Development and Implementation of a Machine Vision System for Error Detection in an Automated Fiber Optic Coilwinder

by

Sieu Tru Duong

B.S. Mechanical Engineering
Massachusetts Institute Of Technology, 1996

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science

at the

Massachusetts Institute of Technology

January 1998
[February 1998]

Signature of Author

_____

Department of Mechanical Engineering
January 15, 1998

Certified by

_____

Dr. Andre Sharon
Executive Officer of the Manufacturing Institute
Thesis Supervisor

Accepted by

_____

Ain A. Sonin
Chairman, Department Committee on Graduate Students

Development and Implementation of a Machine Vision System for Error Detection in an Automated Fiber Optic Coilwinder

by

Sieu Tru Duong

Submitted to the Department of Mechanical Engineering
on January 15, 1998 in partial fulfillment of the
requirements for the Degree of Master of Science in Mechanical Engineering

ABSTRACT

Fiber optic gyroscopes (FOGs) have gained greater acceptance in recent years in both commercial and military applications. FOGs have been shown to be an improved method of providing accurate navigational information and control in guidance systems as compared to conventional mechanical gyroscopes. However, the process used to manufacture FOGs, particularly with the winding of the optical fiber onto the product spool, has a very low throughput. Even with a semi-automated machine, it normally requires up to a week and the full commitment of a technician to wind one such coil.

Efforts are currently being made to fully automate the coil winding process and to increase the rate of production, as well as the quality of the winding on the product spool. However, the intricacies involved in the winding of the complex patterns make touch labor in the manufacturing process a necessity. In order to minimize the need to have an operator constantly monitor the winding process for errors, it was decided that a machine vision system could help mitigate the problem. The vision system would detect winding errors and provide feedback to the coilwinder machine, which would then correct the errors.

The focus of this research is to develop and implement such a machine vision system. The objective of the vision system is to capture an image as the fiber is being wound onto the coil, analyze the image to determine if an error has occurred, and feed the information back to the manufacturing process. The following documents the design, development, and deployment of the vision system.

Thesis Supervisor: Dr. Andre Sharon
Title: Executive Office of the Manufacturing Institute Laboratory

Acknowledgments

Dr. Andre Sharon, the air traffic controller. Thank you for your guidance from takeoff to landing, and for always keeping the project on track and on time. It is not every day that you find someone who would put a project in the hands of a novice. Thank you for giving me the opportunity to work on this great project.

Steve "Air" Lin and Brian "Sonnich-boom" Sonnichsen, the pilots who first took this project off the ground. It was great working with the two of you the first year. Thank you for sharing your knowledge of coilwinding.

Jesse "Simplicity" Darley and Min "O-K" Tsai, the copilots who continued in the great Manufacturing Institute tradition. It has been a blast having the two of you on the cookie crew. It was a relief to have gone through this past year-and-a-half with you guys. Jesse, go enjoy your river mist. Min, she's waiting for you in the weight room.

Pankaj "z-bomber" Lad, the one man pit crew. I owe you big for doing all the dirty work. Thanks for all your help with everything.

Leanna, Nancy, and Joan, the hospitable and lovely flight attendants who have made this trip a much easier one for all of us.

Fred Cote, the mechanic extraordinaire. What can I say that haven't already been said about the great work you are doing. The whole mechanical engineering department will miss you. Thanks you for all your help and patience in the machine shop.

And of course, the Mile High Club: Dave Roberts, Wayne Hsiao, Jeremy Neilson, Guvenc Sisman, Wes Williams, Dave Beal, Tang Tan, and Jason Melvin. It has been a pleasure flying with you all.

A big thanks goes out to the sponsors of this project. Thank you for all the latitude (and longitude) you have given me in completing this research.

I would like to thank all my friends, the mysterious flotation devices under the seat, who are always there if you need them, but hope you won't have to use. Phil, Rob, and Cheryl...this one's for y'all. I hope this proves that I didn't spend that last two years in m37. And Jeffrey, thanks for making these past four months bearable if not sane, and for your support as well.

Last but not least, I want to thank my family for being so patient and supportive through all of this. I'm finally coming home!

Up, up, up and away!!!

TABLE OF CONTENTS

1. INTRODUCTION 5
   1.1. What Is A Gyroscope 5
   1.2. How A Fiber Optic Gyro Works 7
   1.3. The Components Of A FOG 10
   1.4. The Drive to Automation 19
   1.5. Drive for Machine Vision 20
2. VISION BACKGROUND 22
   2.1. Machine Vision System Architecture 22
   2.2. Image Formation 22
   2.3. Image Acquisition and Processing 25
   2.4. Image Analysis 28
   2.5. Image Interpretation 29
3. VISION SYSTEM SPECIFICATIONS 30
   3.1. Different Winding Scenarios 30
   3.2. Type of Errors to Detect 30
   3.3. Performance Specifications 33
   3.4. Operating Assumptions 34
4. VISION SYSTEM HARDWARE 35
   4.1. Vision Hardware Description 35
   4.2. Computer System 35
   4.3. Frame-grabber 36
   4.4. Optics System 36
   4.5. Lighting System 37
   4.6. Vision Stages 39
   4.7. Mounting hardware 42
5. VISION SYSTEM SOFTWARE 44
   5.1. Vision Software Description 44
   5.2. Serial Communication 44
   5.3. Description of Vision System Algorithm 49
6. VISION SYSTEM INTEGRATION 60
   6.1. DOS vs. Windows 60
   6.2. Information to be Communicated 62
   6.3. When and Where to Capture Images 64
7. Conclusion 66
   7.1. Performance Evaluation 66
   7.2. Recommendations 68
8. Appendix A: Neural Network 73
   8.1. Theoretical Background 73
   8.2. Implementation 76
   8.3. Image Acquisition 76
   8.4. Image Preprocessing 77
   8.5. Multilayer perceptron model/training 80
   8.6. Results 82
   8.7. Conclusion 86
9. References 87

1. INTRODUCTION

1.1. What Is A Gyroscope

A gyroscope is a sensor that is used to detect rotation. It finds use in a wide range of applications but is most frequently used in navigation. From submarines to single-engine planes to satellites, gyroscopes make it possible for vehicles to accurately and reliably chart out their courses, either automatically or with the aid of an operator. Gyroscopes perform operations as simple as telling a plane which way is up or as complex as telling a satellite in near-zero gravity if it has deviated a thousandth of a degree off course.

One of the most recent applications of the gyroscope is associated with the Global Positioning System or GPS[1]. Cars and other vehicles can use GPS to find their position on the globe. With the aid of a computer, the locational information from GPS can be used to track a car's progress and help a driver chart a course. Communication with satellites is not without problems: it is expensive and doesn't work well when obscured by obstacles, such as mountains or tall buildings. By using a gyroscope and a simple computer instead of satellite communication for locational information, a vehicle can navigate in almost any set of conditions. The problem to date has been that gyroscopes have been too expensive to find use in consumer automobiles. New technology has made it possible to reduce cost of gyroscopes, enabling their use in GPS. A gyro would sense any turns made by the vehicle and a computer would keep track of the distances and directions traveled. By referencing this information to known maps and knowing a reference starting position, the computer can locate the vehicle on a map contained in memory.

Traditionally, gyroscopes have been mechanically based devices comprised of high precision parts that spin and then twist in response to rotations of the gyroscope[2]. Exactly how these parts turn inside the sensor is well defined by modern dynamic theory. Essentially all mechanical gyroscopes operate on the same basic principle: conservation of angular momentum of a spinning mass. For a discussion of the dynamic theory of a basic mechanical gyroscope, see Crandall[3]. By measuring the torques and forces generated as certain internal parts of a gyroscope move, the rotational motion of the entire gyroscope can

be deduced. Different gyroscopes accomplish these measurements in different ways, depending on the application and accuracy grade of the sensor, but all give a measurement of the rotation of the sensor.

There are two significant disadvantages to using mechanical gyros. The first is that they are relatively expensive to produce, especially the highest accuracy varieties. The high precision components and accurately machined parts used inside these gyros are inherently expensive to produce[2]. Though great effort has been expended to cost-reduce these parts, the parts are still difficult to manufacture and remain an impedance to cost-reducing mechanical gyros. In addition, the electronics and sensors in the gyros add expense due to their high accuracy and low noise requirements. All parties interested would like to see a less expensive gyroscope, but the current cost of the components of the mechanical sensor makes this virtually impossible.

Another drawback of mechanical gyros is that they contain moving parts which are particularly subject to failure over time. Eventually, a bearing may wear out, a part of the structure may fatigue and break, or something could shift out of alignment. Either way, the gyroscope would become useless and would have to be replaced. On vehicles such as satellites, this is not easy and could lead to the destruction of the satellite. If moving parts could be eliminated altogether, these problems could obviously be avoided.

Optical solid state technology has been shown in the last few years to be a viable substitute to the traditional mechanical technology used in gyroscopes. These solid state gyros incorporate no moving parts, relying on light traveling through an optical pathway to detect rotation. There are several ways that the optical pathway can be created. Ring Laser Gyros (RLGs) and Fiber Optic Gyros (FOGs) use two of the possible methods. As suggested by its name, the RLG uses a set of accurate mirrors arranged in a ring around which the light can travel. The FOG uses optical fiber to define the light path. For reasons of greater possible accuracy and lower cost, the FOG seems to be the most promising alternative[1].

FOGs can reach, and possibly exceed the precision of current mechanical gyros and can operate under equal, or even harsher environmental conditions. The drawback to date has been the expense of building these solid state gyros. Only recently has the state of technology begun to progress enough to bring the cost of manufacturing the gyros down to competitive levels. Optical fiber has become less expensive and methods of interfacing the fiber with other components in the gyro have become well

defined. Likewise, the lasers, which act as the light source for the sensor, have recently become far more affordable.

The main factor currently driving the price of manufacturing solid state gyroscopes is assembly time[4]. It can take one person up to three weeks to assemble a working, three-axis gyroscope. If this assembly time can be reduced, through automation and other techniques, the cost of a solid state gyro can be brought significantly below that of a mechanical one. Given that they have no moving parts and could be made inexpensively, the solid state gyroscope could prove to be a superior alternative to its mechanical cousin.

1.2. How A Fiber Optic Gyro Works

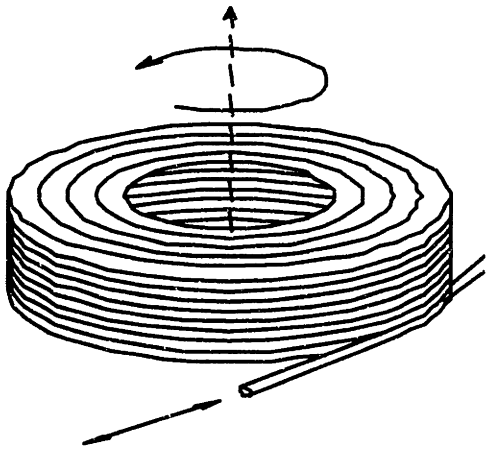


Figure 1.1: A representation of a FOG coil.

The basic theory behind the FOG is fairly simple. Consider a long strand of optical fiber that is wrapped into a coil as shown in Figure 1.1. The fiber is wound from its midpoint outwards so that half the fiber is wound counterclockwise around the coil and the other clockwise. The two ends of the fiber will then be located on the outside of the coil as shown in the figure. Light that is in phase is passed through both ends of the fiber coil. If the coil remains stationary, light coming out of the two ends will be in phase. However, if the coil turns along its axis, as indicated in Figure 1.1, light traveling through the coil in the direction of the rotation will take longer to travel the fiber length than the light traveling against the direction of rotation[5].

A physical example may prove useful in understanding this concept. Consider two balls that are rolled along a plank of a given length from opposite ends, towards each other at equal speeds as shown in Figure 1.2. The balls are rolled along the two edges of the plank so that they don't collide. If the plank is held stationary, the balls will take the same amount of time to reach the ends of the plank. If, on the other hand, the plank is moved towards one of its ends while the balls are moving, the ball traveling in that direction will take longer to reach the end than the ball traveling in the opposing direction. The time difference between when the two balls reach the ends of the plank is related to the velocity of the plank.

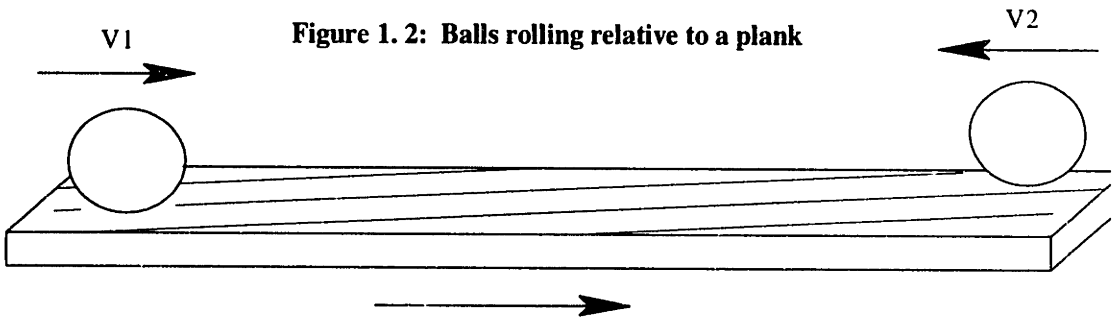
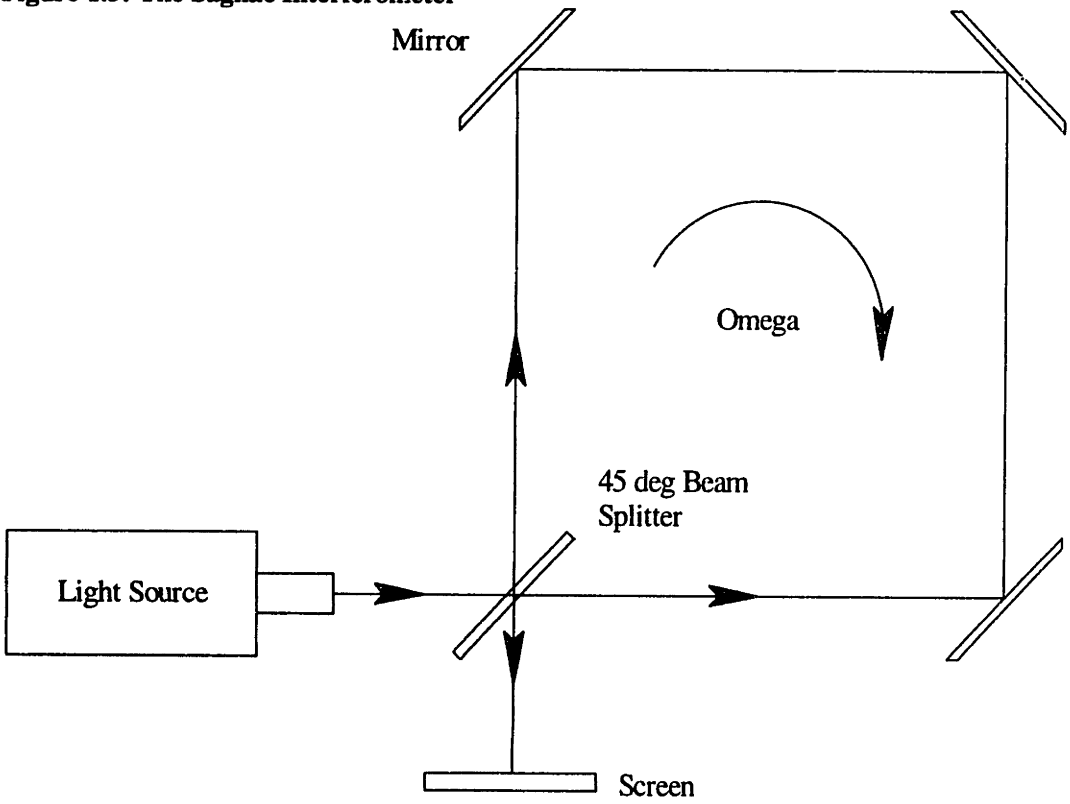


Figure 1. 2: Balls rolling relative to a plank

This is a similar concept to that employed in the fiber optic gyroscope except that light takes the place of the balls and optical fiber takes the place of the plank. The primary conceptual difference is that the fiber is wound into a coil so that rotation rather than translation is measured. The light emerging from opposite ends of the coil can be recombined and measured for phase shift. If the coil has experienced rotation, the light traveling in opposite direction will take different times to traverse the length of the coil and will be shifted in phase when they emerge. This shift is directly related to the rotation rate of the coil.

In 1914, a French researcher by the name of Sagnac first proposed using this concept of phase shift for measurement[6]. He created a simple device consisting of bright light, a 45° beam splitter, and a ring of mirrors. With this device, he was able to successfully prove the above described concept, which is now commonly known as the Sagnac Effect. His device, called a Sagnac interferometer. is shown in Figure 1.3. The entire device was rotated at a velocity Omega and the combined light was shown onto a screen. A fringe pattern was seen on the screen which corresponded directly to the rotation Omega. About sixty years later, in 1976, Vali and Shorthill proposed and implemented the first optical fiber gyroscope[7]. Since then, the technology has grown to the point where FOGs are as viable as their mechanical counterparts.

Figure 1.3: The Sagnac Interferometer



A few, simple mathematical relations are enough to entirely describe the Sagnac effect and relate the relative phase shift of the light to the rotation rate of the coil. When the light beams emerging from the coil are recombined, the beams will interfere, and there will be a power loss corresponding to the degree of phase shift between the beams. This relationship is described by the following equation:

$$P = \frac{1}{2} P_0 (1 + \cos \Delta\phi)^{6} \tag{1.1}$$

where P is the detected output power, P_o is the power input to the coil, and $\Delta\phi$ is the phase difference. By measuring the power output with a photodetector, and knowing the power input, the phase shift can be deduced. The rotation rate is related to the phase shift of the coil through the following relation:

$$\Omega = \frac{2\pi LD}{\lambda_0 c} \Delta\phi^{6} \tag{1.2}$$

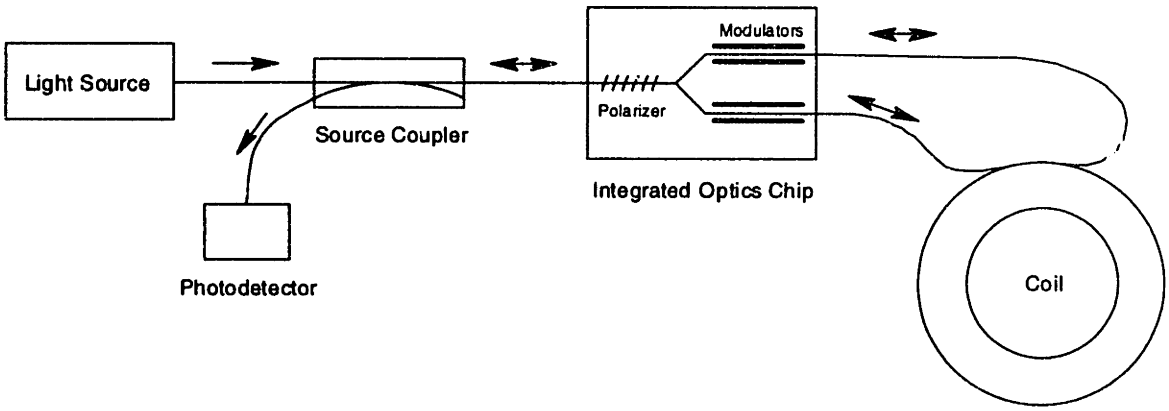
where λ_o and c are the free-space wavelength and velocity of light, respectively, L is the length of fiber, and D is the diameter of the coil. $\Delta\phi$ is the phase shift, as per equation 1, and Ω is the rotation rate of the

coil. Thus, from a measurement of the power output from the recombined beams that have passed through the sensing coil, a measure of the rate of rotation of the coil can be made. The majority of fiber optic gyroscopes use this basic concept in sensing rotation.

1.3. The Components Of A FOG

A diagram of a FOG is presented below.

Figure 1.4: Diagram of Fiber Optic Gyroscope.



Each of the major components of the gyro can be seen in Figure 1.4. These components are the light source, source coupler, integrated optics chip (IOC), coil, and photodetector[2]. Each of these components will be discussed below.

1.3.1. The Light Source

The laser is the light source for the gyro. It provides a constant, coherent beam of light that drives the Sagnac Effect. Laser diodes are commonly used in FOGs as they are small and exhibit adequate performance. To meet power requirements, these diodes are coupled with a rare-earth doped fiber that acts to amplify the beam. Together, the diode and doped fiber act to generate the light beam for the gyro.

1.3.2. The Source Coupler

The source coupler serves the basic function of routing the returning light beam from the coil to the photodetector. The coupler also allows for light to travel from the light source towards the coil. In order to accomplish this, two fibers are fused together along their sides and are reinforced for durability. A fiber lead coming from the laser attaches to the end of one of the fibers and the lead from the coil is attached to the other end of the same fiber. The photodetector lead is attached to the end of the other fiber in the coupler. Finally, the last coupler fiber end is left unattached.

1.3.3. The Integrated Optics Chip (IOC)

In order to split the light beam as it enters the coil, a special beam splitter must be used. This splitter also serves the purpose of recombining the two light beams emerging from the coil. While there are various forms of splitters available, a special type called an Integrated Optics Chip (IOC) is commonly used in gyroscopes. Light pathways are formed within a specially designed chip to efficiently guide the light between the single fiber and the two coil fiber ends. The IOC looks much like a rectangular piece of glass with the fibers attaching to the shorter ends of the rectangle. Where the IOC becomes unique from other beam splitters, is in the phase modulator contained within the IOC. The modulator acts to modulate the frequency of the light beam which aids in the operation of the gyro and significantly increases their accuracy. The IOC also acts to polarize the light beams which also increases operating efficiency.[11]

1.3.4. The Photodetector

In order to detect the phase shift in the light emerging from the coil, two different types of photodetectors are commonly used. Both the PIN and Avalanche Photodiodes operate on the principle of photons of light creating free electrons upon contact with a specially designed surface. The surface is connected to an electrical circuit and the current is measured. By the nature of the interaction, current is proportional to light intensity; thus, providing a measure of the phase shift in the light beams. For various reasons, the PIN variety is the most common in FOGs.[11]

1.3.5. The Coil

The fiber optic coil has the most bearing on this thesis of all the FOG components as the goal of the project was to build a coil winding machine. Therefore, the many unique features of the coil will be covered in much more detail than the other components. Many of the references cited thoroughly discuss the other FOG components.

Among the many factors contributing to FOG performance, the quality of the sensing coil is one of the most important. The Sagnac Effect relies on the fact that the shift in the phase of the light beams is due solely to the rotation of the coil. However, there are other factors that can create this shift. These factors act to change the local index of refraction of the optical fiber. Changes in the index of refraction will locally change the velocity of light in the fiber. And, changes in the velocity of light could lead to an apparent shift in phase of the two propagating beams. Temperature variation and stress are the two most significant factors that can create these local changes in index of refraction. Coils are rigorously tested under thermal variations and severe vibration when determining the performance of the coils. A common measurements called drift, usually expressed in degrees per hour, corresponds to the rotation sensing error over time. The best coils have better than .0005 deg/hr of drift, termed strategic grade, the next best have .05 deg/hr, termed navigation grade, and the crudest coils have 1 deg/hr, termed tactical grade. With current technology, tactical grade coils are quite easy to wind[2]. Navigation grade coils are commonly produced but at high costs and at low production rates. Strategic grade coils can be wound but require a great deal of care and are highly labor intensive. To be used for navigation applications, the gyros must be of navigation grade or better to avoid excess error in the sensor readings.

Different approaches can be taken to deal with these environmental variations. The coil could be carefully insulated and thus protected from thermal and vibration effects. This approach has been tested and works quite well. However, this type of insulation is bulky, and it is desirable to keep the gyroscope as small as possible in most applications. With this in mind, the problem then must be solved by altering the winding of the coil. The approach has been to create patterns of winding that achieve maximum symmetry within the coil. What is meant by symmetry in this case is that points in the fiber equidistant from its midpoint are located adjacent to each other within the coil. By doing this, the light traveling in opposite directions will see the same variations at the same time as it passes through the fiber. Hence, any change experienced by one of the beams should be virtually identical to that of the other, and when recombined, the two propagating light beams should only reflect phase shifts due to rotation.[8] Figure 1.5 provides a clarification of this concept. Consider that the fiber is only folded over once, as shown, rather than wound into a coil. If a temperature or vibration disturbance occurred at point x, light beams A and B traveling in both directions would experience this disturbance at the same time and at the same distance from each end of the fiber. Thus, the light beams emerging from the two ends of the fiber should exhibit the same phase shift caused by the disturbance. The two beams would therefore emerge in phase.

To wind the fiber with such symmetry, the coil must actually be wound from the inside out. More specifically, winding must start with the midpoint of the fiber and proceed outwards towards the fiber ends. Half the fiber will be wound clockwise around the coil and the other half counterclockwise. This will lead to a coil that has the fiber ends exiting the coil on the outside diameter and in opposite directions, as shown

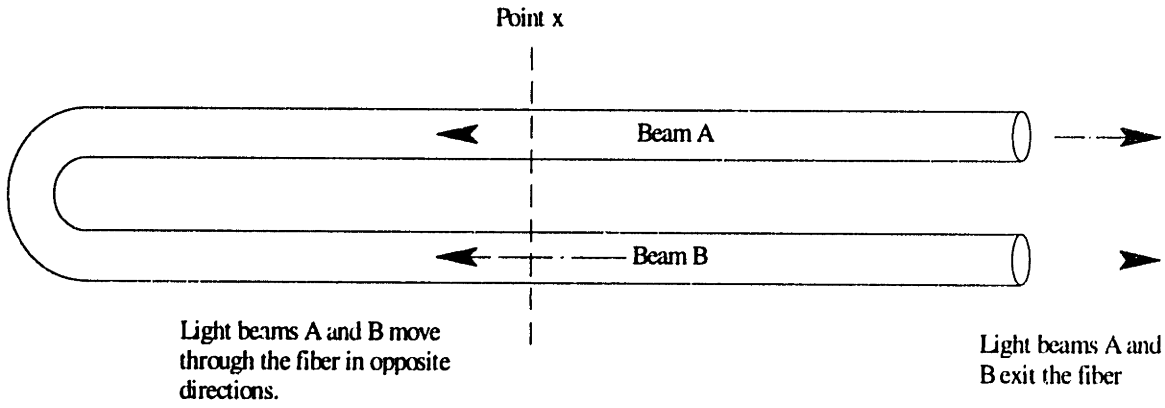


Figure 1.5: Demonstration of symmetry concept.

in figure 1.6. To actually wind a coil this way, a length of fiber must first be wound half onto one supply spool and half on another, with the midpoint of the fiber between the supply spools.[5] During winding, the supply spools will pay out fiber onto the product spool, which holds the coil. At any one time during a wind, one supply spool will be actively paying out fiber while the other is locked in place, rotating as the product spool rotates. An example sequence of winding is shown in Figure 1.7. Note the two supply spools containing half the fiber and the inactive and active positions for each spool.

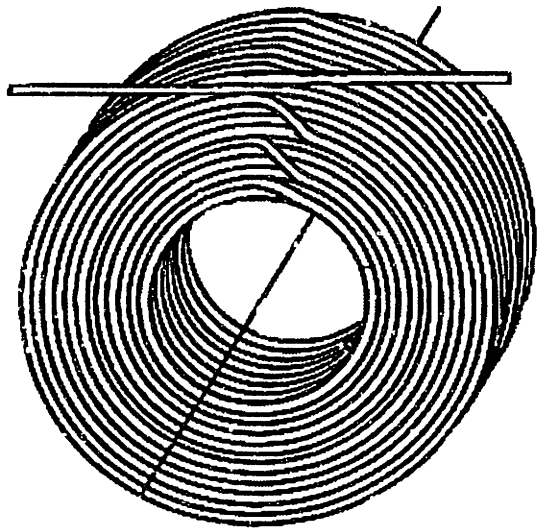


Figure 1.6: A FOG Coil.[9]

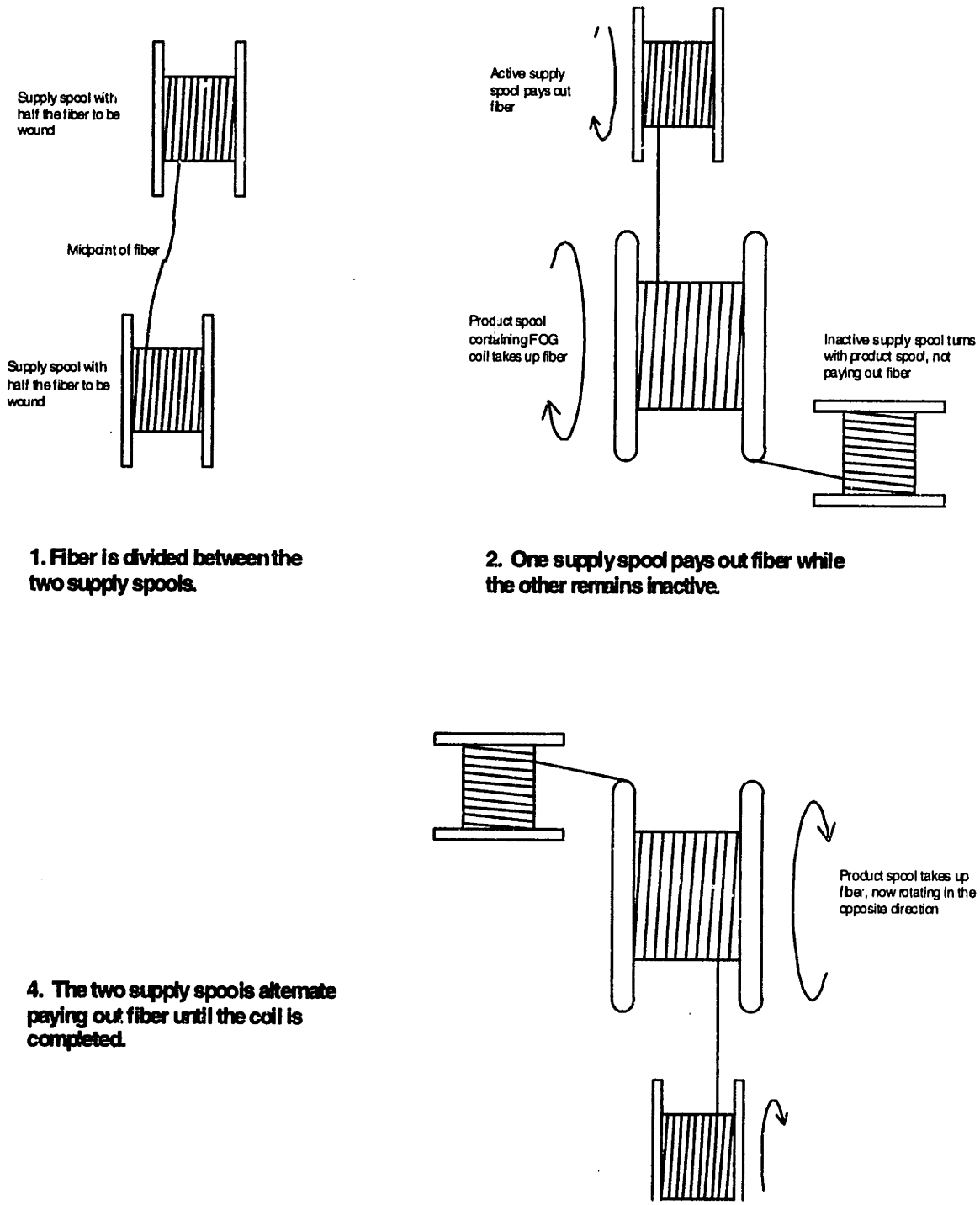


Figure 1.7: Winding Sequence

Keeping in mind that coils are wound in this inside-out manner, there are various patterns of fiber placement that can result in coil symmetry, some exhibiting better results than others. The most basic of these patterns is termed the "quadrupole." Figure 1.8 depicts a cross-section of a coil wound with this pattern. The white circles indicate fibers wound in one direction and the black indicate those fibers wound

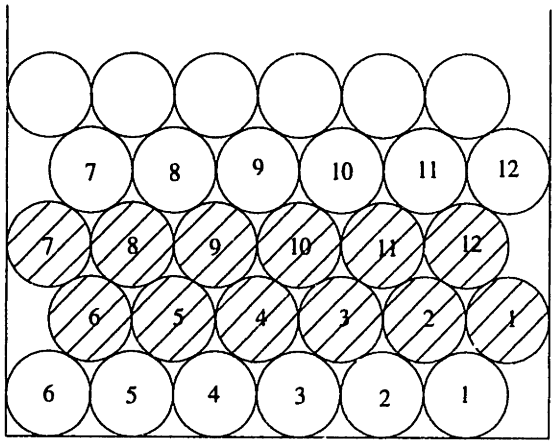
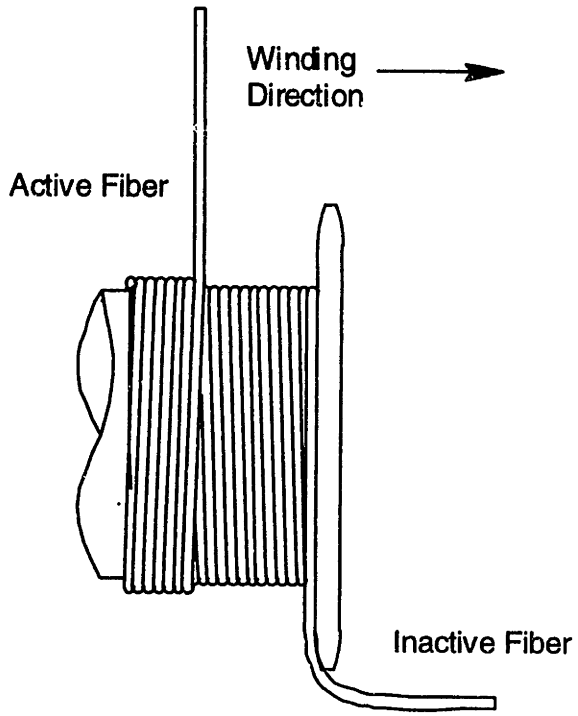


Figure 1.8: Quadrupole Cross Section

in the other. By examination, it can be seen that corresponding turns of the black and white fiber are in very close proximity within the fiber pack. Also note that fibers are placed into the grooves formed by the underlying layer of fiber. This helps in guiding the fiber and achieving a tighter pack. Variations on this basic pattern yield better symmetry which have been shown to improve coil performance. Coils that are wound without attention to this symmetry exhibit far inferior performance.

Regardless of what pattern is used in a coil, local areas of stress within the pack caused by poor winding can be quite detrimental to coil performance. Examples of poor winding would be lumps and voids within the fiber pack and severe variations in tension. Essentially any characteristic of the coil wind that can cause varying pressures on the fibers can lead to variations in index of refraction. What this means is, to obtain optimum performance, the coil must be wound as evenly as possible with minimal variations in tension. Not only must the tension be consistent, it must be low enough not to cause undue stress in the glass of the fiber. It has been shown that coils wound with low tensions yield superior performance[10]. These facts lay out the most basic of restraints for any coil winding machine: it must be able to accurately place fibers during a wind and it must be able to accurately control tension.

Figure 1.9: Helical Winding Technique



To aid in achieving an even wind, a specific winding technique is commonly employed. The simplest way to wind is to place the fiber in a spiral, or helix, allowing it to traverse across the fiber pack naturally as it is wound onto the coil. Fiber follows the groove formed by the pervious layer when traversing in the direction of the helix and crosses those grooves when traversing against the helix as shown in Figure 1.9. The problem is that the fiber can jump to the next groove at any location around the circumference of the coil. The circumferencial location of the fiber can differ from turn to turn. It is easy to see that this phenomenon can cause inconsistency on the current layer, which can only get worse on progressing layers. An example of a cross-section of a coil wound in this way is shown in Figure 1.10. Dark circles correspond to fibers wound in the clockwise direction and light circles correspond to fibers wound counter clockwise. Note the uneven placement of the fibers which can lead to localized stresses. And, these stresses can severely differ from fiber to fiber. Needless to say, coils wound in this way have poor performance.

One alternative is to wind coils in an orthocyclic manner. This winding method was first developed by Phiilips Industries in 1962 and is in common use today.[11] With this method, rather than winding in a helix, fibers are wound in concentric hoops and cross over to adjacent grooves formed by the under-lying layer in well-defined zones. See Figure 1.11 for an illustration of this type of wind. The zone where the fibers cross grooves is commonly termed the "jog zone" and is the only location in the coil where fibers deviate from a straight orientation. Many advantages are gleaned from orthocyclic winding. Local stresses due to crossing over are limited to one small area of the coil. Furthermore, adjacent fibers in this zone will experience similar stresses at the same locations, and because of the symmetry of the coil, this

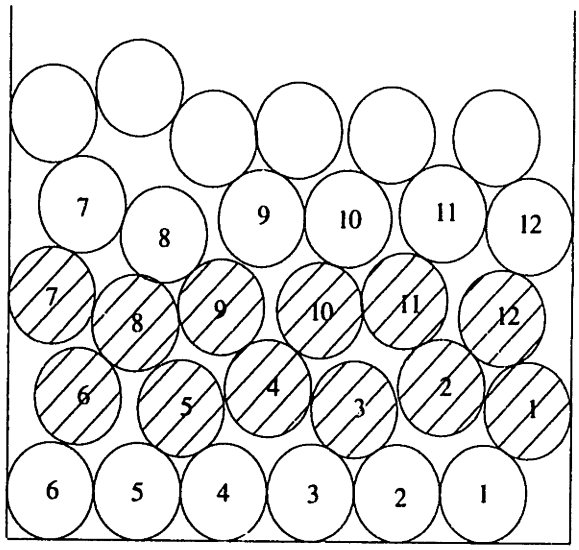
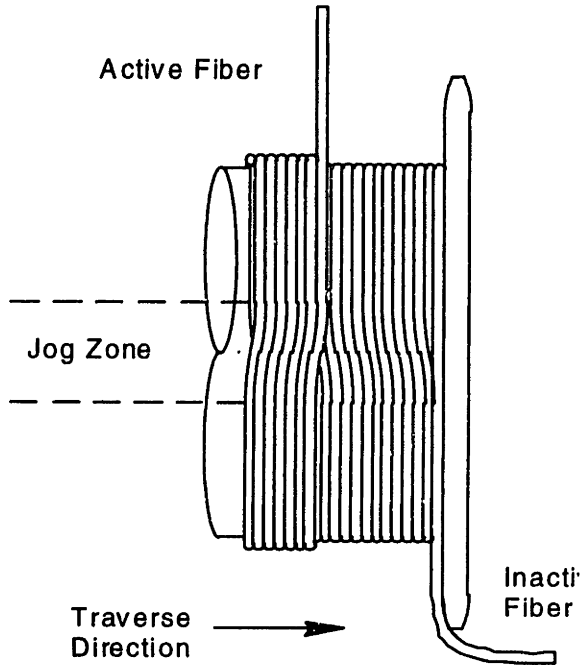


Figure 1.10: Helical Wind Cross Section

should not effect the relative phase of the light emerging from the coil. Another benefit is that the majority of the circumference of the coil is wound with an even, closely packed profile much like that shown in Figure 1.8. Any inconsistencies should then occur in the tight jog zone. Also, the close pack inherent to the orthocyclic wind has been shown to improve thermal variation and vibration resistance of the coil.

Figure 1.11: Orthocyclic Winding



1.4. The Drive to Automation

In review, for optimum performance, coils are wound in specific patterns in an orthocyclic manner, starting from the center of a length of fiber and winding outwards to the ends. Fiber placement and tension control are known to be important factors in winding a high performance coil. Any machine that is created to wind coils must be able to deal with many variables. It must accommodate the different patterns and allow for orthocyclic winding. It must be able to very accurately guide fiber and maintain a constant, low tension. To be successful, it also must allow for fiber diameter and hardware variations. One of the greatest challenges is to be able to wind an entire coil with few or no errors. An error is considered any place in the coil where there is a deviation from the pattern, such as a gap. These errors could degrade the performance of the coil and should be avoided.

The current state of production is generally semi-automatic, requiring the presence of an operator. The operator monitors the winding of a coil at all times and interferes when an error occurs. Generally, the operator would use a small tool to manipulate the fiber into the correct place. With the current machines and control, these errors actually happen quite often and greatly slow the winding times. In addition, with most machines the operator must switch the two supply spools manually between their active and inactive

positions (see Figure 1.4), which takes time and adds risk. Any time the operator must interact with the fiber, there is the possibility of damaging, or even breaking, the fiber. Winding in this manner has resulted in relatively slow winding times, up to a week in many cases, and also yields a significant rejection rate.

There is a definite opportunity in the industry for improvement of the coil winding process. If FOGs are to become competitive, the cost of coils must be reduced without sacrificing quality. To accomplish this goal, winding times must be decreased and rejection rates lowered. This implies that operator intervention must be removed from the process, or at least minimized. In the past, that was not a feasible possibility as the process of coil winding was poorly understood and it was difficult, if not impossible, to design a machine that could account for all the variables. Through experiments that have been completed in recent years, the process of coil winding has become somewhat better characterized and it has become easier to predict the behavior of the fiber during a wind. These experiments have also made it possible to place more accurate specifications on parameters that were previously loosely defined. Along with this improvement in process understanding, the state of technology has greatly improved, facilitating the development of machine automation.

Given that the process and technology are in place, and the opportunity is present, there is a definite drive to automation. It is now possible to design and build a machine that can automatically wind coils and, in the end, reduce the cost per coil. Details on the design and development of an automated coilwinder can be found in Brian Sonnichsen's Masters Thesis entitled "The Design of an Automated Fiber Optic Coil Winder"[12] and Stephen Lin's Masters Thesis entitled "Design and Development of an Automated Fiber Optic Gyroscope Coil Winding Machine"[13].

1.5. Drive for Machine Vision

Although automation may eliminate the need for a technician to run the coilwinder, an operator is still required to monitor the process for errors. Thus, the drive for automation will not be fully complete without an additional element that would oversee the process to insure product quality. The intuitive solution to the problem is to therefore implement a machine vision system.

The drive for a machine vision system is twofold. First, the vision system will provide feedback to a coilwinding process which, like most manufacturing processes, is not perfect nor is free of errors. The

purpose of the vision system is, therefore, to monitor the process for these errors such that corrective measures can be taken and quality can be ensured. The second reason for implementing a machine vision system is to make possible the full automation of the manufacturing process. This would allow for the overall control of the manufacturing process by the computer, thereby eliminating the need for a human operator of having to sit in monotony next to a machine. If the vision system were to be successfully integrated with the main manufacturing process, then the process rate would also increase as well. Corrective measures could be programmed into the software for each error type rather than having an operator spend time determining how to correct those errors.

2. VISION BACKGROUND

2.1. Machine Vision System Architecture

A vision system consists of four main sub-systems: image formation, image acquisition and processing, image analysis, and image interpretation.[14] Image formation creates an environment for the object that is suitable to be captured and analyzed. This subsystem consists of such components as the object under inspection, the camera and optics assembly, and illumination. The next subsystem then converts the analog representation of the image from the camera into a digital format via a framegrabber and image processing board. This subsystem does the appropriate preprocessing and processing of the image before it is transferred to the next subsystem, image analysis. The task of the image analysis subsystem is to extract characteristic features from the processed image, usually making use of object size, position, and orientation. The image interpretation subsystem then makes a pass/fail decision based on the extracted information and the inspection criteria. This subsystem then feeds back the information to the control of the manufacturing process.

2.2. Image Formation

The first step in implementing a machine vision system is image formation. There are several factors that influence the quality of image in the initial set up of the visual environment. Those variables which most affect the image: illumination, optics, and camera, are discussed in this section.

2.2.1. Illumination

One major influence of image quality is illumination. Illumination is often times not uniform due to variations in the environmental lighting. There are, however, different lighting methods available that can be used to reduce those variations so as to produce the best possible environment for extracting information from an image. One type of illumination, front lighting, is often used to highlight surface details and features by shining the light source onto the object. The camera is situated on the same side of the object as the light source and thus captures the light reflected off the surface of the object. However, this method tends to cast strong shadows that often pose a problem for the vision system. The use of diffused lighting

as opposed to a point source, in this case, will help to reduce the shadowing effect. Backlighting is another illumination option where the object is located between the light source and the camera, thereby producing a silhouette of the object. Frosted glass is usually placed over the light source to create diffused light and thus provide a high-contrast outline of the object's boundary as it appears opaque against a white background. The third method of illumination uses strobe lighting as a way to reduce variation in the ambient lighting. The short pulses of light are also useful for capturing images of objects in motion, since it reduces image blurring by making the objects seem to appear to be stationary. However, this illumination option might prove disruptive, if not distracting, to humans who must view it on a regular basis. The fourth lighting alternative is the use of structured lighting. In this method of lighting, special patterns or grids are projected onto an object which help to extract the 3D features in a 2D image. The four illumination methods are shown in Figure 2.1.

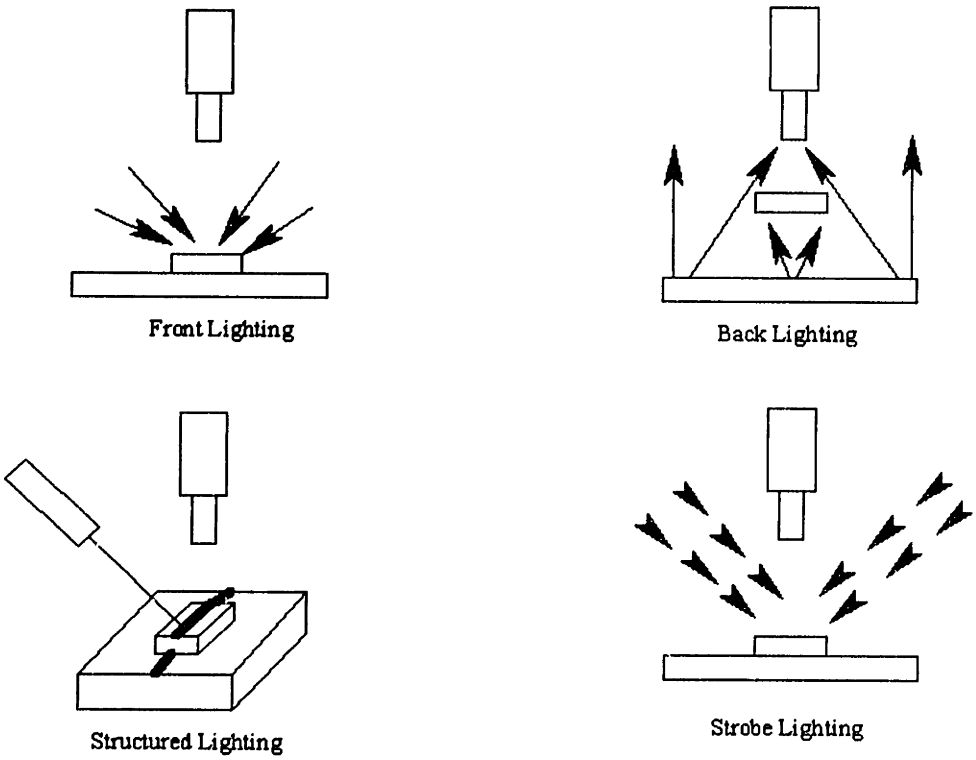


Figure 2.1: Four Types of Illumination Options[15]

2.2.2. Optics

The optics, usually determined by the type of lens and lens settings used, is another factor influencing the image formation. The optics not only controls the clarity of the image but also the size of the object to be analyzed as well. The size of the image is determined by the magnification, m, which sets the ratio between the image size to the size of the actual object

$$m = \frac{S_i}{S_o} = \frac{D_i}{D_o} = D \tag{2.1}$$

where S_i is the size of the image, S_o is the size of the object, D_i is the distance from the image to lens, and D_o is the distance from the object to the lens. More commonly, a focal length, f, is often used to characterize these two settings and is defined as

$$\frac{1}{D_i} + \frac{1}{D_o} = \frac{1}{f} \tag{2.2}$$

Substituting for the magnification, m, the focal length can alternately be expressed as

$$f = \frac{D_o}{(1 + \frac{1}{m})} \tag{2.3}$$

Typically, the focal length and magnification will also determine the 2D viewing area that is captured. This area is referred to as the field of view. A third optical feature, also directly related to the magnification, is called depth of view. The depth of view is the vertical distance that an object can move towards and away from the camera and still have the object of interest be in focus. The depth of view is defined as

$$D_f = \frac{2ap\,(m + 1)}{(m)^2} \tag{2.4}$$

where a is the size of the lens aperture and p is the pixel size.

These optical considerations should be determined during the selection of a lens system. Physical space constraints such as where the camera/lens will be located with respect to the object will determine the focal length of the lens. The size of the object in an image required will decide on whether additional lenses are needed to provide higher magnification. The calculation of the depth of field will provide a good indication of how robust the lens system will be in providing a clear image if the object were to move along the axis of the camera lens.

2.2.3. Camera

A final factor influencing the image formation process is the type of camera used to view physical object. There are basically two types of cameras, the vacuum tube camera or the more commonly used solid state CCD (charge-coupled device) camera. The vacuum tube uses a light sensitive material for image sensing. A charge density, proportional to the duration of the light flux hitting on the surface, is collected on the surface of the light sensitive material. The surface is then scanned to produce an analog voltage signal as output. The advantages of this type of camera device are that it offers high resolution and high sensitivity. However, these devices lack the fast response speed and may produce image burn, geometric distortion, or drift in capturing the image.[16] The more widely used CCD camera does not exhibit the same problems as their vacuum tube counterpart. CCD cameras also offer high speed response time and good image resolution, and thus prove to be a better alternative to the vacuum tube camera. CCD cameras capture an image through separate photodetectors, located at each pixel area, which serve as light sensors. The intensity of light impinging on the photodetector will be reflected in the grayscale value of the pixel.

2.3. Image Acquisition and Processing

With an adequate environment created in the image formation step, the image acquisition and processing subsystem then takes the visual image and transforms the details of a physical object into a set of digitized data. First, the image is taken from the camera in its analog form and passed through a frame grabber, thereby converting the information to a digital format that is compatible for use by the computer. Once in digital form, the image is processed to make it easier to analyze by accenting characteristic features.

2.3.1. Frame grabber

One of the first considerations in the image acquisition process is to select an appropriate framegrabber. The choice of framegrabber will determine the resolution of the image to be captured and should be coordinated with the resolution of the camera. The resolution is the number of pixels that make up an image given a certain field of view and magnification. Some typical image resolutions include 512x480 and 640x480 pixels square. Framegrabbers also have the option of being able to capture images in color, grayscale, or black/white. The larger histogram range of the color and grayscale images will provide for greater detail but will complicate the image processing if the processing were to be done on a pixel by pixel basis. The third factor in considering framegrabbers is the processing speed, often rated in terms of

frames/second. If the framegrabber is to be used for real-time application then a high processing speed is usually recommended. Otherwise, a framegrabber with moderate processing speed will be sufficient.

2.3.2. Image Processing Techniques

Once an image has been converted into digital form, it can now be processed by the computer. There are several techniques, each varying in scale, which can be used in the processing of the images. The first method is point by point processing where each pixel from the original image is mapped onto the corresponding location in the new image. An example of this method is binary inversion where each pixel, if white, is transformed to black and vice versa. Thresholding of an image also implements this technique by setting a pixel to black if it falls below a set value and white if it falls above that same value. Images can also be added to or subtracted from one another to produce a resultant image. Image addition is often used to average out and suppress noise in an image. Image subtraction, on the other hand, is often used to remove known objects or background from an image to see how the object has changed from a previous state (i.e. an object in motion).

A second method of image processing uses a regional approach where the pixel value in the new image is calculated using the values of the neighboring pixels in the region. One popular implementation of this processing technique is filtering where the eight adjacent pixels are used to calculate the new value of the center pixel. Filtering is useful in removing noise or enhancing features, such as edges, in an image. Another popular implementation of the regional approach is the family of functions consisting of skeletonizing, erosion, and dilation. Skeletonizing is the process of obtaining a minimal, or skeletal, representation of an object in an image by iteratively removing neighboring pixels that make up the object.[14] Erosion can be thought of as a single iteration of the skeletonizing operation. Dilation, the inverse operation of erosion, will add neighboring pixels to the existing object.

The third method of image processing uses mathematical operations. Using vector algebra, an image may be translated from one set of coordinates to another. The image could also be rotated to fit another orientation or mirrored along an axis to produce a new image. These processing operations basically map a set of pixel coordinates from the existing image to the new image. Mathematical operators such as AND,

OR, and NOT are also used in this method to alter the image. Erosion and dilation are two examples of using these logic operators where in the first case, the single boundary layer is removed from an object while in the second case a boundary layer is added to an object.

2.3.3. Segmentation

This section will describe in detail the use of the above mentioned techniques in carrying out one of the more important functions in image processing, segmentation. Segmentation is the process of identifying and grouping regions exhibiting similar characteristic features. These regions can typically be distinguished by the pixels gray-scale values or by the different textures exhibited in the image.[14] What results from this process is the creation of boundaries and edges that separate the different regions. It is ideally used to distinguish the objects in the image from the background. Two widely used methods to accomplish segmentation are thresholding and edge detection.

Gray-scale thresholding basically sets the pixels of an object to a value of 0 while the pixels of the background to a value of 1 simply based on the pixel values of the two regions in comparison to the threshold value. The main problem is selecting the appropriate threshold value that will differentiate the object from the background. If the image is bi-modal, one mode representing one end of the histogram for the object and another for the background, then it is a matter of selecting a threshold value in between the two modes. If the image is uni-modal, then additional steps might need to be taken. The image would first need to go through an edge detection operation. Then the pixels near the edge are then averaged to serve as the threshold value. [14]

The edge detection operation, however, is a complex task in and of itself. Two of the more popular approaches to the edge detection process are gradient/difference-based operator and template matching. With the gradient/difference-based operator, the rate of change in the intensity is measured and used to locate edges. The operator assumes that an edge is where there is a large transition in intensity values between two adjacent regions. Thus, if the gradient value (rate of change) is greater than some threshold, then an edge is said to be found. In the template matching approach, edge templates of different

orientation are used to locate edges by matching them at different points in the image. The intensity of the edge is determined by the degree of correlation between the template and the image.[14]

A similar approach to the edge detection, called contour following, is often used to trace the boundary of a particular region. The algorithm starts at a point that is believed to be on the boundary and traces along the positive (+90°) and negative (-90°) directions, both normal to the gradient, based on the surrounding pixel values. The algorithm continues to trace the along the boundary following the maximum gradient path. The algorithm stops if it traces over the original point or if no continuing edge is found.

2.4. *Image Analysis*

Once an image has been processed it can then be analyzed for its characteristic features. Three popular methods used for image analysis include template matching, statistical pattern recognition, and the Hough transform. In template matching, a template of an object is used to locate similar objects in an image. The orientation, both translational and rotational, of the template is varied at each location in the image to look for possible correlations. However, there must exist a template for each different orientation of the object, which makes this approach very computationally intensive. Thus, this method is used mainly in applications where the number of objects and the variations in orientation of the objects in the picture are small.[14]

A second analysis approach, statistical pattern recognition, uses a three-step recognition procedure. First an object is isolated, using a segmentation technique as explained in the previous section. The second step is to extract the characteristic features from the image. Features which are commonly used to differentiate classes of objects include: size, expressed in terms of the objects length, width, perimeter, or area; and shape, measured in terms of its circularity or rectangularity.[14] Then these objects are classified, based on those extracted features, into a predefined set of objects.

The third method of image analysis is called the Hough transform. This method is ideal for isolating curves of a given shape, typically lines, circles, or ellipses in an image. The curves are represented by parametric equations, making this approach very tolerant of slight discontinuities in the boundary of the curve or other

noise corrupting the image.[14] The Hough transform could also be generalized to recognize any curve, which may not be easily be described by an equation, via a look up table. The table would correlate the Hough parameters with a set of boundary coordinates and orientation such that the curve could be traced out.

2.5. *Image Interpretation*

The purpose of this subsystem is to carry out the decision making based on the information extracted from the image analysis procedure. Typically, this step may simply consist of a pass/fail decision based on whether or not a part has satisfied the specifications. In other cases, measurement values may be taken of objects found in the image. In any case, the information from this step of the vision system is typically fed back to control the inspection process or the manufacturing process.

3. VISION SYSTEM SPECIFICATIONS

3.1. Different Winding Scenarios

There are two types of winding configurations that make up a coil. First, there is the continuous winding where the fiber is wound onto consecutive grooves on the coil. The other configuration is the alternating winding where the fiber is first wound onto every other groove (referred to hereafter as alternating A) with the empty grooves being filled in on the second pass (referred to hereafter as alternating B). The combination of these winding configurations makes up the complex winding pattern for a coil.

3.2. Type of Errors to Detect

In the ideal case, no errors in the winding will occur as shown in Figure 3.1. The winding configurations, as mentioned in the previous section, can however give rise to basically four types of errors that the vision system needs to identify and classify. Names are given to these errors which include: gap error, climb error, anti-gap error, and rise error. The first two error types are typically found in the continuous winding mode. A gap error occurs when the fiber skips over one or more grooves leaving a gap in the present layer as shown in Figure 3.2. The second type of error is the climb error, shown in Figure 3.3, where the fiber winds on top of the layer it is presently winding rather than continuing on the current layer. Two other types of errors are specific to the alternating winding mode. The counterpart for the gap is the anti-gap error, pictured in Figure 3.4, which occurs with alternating A winding. Instead of skipping a groove as it is suppose to, the anti-gap winds onto the adjacent groove as if it was performing a continuous wind. The fourth error, rise error, is somewhat similar to the climb error but occurs only with alternating B winding. A rise error, as shown in Figure 3.5, is the winding condition where the fiber is wound to fill in the spaces left over by the alternating A but does not completely fall into the gap. The fiber instead sits on top the two adjacent fibers and is situated above the current layer.

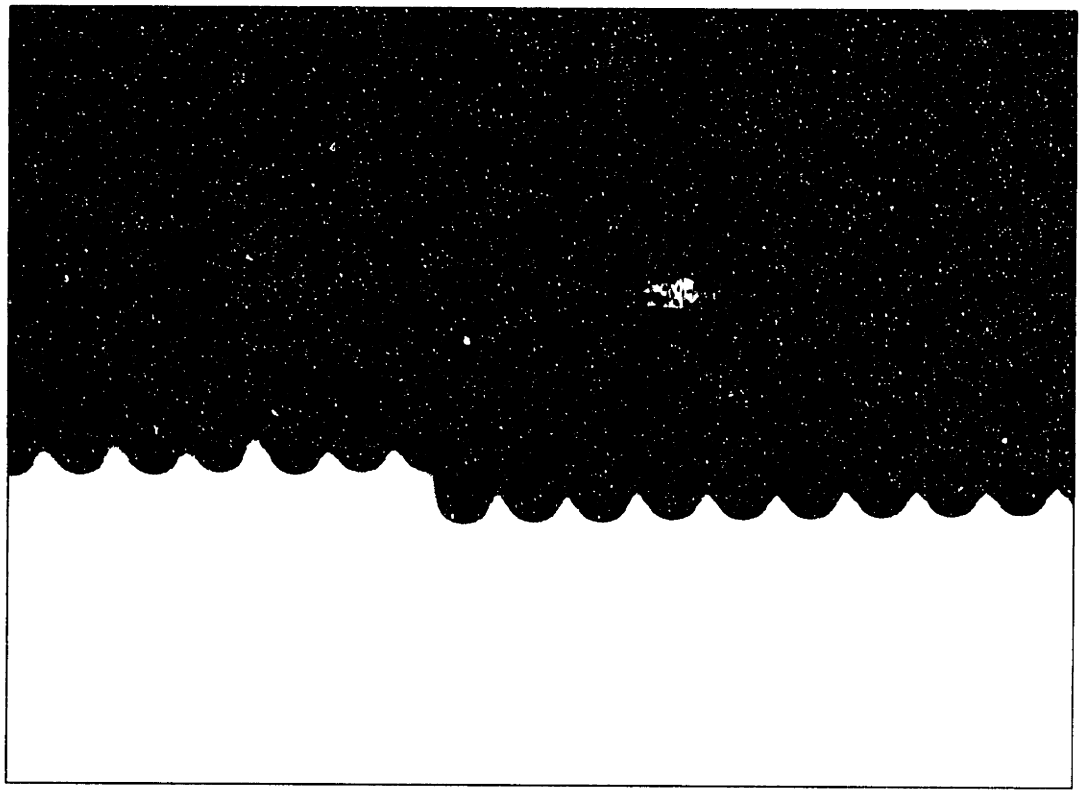


Figure 3.1: Image of Winding with No Error

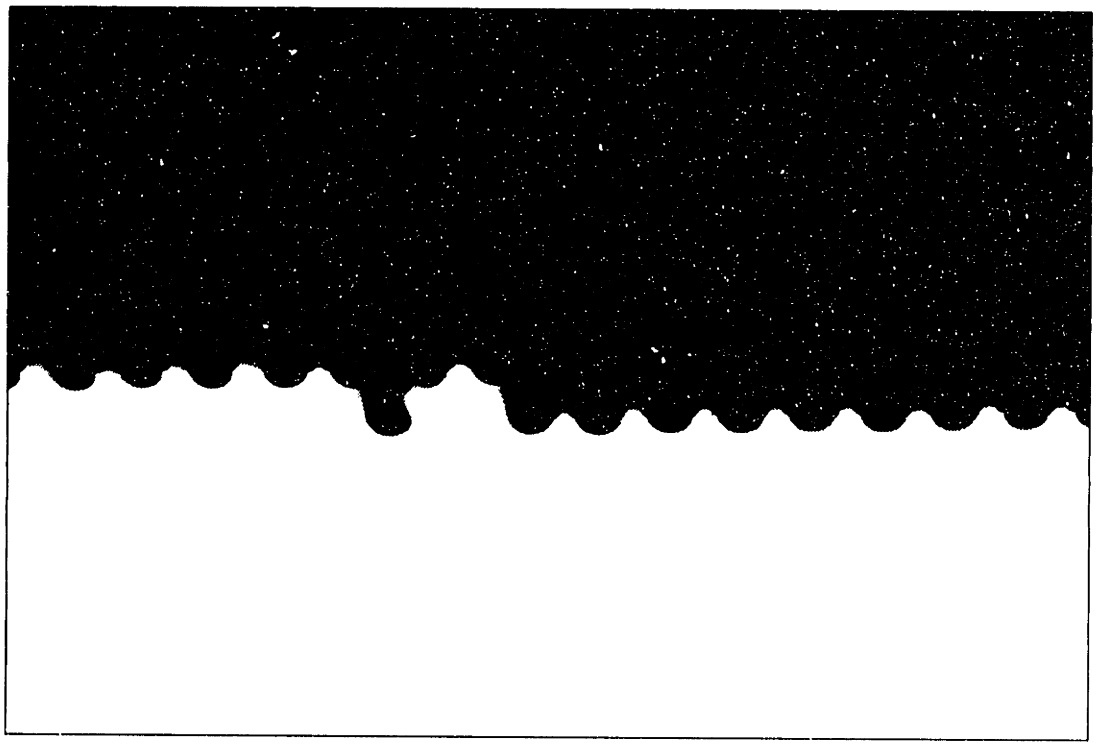


Figure 3.2: Image of Winding with Gap Error

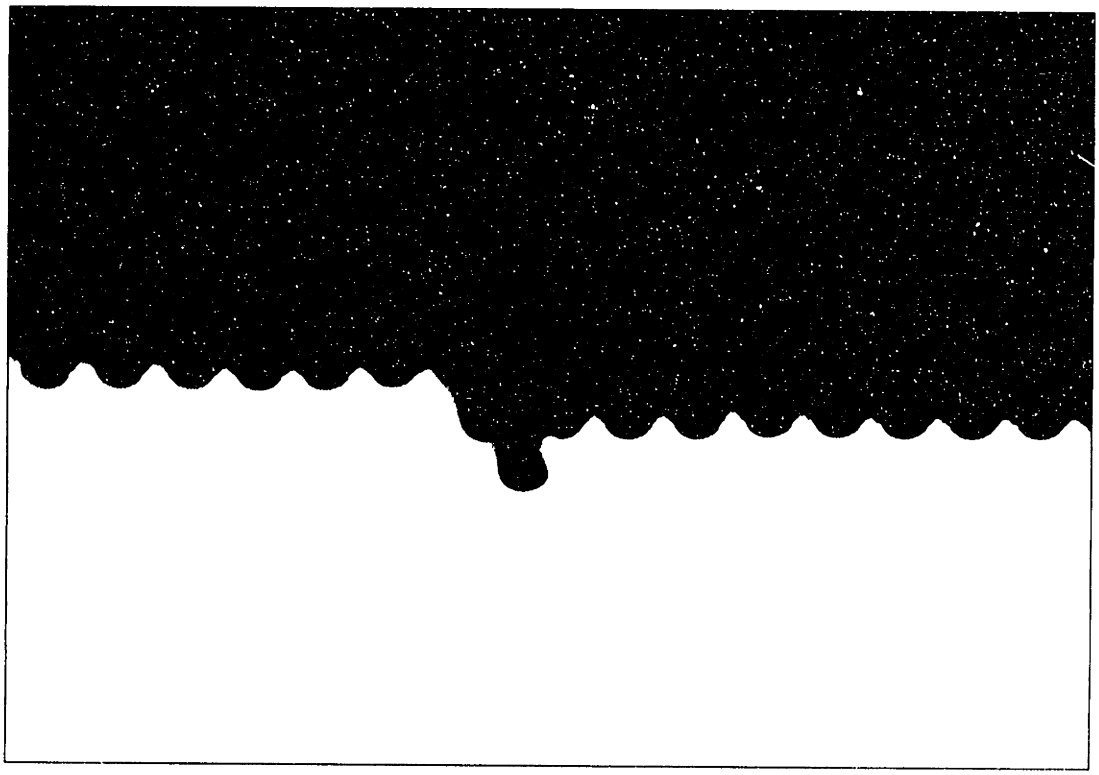


Figure 3.3: Image of Winding with Climb Error

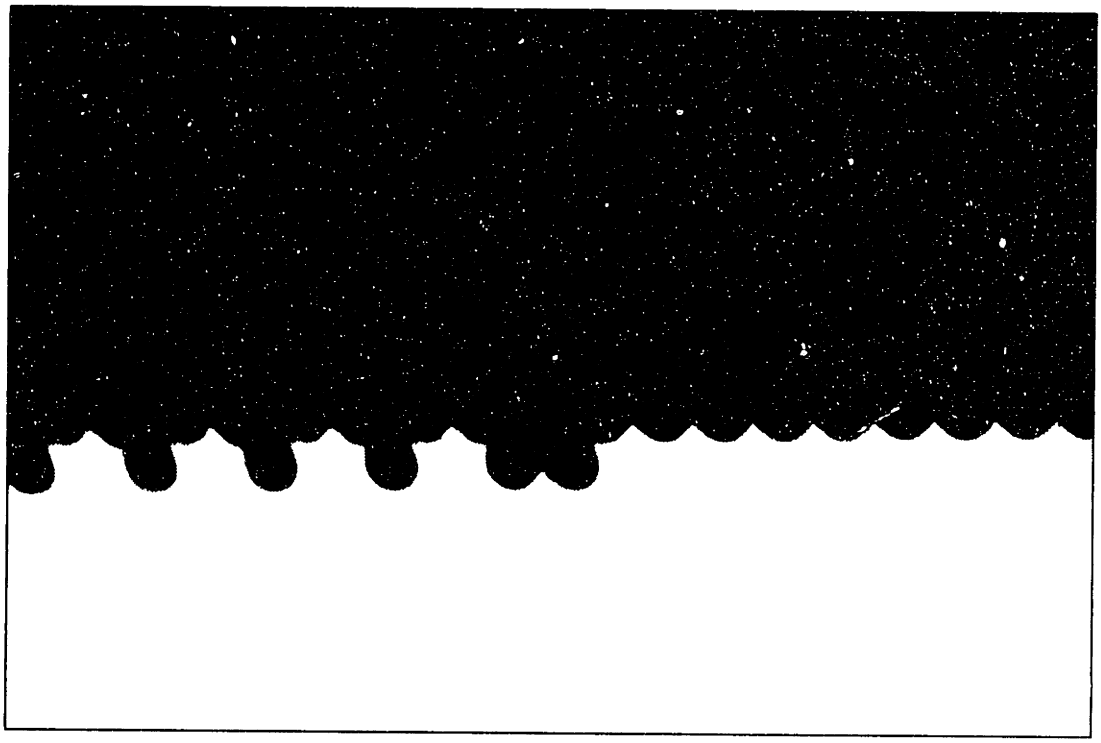


Figure 3.4 Image of Winding with Anti-Gap Error

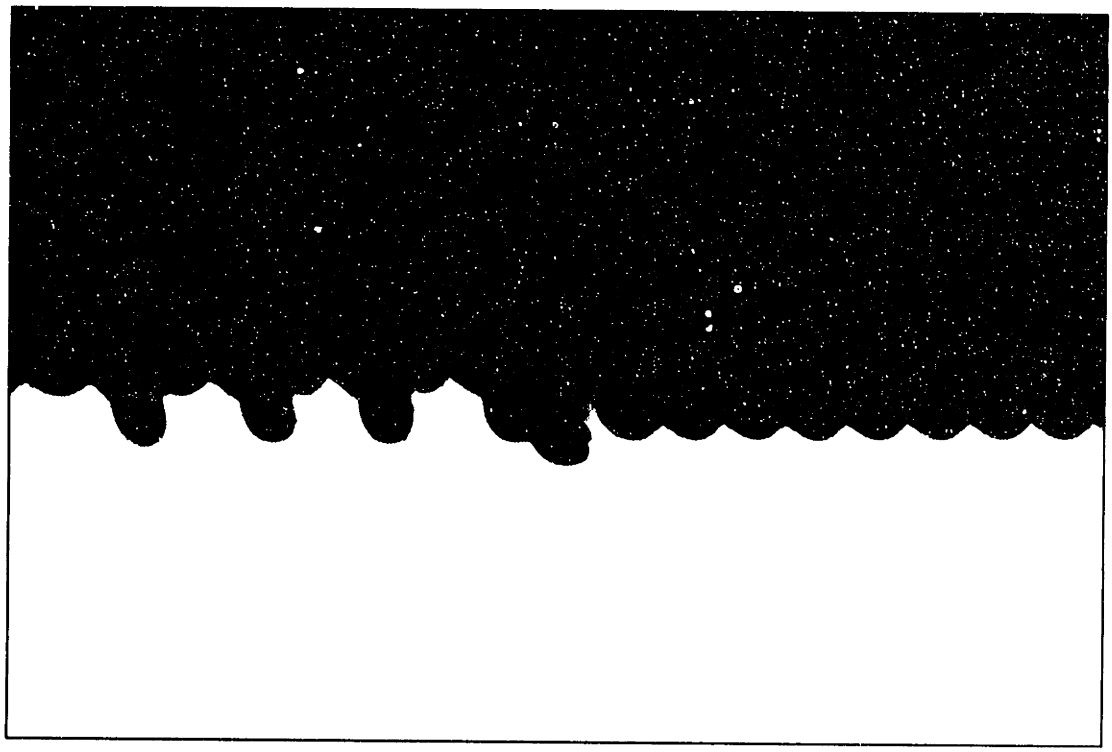


Figure 3.5 Image of Winding with Rise Error

3.3. Performance Specifications

The specifications defined for the vision system are threefold. First, the vision system is expected to be able to detect the four different types of errors as explained in the previous section. The second criterion for the vision system is to have a processing rate of at least 4-5 images per second, which is equivalent to 4-5 images per revolution given the set winding speed of 1 rps. As a third system specification, the vision system is also expected to be able to operate in two modes, automatic and manual analysis. The automatic mode is when the vision system is running continuously with the winder program and is given commands to capture and analyze from within the coilwinder program. The manual mode of operation allows the operator to control when an image is captured with the hit of a keyboard stroke. The manual mode is to allow the operator to test or calibrate the vision system.

3.4. Operating Assumptions

There are several assumptions that are agreed upon by coilwinding machine and the vision system during the image analysis. The first assumption is that the maximum winding speed of the coilwinder will not exceed 1 revolution per second. This will help guarantee that the specification for the vision system to analyze 5 images per second or 5 images per revolution will be satisfied. The second assumption is that the main coilwinder program will notify the vision system of when and where to acquire an image with the following information passed from the coilwinder to the vision system: the type of winding configuration, the direction of the wind, and the current location on the coil. The location is identified as being the first wind, the last wind, and the main body that consists of all intermediate wind in between the first and the last. The third assumption made for the operation of the vision system is that the vision stage will jog concurrently with the winding stage such that the current fiber will always fall within the predefined region of interest (ROI) This is to help decrease the processing time by the vision software in searching for a winding error. The fourth assumption is that the vision system will need to be calibrated before the start of a winding session or if changes are made to the type of fiber (i.e. different fiber diameter) used or to the lens magnification. The last assumption is that the image being captured will have the coil appear dark on a white background. Ideally, the coil will be a completely black silhouette set against a white background while still retaining as close a contour of the coil as possible (i.e. no reflection causing parts of the fiber to appear white).

4. VISION SYSTEM HARDWARE

4.1. Vision Hardware Description

There were many issues faced during the selection of the different hardware components for the vision system. As with the vision software, which had to be gracefully interfaced with the coilwinder program, the vision hardware also needed to be integrated with the coilwinder machine. The following sections will describe in detail the issues and final selection of each integral hardware for the vision system.

The vision hardware can be divided into these 6 major groups:

1) Pentium2, 233MHz, personal computer
2) Framegrabber board
3) Camera, lens, and monitor system
4) Lighting equipment
5) Vision stages
6) Mounting hardware

4.2. Computer System

There are two computer formats that may be used for the vision system, PC or Macintosh. The disadvantages and advantages of each were looked into and a decision was made to use the PC format. The PC format serves as a better real-time operating system, provides for a larger selection of image processing cards, and costs less than that of a Macintosh. The only main advantage that a Macintosh has over the PC is its use of simple memory references in accessing the pixels as compared to the far pointers used in the PC format.[17]

The selection of a personal computer was probably the simplest task in terms of hardware. However, the computer should have the necessary features in order to run the frame grabber board and the vision software. The minimum requirements for each of the other two hardware are described in their corresponding sections of this chapter. Processing speed should also be taken into consideration since the vision system will be used for a real time application and thus would benefit from a fast computing speed.

The personal computer selected for this application is a Gateway 2000 G6-233 series. The computer operates on a Pentium2 processor with 64 Mb RAM and 233Mhz processing speed.

4.3. Frame-grabber

The DT3155 Frame Grabber Board was purchased from Data Translation and is compatible with the Global Lab Acquire program. The board uses a PCI bus interface with an external trigger for capturing images. The board can be set to have a display update rate of 30 frames/second (operating at 60Hz), or 25 frames/second (operating at 50Hz). The resolution of the captured image is 640x480 pixels square and may be scaled down to one-fourth the size in real time acquisition. The image can also be passed through a 256 x 8 bit look-up table (LUT) for thresholding purposes during the image acquisition. The captured images are in 256-level grayscale and can be saved in one of the following formats: TIFF, DT-IRIS, or PCX.

The following are the system requirements needed to operate the frame grabber board

1. IBM PC compatible computer with PCI Bus and 90MHz or faster Pentium processor
2. PCI 32-bit Bus master expansion slot
3. 8 Mb of RAM (300kb required to store each frame in 60Hz format, 732kb required to store each frame in 50 Hz format)
4. Graphics card with a 256 color palette
5. 75 Ohm coaxial cable with BNC connector

4.4. Optics System

The lens selected for the vision system is a Meiji short back zoom lens. It provides for a magnification range of 0.7x-4.5x with an additional eyepiece that further enlarges the image by 10 times. The resulting field of view can range between 25.7-4.0mm, corresponding to the magnification used. This will allow for a good range in the number of fibers that is visible in each image. The working distance, or the distance from the object to the lens, is rated at 95 mm. This distance is very close to that between the coil and the lens, which is 3.75 inches apart. Thus, the lens will prove to be ideal for this particular application.

The lens is attached to a Techni-Quip black and white CCTV camera. The camera allows for images to be captured at both 50Hz and 60Hz, making it compatible with the framegrabber. The image is then transmitted to a Toshiba black and white CCTV monitor where the picture is displayed.

4.5. Lighting System

Lighting was a critical issue for the successful implementation of the vision system. Without proper lighting it was impossible to obtain an image suitable for inspection by the vision system. What was required was a lighting system that could produce an image where the coil would appear black against a white background. Several lighting options were tested and an adequate system was selected. However, there are some problems which still remain with the current lighting system. This section will discuss the issues encountered in the design and setup of the lighting system.

In the prototype model of the coilwinder, a ring light was used as a convenient source of illumination. The ring light did not take up much space and did not interfere with other hardware equipment on the machine. However, the convenience proved to be a downfall in providing the necessary lighting condition required for a good image. Although providing a rather diffused and even lighting, the problem with using the ring light was that it caused reflections off of the optical fiber surface. This caused the fibers to appear translucent at certain locations and thus lost its clear boundary against the background. The images were also difficult to analyze because the histogram of the grayscale values was uni-modal, being fairly evenly distributed along the histogram scale. Thus, attempts to threshold the image would have resulted in the lost of information.

A clear remedy to the problem of the light reflecting off of the surface of the fiber was to use backlighting. With backlighting, the light source is located behind the coil and on the opposite side from the camera. Thus, light will cause the coil to appear as a silhouette against a white background with minimum light reflecting off of the fibers. Although this was a viable solution, the main concern was the space constraint in situating the light source.

The first option was to mount the light source on one of the motion stages directly behind the coil. However, doing so would result in the light source being in constant motion with the winding. Although this would seem to be an advantage, it however, would cause variations in the lighting condition

particularly near the flanges. The light source, if mounted on one of the stages, would also interfere with the different operations of the coilwinder by hindering the motion of the different machine components.

Another option that was looked into was the use of a mirror to guide the light such that it would hit the backside of the coil. The light source would be mounted on the same side of the camera but offset from the coil. The light would shine straight down onto the mirror and reflect onto the coil. The orientation and position of the mirror were adjusted so as to provide the optimal lighting condition. However, there were several problems encountered with this method of lighting. One problem was that the light, which reflected off the mirror, would scatter, thereby resulting in uneven illumination of the coil. It was also very difficult to position the mirror to achieve a suitable lighting environment. The location of the mirror would either interfere with a component of the coilwinder or vice versa where the component would obstruct the light path.

The final solution that came about resembled that of the mirror system described above. However, in place of the mirror, a plexiglass light stick was used instead. The plexiglass was machined into a stick 0.375" wide and 0.175" thick, beveled at a 30 degree angle at one end. Light from a Techni-quip fiber optic illuminator would shine light through a Fostec 0.25" diameter optical fiber bundle and along the length of the light stick as shown in Figure 4.1. As light passed through the light stick, some of that light would escape through the top of the light stick with some light reflecting off the beveled angle at the end. The beveled end provided for a higher intensity light directly behind the coil thereby highlighting the boundary of the coil. The light stick had several advantages over the mirror. First, the light stick provided for even, diffused lighting and thus, provided a higher quality image when viewed under the camera. Second, the light stick was much easier to handle and to position in comparison to the mirror. Third, there was no longer interference with other coilwinder components since the light stick could be coupled to the light source and be mounted on the same side of the camera.

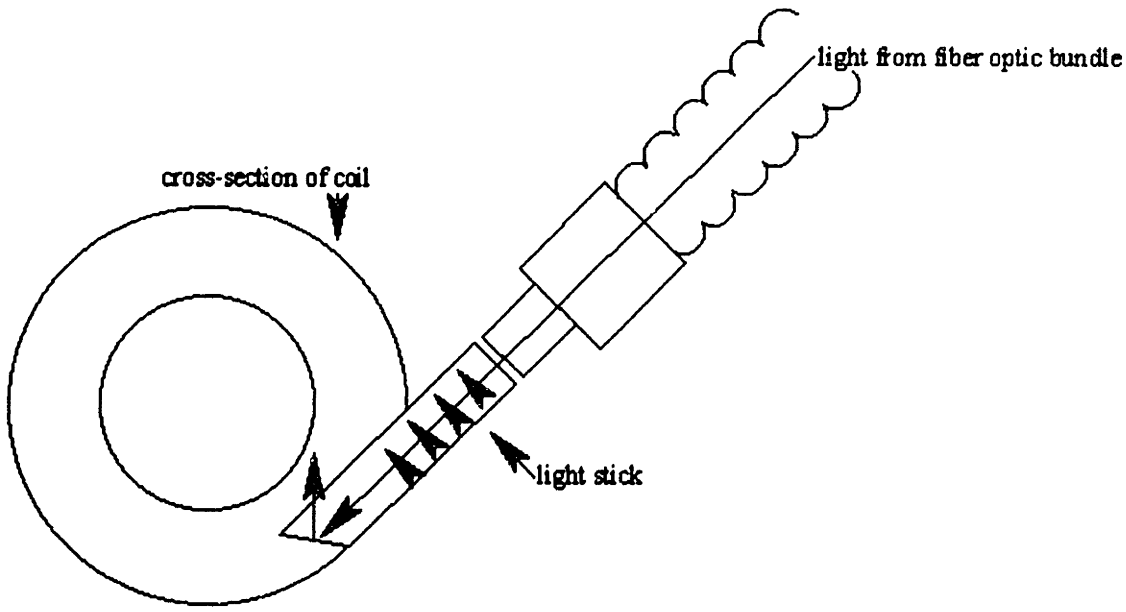


Figure 4.1: Image of Light Set Up

4.6. Vision Stages

4.6.1. Stationary vs. Moving

One of the operating assumptions mentioned in the previous chapter was to have the current fiber that is being wound fall into the region of interest for the vision system to analyze. Several approaches that satisfied this process parameter were considered. The first option was to analyze the entire image and have that serve as the region of interest. It could possibly eliminate the need for the camera to move as long as the entire coil was within the field of view. However, this would require that the vision system analyze the entire coil no matter where the fiber is currently being wound. The main problem with this approach was the poor resolution of the image, which would result from having to fit the width of the coil within the field of view of the vision system. With the width of a captured image being 640 pixel wide, this would result in each fiber diameter (each layer with ~70 turns) being represented by less than 10 pixels. This would basically leave no margin of error on the part of the vision system in finding the error based on the threshold values since each pixel would represent 10% of a fiber width.

Another alternative was to set the camera system onto a pair of manual stages so that only the most current fibers will be monitored. The reason for using two stages rather than one is because the build up of the fiber on the coil is twofold: one along the width of the coil as the fiber is being wound from left to right and vice versa, and the other is the radial build up as each layer is added onto the coil. The use of stages would allow for higher magnification since fewer fibers will be in the field of view of the camera. However, in order to keep the current fiber under constant view both stages needed to be adjusted on a regular basis. This would therefore compromise the drive for full automation of the system. There would also be the problem of achieving repeatability and accuracy if the stages were manually set. Therefore, it was decided that a motorized and controlled method of moving the camera system would be more appropriate in performing the task at hand.

4.6.2. Motion Alternatives

The decision to proceed with a motorized mechanism for the transport of the camera system led to two clear choices, a robotic arm with 2D motion capability or a set of motorized linear stages. Although the robotic arm seemed like an attractive option, it was also thought to be too complex in performing a rather simple task. The motion of the arm would require a large set of x and y coordinates in order to maneuver it to the desired position. The motion stages, on the other hand, provided a more simplistic approach that seemed more suitable for the application. Since the camera would only need to travel in one direction when the fiber is being wound along the coil, only one stage would need to be in operation. Thus, the linear motion of the stage would be more than sufficient in serving that purpose. The build up of the coil as additional layers are added on could be monitored using the second stage. This stage would be required to move only with the start of a new layer. Thus, the two stages do not need to be in motion simultaneously. This would therefore make the control of the camera system much simpler since only one axis is being kept track of at any time.

There were several alternatives in types of stages available. One option that was looked into was a two axes microscope stage which had an opening in the center. The attractive feature of this stage was that the camera could be mounted in the opening and thus be centered over the coil to capture the image. However, the aperture in the stage did not allow sufficient travel without having the camera system interfere with the

motion of the stage or having the stage getting in the field of view of the camera. The sizes of these microscope stages were also rather bulky. Another alternative was an x-y table where, instead of mounting the camera in the center of the stage, the camera would be offset to an edge. However, the severe problem with this approach was that the camera system would come into direct interference with the motion of the x-y table. Another disadvantage was that many of these tables come standard with equal travel in each direction and are coupled symmetrically one on top the other. The sizes of these tables were also very bulky. Although alterations were possible, they were expensive and would have compromised the lead time in acquiring the x-y tables. The last and simplest option was to compound two linear stages. Each stage could be customized to have the required travel and be offset to prevent any interference between the motion of the stages and the camera system. The linear stages could be customized with greater ease in comparison to the microscope stages or the x-y tables.

Some specific issues that came with the selection of the positioning stages include the length of travel of each stage, the resolution of the motion in the stages, and the stability of the stages under loading. Both stages were selected to have 4 inches of total travel with a standard travel of 2 inches to either side of the center point. One of these stages used in monitoring the layer build up, however, had the travel offset such that that it could travel 3" in one direction and 1" in the other. This decision was due to that fact that the center of the coil was offset from the mounting of the stage by an inch. The offsetting of the stage also prevented the camera system from interfering with the stage's motion. The resolution of the stage's travel was another consideration in the selection process. It was decided that a standard resolution of 0.001" in the position was more than adequate to track the range of fiber diameters used in winding. The resolution was also compatible with that of the stages used by the coilwinder and therefore should pose no problem in keeping the current fiber centered in field of view of the camera. The last issue was the stability of the stages due to the loading from the camera. Since the positioning stages were sensitive to torque about the three axes: yaw, roll, and pitch; it was made certain that the stages could handle the required load without any compromise to the performances of the stages.

A third motorized stage was considered for the z-axis that would adjust the focal distance of the lens with respect to the coil. However, it was decided that this third stage need not be motorized but could be instead be substituted for a manual positioning stage. The manual stage was thought to be adequate since the lens did not require to move respect to the coil during the winding process. There was the main concern that the layer build up of the coil would cause the image to become out of focus. However, since the image would be taken only at the point tangent to the coil, the layer buildup should not be of concern since the y-axis would adjust to accommodate for changes in the layer buildup. Thus, the depth of view was not an issue in this particular application. Therefore, the lens focal distance could be set initially and would not have to be adjusted for the remainder of the winding process.

4.7. Mounting hardware

The design of the mounting hardware was strictly constrained by the limited space allocated for the vision system equipment. The design and the location of the hardware also had to take into account the need to center the image of the coil given the range of motion of the vision stages. In addition, the camera mount had to be small enough to accommodate for the integration of an adhesive system onto the vision stages.

The final design of the mounting hardware integrated all the vision equipment into a complete subsystem. The camera mount, which is attached to one of the vision stage, incorporates the fixturing of the light source and light stick as well. The camera is located such that the image of the coil will always be centered given the range of motion for the stages and the range of coil size, both the coil diameter and width. The light mount is offset from the camera and extends from the camera to underneath the one side of the coil. There is a z-axis stage which will allow for the entire camera subsystem to move up and down, thereby adjusting the depth of field. There is also a height adjustment for the light stick by itself, which will ensure that the light stick will not interfere with the coil while the fiber is being wound.

The complete integration of the vision system hardware with the coilwinding machine is shown in Figure 4.2.

Figure 4.2: Photo of Vision System on Coilwinder

5. VISION SYSTEM SOFTWARE

5.1. *Vision Software Description*

The vision system application was developed using the Global Lab Image Development Environment (GLIDE) software in conjunction with a Borland C/C++ compiler. The GLIDE software package includes: the Global Lab Acquire application, which performs the image acquisition functions; Global Lab Image, which is a Windows application that allows for the testing of prototype functions; and the Global Lab Image Processing Library, which is a dynamic link library (DLL) that can be used to develop one's own custom image processing application. The software implements object oriented programming using C++ class functions that create the basic framework in building the Windows application environment. The final vision system application has a menu-driven interface where different functions are performed using the basic click-and-drag.

5.2. *Serial Communication*

One of the major components in development of the vision system software was incorporating a communication protocol that would integrate the vision system with the coilwinder machine.
The two options that were investigated were digital input/output (DIO) and serial communication.
On the standpoint of simplicity, serial communication was the better option in terms of both hardware and software. This section describes in detail how each method operates and the differences between the two communication options.

The two computers, one running the coilwinding machine and the other running the vision system, are interfaced using a standard RS-232 serial cable. It was decided that serial communication was the simplest method for the two computers to interface with one another. The only hardware required for setup was a standard 9 pin RS-232 cable, with a null modem attachment, that could simply be plugged into a serial port of the computer. Since the serial port was readily available and was not used by either computer for other

purposes, the communication therefore did not require any modifications or additions to the computer hardware.

The other communication alternative was the use of digital input/output (DIO). This method of communication was not selected for several reasons. One disadvantage was that it required additional hardware for the computer. The use of DIOs requires the purchase of DIO boards, which are much more costly than the simple RS-232 cable and null modem. One must also consider beforehand the use of DIOs since this would require an additional PCI slot in each of the computers. This may add to the cost or lead time when purchasing the computer for the coilwinder and/or the vision system. Another disadvantage of DIOs is having to deal with the cumbersome wiring of the two boards together whereas with an RS-232 cable, it simply plugs into a standard port. The third disadvantage of using DIOs is the programming of the software.

DIOs transmit bits of information through different channels. Depending on the complexity of the application, the number of channels used may vary significantly. The main problem arises when one needs to check each of the channels to see if new information is coming through. Each channel has a status register that it flags. If a channel has information it wants to send, it sets its "data ready" signal. It then waits for the receiving channel to set its "clear to send" signal before any information is sent through the channel. Thus, the constant monitoring of the channels for changes in the status will compromise the efficiency of the software to do its other tasks.

Serial communication, on the other hand, provides a similar but much simpler means of transmitting information. Like DIOs, there are status registers; however, all information is transmitted through only two channels in the RS-232 cable, the input and output channels. A null modem is used to swap the channels such that the information from the output of one computer goes into the input of the other, and vice versa. Information, usually in the form of a character stream, can be monitored, sent, and received with relative ease.

5.2.1. How Serial Communication Works

Serial communication works similarly to DIO in that there is a constant handshaking procedure that monitors the flow control of information between the two computers. Of the nine pins found on a standard RS-232 DB-9 serial cable, only 5 pins are necessary to carry out the necessary communication protocol between two devices. Two wires are used for transmitting data in each of the two directions, two other wires are used for handshaking, and the last serves as the signal ground. The auxiliary wires include two secondary handshaking lines, a carrier signal line, and a ring indicator line.

The two data pins are the RD and TD line which represents the receive data and transmit data lines respectively. The handshaking lines are bidirectional as well, with the DTR signaling the data terminal ready for when it is ready to receive data and the DSR signaling the data set ready for when there is data to send. The last important pin is the signal ground for the two computers.

With the standard communication interface in place, an agreement must be made between the two computers on the convention in which data is to be transmitted. The setting for these variables are typically done in the initialization of the serial communication ports and must be synchronized for both computers. The settings for the data transmission include the baud rate, the start bit, the data bits, the parity bit, and stop bit. The baud rate is the speed at which data is being sent over the serial line. A rate of 9600 baud would transfer 9600 bits per second. A character will usually take a byte or 8 bits to send over the serial line. However, because the transmission of characters might be asynchronous (not at evenly spaced intervals), it requires that a start bit and a stop bit be sent to mark the start and end of each set of bits which make up a character. The data bits are those which represent the character being sent. It is usually set to 7 bits for standard ASCII characters within the 0-127 range or an 8-bit character for those in the 0-255 range.[18] Lastly, a parity bit is also included to detect possible errors in the transmission of information and may be set to the following: none, even, odd, mark, and space. None parity would eliminate the parity bit from being used. An even parity will check for an even number of logical 1 bits with each character whereas an odd parity would check for an odd number of logical 1 bits. A mark parity will send a logical 1

after the character's data bits while a space parity will send a logical 0. The last two parities are used in cases where a person wants to send a 7-bit character to a device that is expecting a 8-bit character.[18]

Once the data transmission convention has been determined and set for both computers, the computer software then must be programmed to handle the communication. The operation of the serial communication utilizes a special hardware device called a Universal Asynchronous Receiver Transmitter (UART). The UART can be programmed to perform the necessary tasks to transmit and receive data through the serial port. The purpose of the UART is fourfold.[18] First, the UART converts the parallel signals that are to be sent into serial signals for transmission and likewise converts the serial signals received into parallel signals. Second, the UART adds the appropriate start, stop, and parity bits to each character that is to be transmitted. Third, it regulates the baud rate at which characters are being sent and checks the parity bit for errors. Lastly, the UART monitors the handshaking signals for incoming and outgoing data.

In receiving data, the UART first reads the status register and sees if a character has been received. If it has, then the character is read in and the status register is reset to indicate that a character is no longer is waiting to be read. The register continues to update itself with each new character. Similarly, the transmission of character will first check if the UART is ready to transmit a character. If it is, then the status register is set and the character will then be sent. Once sent, the register will reset itself to its idle state value.

The iterative process of checking the status register, reading or sending, and checking the status register again is referred to as polling. The task of having to constantly check the status register becomes a toll on the processing power of the computer especially if the computer is multitasking. This is one of the main disadvantages of using DIOs. Polling might also cause a new character to be lost while the first one is being processed due the time delay required to read in the first character. The baud rate is very much limited in this case and will typically not exceed 1200 baud in order for all characters to be read in by the

computer.[18] An alternative to this method of checking the status of the register is a special signal known as an interrupt.

A hardware interrupt is generated when a character has been received or transmitted or when a handshaking signal changes state. The UART can be programmed to generate an interrupt when a particular event occurs. This eliminates the need for the computer to constantly check the status of a register for that particular event as with the polling method. Therefore, an interrupt can be useful in reducing the processing time especially if several tasks are taking place at once. However, any task running must be designed such that it can be suspended when an interrupt occurs and resumes once the event causing the interrupt has been attended to. [18]

5.2.2. Programming of Serial Communication

The serial communication used to integrate the vision system with the coilwinder is based on the interrupt-driven UART as discussed in the previous section. In order to handle an interrupt-driven input/output, an interrupt service routine (ISR) must be integrated into the software for both computers. The ISR incorporates assembly language programming in order to access the PC's interrupts. The ISR consists of several component functions. One function, using assembly language, initializes the system for interrupt-driven communication. A second function, also utilizing assembly language, serves as the interrupt handler whenever a communication interrupt occurs. A third function, called by the interrupt handler, checks the UART to see what triggered the interrupt and reads in a character if one is available.

The characters are first read into a circular buffer in order to handle faster baud rate without loss of information. The characters are then retrieved on a first in/first out (FIFO) basis for processing. New characters will write over the existing characters, starting from the oldest one, once the buffer fills up. A flow diagram of how the interrupt routine works is illustrated in Figure 5.1.

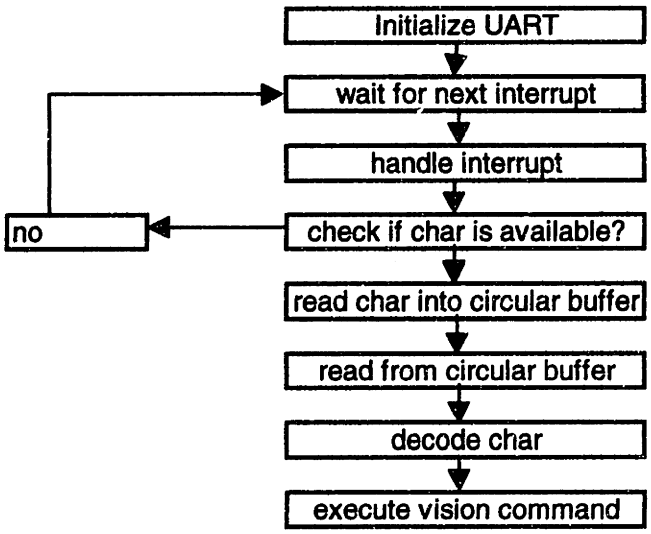


Figure 5.1: Flow diagram of Interrupt Driven Serial Communication

It is important to note that the interrupts should be set back to the original values upon exiting the communication program. Failure to do so will result in the computer to crash if more characters are sent through the serial port when the communication program is shut off. The reason is that the interrupt is still in effect. Thus, any character sent over the serial port will trigger the interrupt.

5.3. *Description of Vision System Algorithm*

The current analysis algorithm came about after several iterations of the image processing code. The main concern in the algorithm development was achieving a highly reliable and repeatable analysis code. The main concern in the algorithm optimization was achieving the desired analysis rate of 5 images per second as given in the project specifications. The following section outlines the pitfalls of some of the algorithms tested and the arrival at the current analysis code.

5.3.1. Image Processing

5.3.1.1. Thresholding

During the beginning of the algorithm development it was thought that thresholding the image from its captured grayscale values into a binary image would simplify the analysis. This method would clearly distinguish the fiber, which would now appear as a black silhouette, from the white background. However, there were several issues that were faced in the development and implementation of the algorithm. One of the main issue with the thresholding method was selecting the threshold value that would retain most of the pertinent information. Selecting too high a threshold value might cause the background to be considered as being a part of the fiber due to shadowing effects. Selecting too low a threshold value, however, might cause part of the fiber to be considered as the background due to light reflecting off the fiber. Typically, a threshold value close to the median of 127 was appropriate although slight changes in the lighting condition would require modification of this value.

5.3.1.2. Image Size vs. Processing Time

The second issue in carrying out the thresholding method was the processing time. It was found that the image processing step, in comparison to the rest of the analysis algorithm, took a large percentage of the total time. One of the first and most intuitive option in reducing the analysis time was to reduce the size of the image that was to be analyzed. There were basically two ways to go about reducing the size of the analysis image. One method was to scale the entire image to a fraction of the original 640x480 image size. The other method was to crop the image to contain the area of the image which had the pertinent information. The former was disregarded as an option mainly due to the fact that with the scaling down of the image would cause a loss of resolution. The number of pixels used to represent a fiber diameter would be reduced and the precision of the analysis would be compromised. The latter posed as a better option because if the current fiber were always to be centered in the image, it would not be a problem to crop just that area for the analysis. No important information would be lost because the analysis is mainly concerned with the current fiber and the closest neighboring fibers. The resolution would still remain the same with the second alternative and thus there would be no loss of information in that respect as well.

An initial set of experiments, comparing the image size versus the analysis time, showed that if the original 640x480 image were to be cropped to a 100x75 size image, the analysis time would decrease from an average of 10-12 seconds per analysis to 2-3 seconds per analysis. Further experiments were conducted to pinpoint the bottleneck in the analysis process and to find solutions to the problem.

Table 5.1 shows the results of the experiments conducted using the Watershed processing method. The Watershed filter is explained further in the subsequent section. The table lists the time required for thresholding and analysis for different image sizes. The size of the ROI was selected such that the current fiber and its nearest neighboring fibers were contained within the region of interest. It was found that it was possible to reduce the ROI to one-third of the size without compromising the lost of detail information in the analysis. The reduction in the size of the ROI shortened the analysis time, as predicted. The experiments also showed that the times required for the thresholding and analysis were found to be the two most time consuming tasks. The bottleneck, however, remained to be the time required to do the thresholding of the image. Thus, the morphological operation was the critical area that needed to be improved upon.

(425x115)	48875 sq. pixels	(225x100)	22500 sq. pixels	(225x75)	16875 sq. pixels
Morph	Analysis	Morph	Analysis	Morph	Analysis
3.57	0.49	1.87	0.17	1.21	0.16
4.73	0.38	2.52	0.11	1.65	0.11
2.91	0.60	1.37	0.33	0.77	0.27
3.13	0.55	1.93	0.22	0.87	0.22

Table 5.1: Image Size vs. Computational Time

5.3.1.3. Roberts Filter vs. Watershed Filter

The third and more involved issue in the thresholding method was selecting an image processing technique which would best serve the application in providing a quick and accurate representation of the grayscale image. The Global Image software package provided some standard image processing tools capable of the tasks. Several tools were tested for feasibility and some were combined to provide improved results. The two filters that brought the greatest contrast in the images were the Roberts filter and the Watershed filter. However, these filters alone were not enough to provide an image suitable for analysis for they both

distorted the image by either making the fibers appear thicker or thinner than they were before thresholding. Thus, a second operation of erosion or dilation, respectively, was needed to correct for the distortions. The addition of a second operation consequently contributed to a longer analysis time requirement.

In comparing the implementation of the Watershed filter to the Roberts filter, the source of the time sink was identified. The Watershed method is based upon a matrix multiplication of a 256 element matrix with that of the image in order to extract the features of an image. This requires intense calculation on the part of the central processing unit and therefore results in a longer analysis time. The Roberts filter is also based upon a matrix multiplication; however, it uses a smaller 3x3 matrix called a kernel. Each pixel is averaged with the values of its 8 neighboring pixels through the Roberts filter. This method was found to greatly reduce the total analysis time. An image could be passed through the filter and be analyzed within 0.25-0.33 second. This meant that at least 3 to 4 images could be analyzed per second as compared to the one second required to analyze the image using the watershed method. Although this was a great improvement, it still failed to meet the specifications for the vision system. Thus, other methods were investigated in order to increase the processing speed.

5.3.1.4. Image Processing Alternatives

An entirely different approach to the analysis algorithm was made through implementing a Hough transform. The Hough transform has the unique feature of extracting particular shapes, especially circles, in an image by using parametric equations describing those shapes. Since the fibers are essentially circles if viewed in cross section, the Hough transform was thought to be an excellent tool in identifying the locations of the fiber. In implementing the Hough transform, it was found that the transform did in fact perform its task remarkably well. It was able to extract the location of the fibers even in the previous layer. However, it did not serve the purpose of reducing the analysis time. In fact, the Hough transform was rather computationally intensive as compared to other morphological operators.

In contemplating the situation at hand, it was decided that the simplest solution in reducing the analysis time was to remove the process that took the most time, the morphological operations. Thus, an attempt was made to analyze the raw grayscale image without preprocessing. Comparison of the pixel values to the threshold was done inline rather than beforehand. The result of this change in the algorithm brought the analysis time down to an average of 0.2 second for each analysis, thereby meeting the vision system specification. However, this reduction in analysis time was also made possible due to changes in the edge detection algorithm as well.

5.3.2. Image Analysis/Edge Detection

The analysis algorithm searches within the predefined region of interest for the current fiber using an edge detection search method. The profile of the coil, distinguished from the background by its grayscale values, is traced until the peak of the current fiber is found. Then depending on the type of wind, the next fiber peak, typically of the previous fiber, is found. The vertical and horizontal distances from the one peak to the next are then measured. The horizontal and vertical distances to the fiber, in front of and on the previous layer to the current fiber, are also found. These distance measurements are then compared to a set of threshold values and an assessment is made to determine the type of error that has occurred if one arises.

In the original algorithm, the current fiber peak was found through an edge detection method that basically traced the contour of the coil silhouette against the white background. The algorithm would look within the region of interest (ROI), going in the opposite direction as the winding. It would start from the top of the ROI, which was usually a white background, and scan down until it hit a black pixel, presumably having found the boundary marking the fiber on the coil. The algorithm would keep track of the maximum y value as it scanned from one side of the ROI to the other. The global maximum y value in the entire ROI was assumed to be the peak of a fiber on the current layer. Then the algorithm would check forward and backward one fiber diameter to confirm whether the fiber is the current fiber. The reason for this check is because the search algorithm might have overshot and misidentify another fiber as being the current fiber, or the peak of a previous fiber might be slightly higher than the current fiber but would still not constitute an error. If the current fiber was found, the algorithm would then proceed to find the two other fiber peaks and obtain the necessary horizontal and vertical distance measurements in making the error assessment.

It was thought that the reason the search algorithm was slow was because it scanned, against the direction of the wind, a large portion of the ROI before the current fiber was found. Thus, a slight modification was made so that the algorithm would search for the first fiber peak going in the same direction as the wind. This helped reduced the time it took to find the current fiber in the analysis. However, the algorithm was still thought to be rather slow because it would require the code to always start the scan from the top of the ROI until it hit a black pixel. Due to the dynamics of the winding (runout), the first black pixel found could shift variably in location. If the coil were centered far from the top of the ROI, it meant the algorithm would take longer to find the current fiber peak. Thus, a third optimization was made to the search algorithm.

The revised method of searching for the current fiber peak takes on the characteristics of a primitive contour tracing method. The algorithm first finds the boundary of the coil and then begins a local tracing of the coil's silhouette. If a neighboring pixel is black, the algorithm will assume it is going towards the peak of a fiber and will continue to search locally upwards. If a neighboring pixel is white, the algorithm will assume it is going towards a trough and will scan horizontally across until it hits a black pixel of presumably the next fiber over. At that point the algorithm will search up towards the peak of the fiber once again. The algorithm will also do a check to see whether or not it has indeed reached a peak and if it is the peak of the current fiber. The basic algorithm is explained in the flow diagram, shown in Figure 5.2, with a pictorial representation shown in Figure 5.3. This search method, in conjunction with the elimination of the image preprocessing, has allowed the algorithm to achieve the 5 images per second analysis rate.

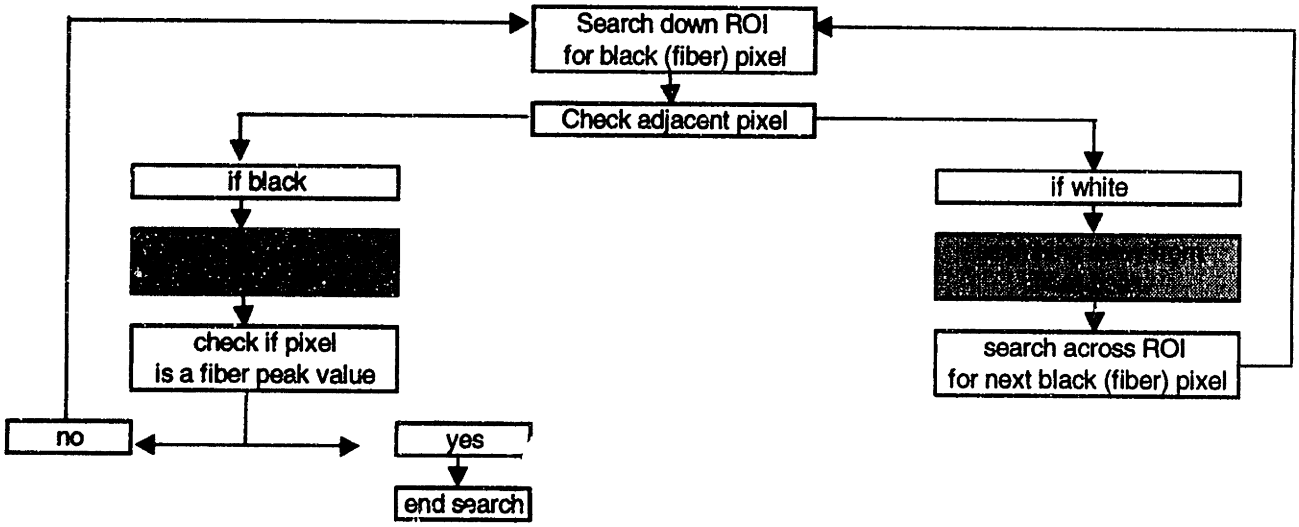


Figure 5.2: Flow Diagram of Contour Tracing Algorithm

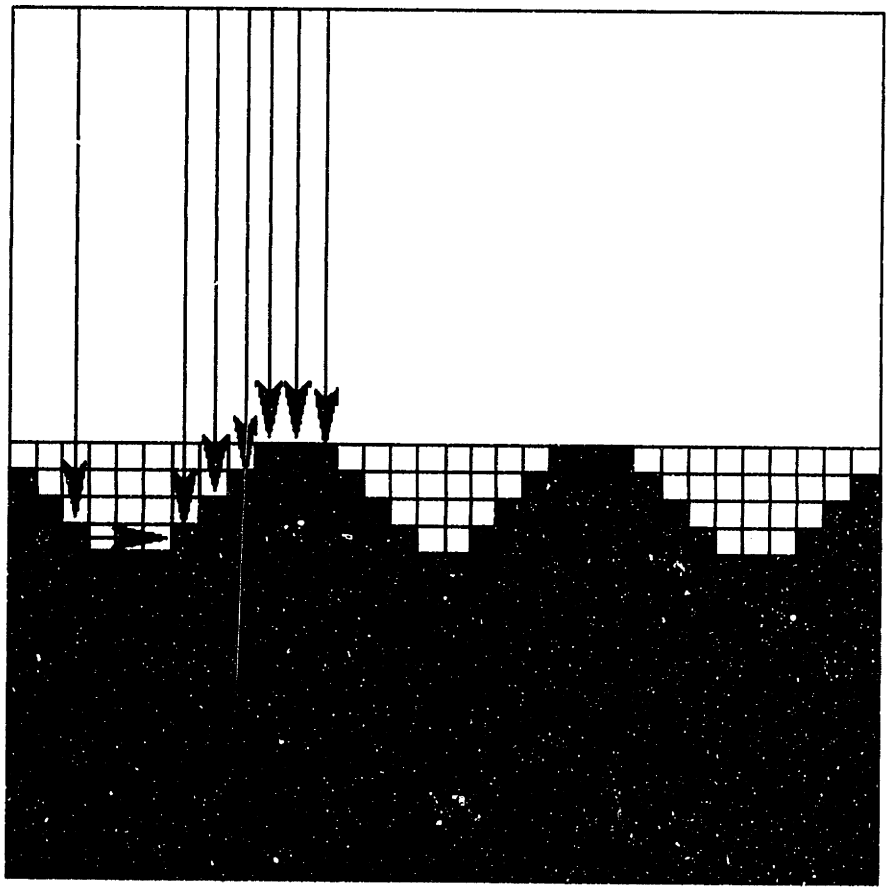


Figure 5.3: Contour Tracing Algorithm

5.3.3. Error Detection Issues

The intricacies in the types of winding patterns that have evolved have brought about a number of error detection issues. These intricacies complicate the error detection algorithm by presenting a larger number of error cases to detect. This section discusses the myriad of issues that need to be addressed in developing the vision system.

The most prominent issue arising from the unique winding patterns is the distinction between continuous and alternating wind. As discussed in the previous chapter, a continuous wind is when the fibers are laid one right next to another while the alternating wind is when a single spacing is left in between consecutive fibers. The difference in the analysis for these two cases is a simple change in the distance when searching for the next or previous fiber. In the continuous case the fibers are one fiber diameter apart whereas in the alternating case the fibers are two fiber diameters apart. However simple the difference is, a distinction between the two cases needs to be made, otherwise an alternating wind could easily be mistaken for a gap error if the vision system considered it to be a continuous wind. Likewise, a continuous wind could be mistaken for an anti-gap error if the vision system was in the alternating analysis mode.

Another special case that the vision software needs to be aware of is the jog zone. The jog zone is where the current fiber winds over the peak of the fiber in the previous layer, as shown in Figure 5.4. This causes the fiber on the current layer to be situated higher and horizontally offset as compared to other parts of the coil. Thus, when the distance measurements are taken with respect to the previous layer, it could lead the vision system into thinking that a climb error has occurred when it otherwise has not. The vertical peak to peak distance for the case of the jog zone is one fiber diameter as compared to 0.866 in the other parts of the coil. The calculation of the distance measurement is shown in Figure 5.5.

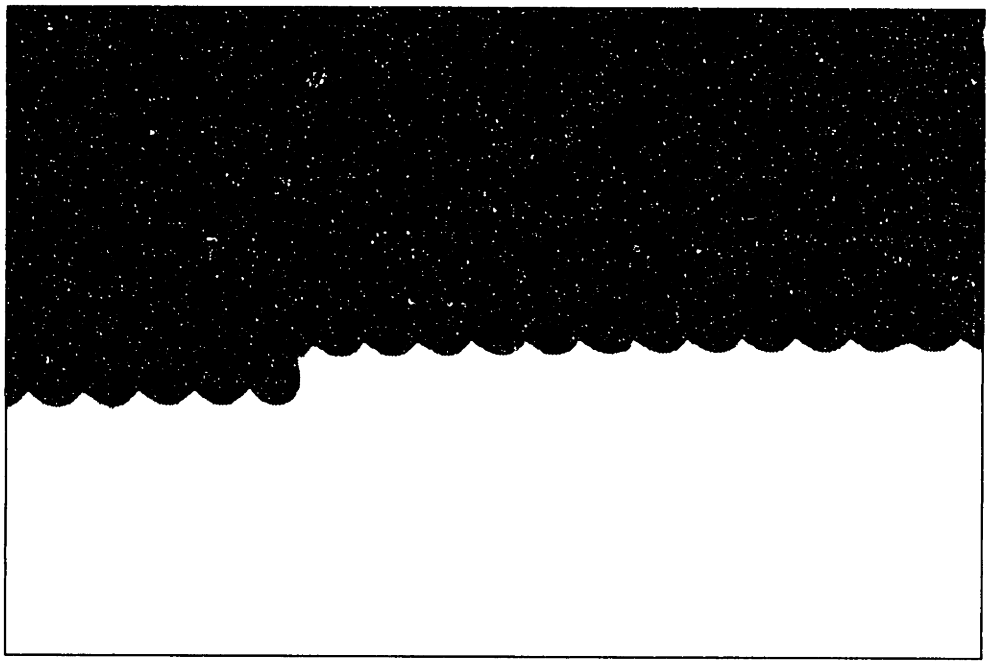
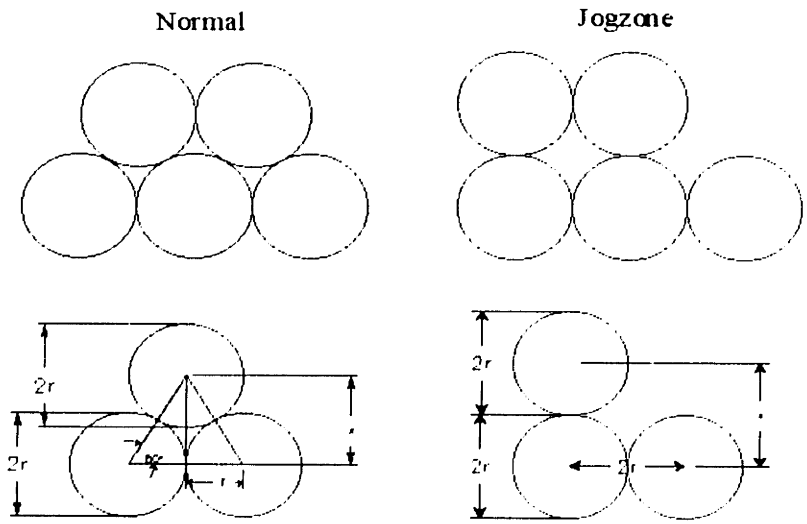


Figure 5.4: Photo of Jog Zone



$\sin 60^\circ = x/2r$

$0.866 = x/2r$

$x = 1.73r$

$x = 2r$

Figure 5.5: Calculation of fiber to fiber distances

A winding factor which may arise more with the error correction than with the error detection is knowing which feed spool is currently being used. In one case, the image is captured 90 degrees after the fiber is first wound onto the coil whereas in the second case, the image is captured 270 degrees after the fiber is first wound onto the coil. This becomes a factor when an error occurs and the coilwinding machine needs to unwind past the error. The coilwinder might not unwind past the error if one case is mistaken for another. This problem could be easily corrected by assuring that the unwinding will always be greater than 270 degrees when an error has been detected.

The wind direction is another important parameter that the coilwinder should make known to the vision system at all times to simplify the analysis of the image. Since the error detection algorithm bases its initial fiber peak search on the direction of the wind, the wind direction parameter should be communicated to the vision system during the analysis. The direction is also important because the analysis algorithm needs to know on which side of the current fiber should it look for the previous fiber and previous layer.

The location on the coil on which the fiber is currently being wound is another factor in the error detection. The main concern is whether the fiber is currently being wound next to the flange of the coil. If the fiber is wound next to the flange on the start of a new layer then it is referred to as the first wind. Similarly, if the fiber is wound next to the flange at the end of a layer then it is referred to as the last wind. Otherwise, the fiber is assumed to be in the middle of the coil, at least one fiber diameter away from either of the two flanges. One reason for making the coil location known to the vision system is because a different distance measurement is required from the vision software in assessing an error. In the case of the first wind, the distance from the current fiber to the flange wall is found, while in the other, the distance from the current fiber to the previous fiber is measured.

Another reason for making the coil location known to the vision system is because the vision system might choose to disregard the error assessment due to the fact that a poor image is typically captured near the flange location. The poor image is due the reflection of light off the metallic surface of the flange wall. This reflection results in the flange boundary being very difficult to identify, even by human observation.

The reflection also causes the image of the fiber to be mirrored along the flange boundary making it appear as if there are fibers where there should not be. The consequence of all this is that the vision system is basically rendered useless near the flange walls. This issue is further discussed in the lighting section of the vision system hardware with possible solutions in the conclusion section of this thesis.

6. VISION SYSTEM INTEGRATION

The main function of the vision system is to provide real-time feedback for the coilwinding machine as a coil is being wound. This is to reduce, if not eliminate, the need for an operator to constantly monitor the coilwinding operation for errors. In the simplest terms, the computer that runs the coilwinding machine will tell the vision system when to capture and analyze an image during the winding process. The vision system will perform the task and will feedback to the coilwinding machine the status of the wind, notifying the winder of any errors that may have occurred. If an error is flagged, the coilwinder will stop winding and take the appropriate corrective measures. Only when the coilwinder is unable to correct the same error, occurring at the same location on the coil, after three tries will a human operator be notified. The integration of the vision system with the coilwinding machine is illustrated in Figure 6.1.

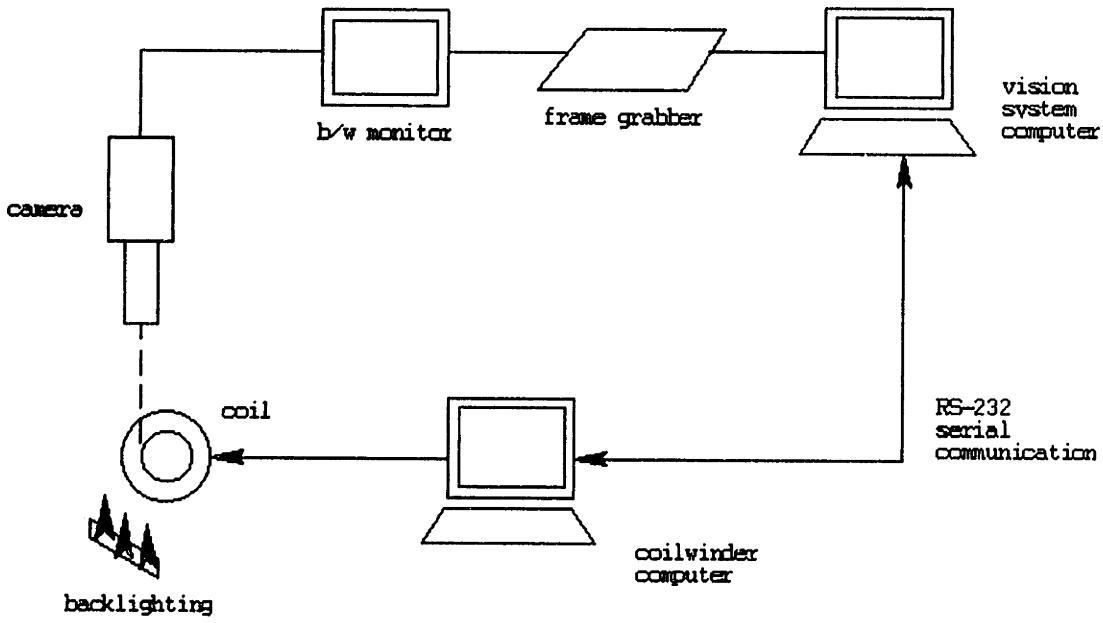


Figure 6.1: Coilwinder/Vision System Network

6.1. DOS vs. Windows

The Windows format of the GLIDE application was one of the issues in interfacing with the DOS format of the coilwinder software. Ideally, it was thought that the coilwinder computer could send a DOS command

through the serial port which would start up the vision system if the vision software were to be a DOS application. This would eliminate the need for the machine operator of having to learn how to run another computer system. However, all vision software requires a graphic user interface and thus must run as a Windows application. No commercially available vision package was found to run under the DOS format. Thus, it was required that the vision system and coilwinder be initialized independently by their corresponding computer.

The second issue of integrating the two computers was the menu driven interface of the vision system application. The click-and-drag feature would allow a function (or macro) to be performed only once. Carrying out additional functions would require that an operator tend to the vision system with each new command. This issue was taken into consideration in the programming of the vision software. The solution was to have the main analysis function run in a continuous loop that would take commands from the coilwinder computer and perform the necessary task. Thus, only one menu item would need to be selected to initialize and run the image analysis. The control and termination of the analysis program would be carried out directly through the coilwinder computer. This will eliminate the need for the operator to attend to both computers.

Although it was feasible to have the main image analysis run in a continuous loop, there still remained the issue of having to calibrate the vision system to accommodate for different fiber diameters and lens magnification. Since direct observation and measurement of the captured image would be required in the calibration process, it was decided that it was best to conduct the calibration on the vision system offline from the coilwinder computer.

The information that is basically required from the calibration is the measurement of how many pixels make up a fiber diameter. The measurement is based on the fiber diameter used for the winding, the depth of focus that is determined by the location of the lens with respect to the coil, and the magnification setting of the lens. A set of calibration values can be performed beforehand for a given set of operating conditions. Values can also be stored for future use if the settings are the same for several coils to be wound.

Otherwise, the calibration should be done prior to the start of any winding session. The information obtained from the calibration should then be entered into the vision.dat file which is read into the coilwinder program upon initialization.

6.2. *Information to be Communicated*

There will be constant communication between the coilwinder and the vision system during the physical winding of the optical fiber onto the mandrel in order to monitor the process for winding errors. There is also the option of leaving the error detection on during the unwinding procedure if the operator feels that will make the process more robust. However, communication between the two computers during other winding operations such as the swapping of the supply units is optional and not necessary for the error detection. The following section describes the type of information that is to be communicated back and forth between the two computers during the winding process.

The first set of information that is passed from the coilwinder to the vision system is the operating parameters. These parameters are entered in the vision.dat file, which is then read into the coilwinder computer upon startup and initiation. These parameters are then passed to the vision system as part of the initialization of the serial communication. The set of parameters include: the threshold values for classifying the errors, given in a percentage of a fiber diameter; the magnification used on the camera lens during the winding operation; and the diameter of the fiber, in microns, currently being used. The vision system will use these values in determining if an error in the winding has occurred. There are three reasons for setting these parameters in the coilwinder computer. The first reason is to keep the same standard convention of setting and initializing winding parameters through the *.dat files. Another reason is to allow flexibility in changing the parameters in the vision system computer rather than have the variables hardcoded into the vision software. The last reason is to keep most of the operations with the coilwinder computer so as to minimize the need for an operator to be complicated with the details of the vision computer.

Once the vision system has been initialized with the winding variables, then most of the communication between the two computers will only occur during the actual winding. Amidst the winding loop, the

coilwinder computer will send a character through the serial port telling the vision system when to capture an image. Each character is coded to represent each of the different type of winding scenarios. The character will tell the vision system the type of wind, the direction of the wind, and where on the mandrel the fiber is being wound. The information will help the vision system determine the type of error to look for and thus which algorithm to use. Figure 6.2 shows the basic framework of the error detection algorithm for the different winding scenarios.

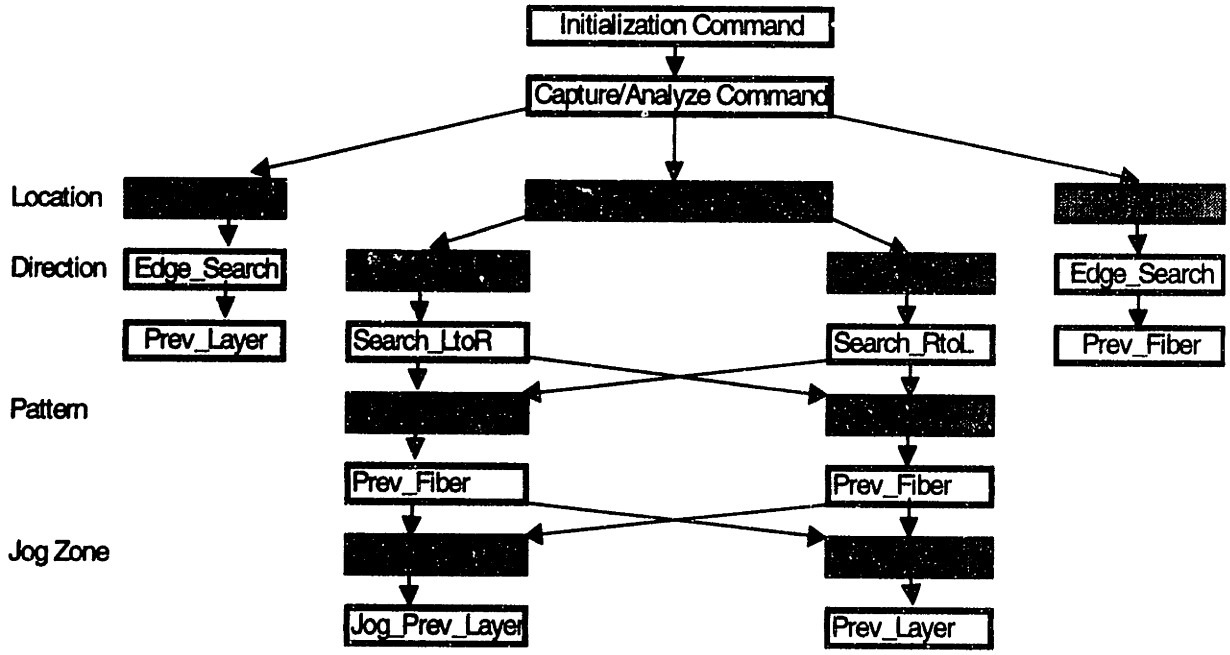


Figure 6.2: Basic Framework of Error Detection Algorithm

After the vision system has captured and analyzed an image, it will return to the coilwinder computer a set of four values. This character stream will tell the coilwinder what type of error was found, if any, the vertical peak to peak distance from the current fiber to the fiber on the previous layer, and the horizontal and vertical peak to peak distance from the current fiber to the previous fiber. The coilwinder computer will use the feedback information to carry out the appropriate corrective measures if an error should occur.

The basic error correction scenario would be to first stop winding and unwind past the error. Then adjustments are made to the winding parameters, mainly shifting the x-stage, to correct for the error. Shifting the x-stage back will hopefully correct for a gap problem while shifting the x-stage forward will mitigate a climb error. If three attempts have been made in trying to correct for an error in the same location without success, then a human operator will be notified to tend to the problem.

6.3. *When and Where to Capture Images*

In determining when and where to capture images to be analyzed by the vision system, there were several issues which had to be addressed. The first issue was the rate at which the vision system will be able to analyze. Although the specifications given for the project required for at least 5 images to be analyzed per second, the feasibility of achieving this was uncertain during the beginning stages of the project. However, after making several iterations of the vision software code (see chapter on vision system software), initial testing of the algorithm showed that the image processing rate was realizable. Thus, given the processing rate and the nominal winding speed of 1 revolution/sec, the vision system could, on average, capture and analyze 5 images/sec or 1 image every 72 degrees.

With a known image processing rate, the second issue was to determine when and where on the coil to capture the images. The simplest approach was to equally space the images based on the rate at which the images could be analyzed. The drawback of this, however, was that the vision system would only look at the same part of the coil with every revolution. This risked the possibility of the vision system failing to detect an error on the parts of the coil in between where the images were being analyzed. The second option was to randomize the location at which the images were to be analyzed so that the vision system would be able to monitor a larger circumferential area of the coil at different locations. However, this method did not seem to be fully compatible with the approach of the vision system software. Since the image analysis is mainly concerned with the current and previous fiber within a region of interest, randomizing when an image is captured may reveal errors but they may be beyond the region of interest in which the image software searches for the errors. In addition, errors might not occur at all between the locations at which the images are being analyzed in an equally spaced analysis interval. The third option was to allow a human operator to manually tell the vision system when and where to capture the image.

However, this would deplete the purpose of implementing the vision system for full automation in the first place. Thus, this option has been left as a stand alone tool to be used mainly for debugging purposes.

After considering the three options, it was decided that the first approach would be the most feasible and easiest to implement. With this option in mind, one last issue had to be addressed. Given an analysis rate of 5 images per second, the question remained as to where the optimal locations are in detecting errors. "Optimal" here refers to the locations that are most likely to have errors arise. From plain observation, it was clear that the jog zone was a problematic location and thus would be an ideal place to look for errors. That subsequently led to selecting the location immediately before and after the jog zone as two other locations to monitor for errors. The two other images were then simply spaced equally apart between the post-jog zone and pre-jog zone locations along the remainder of the circumference of the coil.

7. Results and Conclusion

7.1. Performance Evaluation

The vision system software was implemented in real time to evaluate the performance of the error detection algorithm. All the different error scenarios were simulated to allow for the testing of the algorithm. The following sections summarize the results of these experiments.

7.1.1. Gap Error

Gap errors were found most often during the continuous wind. The error tolerance used for these experiments was 25% of a fiber diameter. Upon detection of the first gap error, the coilwinder would attempt to unwind two turns or the amount of fiber wound if that amount is less than two turns. The coilwinder would simply try to continue the wind from that point on. Upon detection of a second gap error, the coilwinder would again attempt to unwind past the error. At this point, the coilwinder would shift the stage a distance corresponding to 1.25 of a fiber diameter in the direction opposite the wind direction to compensate for the gap error. The coilwinder would continue to resume normal winding once again. If a third gap error arises, the same correction procedure is repeated as with the second error. If after three attempts to correct for the gap error, the coilwinder is still unable to do so, the coilwinder would stop winding and beep for an operator for assistance.

The vision system software was able to detect all the gap errors that arose during the winding. The coilwinder, in conjunction, was able to successfully correct for the gap errors and proceed with continuous winding. The time required for a frame capture ranged between 0-0.06 second while the time required for the analysis ranged between 0-0.06 second as well, based on the computer clock speed. Thus, the maximum time required to detect a gap error would be only 0.12 second, which fell well within the analysis specification.

7.1.2. Climb Error

Climb errors were also monitored during the continuous winding mode. The error tolerance was set to the same level as for the gap errors. Upon the detection of the first climb error, the coilwinder would attempt

to unwind two turns and resume winding from that point on. With the second climb error, the coilwinder would unwind and attempt to correct for the error by shifting the stage in the same direction as the winding direction by a distance of 1.25 times the fiber diameter. The coilwinder would attempt to correct for the error, in a similar manner if a third climb error occurs, before resuming continuous winding.

The vision system was also successful in detecting the climb errors that arose during winding. The time required to capture and analyze an image fell in the same range as that for the gap errors. Thus, the detection of climb errors also met the vision specifications.

7.1.3. Anti-Gap Error

Anti-gap errors were specifically monitored during the winding of the alternating A pattern. The error tolerance was set at 30%, slightly higher than that for the continuous winding, in order to accommodate for the dynamics of the alternating A winding. With the alternating A winding, the fibers were more sensitive to shifting out of place but could be corrected for with the alternating B winding when the gaps are filled in. The correction procedure for anti-gap errors was exactly the same as for the climb error. Errors were compensated for by shifting the stage in the same direction as the wind.

Anti-gap errors were detected and corrected for as with the other errors. The time required to capture and analyze the images also fell within the same range of 0.05-0.12 second. Thus, the algorithm was again successful in satisfying the vision system specifications.

7.1.4. Rise Error

The detection of rise errors was done only with the alternating B winding pattern. The error tolerance was again set at 30%. The correction procedure for rise errors sets it apart from the other types of errors. Upon the detection of the first rise error, the coilwinder would attempt to unwind as with every error. However, to compensate for the error, two steps were taken. First the winding speed was halved during the winding and second, dithering of the tension was turned on during the entire range that the fiber was being rewound until it passed the point at which the error occurred. If a second rise error occurred, the winding speed was further halved and the dithering was again turned on during the rewinding of the fiber. A third error would further reduce the speed of the wind. Only when the rise error has been corrected will the winding resume its regular winding speed and the dithering be turned off.

Although the vision system was able to detect the rise errors, it required more time to do so. The time required to capture an image was the same 0.06 second. The time required to do the analysis was, however, slightly longer with a range between 0.05-0.11 second. Despite this fact, the total time required to detect a rise error still fell within the vision specification of an analysis rate of 5 images per second.

7.1.5. Flange Condition

Due to unavailability of an adequate mandrel, the testing of the flange condition was not carried out. Thus, it is still uncertain as to whether the algorithm will be able to successfully detect errors near the flanges (either the first or last turn) in real time mode. The uncertainty is also compounded by the fact that it is difficult to achieve a good image near the flanges due to inappropriate illumination. However, prior experiments have been done using still images captured near the flange to test the algorithm offline rather than in real time. The algorithm was able to obtain distance measurements in these still images. Thus, the software should have no problem in carrying out the analyses near the flanges provided that an appropriate image can be captured.

7.2. Recommendations

7.2.1. Improved Optics

One of the main problems that still persist is obtaining good images, particularly near the flanges. The problem currently existing is that the light reflects off of the metallic surface of the flange. The result of the reflection is that there is no clear distinction as to where the flange boundary lies. The fibers near the flange are also reflected in the metallic surface making it seem as if there was no boundary at all. This causes the vision system to misinterpret the image near the flanges, not knowing exactly which fiber is the current fiber. It also makes it difficult to measure the distance from the current fiber peak to the flange wall when the wall is not visible. This problem is apparent even by normal human observation.

One solution to this problem is to black anodize the flange wall so that it will help reduce or eliminate the light reflecting off the surface. Although this option was suggested, it was not implemented for the fear that the change in material properties of the flange would affect the performance of the coil during operation in a gyroscope. However, it was noted that marks left over from the machining of the coil,

although unintentional, served to benefit the vision system. Light was found to be trapped in those machining marks rather than being directly reflected off the surface, somewhat alleviating the reflection problem.

Different alternatives in terms of light were also investigated. Color filters were first used in attempts to eliminate light of a certain frequency from being transmitted. This would help reduce the intensity of the light being reflected of the flange while still providing good contrast between the coil and the background. Although this method did reduce the reflection there was still no distinction as to where the flange wall was in the image. A similar test was done using polarized lighting to help reduce the intensity level of the reflection. This method was not successful since it reduced the light intensity throughout the image and not just at the flange location. It therefore merely created a darker image for the vision system to analyze making it even more difficult in selecting a threshold value due to the shift in the gray-level histogram.

One option which could possibly solve the problem of reflections, but was not tested, is the use of strobe lighting. Strobe lighting has been found to reduce variation in the ambient lighting while at the same time reduce blurring effects due to the coil in motion. However, this option was not looked into for the main reason that strobe lighting might prove to be distracting in plain sight especially if the coilwinder were to be located where human operators would work.

Another problem that arose with the optics was the apparent flickering of the image. At first, the problem was thought to be caused by the vibration of camera system. However, even with the vision system stationary, the problem still persisted. After careful inspection, the problem was instead traced to the light source. Varying the intensity of the light source was found to induce flickering in the image. One explanation for the flickering could be the ac characteristics of the electrical power supply. Voltage variations in the power source could trigger flickering. Humans often do not notice this flickering when viewing the light, however, the camera sensors are more sensitive to these variations.[14] Thus, when an image is captured onto the screen, it exhibits the flickering characteristics. Using a dc power source should help to solve this problem.

7.2.2. DOS/Windows Serial Communications

It was found that there was a discrepancy between the DOS versus the Windows format in the programming of the serial communications. Although the same basic code was used for both formats, it was found that if the same computer were to switch from the Windows communication program to that of the DOS format, the computer would not communicate properly. However, the computer would function properly when the change was from DOS to Windows. No problem would be encountered either if the computer used the same format throughout. It was assumed that operating in the Windows format added certain overhead that was not reset when operating in the DOS mode. Thus, terminating the windows program would leave those settings changed. Restarting the windows program would reset it properly but starting a DOS program right after would keep those changes. The symptom of this problem is when the program is unable to receive characters from the other computer. This would result in having to reboot the computer that is affected.

7.2.3. Neural Network Approach to Vision System

From unmanned vehicle guidance system to manufacturing line quality assurance, neural network is definitely making its place in research and industrial applications. Neural network has also gained popularity in recent years for its use in machine vision applications. Neural network is a relatively new approach where a vision system can be trained to recognize patterns. As the system is presented with more patterns, it goes through a learning process, which thereby makes the system more adept in identifying and classifying the patterns. Once the system has been trained to recognize all possible patterns, the system is then able to provide high performance quality and speed in classifying the patterns in the actual application. This presents an attractive alternative for machine vision applications because they often operate in real-time mode.

The use of neural networks was thought to be appropriate for possibly improving the current vision system. There were certain features of the existing vision system that could be better implemented using neural networks. One feature that would be incorporated into the learning vision system is the operation of the vision software without the need for calibration. The operator will not need to recalibrate the vision system

if changes are made to the magnification setting on the lens or if a different fiber diameter is used in the winding process. This would basically eliminate the need for the operator to have to learn how to operate the vision system computer directly because all commands will be issued through the coilwinder computer. Another improvement that would come about with the implementation of the neural network is making the vision system robust against translational variations in the image. Because of the dynamics of the machine, due to radial and axial runout in the machined components or misalignment of the hardware, translation of the images on the screen is inevitable. The translation will in turn compromise the analysis speed of the algorithm. Thus, a neural network was developed in hopes of solving or eliminating the above mentioned problems.

Appendix B details the development and test of the neural network vision system. The neural network was implemented offline rather than operating in real time mode. Basically, images of three different winding errors: no error, gap error, and climb error were captured. The parameters that were varied in the images include the magnification of the camera lens, the x-position of the vision stage, and the y-position of the vision stage. The different magnifications were to simulate the scaling of the image either through the use of different fiber diameters or different magnifications. The different x-y position settings would simulate the translation of the image. Thus, the vision system was trained to accommodate for size and translational variations in the images.

The result of the neural network experiments showed that a learning vision system does in fact improve the performance of the system. Once trained, the vision system was able to correctly identify all of the 36 images of different error types and parameter settings. The time that was required for the neural network to identify and classify all 36 images was 0.5 second, or approximately 0.015 second per image. However, this does not take into account the 11.5 seconds required to read in the image files and to preprocess the images. Even with the preprocessing, the neural network was still able to identify 3 images per second. The results are promising considering this was the first iteration of the neural network algorithm. A different approach could be taken to eliminate the need for preprocessing. This would increase the time

required to train the vision system but should not compromise the speed of the analysis once the system has been trained.

The neural network also showed promise when it was tested for robustness. Nine additional images taken at 1.25x magnification, a value that the neural network was not trained for, were used to test the neural network. The results from this set of experiments showed that the neural network was able to correctly identify the three different error types found in the images. This therefore meant that the neural network vision system would not need to be recalibrated if a new magnification or fiber diameter were to be used. The neural network also proved to be robust against noise in the images. It was noticed that these nine new images contained shadows in the lower region of the picture. Despite this fact, the neural network was not wholly affected, but prevailed in identifying the errors.

8. Appendix A: Neural Network

8.1. Theoretical Background

A single perceptron is an element that computes the weighted w_i sum of the input elements x_i and compares that value to a threshold θ

$$\sum_{i=1}^{N} w_i x_i - \theta \tag{8.1}$$

The result is then passed through a function, f(v) (i.e. hard-limiter, nonlinearity function, or threshold logic), such that the output is bounded, typically between 0 and 1, thus classifying the input into one of two classes.

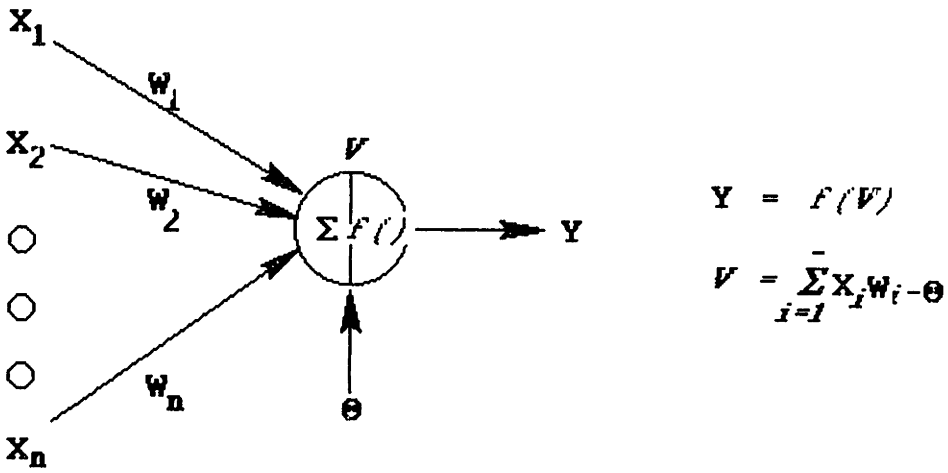


Figure 8.1: Model of a single perceptron

The weights used in the above equation are adjusted according to the following equation:

$$w_i(t+1) = w_i(t) + \eta[d(t) - y(t)]x_i(t) \tag{8.2}$$

Where η is the selected gain, y(t) is the actual output, and d(t) is the desired output. The term [d(t) – y(t)] is the error between the desired and actual values.

A system of single perceptrons can be grouped together to form a neural network of multilayer perceptrons. This system can be organized into different layers that connect the inputs with the outputs. These intermediate layers between the input and output are referred to as hidden layers. Figure 8.2 shows a multilayer perceptron model with three hidden layer.

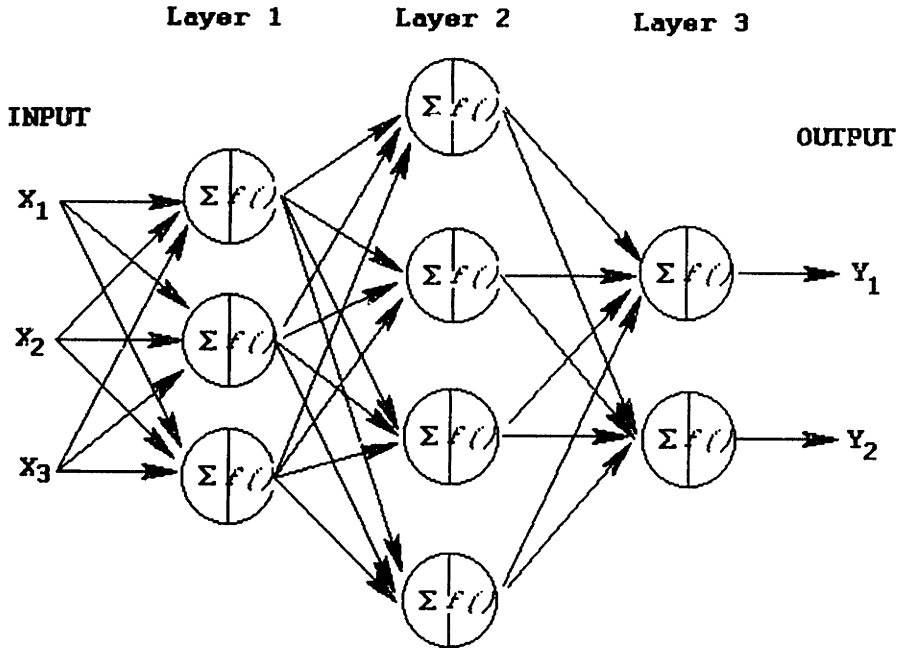


Figure 8.2: Model of a multilayer perceptron

The outputs of each layer are computed as follows:

$$v_j^m = \sum_{i=1}^{N} w_{ji}^m i_i^m \tag{8.3}$$

$$o_j^m = f(v_j^m) = i_j^{m+1} \tag{8.4}$$

$$i_i^m = o_i^{m-1} \tag{8.5}$$

where v_j^m is the state of unit j in layer m

w_{ji}^m is the weight of the connection from unit I (layer m-1) to unit j in layer m

o_j^m is the output from unit j in layer m,

i_i^m is the input to unit layer m from unit I of layer m-1, and

f() can be a hard limiter, threshold logic, or a nonlinearity function

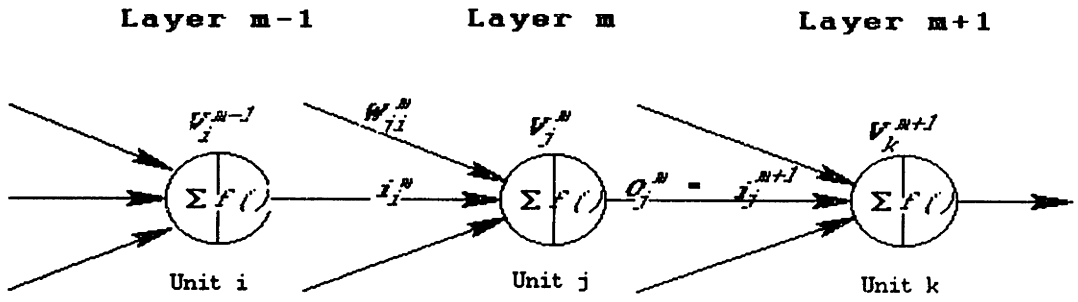


Figure 8.3: Notation for multilayer perceptron model

The weights in the system are adjusted so as to minimize the total error defined by

$$E = \frac{1}{2}\sum (t_l - o_l^N)^2 \tag{8.6}$$

where t_l is the desired output and o_l^N is the actual output from the last layer.

To minimize the sum-squared-error, the steepest descent least-mean-square (LMS) rule is used to adjust the weight ΔW with respect to the back-propagation of the error δ. The following three formulas summarize the calculation of the error and change in weights:

$$\Delta W_{ji}^m = \eta \delta_j^m i_i^m \tag{8.7}$$

For the output layer (m=N)

$$\delta_j^N = (t_j - o_j^N) f'(v_j^m) \tag{8.8}$$

For the hidden layers (1≤ m ≤ N-1)

$$\delta_j^m = \sum_k \delta_k^{m+1} w_{kj}^{m+1} f'(v_j^m) \tag{8.9}$$

The procedure in the training a neural network using back-propagation is to first process the inputs through the network. Then compare the actual output with that of the desired. The difference between the desired and actual output, the error, is then transmitted starting from the final layer back to the first layer. The error values are used to adjust the weights accordingly as it back-propagates to the first layer. The next training set is then processed and the procedure above is repeated until the output error falls below that of the desired threshold.

8.2. Implementation

Implementing a neural network using raw images would require immense computation. A simple 20x20 image would consist of 400 inputs into a neural network. Thus, to process the captured 640x480 image would seem impractical if not impossible. Several options were looked into in order to minimize the number of inputs used in the neural network. Scaling down of the image was one possibility. However, with the reduction of the image size to a manageable size, say 10x10, there would result in the lost of resolution and thus the lost of significant information. Another option was to crop the image to extract the prominent features. However, to accommodate for translational and magnification adjustments in the images, the required field of view (100x100) was still too large to be used as inputs to a neural network.

After researching different journal articles, one method which seemed fairly feasible was adopted for use.[19] The method involved preprocessing of the image using 2D Fourier transform and selecting a 5x5 matrix (25 low-frequency components) from the transform. The advantages to this method are that FT is a standard image processing technique, it reduces the number of neural network inputs with little loss of information, and invariances (translational, rotational, and scaling) can be built into the Fourier transform. The following sections outline the procedure in implementing the multilayer perceptron.

8.3. Image Acquisition

Profiles of the coil, as the fiber was being wound, were captured using a CCD camera producing 640x480, 256-level grayscale images. Each of the 640x480 bytes thus represented the intensity of the corresponding pixel which ranges from 0 to 255. However, for the most part, the profiles were black set against a white

background. For each category of winding error, images were taken with the camera located at 3 different x coordinates, 3 different y coordinates, and at 2 magnifications. The different x and y coordinates would train the vision system to account for translation while the different magnifications would train the system to account for scaling of the images. Thus, the total number of images captured was (3x3x3x2) 54. Of those 54 images, 18 images were used for the training set, while the 36 other images were used for the test set. The settings for the test variables are listed in Table 8.1.

Error Type	Climb Error	No Error	Gap Error
X- position	X_1 = -224.5	X_2 = -234.5	X_3 = -244.5
Y- position	Y_1 = 83.55	Y_2 = 93.55	Y_3 = 103.55
Magnification	1x	1.5x	(1.25x)

Table 8.1: Settings for test variables during image acquisition

8.4. Image Preprocessing

The captured 640x480 images were reduced in size by 5x resulting in images with dimensions of 128x96 pixels. The decision to reduce the image to this size was to reduce the amount of memory space needed to store the images, to allow for ease of file transfer, and to reduce the time required to process the image in the latter stages of preprocessing. Although the size of the image was reduced in size it still retained a fairly high resolution to distinguish among the three categories of errors from the images

The images were then read into Matlab using Matlab's Image Processing Toolbox. The pixels' grayscale values were mapped as intensity values and stored in a matrix format. A 2D Fourier Transform was then applied to each of the images that resulted in a spectrum plot with its corresponding matrix of complex values. From the spectrum, a 5x5 area was selected from the bottom right hand corner. These spectral values were then stor in a 1D array with 25 entries. The complex modulus (magnitude) was then computed for all 25 entries and normalized so that the values fell between 0 and 1. Each set of 25 values served as a set of inputs for the multilayer perceptron. The preprocessing procedure is show in Figures 8.4-8.6.

The following outlines the procedure used to create the training and test sets of images

1. Capture image using CCD camera in the 640x480, 256-level grayscale format
2. Reduce image by 5x resulting in a 128x96 image
3. Read in image using Matlab's Image Processing Toolbox
4. Implement a 2D Fourier Transform on the image and calculate the complex modulus (magnitude) of each pixel (FT values take the form of complex numbers)
5. Extract a 5x5 matrix (lower right corner) from the transformed data
6. Normalize the pixel values so that they fall between 0 and 1
7. Use these 25 normalized pixel values as the inputs into the neural network

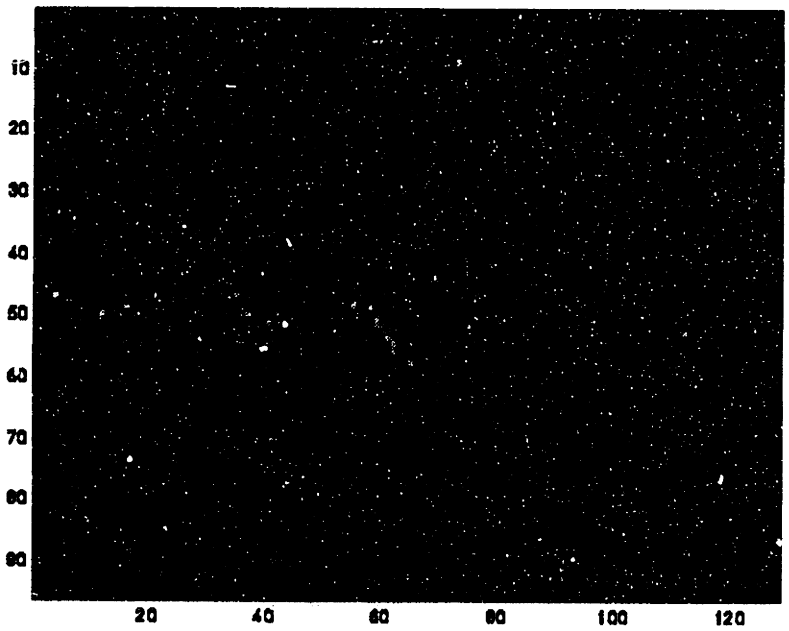


Figure 8.4: Reading Image into Matlab

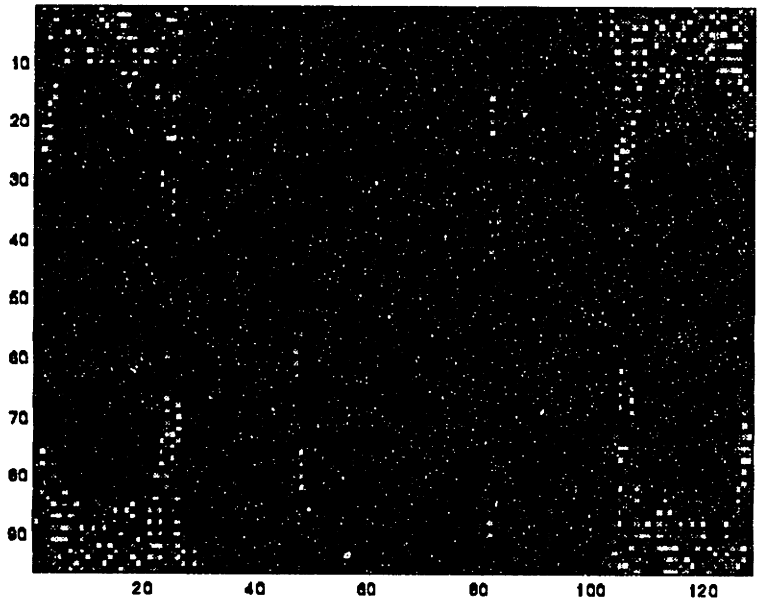


Figure 8.5: 2D Fourier Transform

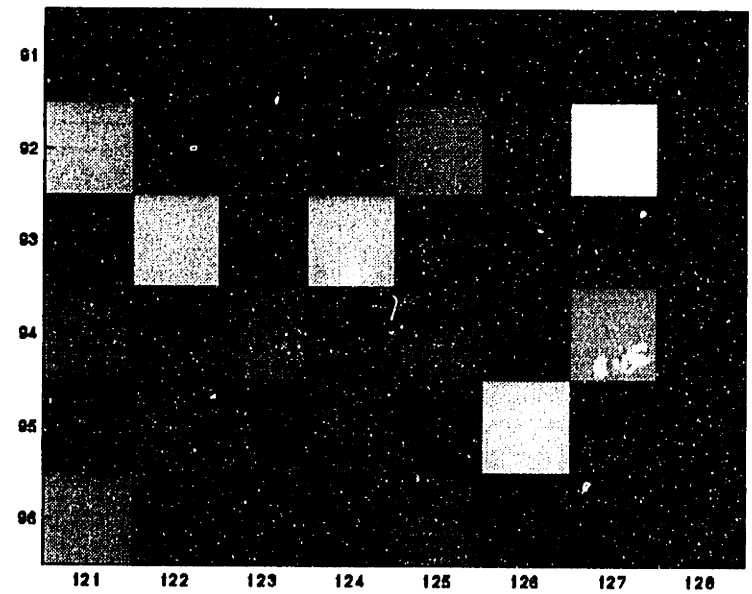


Figure 8.6: Close-up of Pixel Values Selected as Inputs for Neural Network

8.5. *Multilayer perceptron model/training*

The multilayer perceptron consisted of an input layer with 25 nodes, a hidden layer of five nodes, and an output layer of 3 nodes as show in Figure 8.7. The 25 input nodes were the normalized pixel values extracted from the image in the 5x5 square area. Each of the three output nodes represented the three different error types to be identified. A climb error was denoted as [1 0 0]', no error was denoted as [0 1 0]', and a gap error was denoted as [0 0 1]'. These were the desired output for the neural network. The five nodes in the hidden layer were selected somewhat arbitrarily as a transition between the input and output layers. Since the number of input nodes was fairly large, this meant the system would have a large number of weights regardless of the number of hidden nodes. Thus, a large number of hidden-layer nodes to obtain sufficient flexibility for the multilayer perceptron in approximating the classification function was not required.[20]

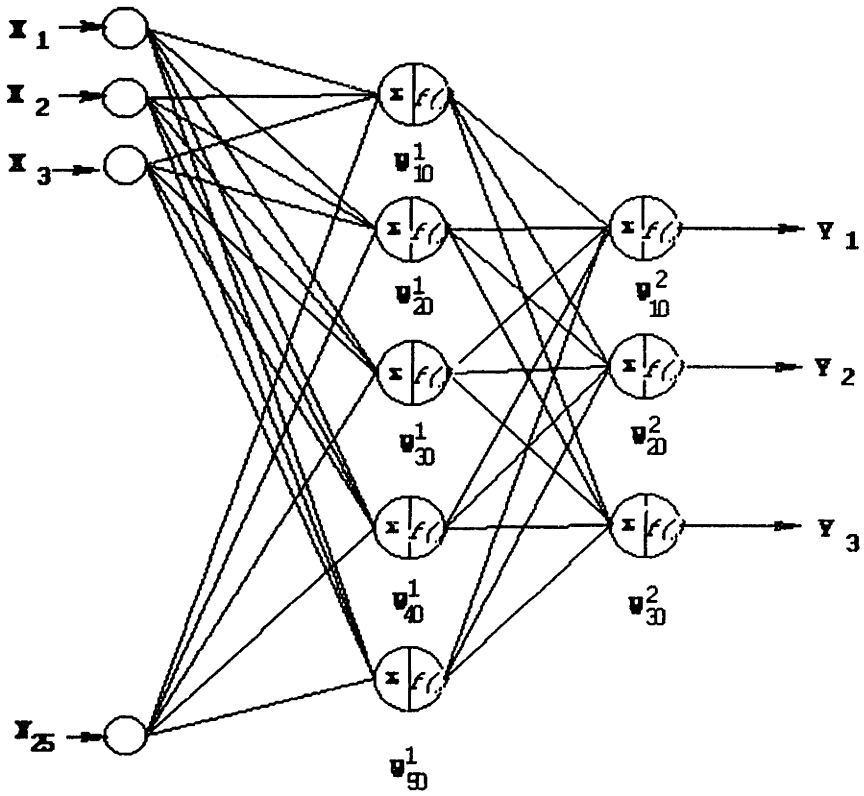


Figure 8.7: Model of the Neural Network Vision System

Each of the 18 images in the training set were presented to the network and the weights were adjusted using error back-propagation after each iteration. The initial training session (a training session is the complete training using all 18 training sets) had a learning rate of 1, a sum-squared–error (SSE) target of 0.005, and a test period of 1000 epoch. However, the network failed to converge using these settings. The learning rate was varied in subsequent training sessions in attempts to have the system converge. That too was unsuccessful although an "optimal" learning rate was found for the system.

As a second course of action, a longer training period was used. The training period was extended to 15000 epochs. Using these parameters the neural network did in fact converge. A smaller sum-squared-error of 0.0005 was then set as a means of making the system more robust. Fig. 8.8 shows the final simulation of the neural network training session. Convergence with an SSE of 0.0005 was achieved after 14029 epochs using a learning rate of 1.5. It took approximately 6 seconds to preprocess all of the 18 images. The training time itself was approximately 332 seconds.

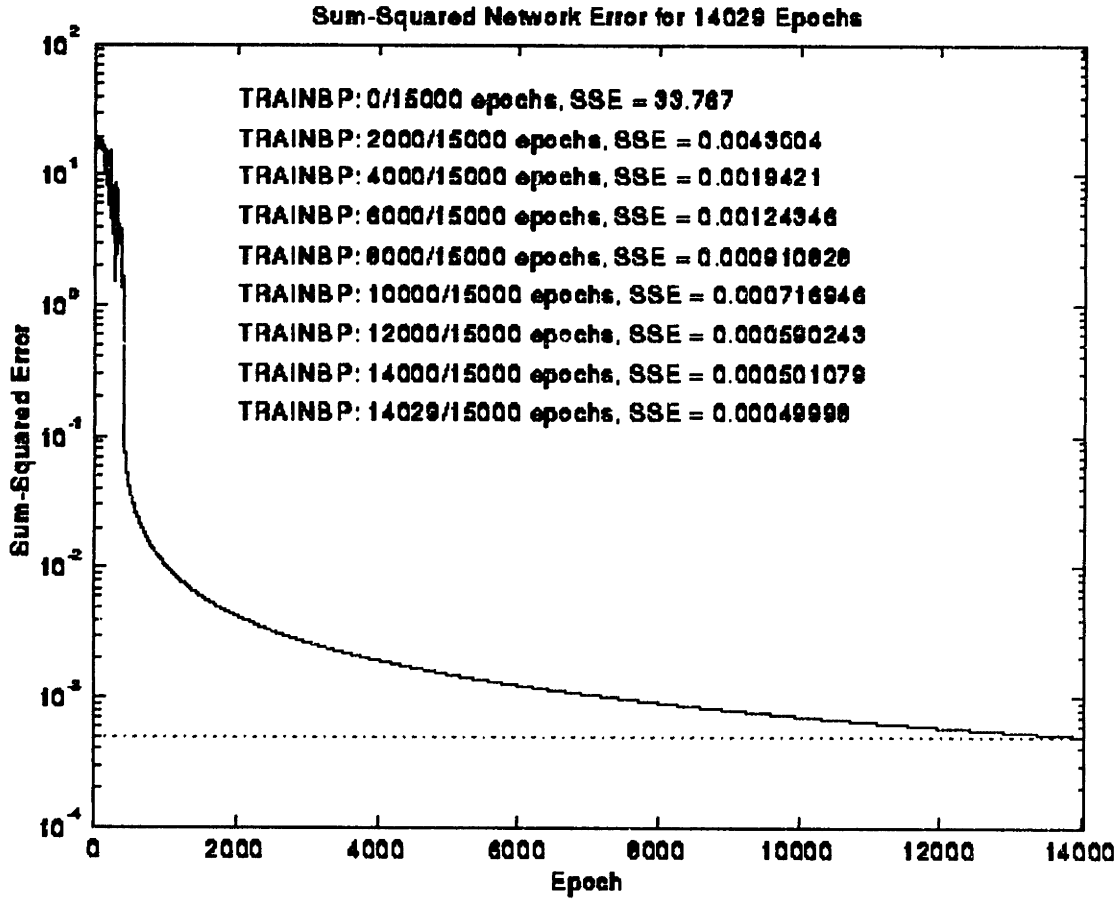


Figure 8.8: Simulation of the Neural Network Training Session

8.6. Results

Using the set of weights and threshold values obtained from the training session, 36 new images were put through the multilayer perceptron for error identification and classification. The 36 images varied in the type of error, x and y location, as well as magnification. The neural network was able to correctly classify all of the 36 images with relative ease giving 100% success rate. The output of the 36 test sets are shown in Table 8.2. The time required for the neural network to identify 36 images was only 0.5 second. The preprocessing of the 36 images, however, took approximately 11.5 seconds in total.

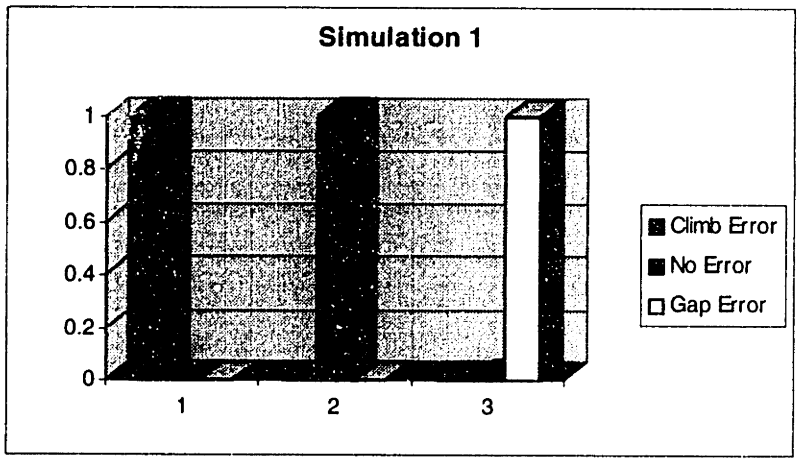


Figure 8.9: Visual Representation of Output

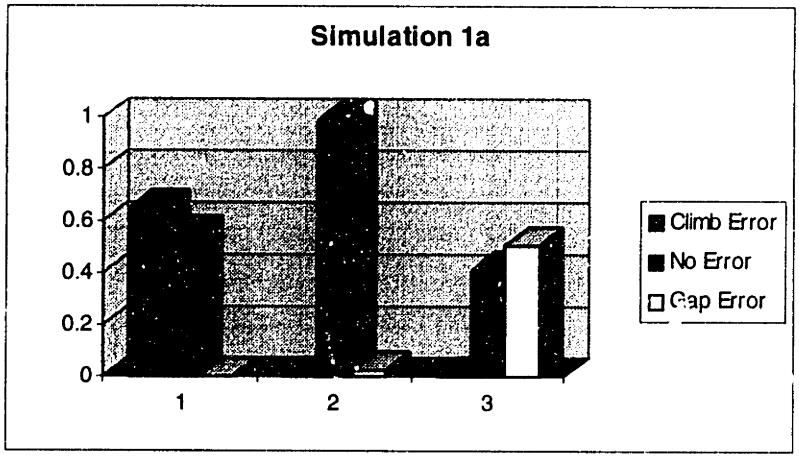


Figure 8.10: Visual Representation of Neural Network Results

	Climb Error	No Error	Gap Error
Vx = -224.5	0.9968	0.0027	0.0001
Vy = 93.55	0.0012	0.997	0.0073
Magnification = 1	0.0003	0.0021	0.9946
Vx = -224.5	0.9985	0.0025	0.0002
Vy = 93.55	0.0051	0.9975	0.0003
Magnification = 1.5	0	0.0018	0.9991
Vx = -244.5	0.996	0.0019	0.0001
Vy = 93.55	0.002	0.9952	0.0022
Magnification = 1	0.0002	0.0034	0.9982
Vx = -244.5	0.9995	0.0029	0.0048
Vy = 93.55	0.0022	0.9976	0
Magnification = 1.5	0	0.0012	0.9973
Vx = -234.5	0.9942	0.0034	0.0001
Vy = 103.55	0.0031	0.998	0.0048
Magnification = 1	0.0002	0.0009	0.9963
Vx = -234.5	0.999	0.0032	0.0028
Vy = 103.55	0.0031	0.9952	0
Magnification = 1.5	0	0.0025	0.9989
Vx = -224. 5	0.9921	0.0033	0.0001
Vy = 103.55	0.0025	0.996	0.0068
Magnification = 1	0.0005	0.002	0.9951
Vx = -224.5	0.9987	0.0025	0.0002
Vy = 103.55	0.0042	0.9972	0.0004
Magnification = 1.5	0	0.002	0.9994
Vx = -234.5	0.9972	0.0023	0.0001
Vy = 83.55	0.0016	0.9983	0.0047
Magnification = 1	0.0002	0.0013	0.9964
Vx = -234.5	0.999	0.0018	0.0156
Vy = 83.55	0.0031	0.9887	0
Magnification = 1.5	0	0.0097	0.9984
Vx = -244.5	0.9982	0.0019	0.0001
Vy = 83.55	0.0011	0.996	0.0038
Magnification = 1	0.0001	0.0029	0.9965
Vx = -244.5	0.9995	0.003	0.0332
Vy = 83.55	0.0021	0.9959	0
Magnification = 1.5	0	0.0018	0.9954

Table 8.2: Results of 36 Test Images Identified by the Neural Network

To further test the robustness of the neural network, nine additional images were captured at settings that were not included in the training set. The x, y positions were set within the range used in the training set

while the magnification was set to 1.25x, which was also within the range of values used in the training set. The results are tabulated in Table 8.3. All of the nine images were correctly classified for errors. However, the degree of certainty in identifying the errors was not as great as with the original set of images. For one of the climb errors, the output was approximately [0.65 0.55 0]' instead of the expected [1 0 0]'. Likewise, for one of the gap error, the output was approximately [0 0.4 0.5]' instead of the expected [0 0 1]'. Although the ratio of the output matrix did not seem to correspond well to the desired, the output value representing the error type was still found to be the largest among the three values. In the case of the climb error, it was the first value while in the case of the gap error it was the last value.

	Climb Error		No Error		Gap Error
Vx = -224.5	0.6483	Vx = -224.5	0.0011	Vx = -224.5	0.0007
Vy = 93.55	0.5493	Vy = 103.55	0.9815	Vy = 83.55	0.4106
Magnification = 1.25	0.0001	Magnification = 1.25	0.0233	Magnification = 1.25	0.5025
Vx = -234.5	0.999	Vx = -234.5	0.0023	Vx = -234.5	0.0033
Vy = 93.55	0.0027	Vy = 103.55	0.9972	Vy = 83.55	0.0078
Magnification = 1.25	0	Magnification = 1.25	0.0021	Magnification = 1.25	0.8831
Vx = -244.5	0.9994	Vx = -244.5	0.002	Vx = -244.5	0.0013
Vy = 93.55	0.0021	Vy = 103.55	0.9987	Vy = 83.55	0.0016
Magnification = 1.25	0	Magnification = 1.25	0.0012	Magnification = 1.25	0.9851

Table 8.3: Results of 9 Test Images at 1.25x Magnification

There are two explanations for the discrepancy found in the output matrix. First, the neural network was not trained with images taken at 1.25x magnification. Therefore, one would not expect the neural network to able to discriminate between the different error types. However, the neural network was robust enough to extrapolate the new images and classify them using the existing information obtained during the training session. The second reason for the discrepancy in the output values is that the new images contained noise. It was found, after the fact, that the new set of images had shadows on the lower edge of the picture. This could have affected the performance of the neural network. However, despite the noisy image, the neural network prevailed in identify the different errors.

8.7. Conclusion

It was found that a multilayer perceptron could be trained to successfully identify and classify different types of winding errors that arise during the operation of an automated coilwinder. The training required only 18 images for the neural network to converge to the specified sum-squared-error value. Once trained, the neural network was able to identify and classify a test set of 36 images with 100% success. The network was successful despite translational variances in the image and also changes in the magnification as well. The neural network was also able to classify a set of nine additional images, for which it was not trained, with 100% success. This therefore proved the robustness of the neural network vision system in being able to operate under conditions that it is not fully familiar with and operate without the need for recalibration.

9. References

1. "Fiber-Optic Gyroscopes," Craig T. Herdman, Honeywell Technology Center.
2. "Fiber-Optic Gyroscopes. History, Theory and Application to Naval Systems," LCDR M.H. Nauck, Multi-RedCom Technical Training Session, 1997.
3. Crandall, Karnopp, Kurtz, Pridmore-Brown, *Dynamics of Mechanical and Electromechanical Systems*, Robert E. Krieger Publishing Company Inc., FL 1982.
4. "Development Issues for Quadrupole-Pattern Optical-Fiber Coil-Winding Machinery for Interferometric Fiber Optic Gyro Manufacture and Automation," Thomas L. Defazio, Draper Laboratories, 1992.
5. "Fundamentals of the Interferometric Fiber-Optic Gyroscope", Herve C. Lefevre, Photonetics.
6. "Inertial Rotation Sensing Using a Fiber Sagnac Interferometer," James Lawrence Davis, Thesis (Ph.D.), MIT, Department of Electrical Engineering. 1981.
7. V. Vali and R.W. Shorthill, "Fiber Ring Interferometer," Applied Optics, 1976 (pp. 15, 1099).
8. United States Patent, "RPM Measuring Device Utilizing an Optical Fiber Coil and Winding Method for Making the Coil," Klaus U. Baron, Heidelberg, Eberhard Kiesel, Edingen, Patent Number 4,781,461.
9. United States Patent, "Fiber Optic Sensing Coil," Litton Systems Canada Limited, Patent Number 4,793,708, Dec. 27, 1988.
10. "Exploratory Studies of Optical Fiber Gyro Coil Winding Automation," Kendall L. Belsley, Ronald H. Smith, Optelecom, Inc.
11. "The Orthocyclic Method of Coil Winding," W.L.L. Lenders, Phillips Technical Review, Volume 23, October, 1962, No. 12 (pp. 365-404).
12. "The Design of an Automated Fiber Optic Coil Winder", Brian Sonnichsen, Masters Thesis, MIT, Mechanical Engineering Department. 1997.
13. "Design and Development of an Automated Fiber Optic Gyroscope Coil Winding Machine", Stephen Lin, Masters Thesis, MIT, Mechanical Engineering Department. 1997.
14. Vernon, David. *Machine Vision: Automated Visual Inspection and Robot Vision.* Prentice Hall. London. 1991
15. Shugan, Beth. "Introduction to Machine Vision". Vision Magazine vol.2. 1996 p.6.
16. Galbiati, Louis. *Machine Vision and Digital Image Processing Fundamentals.* Prentice Hall. New Jersey. 1990.
17. McKee, James. *Machine Vision Adhesive Inspection Technology Assessment* vol. 2. 1991.
18. Gofton, Peter. *Mastering Serial Communications.* Sybex. San Francisco. 1986.
19. Embrechts, Harrson, Kraft, Millard, Sankaran. *Intelligent Engineering Systems through Artificial Neural Networks* (vol. 5). "Preprocessing for the Inspection of Printed Wiring Boards with Neural Nets." 1995.
20. Welstead, Stephen. *Neural Network and Fuzzy Logic Application in C/C++.* John Wiley & Son, Inc. New York. 1994.