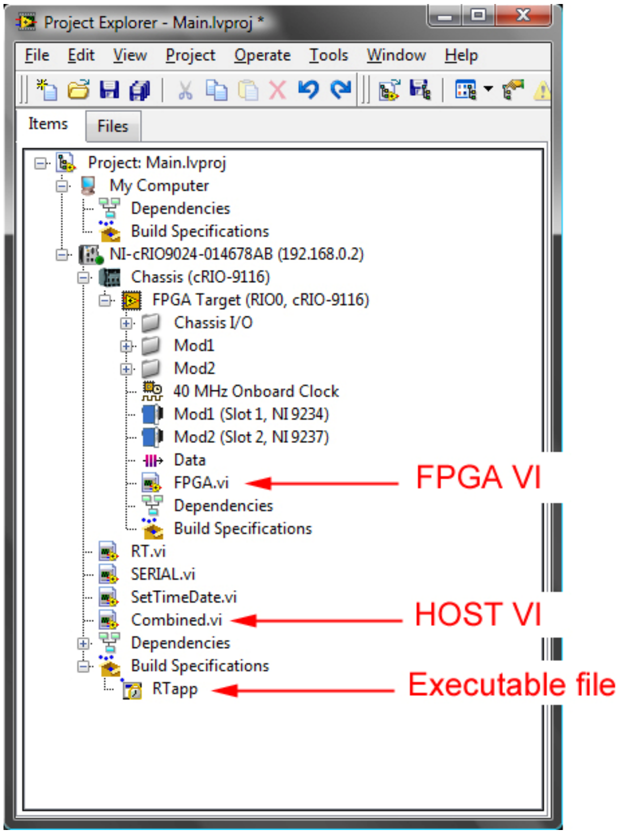


Figure A.1 LabVIEW Welcome Page

From here, the user may browse to find the proper file path under which the project exists. Project files have the suffix ".lvproj" and all the specific VIs created for the data acquisition device are found under *Main.lvproj*. By opening this project the following screen will appear (shown in Figure A.2)


Project Explorer - Main.lvproj *
File Edit View Project Operate Tools Window Help
Items
Files

- Project: Main.lvproj
  - My Computer
    - Dependencies
    - Build Specifications
  - NI-cRIO9024-014678AB (192.168.0.2)
    - Chassis (cRIO-9116)
      - FPGA Target (RIO0, cRIO-9116)
        - Chassis I/O
        - Mod1
        - Mod2
        - 40 MHz Onboard Clock
        - Mod1 (Slot 1, NI 9234)
        - Mod2 (Slot 2, NI 9237)
        - Data
        - FPGA.vi
        - Dependencies
        - Build Specifications
    - RT.vi
    - SERIAL.vi
    - SetTimeDate.vi
    - Combined.vi
    - Dependencies
    - Build Specifications
      - RTapp

FPGA VI

HOST VI

Executable file

Figure A.2 Project Window

From here all the programs under the project can be accessed simply by double clicking on them. As mentioned in chapter 4, the two VIs actually deployed on the cRIO are the FPGA.vi and Combined.vi (Host VI), and they are pointed out in the above figure. As will be discussed later, an executable can be created in this window, under *Build Specifications*. Also under the project are several other VIs which are not deployed onto the cRIO. They were used during development and were left in the project folder as they may be useful at some point. RT and Serial.vi read the strain gauges and anemometer, respectively and exclusively. In other words the RT VI contains no code to be able to read the anemometer; conversely the Serial VI has no code for obtaining strain readings.

These two VIs were later combined into a single VI that had both features, hence Combined.vi. These two VIs work properly and could be used if only one or the other measurements were of interest. The other VI in the project file is SetTimeDate.vi, which can be used to set, exactly as described, the time and date on the cRIO. This was used once and should not need to be run again, unless the onboard clock becomes asynchronous with real-time. There is a bug with this VI however, and if the user desires to use this VI, please refer to section A.3.

After opening up a specific VI, the front panel will appear. From here the user can specify any inputs allowed by the front panel (see chapter 4 discussion on front panels and block diagrams)

A.2 How to Deploy the Program

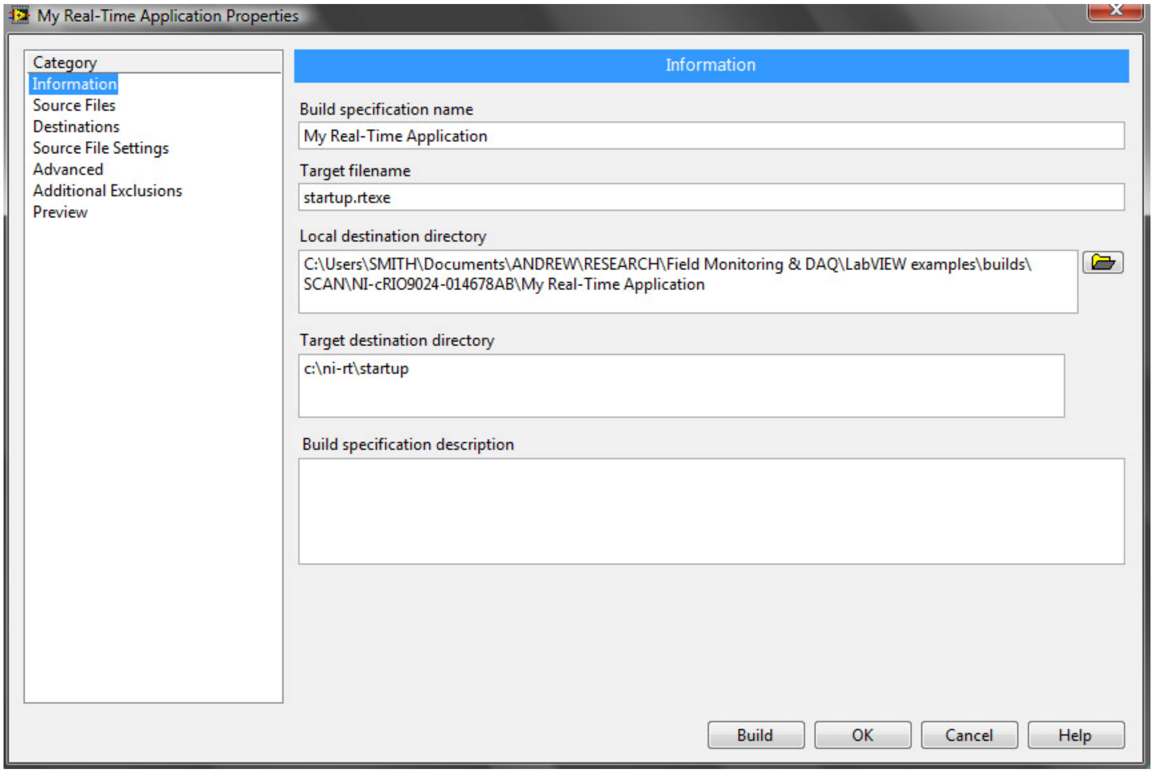
There are two ways in which someone can deploy a VI to run on a remote target such as the cRIO data acquisition device. The laptop or PC must be first connected via ethernet to link with the remote target (see section 3.5).

The first method is to simply open up a VI and click “run” (a little arrow icon in the menu bar of the front panel). The VI is now running on the cRIO. At this time the laptop can be disconnected from the cRIO. An error message will appear indicating that communication has been lost with remote target.

This downfall to this type of deployment is that the VI loaded onto the cRIO is stored in *volatile* memory. This means that if power is lost to the cRIO, even momentarily, the VI running on the cRIO will be wiped. Someone would need to come and reload the VI onto the cRIO. In a remote application this is a bad method of deployment that could result in long periods of data never being recorded.

The better method is to store the deployed VI on the *non-volatile* memory by creating an *executable,* which acts as a start-up feature and starts the VI automatically. An executable "RTapp.exe" has already been created for the currently deployed settings and is the last icon in Figure A.2. An executable is only as good as the current VI loaded onto the cRIO. If code modification is made to any deployed VI, and needs to be reloaded onto the cRIO, a new executable will need to be created also.

This can be done by right-clicking *Build Specifications* in the project window (Figure A.2). Select "New" and then "Real-Time Application". The following window will appear (Figure A.3).



Build specification name
My Real-Time Application

Target filename
startup.rtexe

Local destination directory
C:\Users\SMITH\Documents\ANDREW\RESEARCH\Field Monitoring & DAQ\LabVIEW examples\builds\SCAN\NI-cRIO9024-014678AB\My Real-Time Application

Target destination directory
c:\ni-rt\startup

Build specification description

Figure A.3 Executable Window

Along the left pane is category selection. The default screen is *Information.* This allows the user to specify the executable name and file directory for both local storage and storage on the cRIO. Next, select *Source Files,* on this pane, the user can select the startup VI. The startup VI should be, in this case, *Combined.vi.* This specifies which VI will start up in conjunction with the cRIO powering on. Next, select *Source File Settings.* This allows the user to specify any settings for the individual VIs in the project window. For instance, under the current setup, *Combined.vi* is set to automatically start, however *FPGA.vi* is also part of the data acquisition program structure. In this window *FPGA.vi* can be specified to run automatically when called by *Combined.vi.* Basically this allows the user to tag on any additional VIs that run in conjunction with the main startup VI. Also in this pane, the user can specify to keep the front panel and block diagram values, such as default inputs. This means that anytime the cRIO reboots, it takes the default input values and applies them to the program.

After all settings have been made, click “Build”. This now creates the executable and saves it to the local directory (your PC), and the executable icon shows up in the project window. The executable must be deployed (or saved) to the cRIO’s non-volatile memory. This can be done by right-clicking on the executable and doing one of two things, either select *Run As Startup* or *Deploy. Run As Startup* will deploy the executable and reboot the cRIO for automatic start-up. Otherwise, selecting *Deploy* will save the executable to the cRIO but will not reboot. Rebooting can be done later then, by opening the program *Measurement & Automation Explorer (MAX).* This software is part of the LabVIEW package and is used change cRIO settings, such as date, time, IP address, etc… When *MAX* is opened, expand *remote systems* under the configuration menu. Select the cRIO and the settings menu will appear on the right. Along the top right menu is the *Reboot* select.

A.3 Known Program Issues

If an issue related to the National Instruments hardware or software (including the cRIO and LabVIEW) arises, the best source of information is to go directly to the National Instruments website at *http://www.ni.com/*. Here you will find a vast array of tools and support.

Strain data channels flipped at an instant in time and stuck there.

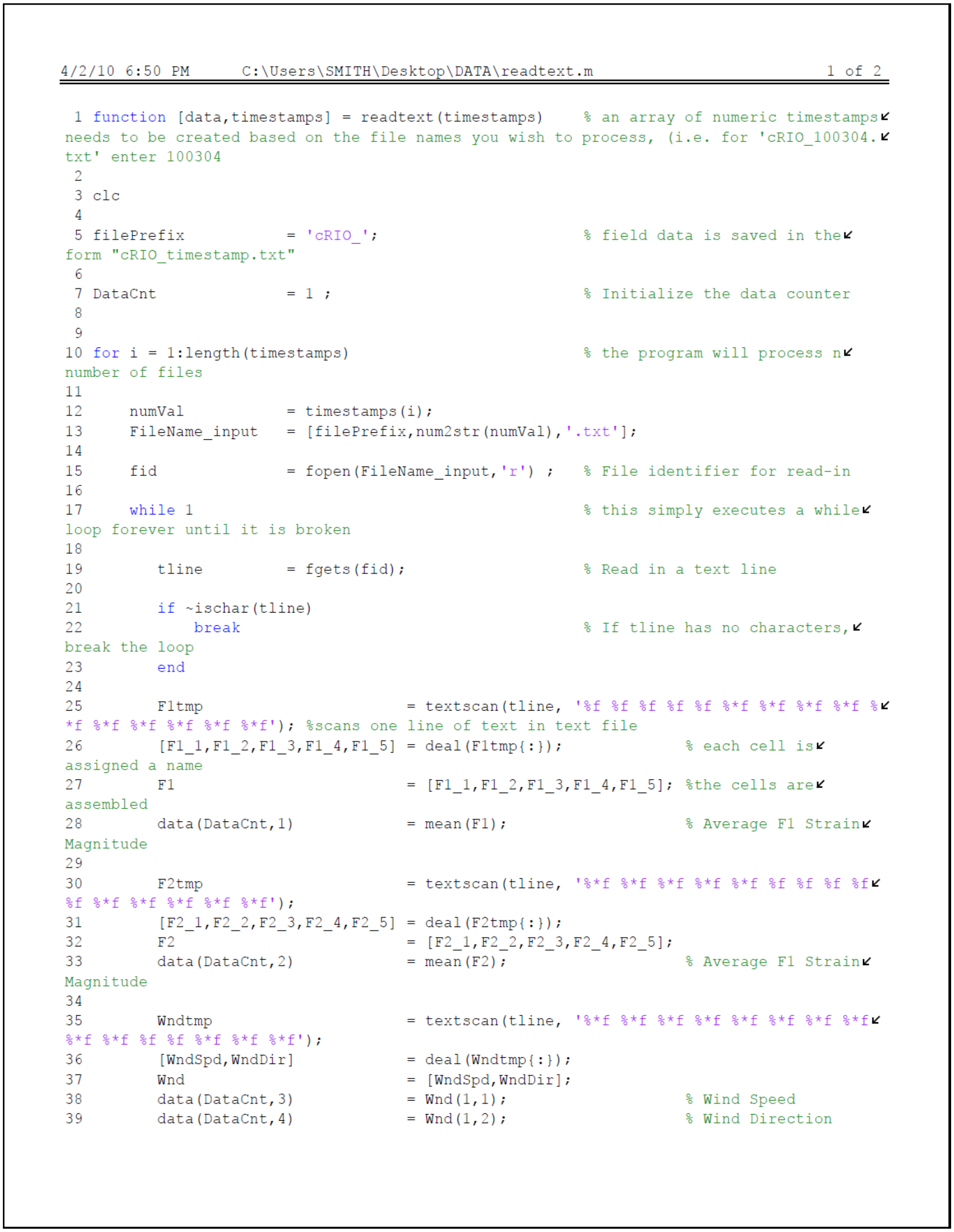
For whatever reason, the first two weeks of deployment saw an occasional flip in strain signals. Since then modifications have been made to the VIs currently running on the cRIO, including some timing features to control the loop iterations. After several weeks of uninterrupted data flow, this has seemed to resolve the issue. Should it occur, however, simply reboot the system, and the strain channels will correct themselves.

After setting the date/time using *SetTimeDate.vi* the cRIO internal clock progresses twice as fast as real time.

For whatever reason, this VI has a glitch that accelerates the internal clock on the cRIO. The date and time have already been set on the cRIO and shouldn’t need resetting. If however use of that VI is needed, reboot the cRIO immediately following the date/time set. Rebooting will keep the newly adjusted date and time, but will also return the internal clock to normal operation.

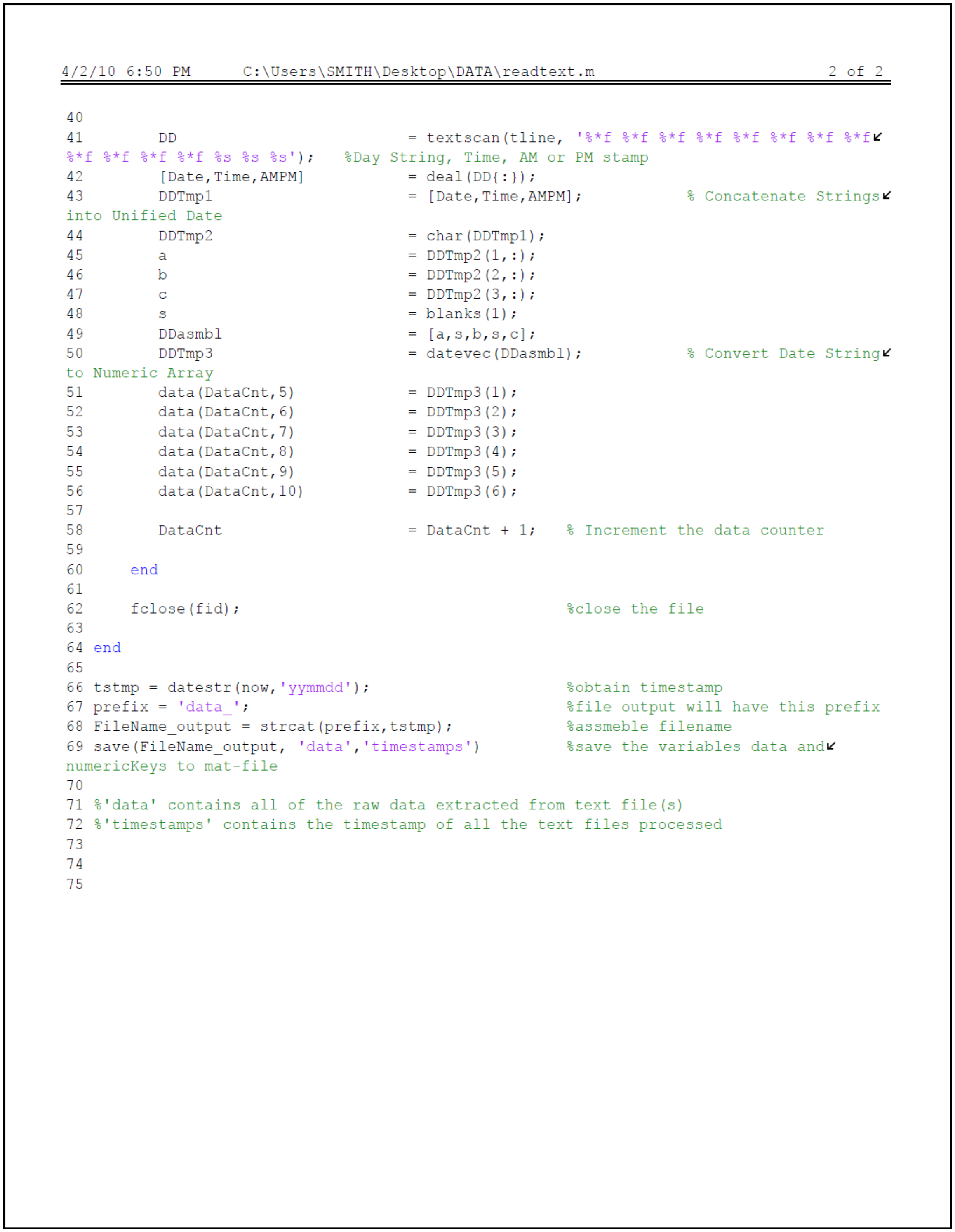
Appendix B: Data Synthesis Algorithms (Matlab m-files)

B.1 readtext.m



4/2/10 6:50 PM C:\Users\SMITH\Desktop\DATA\readtext.m 1 of 2

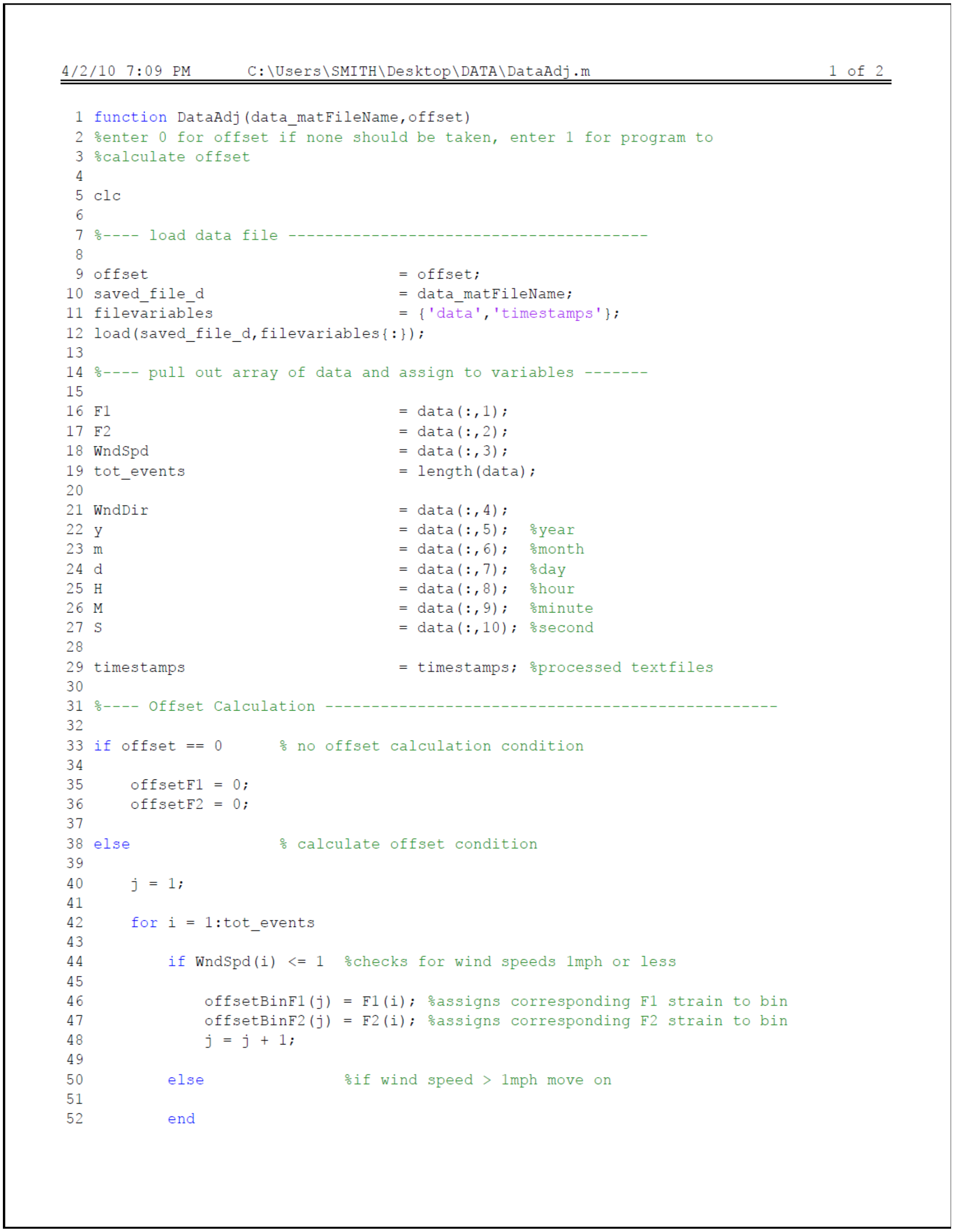
```
function [data,timestamps] = readtext(timestamps)    % an array of numeric timestamps↙
needs to be created based on the file names you wish to process, (i.e. for 'cRIO_100304.↙
txt' enter 100304

clc

filePrefix           = 'cRIO_';                      % field data is saved in the↙
form "cRIO_timestamp.txt"

DataCnt              = 1 ;                           % Initialize the data counter


for i = 1:length(timestamps)                         % the program will process n↙
number of files

    numVal           = timestamps(i);
    FileName_input   = [filePrefix,num2str(numVal),'.txt'];

    fid              = fopen(FileName_input,'r') ;   % File identifier for read-in

    while 1                                          % this simply executes a while↙
loop forever until it is broken

       tline         = fgets(fid);                   % Read in a text line

       if ~ischar(tline)
           break                                     % If tline has no characters,↙
break the loop
       end

       F1tmp                      = textscan(tline, '%f %f %f %f %f %*f %*f %*f %*f %↙
*f %*f %*f %*f %*f %*f'); %scans one line of text in text file
       [F1_1,F1_2,F1_3,F1_4,F1_5] = deal(F1tmp{:});             % each cell is↙
assigned a name
       F1                         = [F1_1,F1_2,F1_3,F1_4,F1_5]; %the cells are↙
assembled
       data(DataCnt,1)            = mean(F1);                   % Average F1 Strain↙
Magnitude

       F2tmp                      = textscan(tline, '%*f %*f %*f %*f %*f %f %f %f %f↙
%f %*f %*f %*f %*f %*f');
       [F2_1,F2_2,F2_3,F2_4,F2_5] = deal(F2tmp{:});
       F2                         = [F2_1,F2_2,F2_3,F2_4,F2_5];
       data(DataCnt,2)            = mean(F2);                   % Average F1 Strain↙
Magnitude

       Wndtmp                     = textscan(tline, '%*f %*f %*f %*f %*f %*f %*f %*f↙
%*f %*f %f %f %*f %*f %*f');
       [WndSpd,WndDir]            = deal(Wndtmp{:});
       Wnd                        = [WndSpd,WndDir];
       data(DataCnt,3)            = Wnd(1,1);                   % Wind Speed
       data(DataCnt,4)            = Wnd(1,2);                   % Wind Direction
```



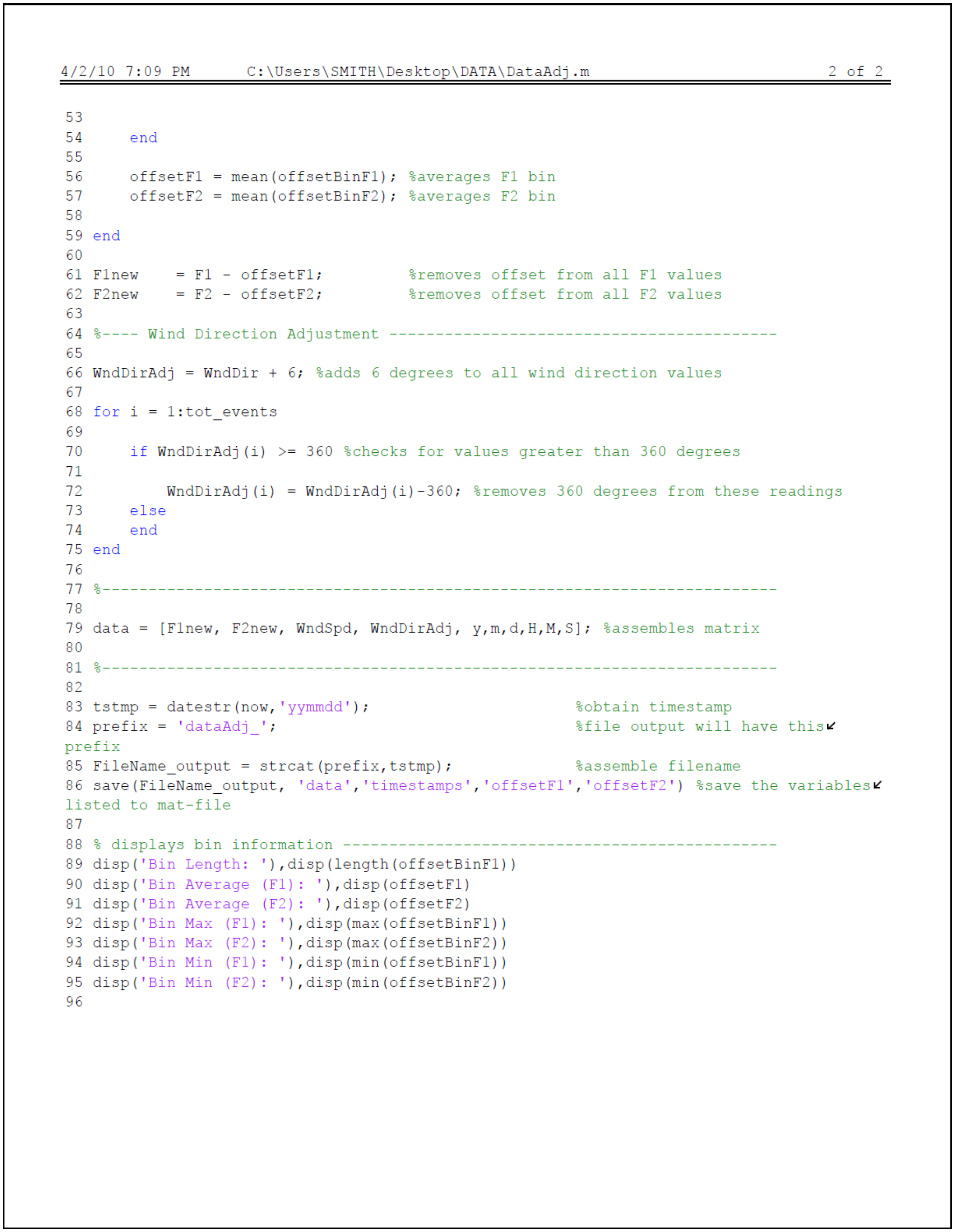
4/2/10 6:50 PM   C:\Users\SMITH\Desktop\DATA\readtext.m   2 of 2

```

        DD                         = textscan(tline, '%*f %*f %*f %*f %*f %*f %*f %*f↙
%*f %*f %*f %*f %s %s %s');   %Day String, Time, AM or PM stamp
        [Date,Time,AMPM]           = deal(DD{:});
        DDTmp1                     = [Date,Time,AMPM];           % Concatenate Strings↙
into Unified Date
        DDTmp2                     = char(DDTmp1);
        a                          = DDTmp2(1,:);
        b                          = DDTmp2(2,:);
        c                          = DDTmp2(3,:);
        s                          = blanks(1);
        DDasmbl                    = [a,s,b,s,c];
        DDTmp3                     = datevec(DDasmbl);           % Convert Date String↙
to Numeric Array
        data(DataCnt,5)            = DDTmp3(1);
        data(DataCnt,6)            = DDTmp3(2);
        data(DataCnt,7)            = DDTmp3(3);
        data(DataCnt,8)            = DDTmp3(4);
        data(DataCnt,9)            = DDTmp3(5);
        data(DataCnt,10)           = DDTmp3(6);

        DataCnt                    = DataCnt + 1;   % Increment the data counter

    end

    fclose(fid);                                    %close the file

end

tstmp = datestr(now,'yymmdd');                      %obtain timestamp
prefix = 'data_';                                   %file output will have this prefix
FileName_output = strcat(prefix,tstmp);             %assmeble filename
save(FileName_output, 'data','timestamps')          %save the variables data and↙
numericKeys to mat-file

%'data' contains all of the raw data extracted from text file(s)
%'timestamps' contains the timestamp of all the text files processed



```

B.2 DataAdj.m



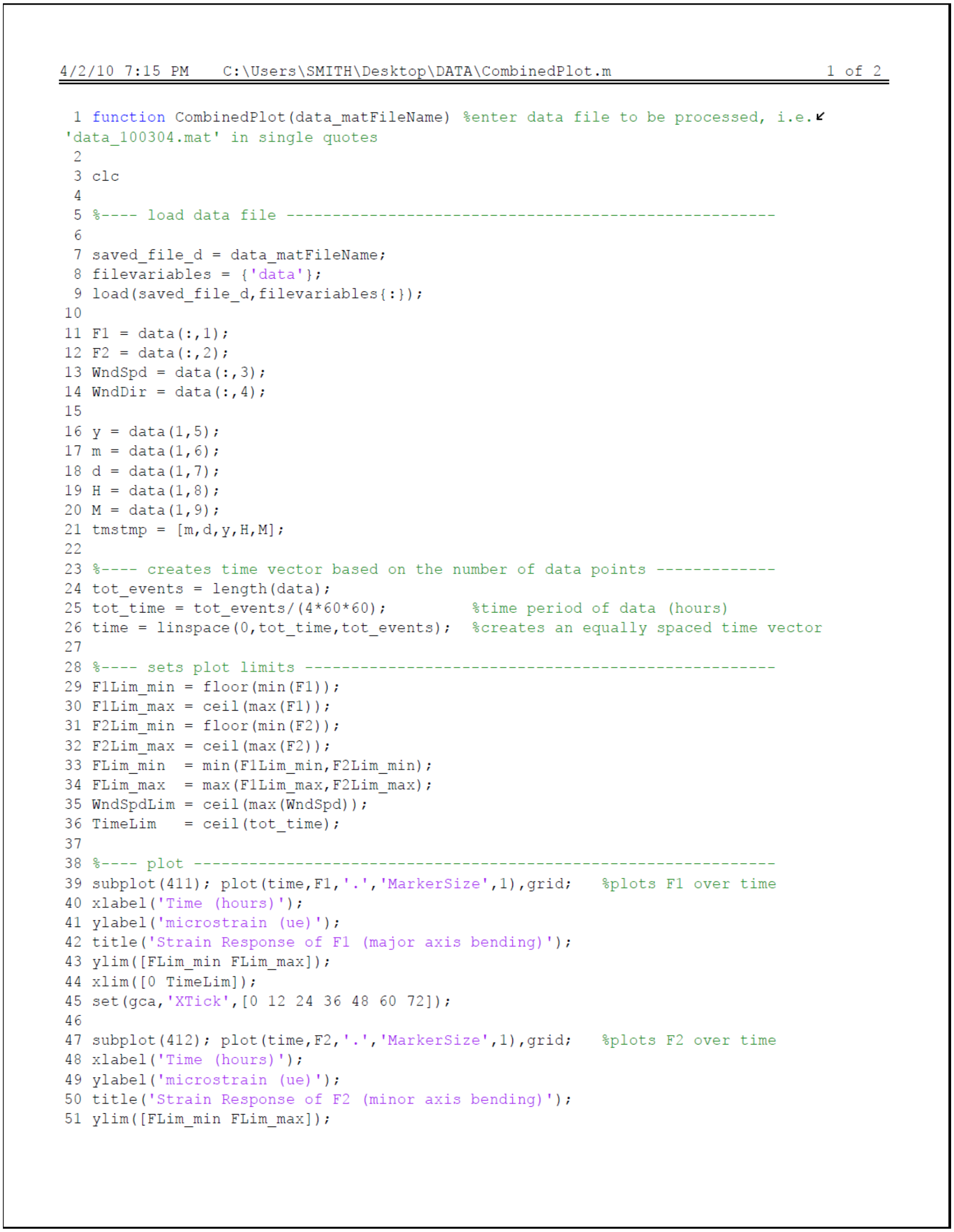
```
4/2/10 7:09 PM      C:\Users\SMITH\Desktop\DATA\DataAdj.m                   1 of 2

function DataAdj(data_matFileName,offset)
%enter 0 for offset if none should be taken, enter 1 for program to
%calculate offset

clc

%---- load data file ---------------------------------------

offset                           = offset;
saved_file_d                     = data_matFileName;
filevariables                    = {'data','timestamps'};
load(saved_file_d,filevariables{:});

%---- pull out array of data and assign to variables -------

F1                               = data(:,1);
F2                               = data(:,2);
WndSpd                           = data(:,3);
tot_events                       = length(data);

WndDir                           = data(:,4);
y                                = data(:,5);  %year
m                                = data(:,6);  %month
d                                = data(:,7);  %day
H                                = data(:,8);  %hour
M                                = data(:,9);  %minute
S                                = data(:,10); %second

timestamps                       = timestamps; %processed textfiles

%---- Offset Calculation -------------------------------------------------

if offset == 0      % no offset calculation condition

    offsetF1 = 0;
    offsetF2 = 0;

else                % calculate offset condition

    j = 1;

    for i = 1:tot_events

        if WndSpd(i) <= 1  %checks for wind speeds 1mph or less

            offsetBinF1(j) = F1(i); %assigns corresponding F1 strain to bin
            offsetBinF2(j) = F2(i); %assigns corresponding F2 strain to bin
            j = j + 1;

        else               %if wind speed > 1mph move on

        end
```



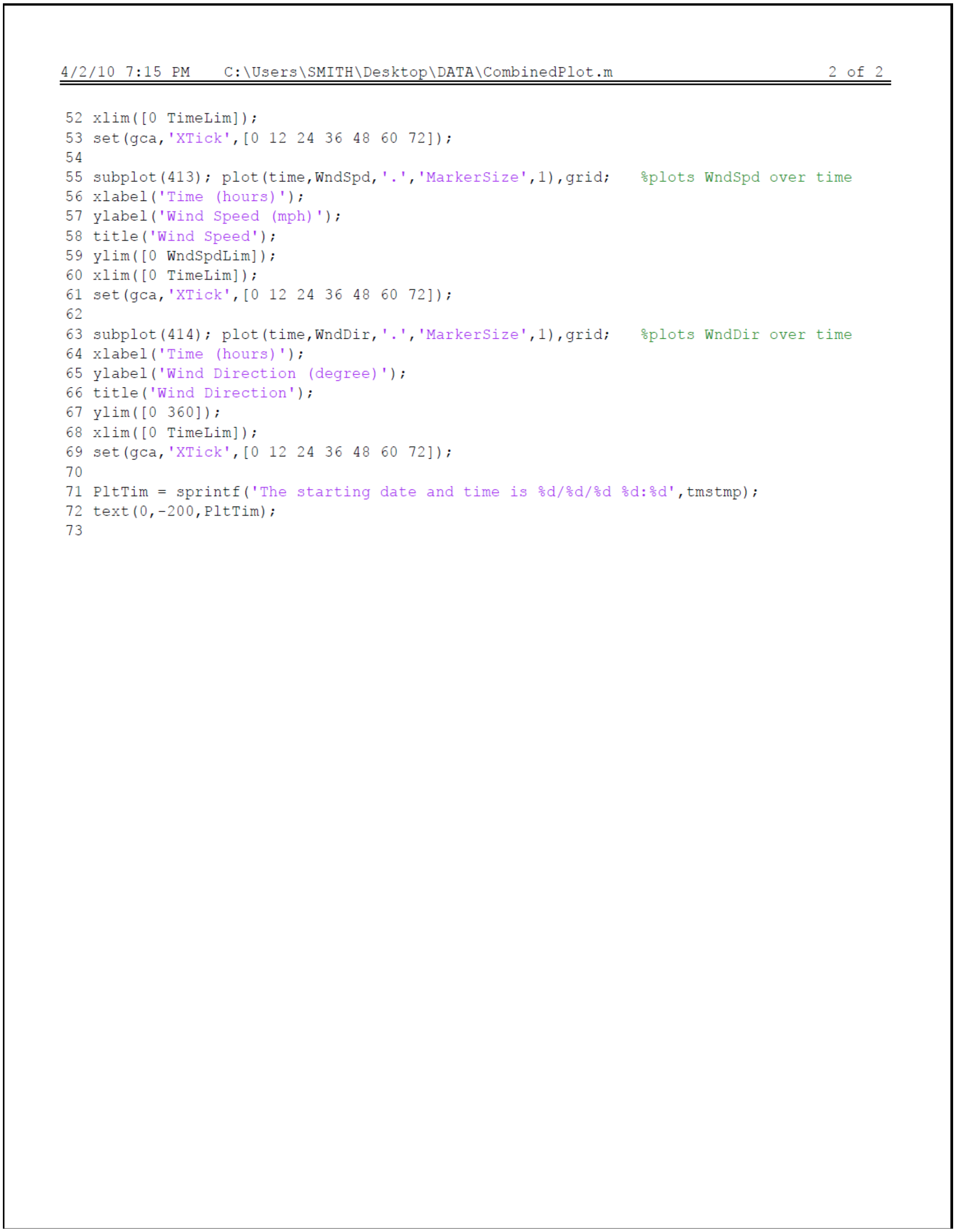
4/2/10 7:09 PM   C:\Users\SMITH\Desktop\DATA\DataAdj.m   2 of 2

```
    end

    offsetF1 = mean(offsetBinF1); %averages F1 bin
    offsetF2 = mean(offsetBinF2); %averages F2 bin

end

F1new    = F1 - offsetF1;         %removes offset from all F1 values
F2new    = F2 - offsetF2;         %removes offset from all F2 values

%---- Wind Direction Adjustment ------------------------------------------

WndDirAdj = WndDir + 6; %adds 6 degrees to all wind direction values

for i = 1:tot_events

    if WndDirAdj(i) >= 360 %checks for values greater than 360 degrees

        WndDirAdj(i) = WndDirAdj(i)-360; %removes 360 degrees from these readings
    else
    end
end

%-------------------------------------------------------------------------

data = [F1new, F2new, WndSpd, WndDirAdj, y,m,d,H,M,S]; %assembles matrix

%-------------------------------------------------------------------------

tstmp = datestr(now,'yymmdd');                      %obtain timestamp
prefix = 'dataAdj_';                                %file output will have this↙
prefix
FileName_output = strcat(prefix,tstmp);             %assemble filename
save(FileName_output, 'data','timestamps','offsetF1','offsetF2') %save the variables↙
listed to mat-file

% displays bin information -----------------------------------------------
disp('Bin Length: '),disp(length(offsetBinF1))
disp('Bin Average (F1): '),disp(offsetF1)
disp('Bin Average (F2): '),disp(offsetF2)
disp('Bin Max (F1): '),disp(max(offsetBinF1))
disp('Bin Max (F2): '),disp(max(offsetBinF2))
disp('Bin Min (F1): '),disp(min(offsetBinF1))
disp('Bin Min (F2): '),disp(min(offsetBinF2))

```

B.3 CombinedPlot.m



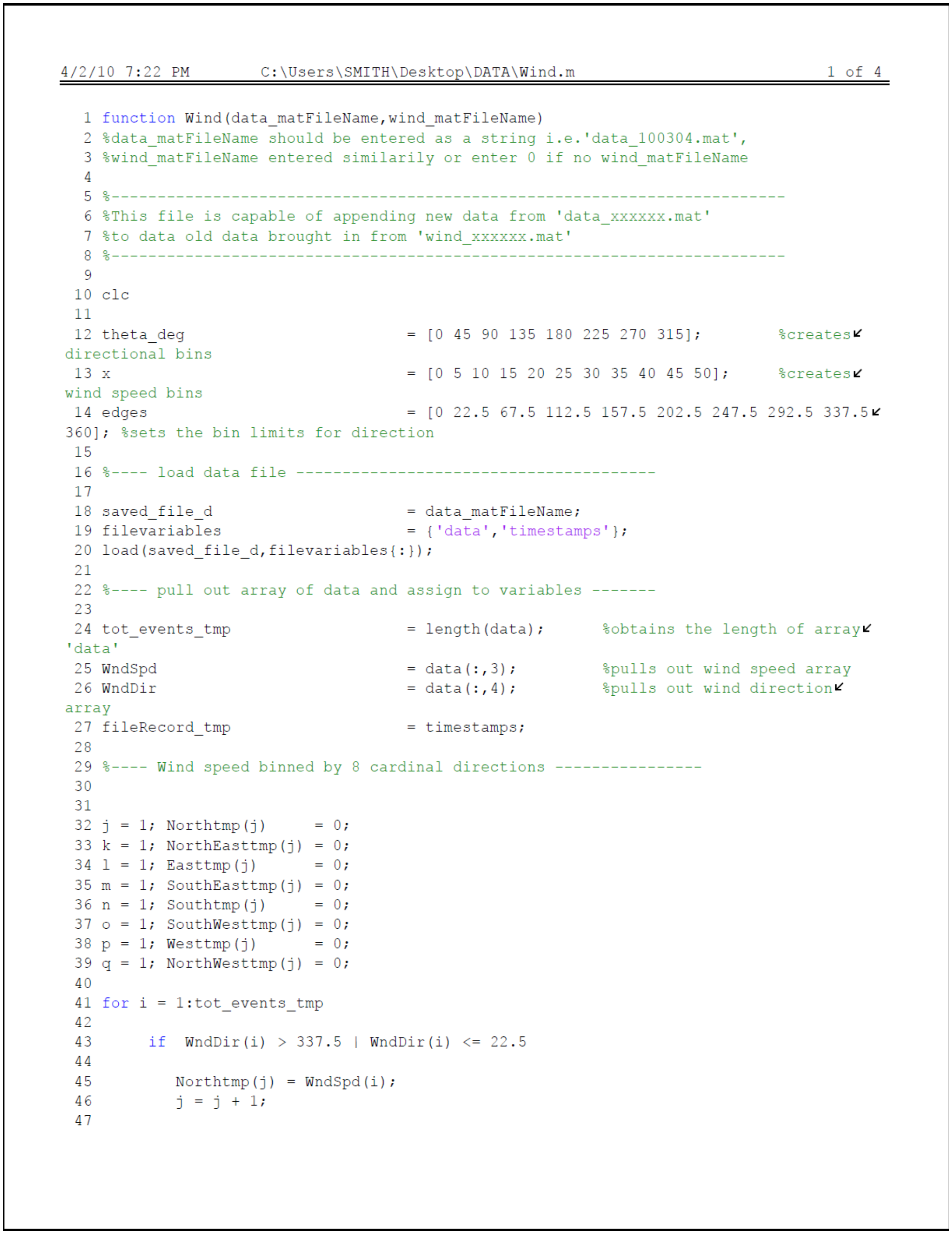
```
4/2/10 7:15 PM    C:\Users\SMITH\Desktop\DATA\CombinedPlot.m                     1 of 2

function CombinedPlot(data_matFileName) %enter data file to be processed, i.e.↙
'data_100304.mat' in single quotes

clc

%---- load data file -----------------------------------------------------

saved_file_d = data_matFileName;
filevariables = {'data'};
load(saved_file_d,filevariables{:});

F1 = data(:,1);
F2 = data(:,2);
WndSpd = data(:,3);
WndDir = data(:,4);

y = data(1,5);
m = data(1,6);
d = data(1,7);
H = data(1,8);
M = data(1,9);
tmstmp = [m,d,y,H,M];

%---- creates time vector based on the number of data points -------------
tot_events = length(data);
tot_time = tot_events/(4*60*60);         %time period of data (hours)
time = linspace(0,tot_time,tot_events);  %creates an equally spaced time vector

%---- sets plot limits ---------------------------------------------------
F1Lim_min = floor(min(F1));
F1Lim_max = ceil(max(F1));
F2Lim_min = floor(min(F2));
F2Lim_max = ceil(max(F2));
FLim_min  = min(F1Lim_min,F2Lim_min);
FLim_max  = max(F1Lim_max,F2Lim_max);
WndSpdLim = ceil(max(WndSpd));
TimeLim   = ceil(tot_time);

%---- plot ---------------------------------------------------------------
subplot(411); plot(time,F1,'.','MarkerSize',1),grid;    %plots F1 over time
xlabel('Time (hours)');
ylabel('microstrain (ue)');
title('Strain Response of F1 (major axis bending)');
ylim([FLim_min FLim_max]);
xlim([0 TimeLim]);
set(gca,'XTick',[0 12 24 36 48 60 72]);

subplot(412); plot(time,F2,'.','MarkerSize',1),grid;    %plots F2 over time
xlabel('Time (hours)');
ylabel('microstrain (ue)');
title('Strain Response of F2 (minor axis bending)');
ylim([FLim_min FLim_max]);
```



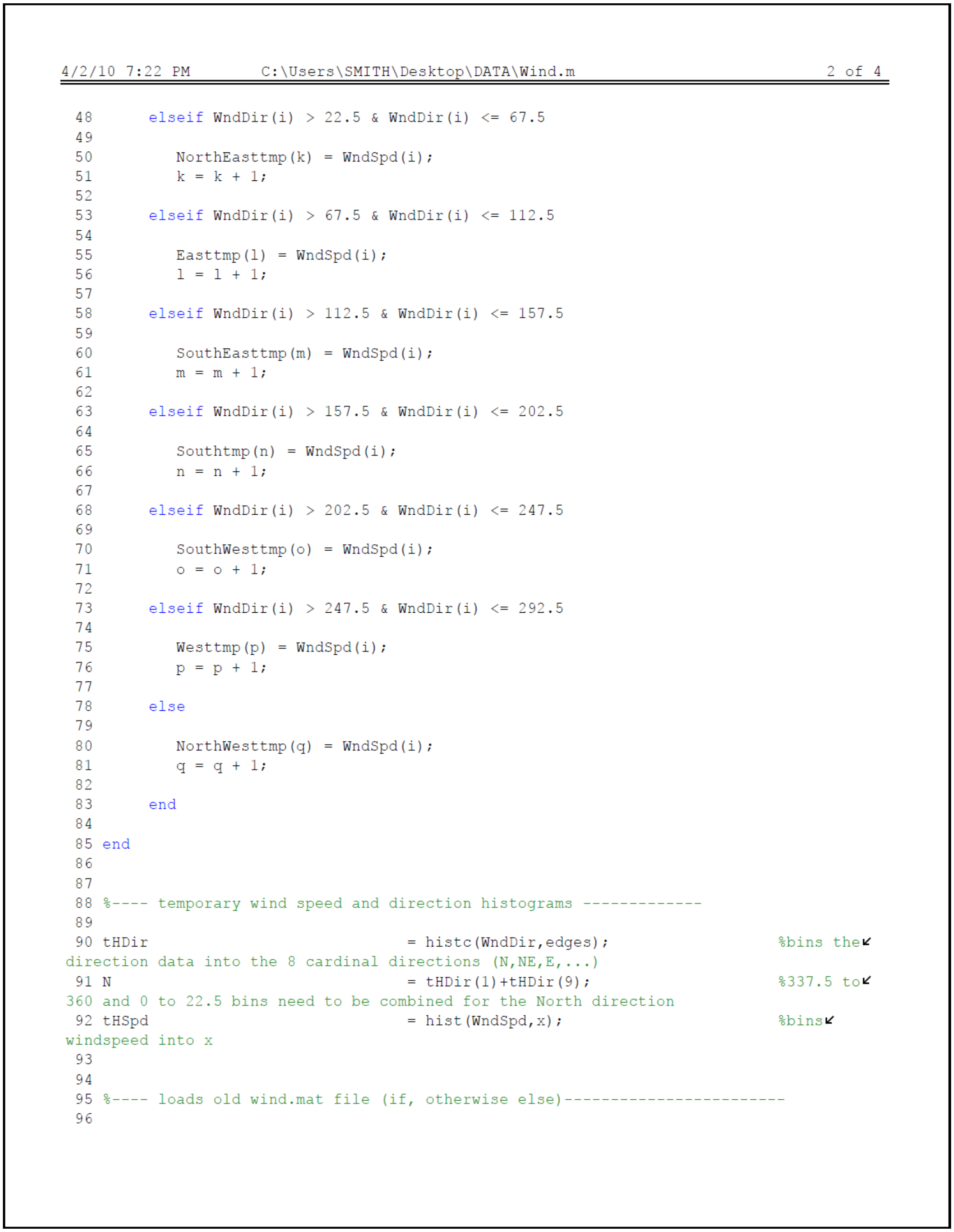
4/2/10 7:15 PM   C:\Users\SMITH\Desktop\DATA\CombinedPlot.m   2 of 2

```
xlim([0 TimeLim]);
set(gca,'XTick',[0 12 24 36 48 60 72]);

subplot(413); plot(time,WndSpd,'.','MarkerSize',1),grid;   %plots WndSpd over time
xlabel('Time (hours)');
ylabel('Wind Speed (mph)');
title('Wind Speed');
ylim([0 WndSpdLim]);
xlim([0 TimeLim]);
set(gca,'XTick',[0 12 24 36 48 60 72]);

subplot(414); plot(time,WndDir,'.','MarkerSize',1),grid;   %plots WndDir over time
xlabel('Time (hours)');
ylabel('Wind Direction (degree)');
title('Wind Direction');
ylim([0 360]);
xlim([0 TimeLim]);
set(gca,'XTick',[0 12 24 36 48 60 72]);

PltTim = sprintf('The starting date and time is %d/%d/%d %d:%d',tmstmp);
text(0,-200,PltTim);

```

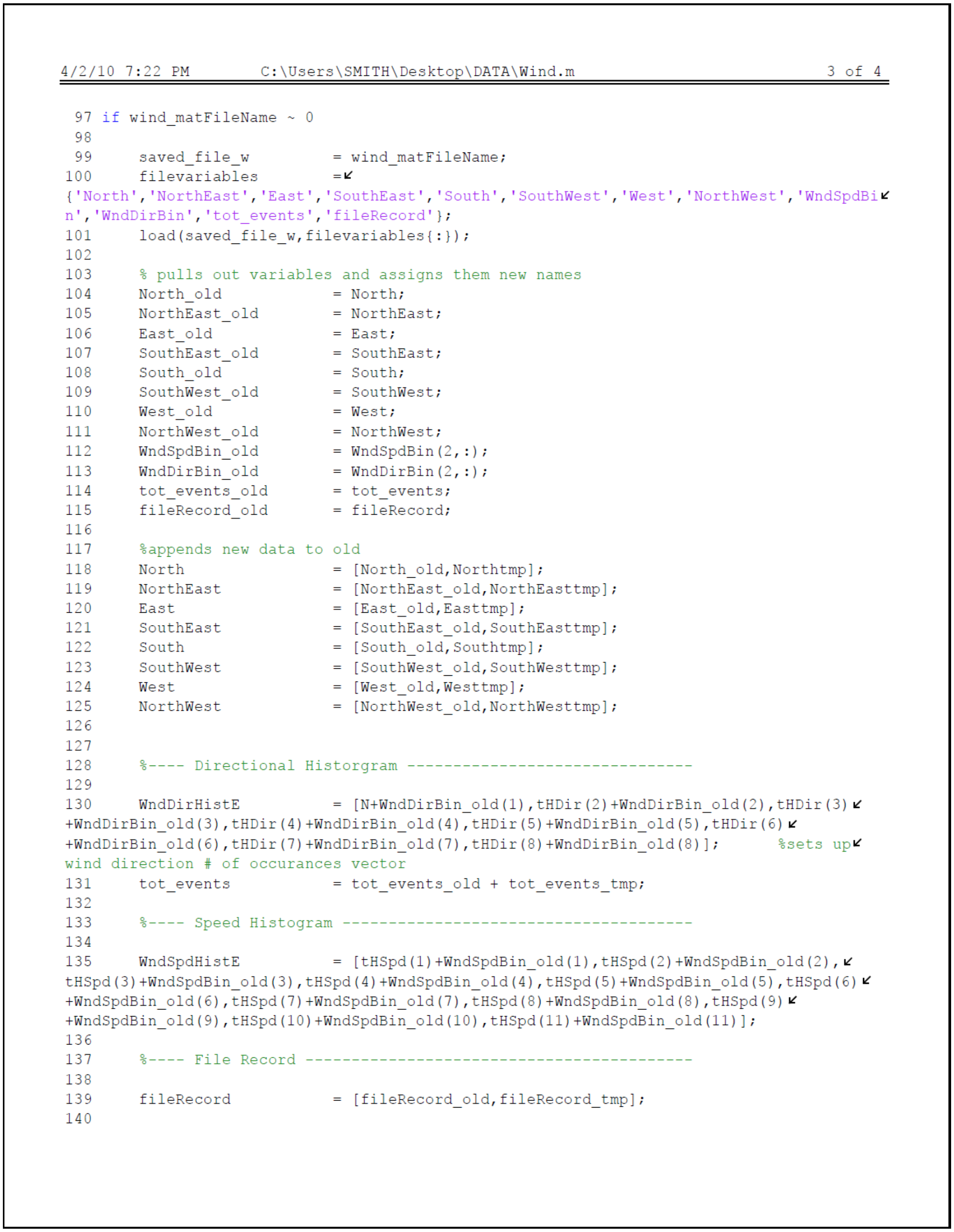
B.4 Wind.m



```
4/2/10 7:22 PM        C:\Users\SMITH\Desktop\DATA\Wind.m                    1 of 4

function Wind(data_matFileName,wind_matFileName)
%data_matFileName should be entered as a string i.e.'data_100304.mat',
%wind_matFileName entered similarily or enter 0 if no wind_matFileName

%-------------------------------------------------------------------------
%This file is capable of appending new data from 'data_xxxxxx.mat'
%to data old data brought in from 'wind_xxxxxx.mat'
%-------------------------------------------------------------------------

clc

theta_deg                        = [0 45 90 135 180 225 270 315];        %creates↙
directional bins
x                                = [0 5 10 15 20 25 30 35 40 45 50];     %creates↙
wind speed bins
edges                            = [0 22.5 67.5 112.5 157.5 202.5 247.5 292.5 337.5↙
360]; %sets the bin limits for direction

%---- load data file ---------------------------------------

saved_file_d                     = data_matFileName;
filevariables                    = {'data','timestamps'};
load(saved_file_d,filevariables{:});

%---- pull out array of data and assign to variables -------

tot_events_tmp                   = length(data);      %obtains the length of array↙
'data'
WndSpd                           = data(:,3);         %pulls out wind speed array
WndDir                           = data(:,4);         %pulls out wind direction↙
array
fileRecord_tmp                   = timestamps;

%---- Wind speed binned by 8 cardinal directions ----------------


j = 1; Northtmp(j)      = 0;
k = 1; NorthEasttmp(j)  = 0;
l = 1; Easttmp(j)       = 0;
m = 1; SouthEasttmp(j)  = 0;
n = 1; Southtmp(j)      = 0;
o = 1; SouthWesttmp(j)  = 0;
p = 1; Westtmp(j)       = 0;
q = 1; NorthWesttmp(j)  = 0;

for i = 1:tot_events_tmp

    if  WndDir(i) > 337.5 | WndDir(i) <= 22.5

        Northtmp(j) = WndSpd(i);
        j = j + 1;

```

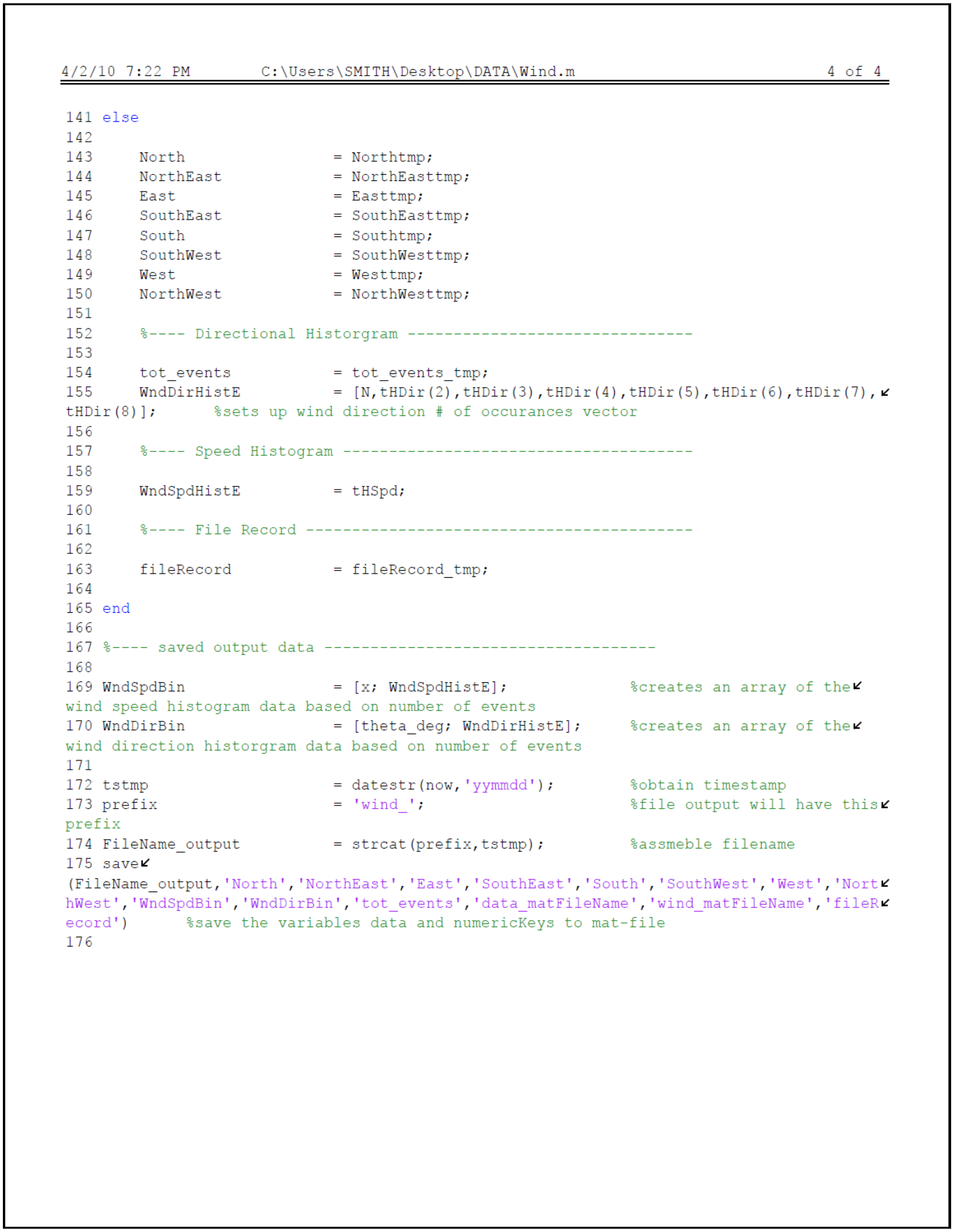


```
     elseif WndDir(i) > 22.5 & WndDir(i) <= 67.5

        NorthEasttmp(k) = WndSpd(i);
        k = k + 1;

     elseif WndDir(i) > 67.5 & WndDir(i) <= 112.5

        Easttmp(l) = WndSpd(i);
        l = l + 1;

     elseif WndDir(i) > 112.5 & WndDir(i) <= 157.5

        SouthEasttmp(m) = WndSpd(i);
        m = m + 1;

     elseif WndDir(i) > 157.5 & WndDir(i) <= 202.5

        Southtmp(n) = WndSpd(i);
        n = n + 1;

     elseif WndDir(i) > 202.5 & WndDir(i) <= 247.5

        SouthWesttmp(o) = WndSpd(i);
        o = o + 1;

     elseif WndDir(i) > 247.5 & WndDir(i) <= 292.5

        Westtmp(p) = WndSpd(i);
        p = p + 1;

     else

        NorthWesttmp(q) = WndSpd(i);
        q = q + 1;

     end

end


%---- temporary wind speed and direction histograms -------------

tHDir                            = histc(WndDir,edges);                 %bins the↙
direction data into the 8 cardinal directions (N,NE,E,...)
N                                = tHDir(1)+tHDir(9);                   %337.5 to↙
360 and 0 to 22.5 bins need to be combined for the North direction
tHSpd                            = hist(WndSpd,x);                      %bins↙
windspeed into x


%---- loads old wind.mat file (if, otherwise else)------------------------

```



4/2/10 7:22 PM C:\Users\SMITH\Desktop\DATA\Wind.m 3 of 4

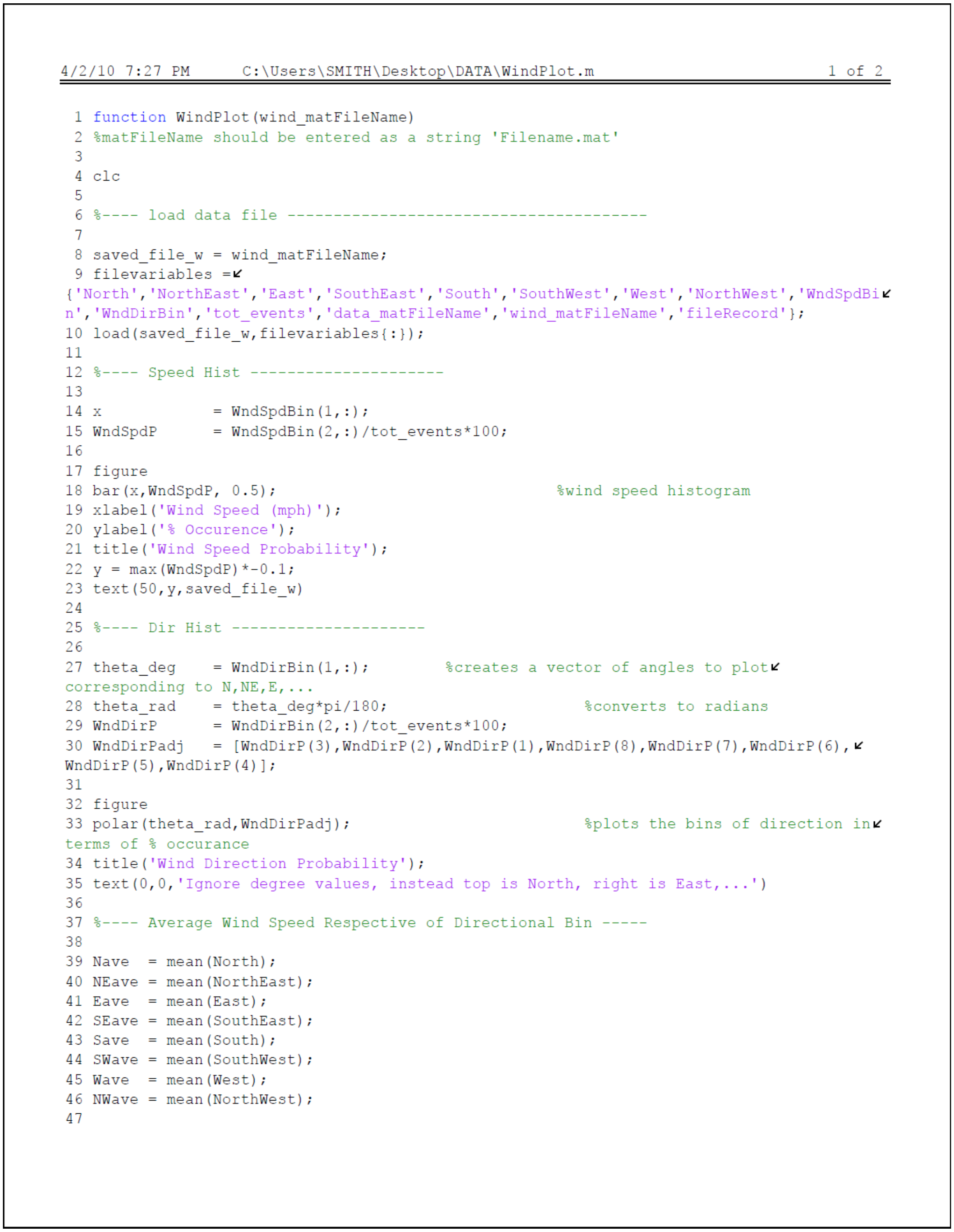
```
if wind_matFileName ~ 0

    saved_file_w          = wind_matFileName;
    filevariables         =↙
{'North','NorthEast','East','SouthEast','South','SouthWest','West','NorthWest','WndSpdBi↙
n','WndDirBin','tot_events','fileRecord'};
    load(saved_file_w,filevariables{:});

    % pulls out variables and assigns them new names
    North_old             = North;
    NorthEast_old         = NorthEast;
    East_old              = East;
    SouthEast_old         = SouthEast;
    South_old             = South;
    SouthWest_old         = SouthWest;
    West_old              = West;
    NorthWest_old         = NorthWest;
    WndSpdBin_old         = WndSpdBin(2,:);
    WndDirBin_old         = WndDirBin(2,:);
    tot_events_old        = tot_events;
    fileRecord_old        = fileRecord;

    %appends new data to old
    North                 = [North_old,Northtmp];
    NorthEast             = [NorthEast_old,NorthEasttmp];
    East                  = [East_old,Easttmp];
    SouthEast             = [SouthEast_old,SouthEasttmp];
    South                 = [South_old,Southtmp];
    SouthWest             = [SouthWest_old,SouthWesttmp];
    West                  = [West_old,Westtmp];
    NorthWest             = [NorthWest_old,NorthWesttmp];


    %---- Directional Historgram -------------------------------

    WndDirHistE           = [N+WndDirBin_old(1),tHDir(2)+WndDirBin_old(2),tHDir(3)↙
+WndDirBin_old(3),tHDir(4)+WndDirBin_old(4),tHDir(5)+WndDirBin_old(5),tHDir(6)↙
+WndDirBin_old(6),tHDir(7)+WndDirBin_old(7),tHDir(8)+WndDirBin_old(8)];      %sets up↙
wind direction # of occurances vector
    tot_events            = tot_events_old + tot_events_tmp;

    %---- Speed Histogram --------------------------------------

    WndSpdHistE           = [tHSpd(1)+WndSpdBin_old(1),tHSpd(2)+WndSpdBin_old(2),↙
tHSpd(3)+WndSpdBin_old(3),tHSpd(4)+WndSpdBin_old(4),tHSpd(5)+WndSpdBin_old(5),tHSpd(6)↙
+WndSpdBin_old(6),tHSpd(7)+WndSpdBin_old(7),tHSpd(8)+WndSpdBin_old(8),tHSpd(9)↙
+WndSpdBin_old(9),tHSpd(10)+WndSpdBin_old(10),tHSpd(11)+WndSpdBin_old(11)];

    %---- File Record ------------------------------------------

    fileRecord            = [fileRecord_old,fileRecord_tmp];

```



4/2/10 7:22 PM   C:\Users\SMITH\Desktop\DATA\Wind.m   4 of 4

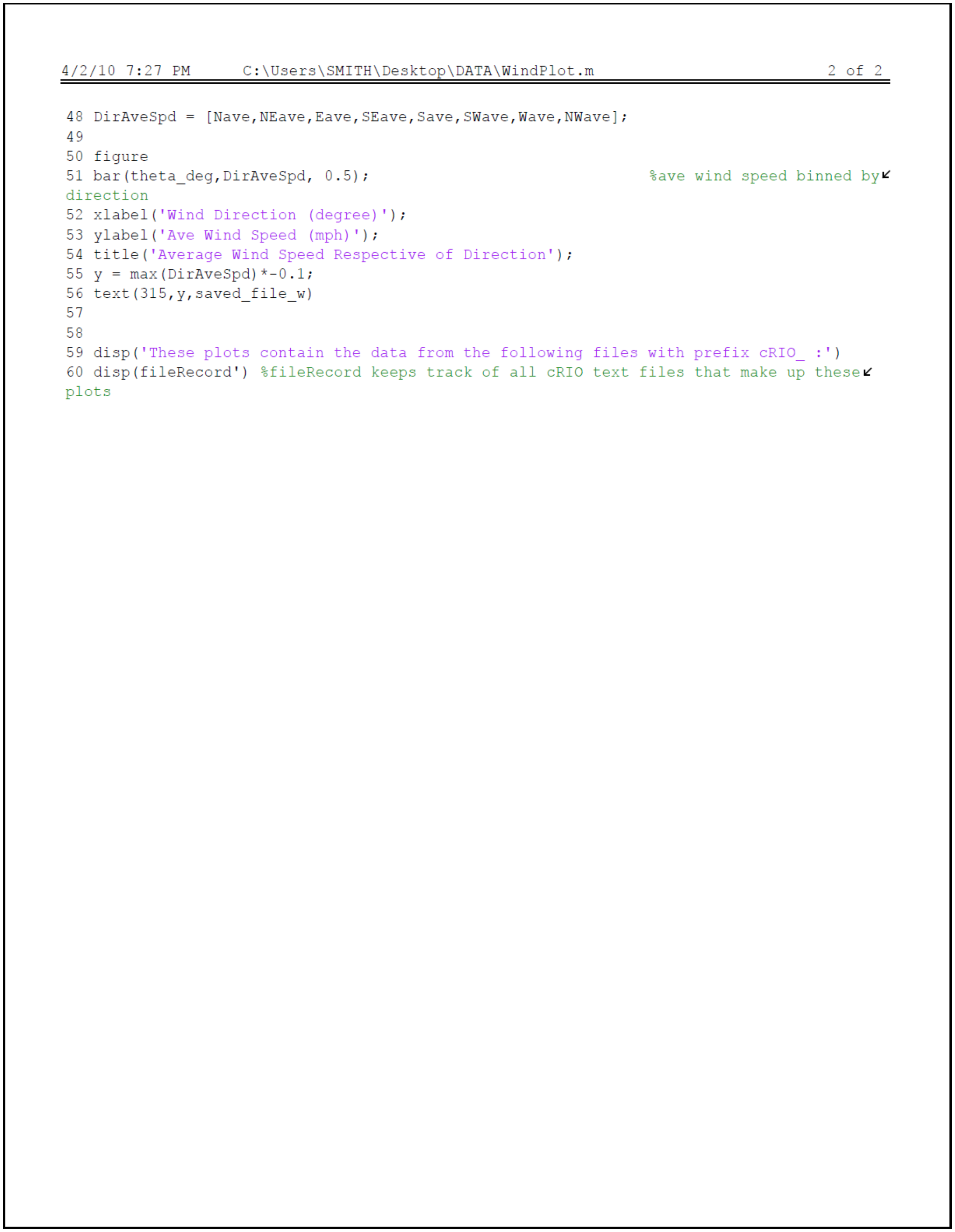
```
else

    North                 = Northtmp;
    NorthEast             = NorthEasttmp;
    East                  = Easttmp;
    SouthEast             = SouthEasttmp;
    South                 = Southtmp;
    SouthWest             = SouthWesttmp;
    West                  = Westtmp;
    NorthWest             = NorthWesttmp;

    %---- Directional Historgram -------------------------------

    tot_events            = tot_events_tmp;
    WndDirHistE           = [N,tHDir(2),tHDir(3),tHDir(4),tHDir(5),tHDir(6),tHDir(7),↙
tHDir(8)];       %sets up wind direction # of occurances vector

    %---- Speed Histogram --------------------------------------

    WndSpdHistE           = tHSpd;

    %---- File Record ------------------------------------------

    fileRecord            = fileRecord_tmp;

end

%---- saved output data -------------------------------------

WndSpdBin                 = [x; WndSpdHistE];            %creates an array of the↙
wind speed histogram data based on number of events
WndDirBin                 = [theta_deg; WndDirHistE];     %creates an array of the↙
wind direction historgram data based on number of events

tstmp                     = datestr(now,'yymmdd');        %obtain timestamp
prefix                    = 'wind_';                      %file output will have this↙
prefix
FileName_output           = strcat(prefix,tstmp);         %assmeble filename
save↙
(FileName_output,'North','NorthEast','East','SouthEast','South','SouthWest','West','Nort↙
hWest','WndSpdBin','WndDirBin','tot_events','data_matFileName','wind_matFileName','fileR↙
ecord')       %save the variables data and numericKeys to mat-file

```

B.5 WindPlot.m



4/2/10 7:27 PM C:\Users\SMITH\Desktop\DATA\WindPlot.m 1 of 2

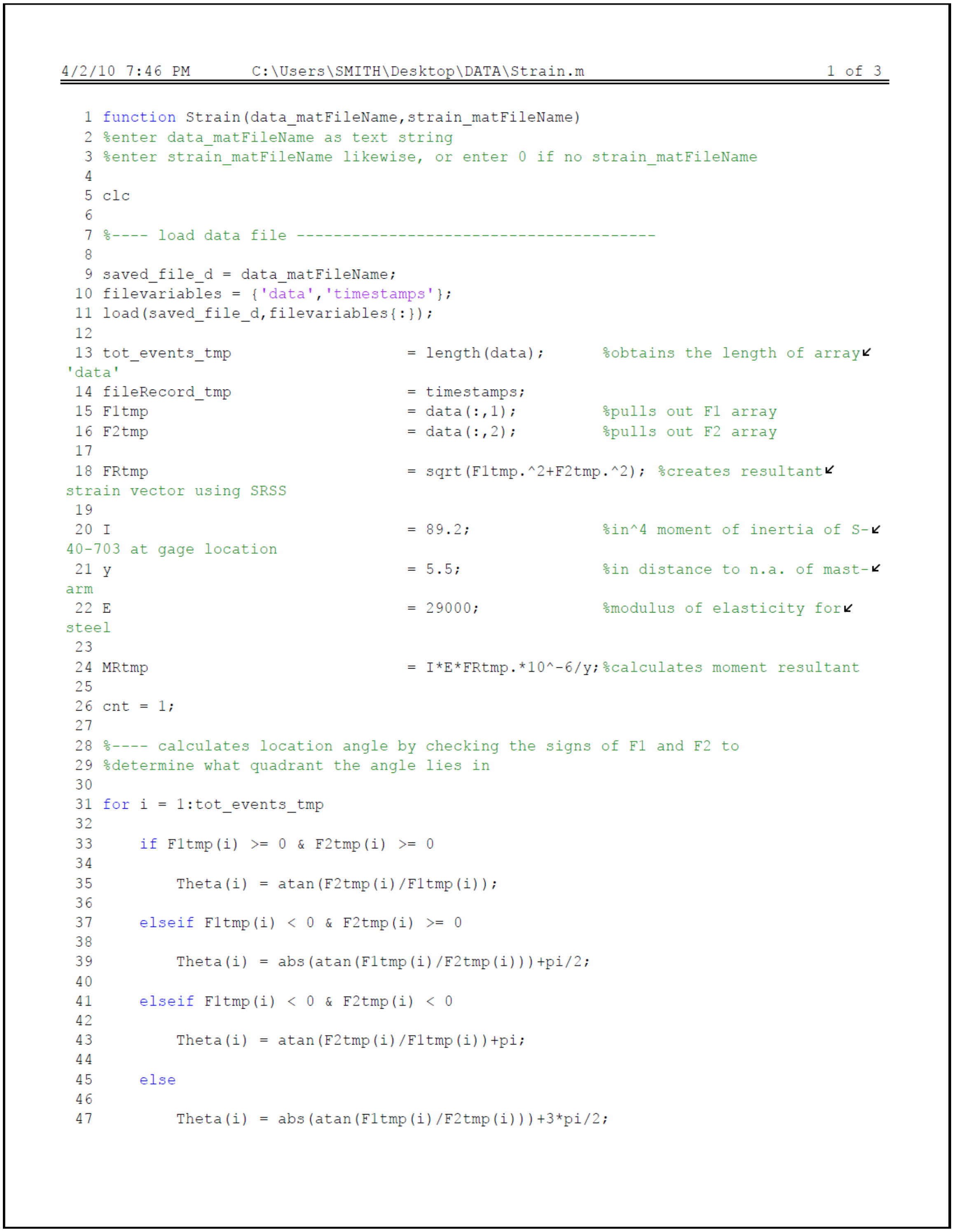
```
function WindPlot(wind_matFileName)
%matFileName should be entered as a string 'Filename.mat'

clc

%---- load data file ---------------------------------------

saved_file_w = wind_matFileName;
filevariables ={'North','NorthEast','East','SouthEast','South','SouthWest','West','NorthWest','WndSpdBin','WndDirBin','tot_events','data_matFileName','wind_matFileName','fileRecord'};
load(saved_file_w,filevariables{:});

%---- Speed Hist ---------------------

x            = WndSpdBin(1,:);
WndSpdP      = WndSpdBin(2,:)/tot_events*100;

figure
bar(x,WndSpdP, 0.5);                              %wind speed histogram
xlabel('Wind Speed (mph)');
ylabel('% Occurence');
title('Wind Speed Probability');
y = max(WndSpdP)*-0.1;
text(50,y,saved_file_w)

%---- Dir Hist ---------------------

theta_deg    = WndDirBin(1,:);           %creates a vector of angles to plot corresponding to N,NE,E,...
theta_rad    = theta_deg*pi/180;                     %converts to radians
WndDirP      = WndDirBin(2,:)/tot_events*100;
WndDirPadj   = [WndDirP(3),WndDirP(2),WndDirP(1),WndDirP(8),WndDirP(7),WndDirP(6),WndDirP(5),WndDirP(4)];

figure
polar(theta_rad,WndDirPadj);                         %plots the bins of direction in terms of % occurance
title('Wind Direction Probability');
text(0,0,'Ignore degree values, instead top is North, right is East,...')

%---- Average Wind Speed Respective of Directional Bin -----

Nave  = mean(North);
NEave = mean(NorthEast);
Eave  = mean(East);
SEave = mean(SouthEast);
Save  = mean(South);
SWave = mean(SouthWest);
Wave  = mean(West);
NWave = mean(NorthWest);

```



4/2/10 7:27 PM C:\Users\SMITH\Desktop\DATA\WindPlot.m 2 of 2

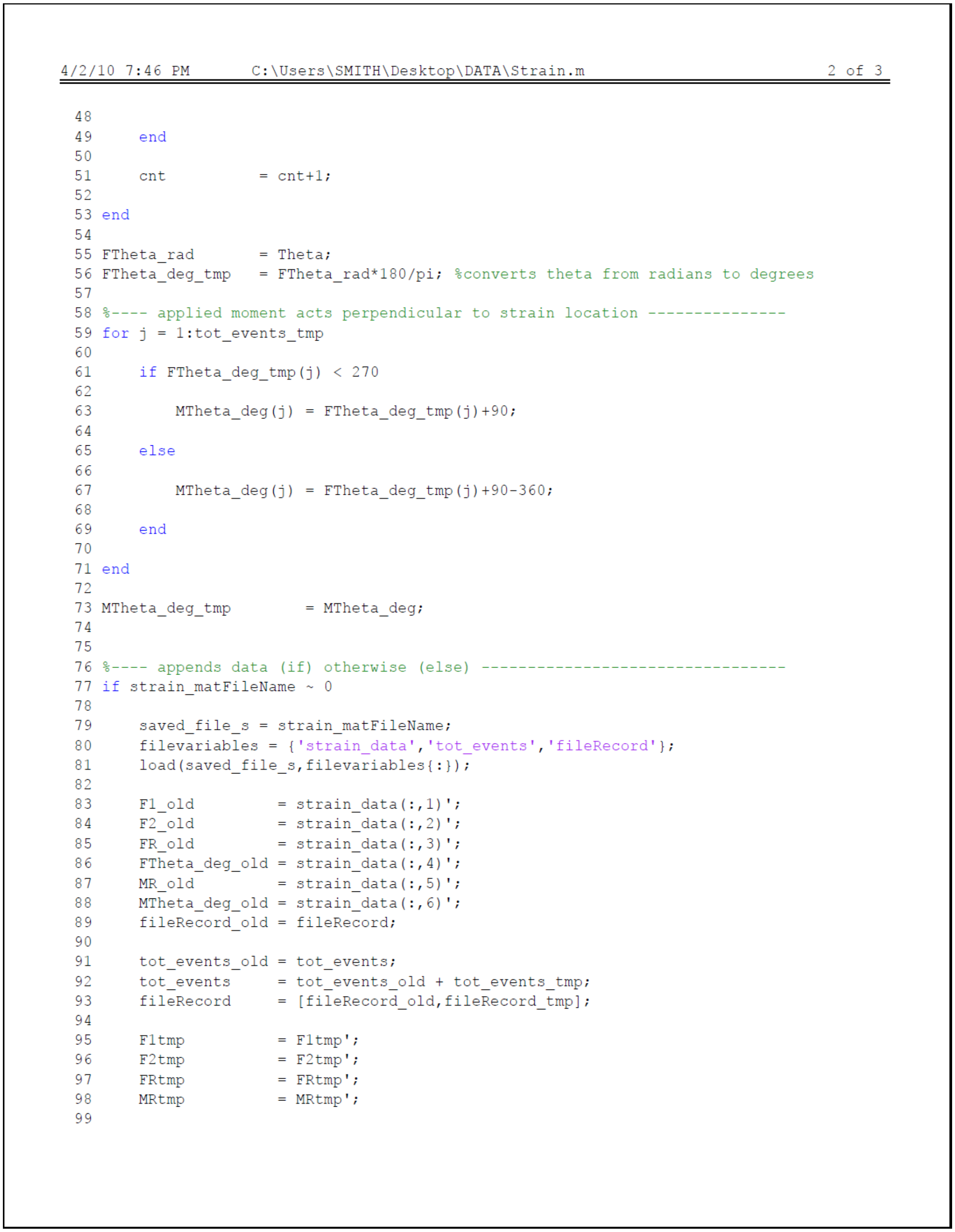
```
DirAveSpd = [Nave,NEave,Eave,SEave,Save,SWave,Wave,NWave];

figure
bar(theta_deg,DirAveSpd, 0.5);                              %ave wind speed binned by direction
xlabel('Wind Direction (degree)');
ylabel('Ave Wind Speed (mph)');
title('Average Wind Speed Respective of Direction');
y = max(DirAveSpd)*-0.1;
text(315,y,saved_file_w)


disp('These plots contain the data from the following files with prefix cRIO_ :')
disp(fileRecord') %fileRecord keeps track of all cRIO text files that make up these plots
```

B.6 Strain.m



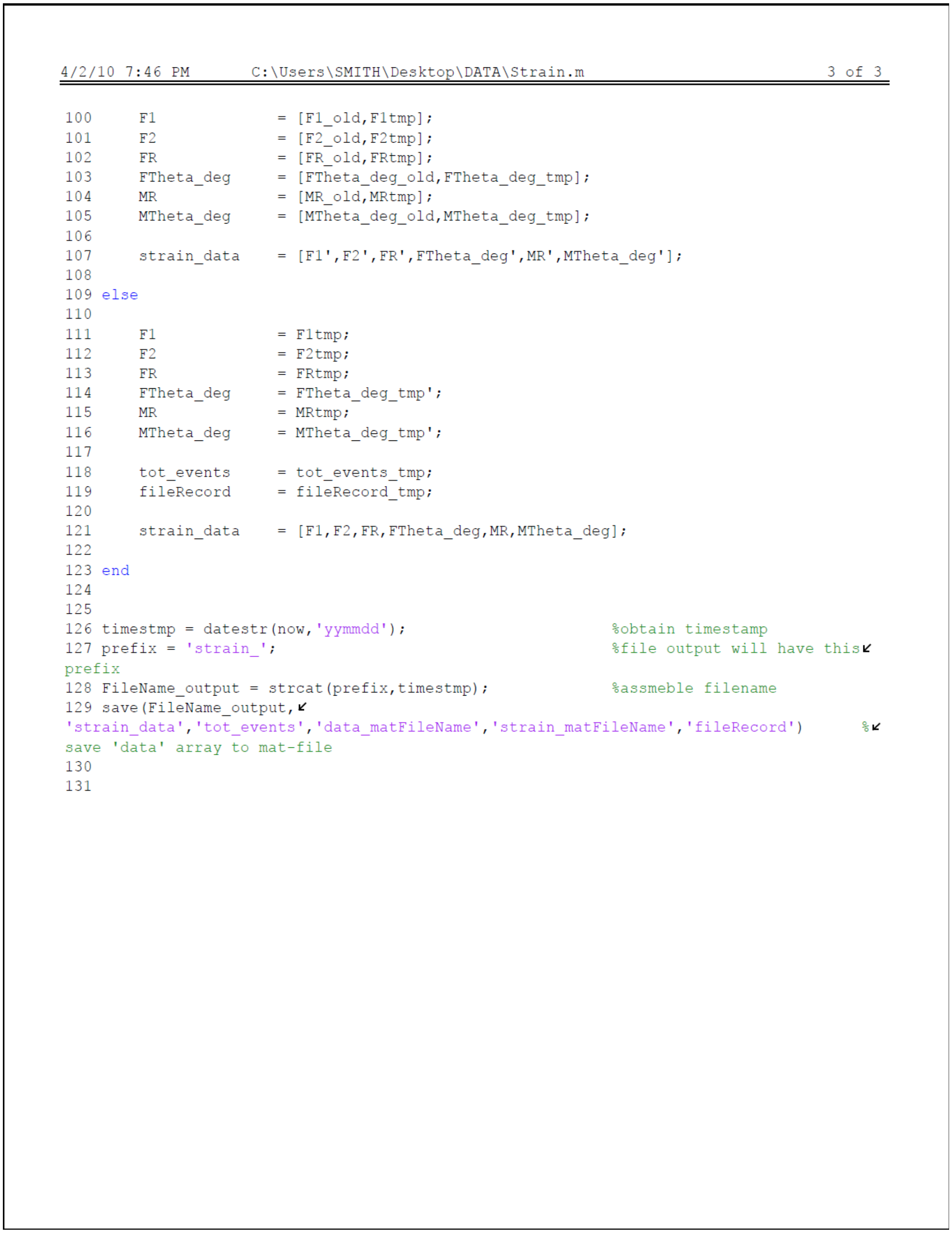
4/2/10 7:46 PM C:\Users\SMITH\Desktop\DATA\Strain.m 1 of 3

```
function Strain(data_matFileName,strain_matFileName)
%enter data_matFileName as text string
%enter strain_matFileName likewise, or enter 0 if no strain_matFileName

clc

%---- load data file ---------------------------------------

saved_file_d = data_matFileName;
filevariables = {'data','timestamps'};
load(saved_file_d,filevariables{:});

tot_events_tmp                   = length(data);      %obtains the length of array↙
'data'
fileRecord_tmp                   = timestamps;
F1tmp                            = data(:,1);         %pulls out F1 array
F2tmp                            = data(:,2);         %pulls out F2 array

FRtmp                            = sqrt(F1tmp.^2+F2tmp.^2); %creates resultant↙
strain vector using SRSS

I                                = 89.2;              %in^4 moment of inertia of S-↙
40-703 at gage location
y                                = 5.5;               %in distance to n.a. of mast-↙
arm
E                                = 29000;             %modulus of elasticity for↙
steel

MRtmp                            = I*E*FRtmp.*10^-6/y;%calculates moment resultant

cnt = 1;

%---- calculates location angle by checking the signs of F1 and F2 to
%determine what quadrant the angle lies in

for i = 1:tot_events_tmp

    if F1tmp(i) >= 0 & F2tmp(i) >= 0

        Theta(i) = atan(F2tmp(i)/F1tmp(i));

    elseif F1tmp(i) < 0 & F2tmp(i) >= 0

        Theta(i) = abs(atan(F1tmp(i)/F2tmp(i)))+pi/2;

    elseif F1tmp(i) < 0 & F2tmp(i) < 0

        Theta(i) = atan(F2tmp(i)/F1tmp(i))+pi;

    else

        Theta(i) = abs(atan(F1tmp(i)/F2tmp(i)))+3*pi/2;
```



4/2/10 7:46 PM C:\Users\SMITH\Desktop\DATA\Strain.m 2 of 3

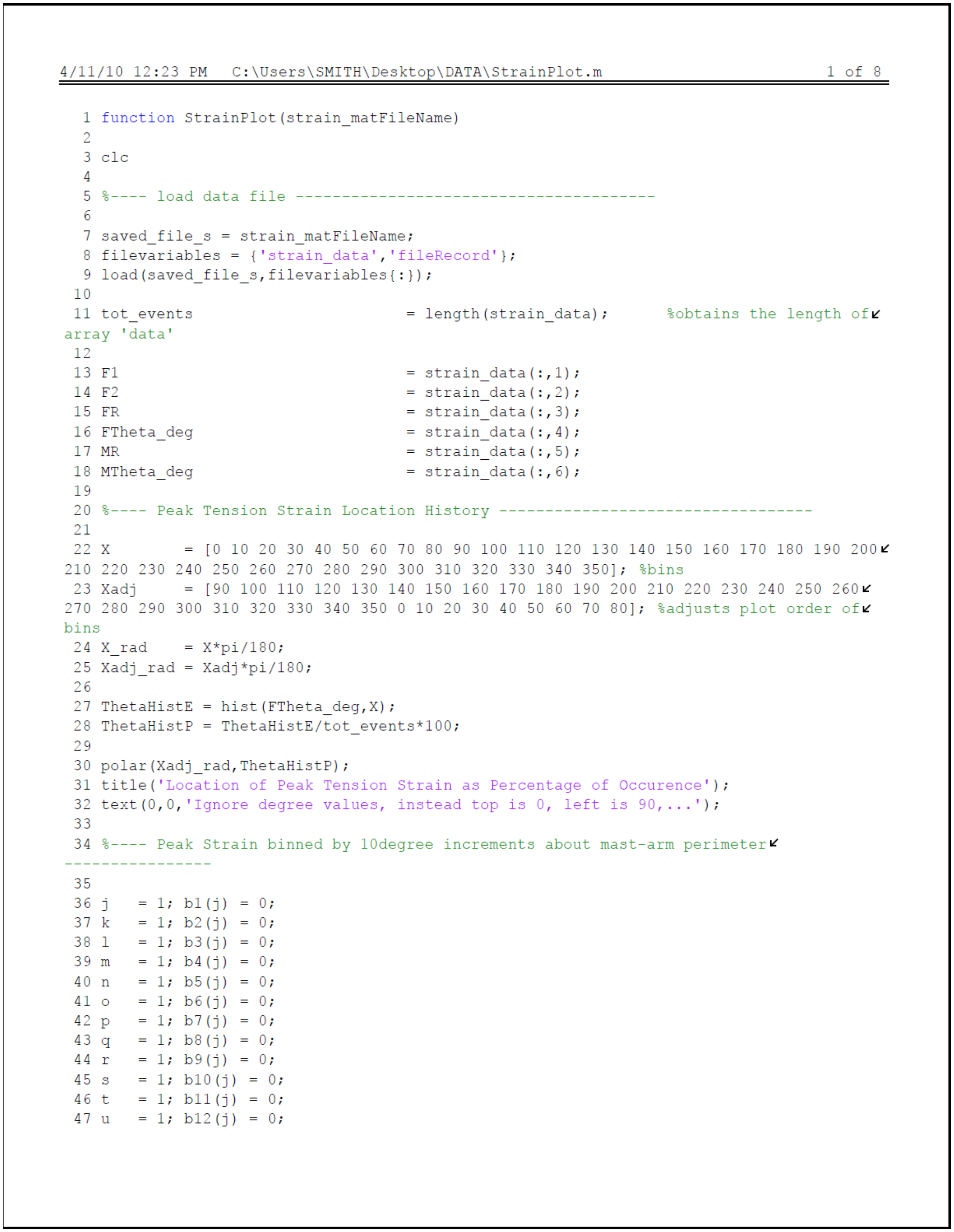
```

    end

    cnt          = cnt+1;

end

FTheta_rad       = Theta;
FTheta_deg_tmp   = FTheta_rad*180/pi; %converts theta from radians to degrees

%---- applied moment acts perpendicular to strain location ---------------
for j = 1:tot_events_tmp

    if FTheta_deg_tmp(j) < 270

        MTheta_deg(j) = FTheta_deg_tmp(j)+90;

    else

        MTheta_deg(j) = FTheta_deg_tmp(j)+90-360;

    end

end

MTheta_deg_tmp        = MTheta_deg;


%---- appends data (if) otherwise (else) ---------------------------------
if strain_matFileName ~ 0

    saved_file_s = strain_matFileName;
    filevariables = {'strain_data','tot_events','fileRecord'};
    load(saved_file_s,filevariables{:});

    F1_old         = strain_data(:,1)';
    F2_old         = strain_data(:,2)';
    FR_old         = strain_data(:,3)';
    FTheta_deg_old = strain_data(:,4)';
    MR_old         = strain_data(:,5)';
    MTheta_deg_old = strain_data(:,6)';
    fileRecord_old = fileRecord;

    tot_events_old = tot_events;
    tot_events     = tot_events_old + tot_events_tmp;
    fileRecord     = [fileRecord_old,fileRecord_tmp];

    F1tmp          = F1tmp';
    F2tmp          = F2tmp';
    FRtmp          = FRtmp';
    MRtmp          = MRtmp';

```



4/2/10 7:46 PM C:\Users\SMITH\Desktop\DATA\Strain.m 3 of 3

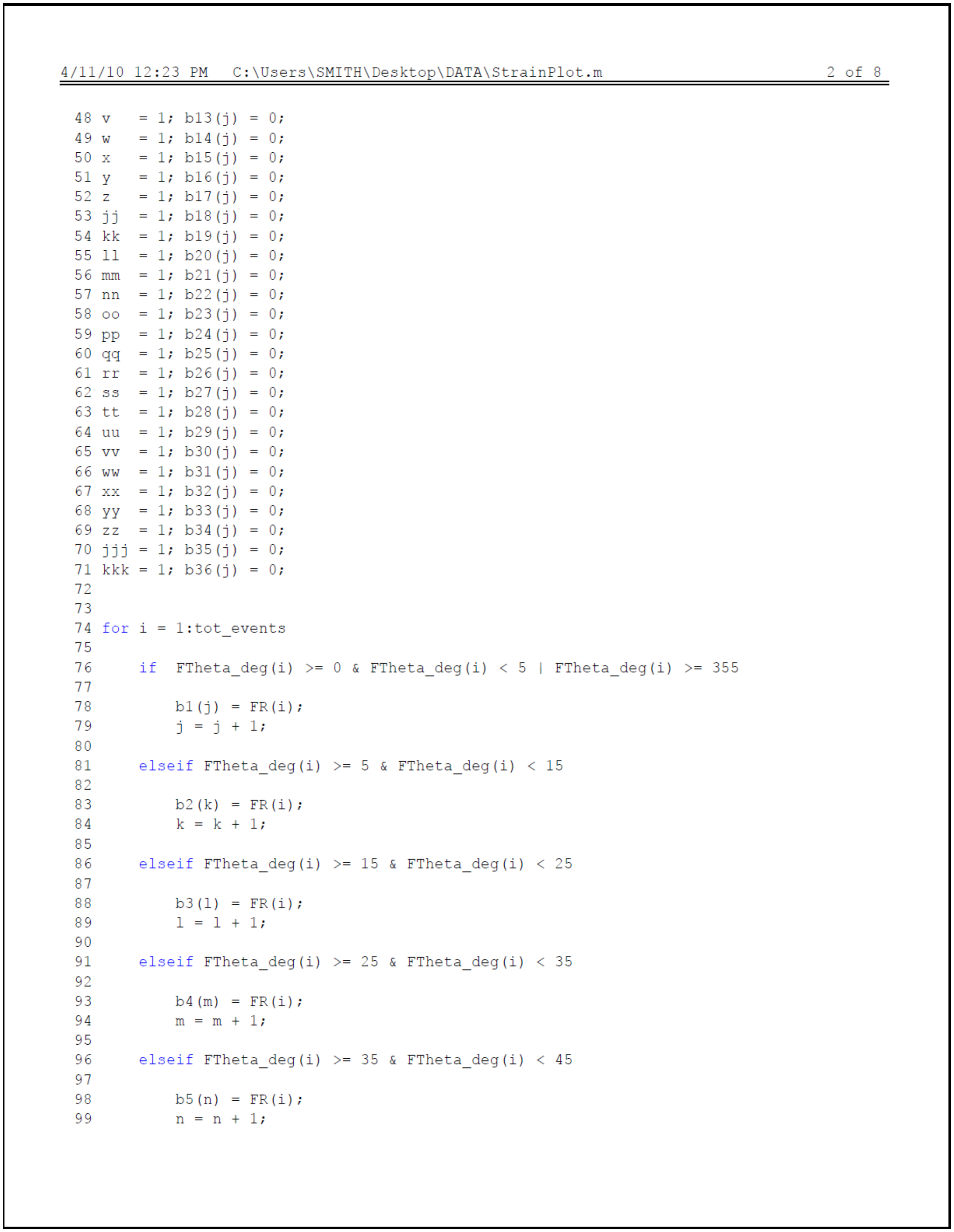
```
    F1              = [F1_old,F1tmp];
    F2              = [F2_old,F2tmp];
    FR              = [FR_old,FRtmp];
    FTheta_deg      = [FTheta_deg_old,FTheta_deg_tmp];
    MR              = [MR_old,MRtmp];
    MTheta_deg      = [MTheta_deg_old,MTheta_deg_tmp];

    strain_data     = [F1',F2',FR',FTheta_deg',MR',MTheta_deg'];

else

    F1              = F1tmp;
    F2              = F2tmp;
    FR              = FRtmp;
    FTheta_deg      = FTheta_deg_tmp';
    MR              = MRtmp;
    MTheta_deg      = MTheta_deg_tmp';

    tot_events      = tot_events_tmp;
    fileRecord      = fileRecord_tmp;

    strain_data     = [F1,F2,FR,FTheta_deg,MR,MTheta_deg];

end


timestmp = datestr(now,'yymmdd');                       %obtain timestamp
prefix = 'strain_';                                     %file output will have this↙
prefix
FileName_output = strcat(prefix,timestmp);              %assmeble filename
save(FileName_output,↙
'strain_data','tot_events','data_matFileName','strain_matFileName','fileRecord')      %↙
save 'data' array to mat-file


```

B.7 StrainPlot.m

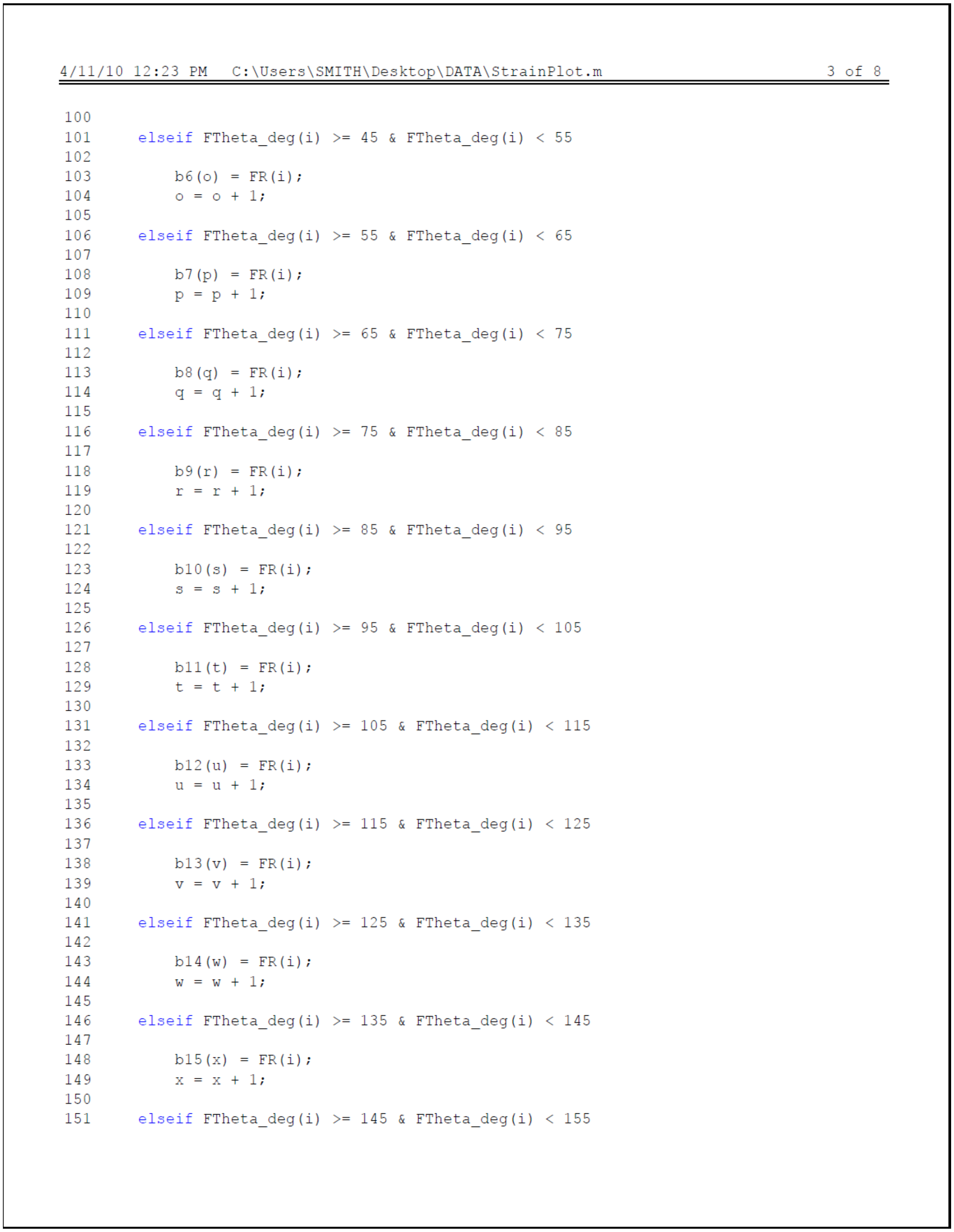


4/11/10 12:23 PM C:\Users\SMITH\Desktop\DATA\StrainPlot.m 1 of 8

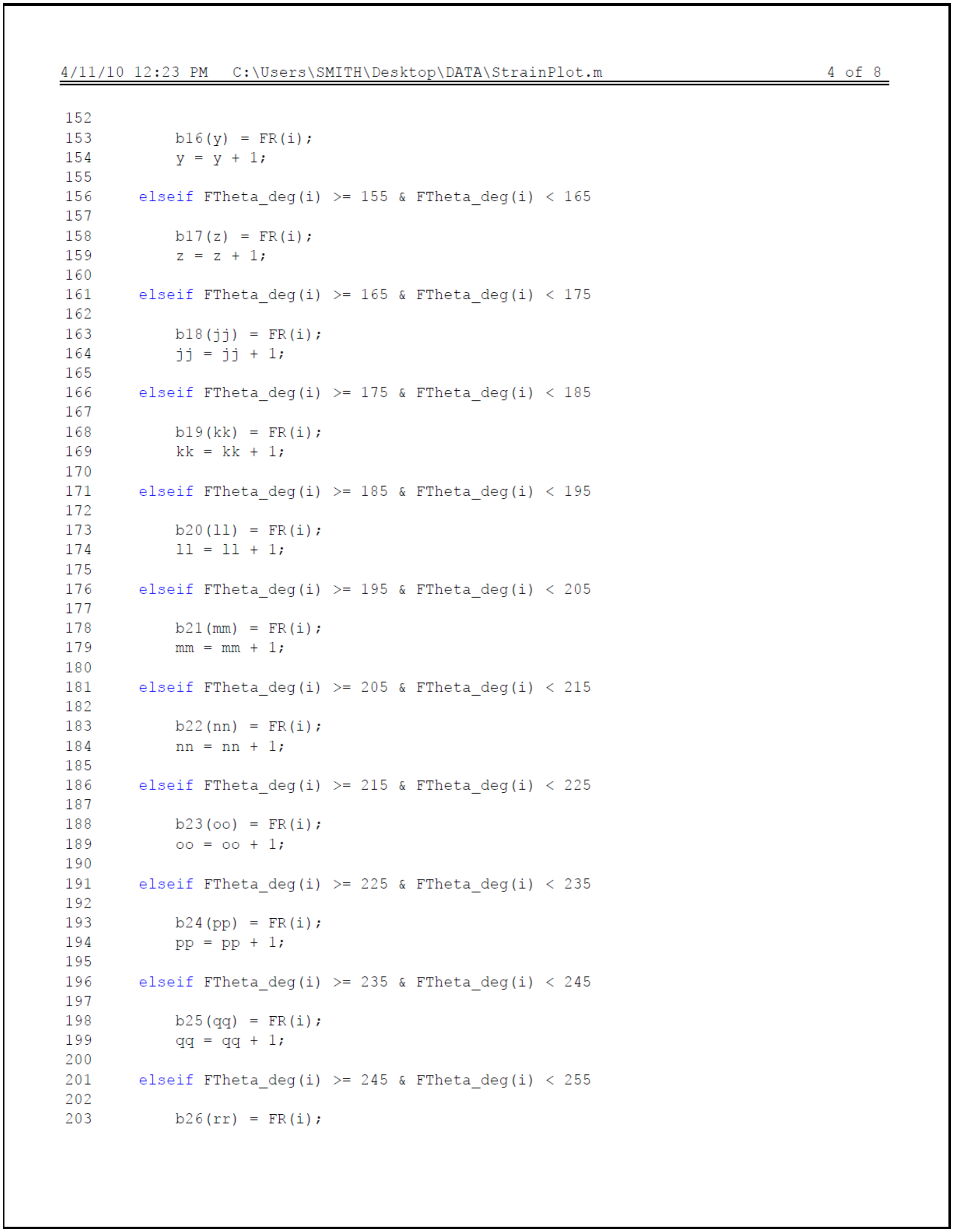
```
function StrainPlot(strain_matFileName)

clc

%---- load data file ---------------------------------------

saved_file_s = strain_matFileName;
filevariables = {'strain_data','fileRecord'};
load(saved_file_s,filevariables{:});

tot_events                        = length(strain_data);      %obtains the length of array 'data'

F1                                = strain_data(:,1);
F2                                = strain_data(:,2);
FR                                = strain_data(:,3);
FTheta_deg                        = strain_data(:,4);
MR                                = strain_data(:,5);
MTheta_deg                        = strain_data(:,6);

%---- Peak Tension Strain Location History ----------------------------------

X        = [0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350]; %bins
Xadj     = [90 100 110 120 130 140 150 160 170 180 190 200 210 220 230 240 250 260 270 280 290 300 310 320 330 340 350 0 10 20 30 40 50 60 70 80]; %adjusts plot order of bins
X_rad    = X*pi/180;
Xadj_rad = Xadj*pi/180;

ThetaHistE = hist(FTheta_deg,X);
ThetaHistP = ThetaHistE/tot_events*100;

polar(Xadj_rad,ThetaHistP);
title('Location of Peak Tension Strain as Percentage of Occurence');
text(0,0,'Ignore degree values, instead top is 0, left is 90,...');

%---- Peak Strain binned by 10degree increments about mast-arm perimeter----------------

j    = 1; b1(j) = 0;
k    = 1; b2(j) = 0;
l    = 1; b3(j) = 0;
m    = 1; b4(j) = 0;
n    = 1; b5(j) = 0;
o    = 1; b6(j) = 0;
p    = 1; b7(j) = 0;
q    = 1; b8(j) = 0;
r    = 1; b9(j) = 0;
s    = 1; b10(j) = 0;
t    = 1; b11(j) = 0;
u    = 1; b12(j) = 0;
```



```
v   = 1; b13(j) = 0;
w   = 1; b14(j) = 0;
x   = 1; b15(j) = 0;
y   = 1; b16(j) = 0;
z   = 1; b17(j) = 0;
jj  = 1; b18(j) = 0;
kk  = 1; b19(j) = 0;
ll  = 1; b20(j) = 0;
mm  = 1; b21(j) = 0;
nn  = 1; b22(j) = 0;
oo  = 1; b23(j) = 0;
pp  = 1; b24(j) = 0;
qq  = 1; b25(j) = 0;
rr  = 1; b26(j) = 0;
ss  = 1; b27(j) = 0;
tt  = 1; b28(j) = 0;
uu  = 1; b29(j) = 0;
vv  = 1; b30(j) = 0;
ww  = 1; b31(j) = 0;
xx  = 1; b32(j) = 0;
yy  = 1; b33(j) = 0;
zz  = 1; b34(j) = 0;
jjj = 1; b35(j) = 0;
kkk = 1; b36(j) = 0;


for i = 1:tot_events

    if  FTheta_deg(i) >= 0 & FTheta_deg(i) < 5 | FTheta_deg(i) >= 355

        b1(j) = FR(i);
        j = j + 1;

    elseif FTheta_deg(i) >= 5 & FTheta_deg(i) < 15

        b2(k) = FR(i);
        k = k + 1;

    elseif FTheta_deg(i) >= 15 & FTheta_deg(i) < 25

        b3(l) = FR(i);
        l = l + 1;

    elseif FTheta_deg(i) >= 25 & FTheta_deg(i) < 35

        b4(m) = FR(i);
        m = m + 1;

    elseif FTheta_deg(i) >= 35 & FTheta_deg(i) < 45

        b5(n) = FR(i);
        n = n + 1;
```

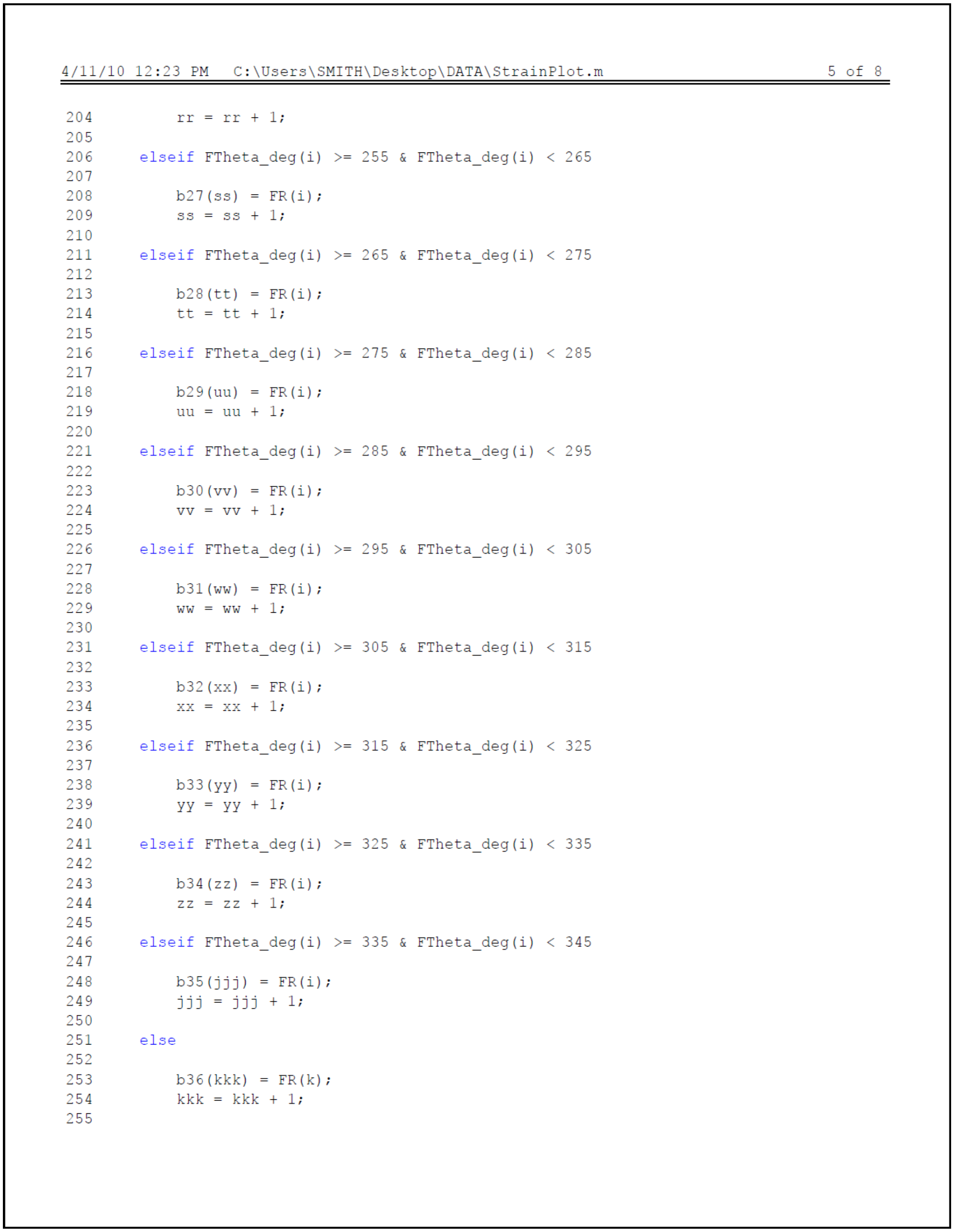


```

    elseif FTheta_deg(i) >= 45 & FTheta_deg(i) < 55

        b6(o) = FR(i);
        o = o + 1;

    elseif FTheta_deg(i) >= 55 & FTheta_deg(i) < 65

        b7(p) = FR(i);
        p = p + 1;

    elseif FTheta_deg(i) >= 65 & FTheta_deg(i) < 75

        b8(q) = FR(i);
        q = q + 1;

    elseif FTheta_deg(i) >= 75 & FTheta_deg(i) < 85

        b9(r) = FR(i);
        r = r + 1;

    elseif FTheta_deg(i) >= 85 & FTheta_deg(i) < 95

        b10(s) = FR(i);
        s = s + 1;

    elseif FTheta_deg(i) >= 95 & FTheta_deg(i) < 105

        b11(t) = FR(i);
        t = t + 1;

    elseif FTheta_deg(i) >= 105 & FTheta_deg(i) < 115

        b12(u) = FR(i);
        u = u + 1;

    elseif FTheta_deg(i) >= 115 & FTheta_deg(i) < 125

        b13(v) = FR(i);
        v = v + 1;

    elseif FTheta_deg(i) >= 125 & FTheta_deg(i) < 135

        b14(w) = FR(i);
        w = w + 1;

    elseif FTheta_deg(i) >= 135 & FTheta_deg(i) < 145

        b15(x) = FR(i);
        x = x + 1;

    elseif FTheta_deg(i) >= 145 & FTheta_deg(i) < 155
```



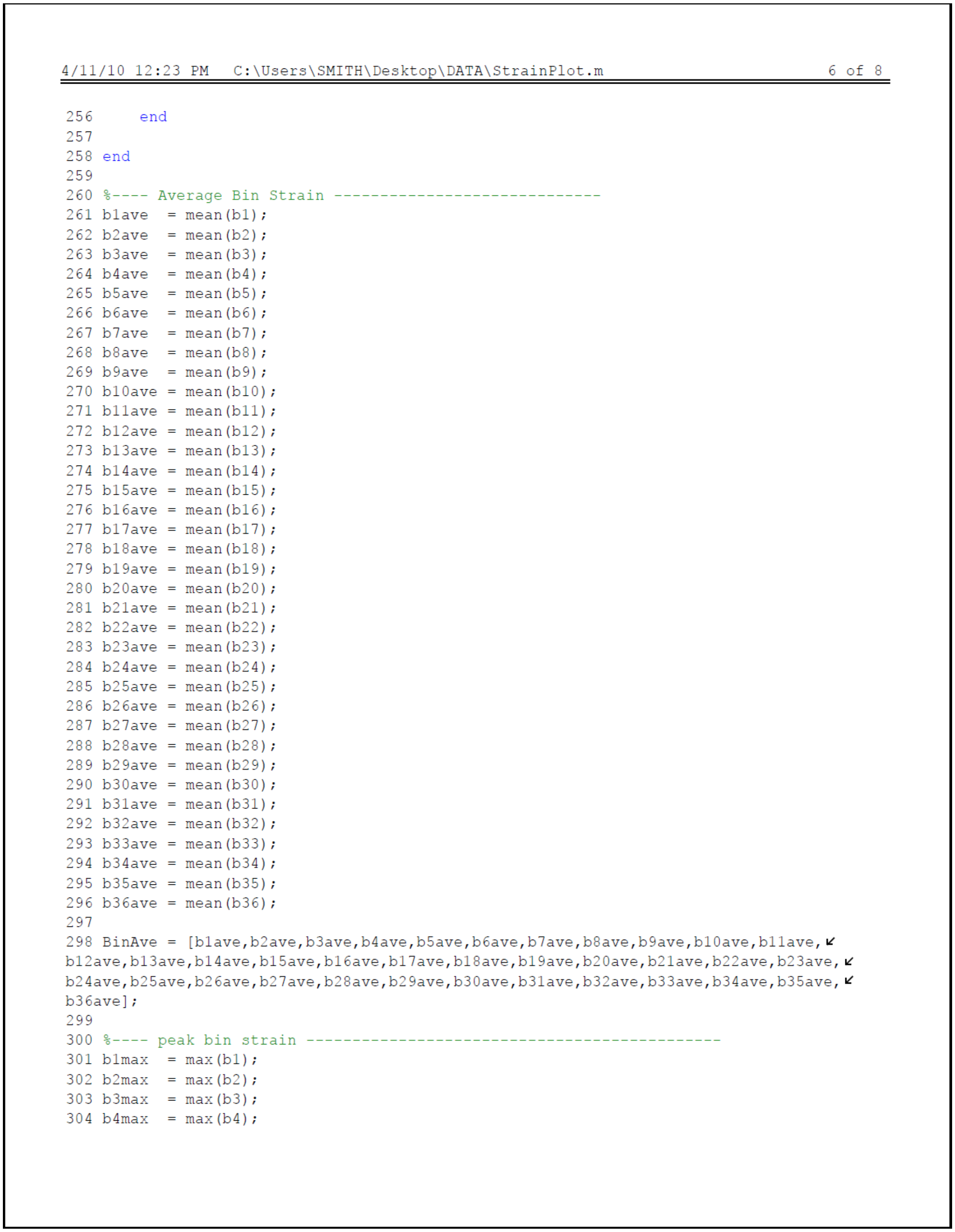
4/11/10 12:23 PM   C:\Users\SMITH\Desktop\DATA\StrainPlot.m   4 of 8

```

        b16(y) = FR(i);
        y = y + 1;

    elseif FTheta_deg(i) >= 155 & FTheta_deg(i) < 165

        b17(z) = FR(i);
        z = z + 1;

    elseif FTheta_deg(i) >= 165 & FTheta_deg(i) < 175

        b18(jj) = FR(i);
        jj = jj + 1;

    elseif FTheta_deg(i) >= 175 & FTheta_deg(i) < 185

        b19(kk) = FR(i);
        kk = kk + 1;

    elseif FTheta_deg(i) >= 185 & FTheta_deg(i) < 195

        b20(ll) = FR(i);
        ll = ll + 1;

    elseif FTheta_deg(i) >= 195 & FTheta_deg(i) < 205

        b21(mm) = FR(i);
        mm = mm + 1;

    elseif FTheta_deg(i) >= 205 & FTheta_deg(i) < 215

        b22(nn) = FR(i);
        nn = nn + 1;

    elseif FTheta_deg(i) >= 215 & FTheta_deg(i) < 225

        b23(oo) = FR(i);
        oo = oo + 1;

    elseif FTheta_deg(i) >= 225 & FTheta_deg(i) < 235

        b24(pp) = FR(i);
        pp = pp + 1;

    elseif FTheta_deg(i) >= 235 & FTheta_deg(i) < 245

        b25(qq) = FR(i);
        qq = qq + 1;

    elseif FTheta_deg(i) >= 245 & FTheta_deg(i) < 255

        b26(rr) = FR(i);
```



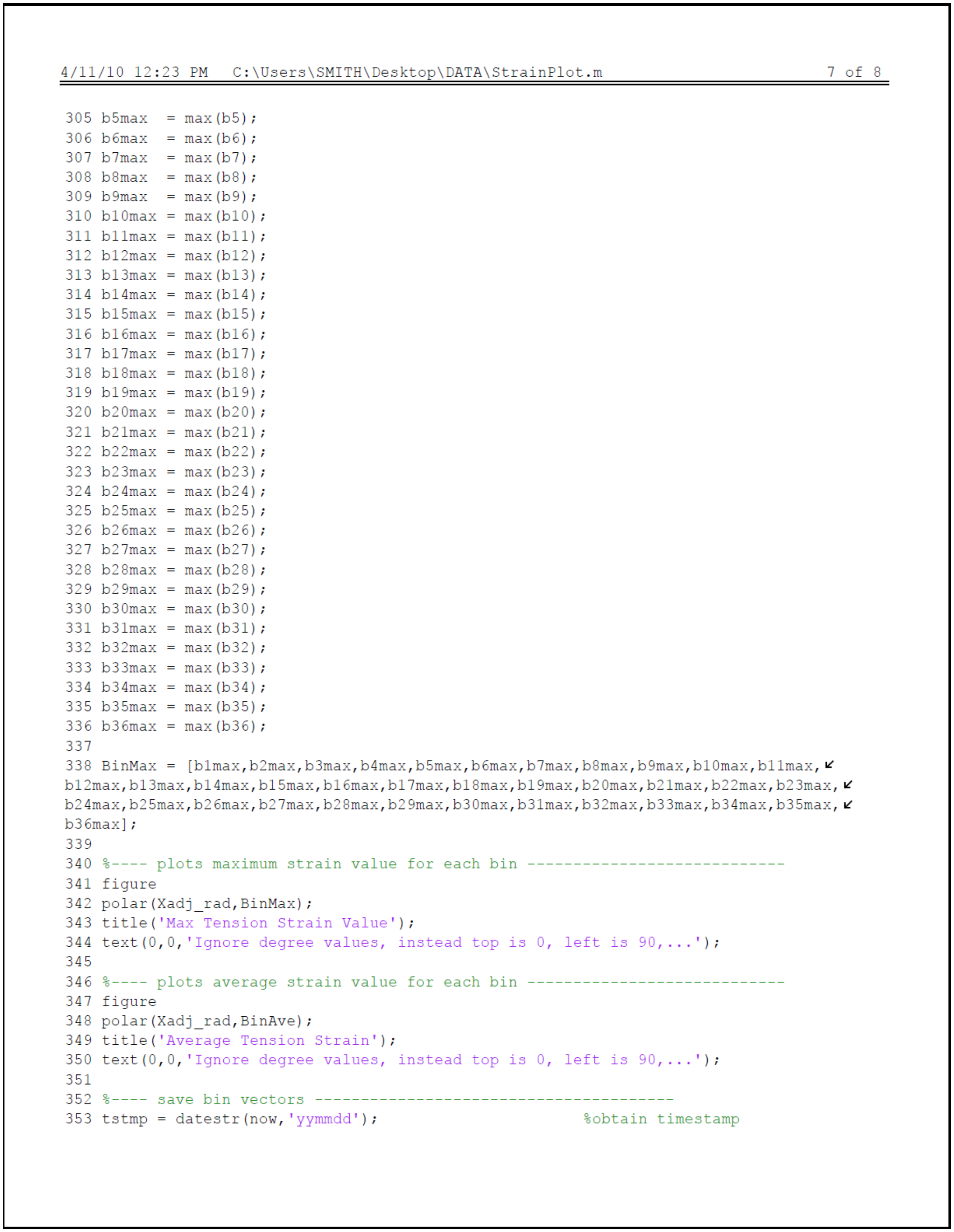
4/11/10 12:23 PM   C:\Users\SMITH\Desktop\DATA\StrainPlot.m   5 of 8

```
        rr = rr + 1;

    elseif FTheta_deg(i) >= 255 & FTheta_deg(i) < 265

        b27(ss) = FR(i);
        ss = ss + 1;

    elseif FTheta_deg(i) >= 265 & FTheta_deg(i) < 275

        b28(tt) = FR(i);
        tt = tt + 1;

    elseif FTheta_deg(i) >= 275 & FTheta_deg(i) < 285

        b29(uu) = FR(i);
        uu = uu + 1;

    elseif FTheta_deg(i) >= 285 & FTheta_deg(i) < 295

        b30(vv) = FR(i);
        vv = vv + 1;

    elseif FTheta_deg(i) >= 295 & FTheta_deg(i) < 305

        b31(ww) = FR(i);
        ww = ww + 1;

    elseif FTheta_deg(i) >= 305 & FTheta_deg(i) < 315

        b32(xx) = FR(i);
        xx = xx + 1;

    elseif FTheta_deg(i) >= 315 & FTheta_deg(i) < 325

        b33(yy) = FR(i);
        yy = yy + 1;

    elseif FTheta_deg(i) >= 325 & FTheta_deg(i) < 335

        b34(zz) = FR(i);
        zz = zz + 1;

    elseif FTheta_deg(i) >= 335 & FTheta_deg(i) < 345

        b35(jjj) = FR(i);
        jjj = jjj + 1;

    else

        b36(kkk) = FR(k);
        kkk = kkk + 1;

```

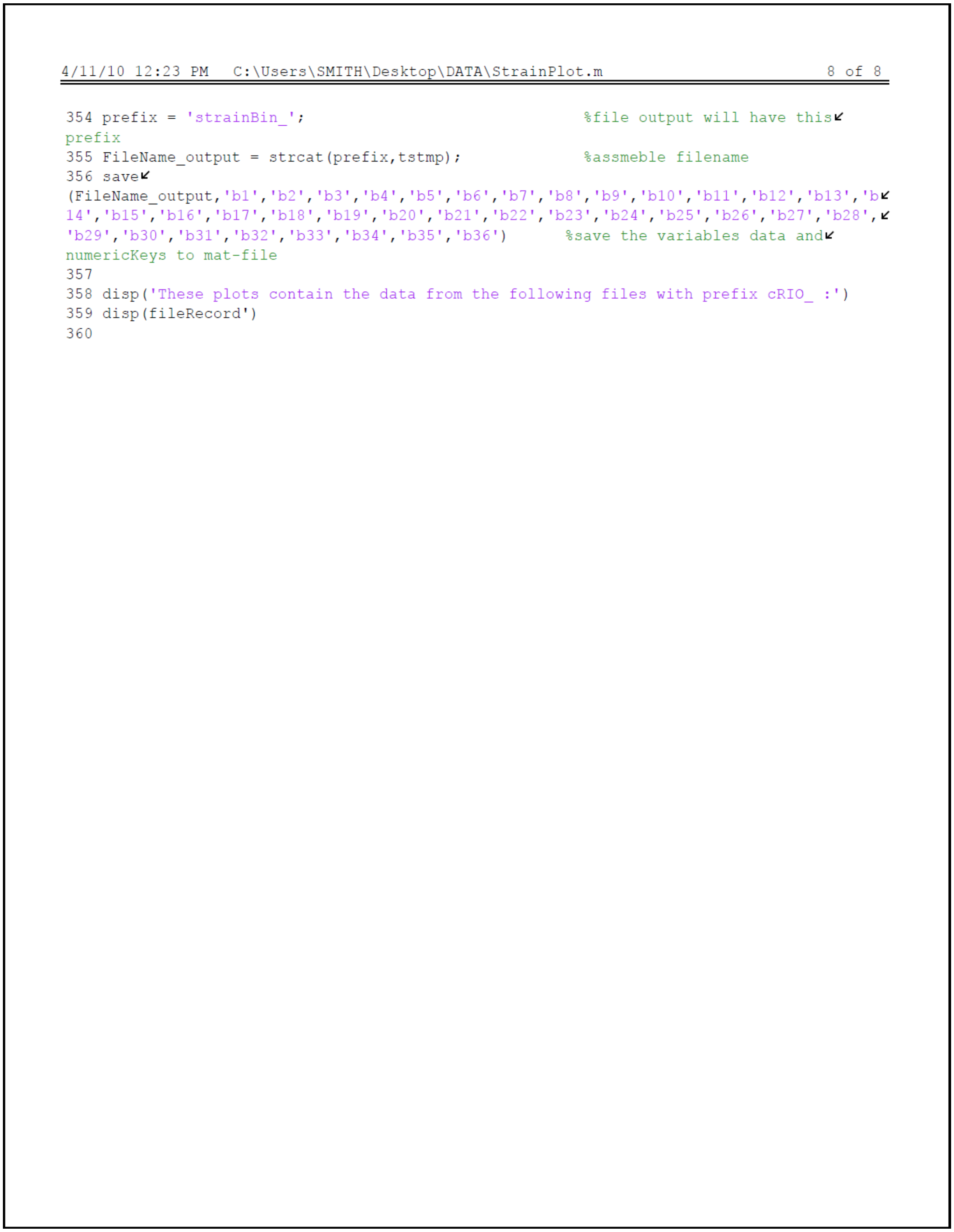


4/11/10 12:23 PM   C:\Users\SMITH\Desktop\DATA\StrainPlot.m   6 of 8

```
    end

end

%---- Average Bin Strain ----------------------------
b1ave  = mean(b1);
b2ave  = mean(b2);
b3ave  = mean(b3);
b4ave  = mean(b4);
b5ave  = mean(b5);
b6ave  = mean(b6);
b7ave  = mean(b7);
b8ave  = mean(b8);
b9ave  = mean(b9);
b10ave = mean(b10);
b11ave = mean(b11);
b12ave = mean(b12);
b13ave = mean(b13);
b14ave = mean(b14);
b15ave = mean(b15);
b16ave = mean(b16);
b17ave = mean(b17);
b18ave = mean(b18);
b19ave = mean(b19);
b20ave = mean(b20);
b21ave = mean(b21);
b22ave = mean(b22);
b23ave = mean(b23);
b24ave = mean(b24);
b25ave = mean(b25);
b26ave = mean(b26);
b27ave = mean(b27);
b28ave = mean(b28);
b29ave = mean(b29);
b30ave = mean(b30);
b31ave = mean(b31);
b32ave = mean(b32);
b33ave = mean(b33);
b34ave = mean(b34);
b35ave = mean(b35);
b36ave = mean(b36);

BinAve = [b1ave,b2ave,b3ave,b4ave,b5ave,b6ave,b7ave,b8ave,b9ave,b10ave,b11ave,↙
b12ave,b13ave,b14ave,b15ave,b16ave,b17ave,b18ave,b19ave,b20ave,b21ave,b22ave,b23ave,↙
b24ave,b25ave,b26ave,b27ave,b28ave,b29ave,b30ave,b31ave,b32ave,b33ave,b34ave,b35ave,↙
b36ave];

%---- peak bin strain ---------------------------------------------
b1max  = max(b1);
b2max  = max(b2);
b3max  = max(b3);
b4max  = max(b4);
```



```
b5max  = max(b5);
b6max  = max(b6);
b7max  = max(b7);
b8max  = max(b8);
b9max  = max(b9);
b10max = max(b10);
b11max = max(b11);
b12max = max(b12);
b13max = max(b13);
b14max = max(b14);
b15max = max(b15);
b16max = max(b16);
b17max = max(b17);
b18max = max(b18);
b19max = max(b19);
b20max = max(b20);
b21max = max(b21);
b22max = max(b22);
b23max = max(b23);
b24max = max(b24);
b25max = max(b25);
b26max = max(b26);
b27max = max(b27);
b28max = max(b28);
b29max = max(b29);
b30max = max(b30);
b31max = max(b31);
b32max = max(b32);
b33max = max(b33);
b34max = max(b34);
b35max = max(b35);
b36max = max(b36);

BinMax = [b1max,b2max,b3max,b4max,b5max,b6max,b7max,b8max,b9max,b10max,b11max,↙
b12max,b13max,b14max,b15max,b16max,b17max,b18max,b19max,b20max,b21max,b22max,b23max,↙
b24max,b25max,b26max,b27max,b28max,b29max,b30max,b31max,b32max,b33max,b34max,b35max,↙
b36max];

%---- plots maximum strain value for each bin ----------------------------
figure
polar(Xadj_rad,BinMax);
title('Max Tension Strain Value');
text(0,0,'Ignore degree values, instead top is 0, left is 90,...');

%---- plots average strain value for each bin ----------------------------
figure
polar(Xadj_rad,BinAve);
title('Average Tension Strain');
text(0,0,'Ignore degree values, instead top is 0, left is 90,...');

%---- save bin vectors ---------------------------------------
tstmp = datestr(now,'yymmdd');                      %obtain timestamp
```



4/11/10 12:23 PM   C:\Users\SMITH\Desktop\DATA\StrainPlot.m   8 of 8

```
prefix = 'strainBin_';                                 %file output will have this↙
prefix
FileName_output = strcat(prefix,tstmp);                %assmeble filename
save↙
(FileName_output,'b1','b2','b3','b4','b5','b6','b7','b8','b9','b10','b11','b12','b13','b↙
14','b15','b16','b17','b18','b19','b20','b21','b22','b23','b24','b25','b26','b27','b28',↙
'b29','b30','b31','b32','b33','b34','b35','b36')      %save the variables data and↙
numericKeys to mat-file

disp('These plots contain the data from the following files with prefix cRIO_ :')
disp(fileRecord')

```