DEVELOPMENT OF A LASER BASED

TEXTURE MEASURING SYSTEM

RESEARCH PROJECT #2981

2981-1

PROJECT TITLE: DEVELOPMENT AND IMPLEMENTATION OF A PAVEMENT AND SURFACE TEXTURE MEASUREMENT SYSTEM

THE UNIVERSITY OF TEXAS AT ARLINGTON TRANSPORTATION INSTRUMENTATION LABORATORY

Research Supervisor: Roger S. Walker, Ph.D., P.E
Serial No. 3154

Date: January 2001

2981-1

Notice – The United States Government and the State of Texas do not endorse products or manufacturers. Trade or manufacturers' names appear solely because they are considered essential to the object of the report.

IMPLEMENTATION STATEMENT

The project has been successful in developing a high-speed instrumentation system for macrotexture measurements and a laboratory system for static field and laboratory microtexture measurements. Both systems have been made available to TxDOT for implementation.

The high speed texture system was designed to work with the new TxDOT VNET concept for PMIS data collection activities.

DISCLAIMER(S)

ACKNOWLEDGEMENTS

The author would like to acknowledge students and staff personnel at the Transportation Instrumentation Laboratory facility at the University of Texas at Arlington. Special acknowledgement of Dr. Stephen Underwood for his work on the microtexture scanner system and of Brian Burgess for his work with the PCLIM. Also, recognition is due to Wenming Kuo and Ashutosh Nagarkar for their work with the TXTR and WinTXTR programs. Finally, acknowledgment should be given Brian Michalk, Project Director, and Carl Bertand of TxDOT for making this project possible.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction and Project Background

1.0 Introduction

This report details findings from Research Project 2981, "Development and Implementation of a Pavement and Surface Texture Measurement System" sponsored by the Texas Department of Transportation (TxDOT).

1.1 Project Objectives

The project was initiated in order to provide an automated means by which pavement and surface measurements could be collected using a laser based measuring system and preferably at highway speeds. The measurement system to be designed and prototyped was to use non-contact sensors in order to accommodate the highway speeds. After obtaining the raw measurements readings, the system was to measure and quantify aggregate and pavement texture values. Efforts would then be to verify or correlate the statistic or statistics from the texture data to existing standards. The research would address measurements for both microtexture and macrotexture. In the original objectives the system was designed to address the possible implementation of the instrumentation either on the existing skid systems, the Siometer/rut bar systems, or on a stop and go operation such as the Falling Weight Deflectometer or FWD.

During the course of the project the Siometer/Rutbar systems were replaced with the Texas Profiler systems. Additionally, TxDOT changed its data collection procedure so that most of the real-time instrumentation systems that are planned for use with the PMIS data collection would work directly with Windows using sockets for communications. Thus the objectives were modified accordingly.

1.2 Project Accomplishments

The project has been successful in developing a high-speed instrumentation system for macrotexture measurements and a laboratory system for static field and laboratory micro-texture measurements. Both systems have been made available to TxDOT for implementation. This report provides details on these two systems.

The high speed system is based around the Selcom laser. Specifically, it is designed to capture the raw digital laser readings directly from the Selcom laser, without further D/A, or A/D operations. It directly interfaces to the Selcom 78Khz texture laser system and is capable of accommodating much higher data rates should sampling rate improvements be made by Selcom and using their standard interface. The system developed in the project includes a laser power and interface module, and a data collection board that permits the direct inputs into a PC or embedded PC architecture via the ISA or PC104 bus from the Selcom laser. A complete description of the system is included in the report.

The high speed system was initially developed using a DOS based embedded PC. The objectives of the project were later changed by TxDOT requiring the system to run under a Windows based system. The Windows based system was structured so that the embedded texture processor would provide a server function, reporting results to a client on a separate system via the Ethernet and sockets. The client-server concept and software procedures are described in the report.

A slow speed microtexture measuring system was also developed. Initially, it was planned to use a faster microtexture measuring system using an Aromat high-resolution laser, and a displacement control system offered by Motion Dynamics. It was planned to place this system on stop and go measurement systems, such as the FWD. For instance, following each set of FWD measurements, microtexture readings could also be provided while stationary. Unfortunately, Motion Dynamics could not deliver the system ordered. This resulted in a year's delay in obtaining the microtexture system as well as, resulted in the purchase of a much slower high-resolution measurement system than originally anticipated. The slow speed system was obtained and tested. It soon became obvious that the system would only be practical for limited field use, although it appears well suited for laboratory measurements. The microtexture system is available for implementation and described in the report.

Several summary statistics were developed and investigated for quantifying the texture readings. The statistics were then used to investigate their suitability for estimating pavement friction as measured by skid number.

1.3 Background

The Texas Department of Transportation (TxDOT) skid systems are presently being used to collect wet skid resistance data. These measurements are based on skid number and polish values. It is generally believed that the skid number is a combination of the pavement surface microtexture and macrotexture. There has not been a good correlation obtained from the polish values and skid numbers and pavement performance.

The skid resistance, or frictional properties, of a pavement surface in contact with a given tire under similar conditions has been shown to be largely determined by the texture of the surface [1-5]. It is generally believed that texture consists of two components: macrotexture and microtexture.

Macrotexture is defined as the deviations of a pavement surface from a true planar surface with the characteristic dimensions of wavelength and amplitude from .02 in. (0.5 mm) up to those that no longer affect tire-pavement interaction [6]. It is generally believed that macrotexture provides the drainage on wet pavement and determines the rate that skid resistance decreases with increasing speed. Pavements with more macrotexture typically show a smaller rate of decline of skid resistance with increasing speed.

Similarly, surface microtexture is defined as the deviations of a pavement surface from a true planar surface with characteristic dimensions of wavelength and amplitude less than 0.02 inch (0.5 mm) [6]. The sharp, fine particles in a surface which make up the microtexture penetrate the thin water film and permit contact of the tire and the pavement surface. At low vehicle speed the microtexture is considered by many the predominant factor in determining the skid resistance of a pavement. Macrotexture is somewhat easier to measure because of the greater size.

It is apparent that both macrotexture and microtexture should be considered when determining the skid resistance of a pavement. Measures of macrotexture have been reported using optical techniques, pavement profiles, and the sand-patch method [5,7]. Microtexture measures are often reported using the British pendulum number [5]. Researchers have reported on results obtained from profiles using optical techniques or a mechanical stylus but these profiles are difficult to obtain [1,5]. Lasers provide very accurate distance measurements and are now being used for many such measurements. A complete compilation of all the many studies for macro and micro texture are not included in this report. A good report of much of the early work, as well as recent work, is provided in Reference [14].

1.4 Previous Texture Research at UTA Using Laser

A pilot study was conducted [11] several years ago at The University of Texas at Arlington with the TxDOT, where one of the first Selcom texture lasers was tested to determine its possible use for measuring texture. The purpose of the study was to investigate the use of the Selcom 2301 laser as a means of obtaining macro- and microtexture measurements. Efforts were then made to relate these measurements to skid resistance numbers obtained from skid calibration sections at the Central/Western Field Test and Evaluation Center at the Texas Transportation Institute in Bryan, Texas. The study was not extensive and resources were not available to obtain additional skid data other than the skid numbers provided for the three skid sections. This study was made possible by Selcom as they provided a laser for testing at no cost.

The 2301 Selcom laser was used to measure pavement texture in the study. The Selcom 2301 used had a standoff distance of 3.74 in. (95mm), a measurement range of .315 in. (8mm), and resolution of .079 mils(0.002mm).

1.5 Experiment Conducted

The actual data collected in the field was obtained from the three skid sites at the Texas Transportation Institute at Bryan, Texas. These sites were selected because recently obtained skid numbers using the locked-wheel method [6] and ribbed tires were available for comparison.

The maximum update rate of the output from the Optocator Interface module was 8 KHz. The data was collected at approximately 1 mile per hour thus providing 457 samples per inch (18 samples per mm). Prior to actually collecting data on pavement surfaces, a simple laboratory test was conducted to test the resolution of the lasers. Four different types of sandpaper were used with grits of 36, 50, 80, and 120. The sandpaper was passed under the laser at a speed that approximated 1 mph (1.6 km/hr) and data was sampled at a rate of 457 samples/in. (18 samples per mm). The results obtained clearly show that sandpaper with different size texture could be differentiated using the laser as a measuring device.

In preparation for collecting the data, a representative 15 foot (4.6m) section of the particular pavement type to be measured was selected. A trace was marked to obtain repeat runs and 2 markers were placed on the path at approximately 10 and 11 feet (3 and 3.4m) to validate the collection speed. The laser probe was mounted on a section of an Ames profilograph. Five repeat runs on each of the sections were obtained for analysis.

1.6 Data Processing

The first analysis technique applied to the data involved filtering and determining the root mean square (RMS) of the filtered data. In order to obtain a measure of the microtexture, the data was passed through a software filter with a cutoff frequency of 0.5mm. The root mean square of the filtered data was then determined. Similarly, to determine the RMS value associated with the macrotexture, the data was passed through a band-pass Butterworth filter with a low frequency cutoff of 2 1/2 inches (63.5mm) and high frequency cutoff of 19.7 mils (0.5mm). The band-pass filter was used instead of a low-pass filter so that DC and very low frequency components would be removed. The values obtained using this technique to calculate the RMS for microtexture appeared high compared to results reported in the literature so another method of processing the data was examined.

The second technique considered had been used by researchers at Penn State [3,5]. This method calculates the root mean square texture heights of the microtexture and macrotexture. These values are computed from the power spectrum of the data. Using this relation, microtexture is defined in terms of the width of asperities or wavelength. This method produced results consistent with the first method as discussed in the report.

The magnitude of microtexture RMS values were found to be larger than those reported in the Pennsylvania State research. However, it was noted that the profile measuring device used a mechanical stylus with a point size larger than the 3.94 mils (0.1mm) spot size of the laser used in this study. The stylus did not provide the same resolution as did the laser and thus would have the effect of averaging data points. In fact, several sections of the data were rerun with 4-point averaging of the raw data before the PSD was calculated and results in the range of those reported by Pennsylvania State research report were obtained.

Another discrepancy observed was the relatively large RMS values obtained from the jenite with sand pavement. According to SN_0 calculations a microtexture RMS value between concrete and jenite was expected but was not observed. It should be noted, however, that the pavement had considerable cracking and provided significantly more invalid data than did the other pavements as the laser dropped out of the measurement range.

Tables 1.1 and 1.2 provide the results of calculations for micro and macrotexture for one of the sections. The values given here are expressed in mils. Again, a high correlation between the filtered and power spectral density values is observed, indicating that either method could probably be used. Macrotexture values were not considered further because the limited TTI skid data did not provide enough data to accurately study the relationship of macrotexture to PSD or skid numbers.

Microtexture Calculations in Mils/1000

	Concrete		**Jenite**		**Jenite w/ Sand**	
	Filtered	**PSD**	**Filtered**	**PSD**	**Filtered**	**PSD**
	12.0	10.4	6.8	6.4	10.7	9.64
	12.6	11.5	5.3	4.8	9.2	8.3
	8.0	7.0	4.6	4.2	12.8	11.6
	8.7	7.5	5.4	4.9	10.8	9.6
	8.3	7.2	7.4	6.9	11.2	10.3
Overall	**9.9**	**8.7**	**5.9**	**5.5**	**11.0**	**9.9**

Table 1.1 Results from 2 methods of microtexture calculations in mils/1000

MacroTexture Calculations in mils

	Concrete		**Jenite**		**Jenite w/Sand**	
	Filtered	**PSD**	**Filtered**	**PSD**	**Filtered**	**PSD**
	92.1	113.8	77.6	95.3	165.7	207.9
	102.8	129.1	70.4	86.6	181.1	226.4
	76.0	94.5	75.2	92.9	207.8	259.4
	82.2	102.0	79.1	97.6	193.3	260.6
	79.9	99.2	84.5	100.0	168.1	210.2
Overall	**86.6**	**107.9**	**76.8**	**94.5**	**183.1**	**233.1**

Table 1.2 Results from 2 methods of macrotexture calculations in mils

1.7 Conclusions

Results obtained by this initial effort indicated that a more exhaustive study was needed. Many more test sections with skid numbers obtained at several speeds are needed to better relate micro and macrotexture to pavement skid.

In the next chapter the microtexture measurement system is described. This includes information for constructing additional units by TxDOT, as well as operating instructions. Chapter 3 provides details of the macrotexture system. Similar to Chapter 2, details are provided so that additional systems can be developed. The Windows based software and procedures for using the macrotexture system is provided in Chapter 4. Chapter 5 provides details in statistics used for attempting to classify the macrotexture measurements. Chapter 6 summarizes the report and indicates areas for additional studies.

CHAPTER 2

Microtexture Measuring System

2.0 Introduction

This chapter provides a description of the laser system used for microtexture measurements and has been provided to TxDOT for implementation. The system is not for highway speeds because of the measurement distance between the laser and target (pavement), and the sampling rate. However, the resolution selectable to 0.2 micrometer provides for a high definition of the surface of the target. The system can easily be implemented in either a lab environment or on a section of pavement. The microtexture system consists of basically three main components, the laser, the computer controlled laser holder or translation table, and the PC. Figure 2.1 illustrates a block diagram of the system developed which has been named the 3D Microtexture Scanner System. The next section provides greater details on this system and its operation. Figures 2.2 and 2.3 depict the 3D Scanner System. Figure 2.4 depicts the laser translation assembly. Illustrated in Figures 2.5A, B, and C are three plots. The first is a picture of a line joint on the floor in the laboratory. The laser was placed over the joint and 100 scans in each the x and y directions were made where each step was 100 micrometer. Thus a 10mm square was scanned (approx 0.4 inch). The resolution of the laser was 10 micrometers or about 0.4 mils. The second picture illustrates one of the scans as the laser moved across the crack. The final figure is the actual scanned data displayed as a 3D plot.

First the main components of the scanner system are described. Next procedures for setting up the scanner are given. Finally the operating procedures for running the data acquisition program are given. Table 2.1 provides an overview of the main scanner components and their specifications.

Parts List for 3D Microtexture Scanner

	Parts List #'s - Newport Corporation
2	UTM 150 CC1DD Motorized Translation Table
1	MM2000-OPT 0102NNNN03NNN Motion Controller
1	MMD31-OPT 6363NNNN01 Motion Driver
1	M-423 Manual Translation Stage
1	AJS-2 Adjustment Screw
1	360-90 Angle Bracket
	Parts List #'s – Matsushita Electric Works, Ltd.
1	LM200 ANL2334(5) Laser Analog Sensor
1	Controller (see schematic)

Table 2.1 Parts List for 3D Microtexture Scanner

3D Microtexture Scanner System

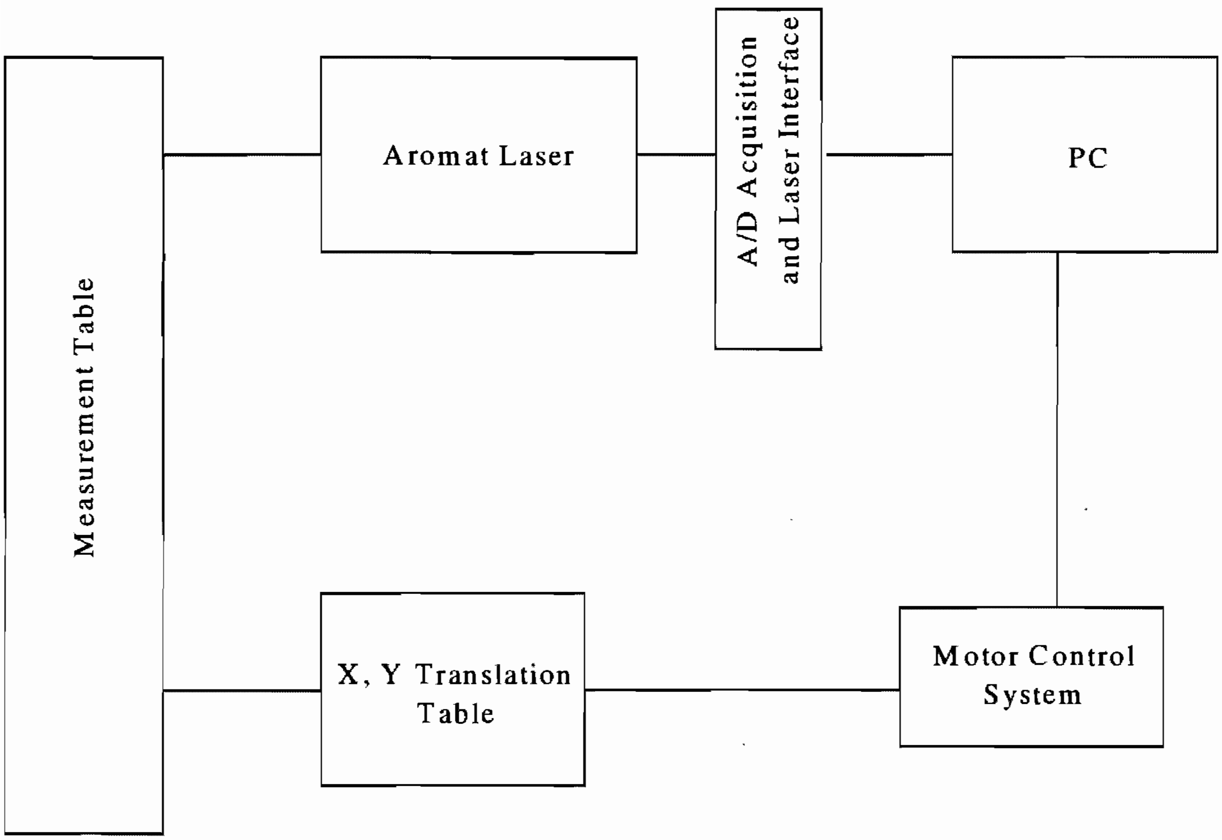


Figure 2.1 3D Microtexture Scanner System

3D Microtexture Scanner Assembly

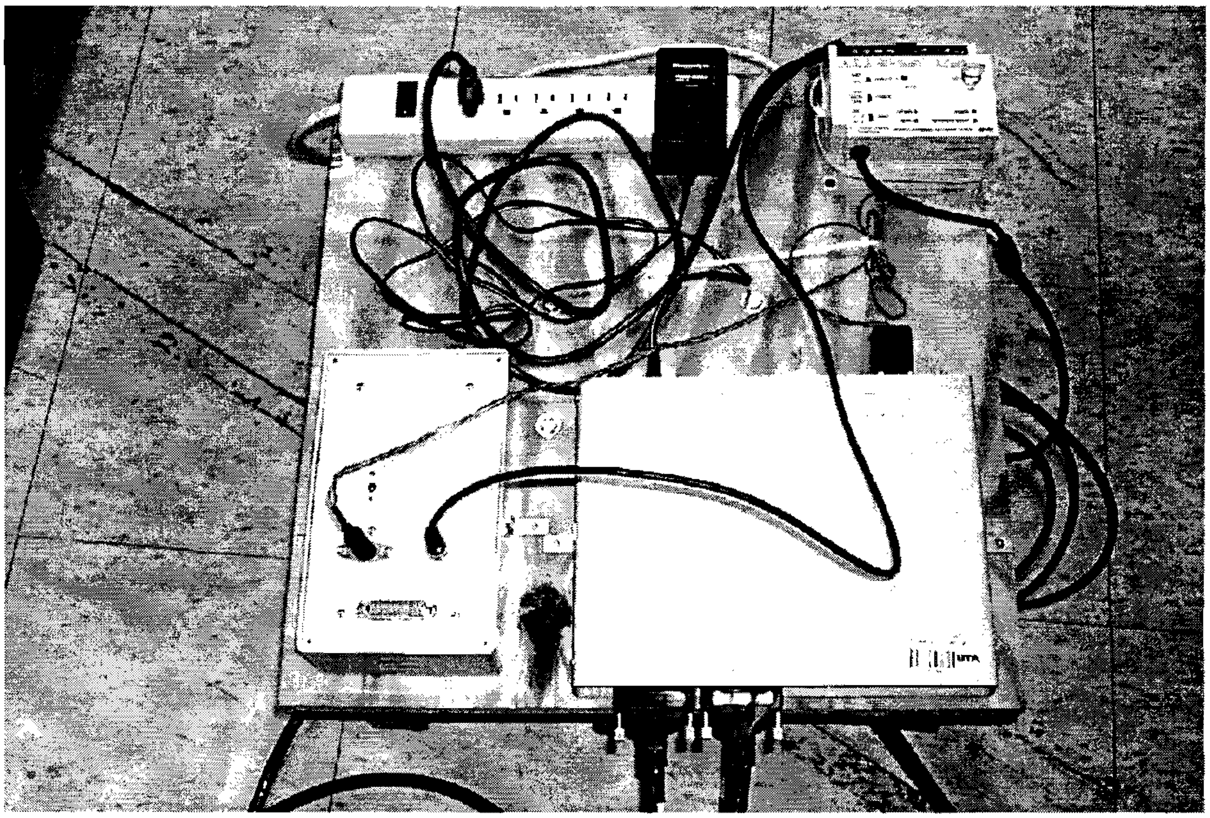


Figure 2.2 3D Microtexture Scanner System Assembly

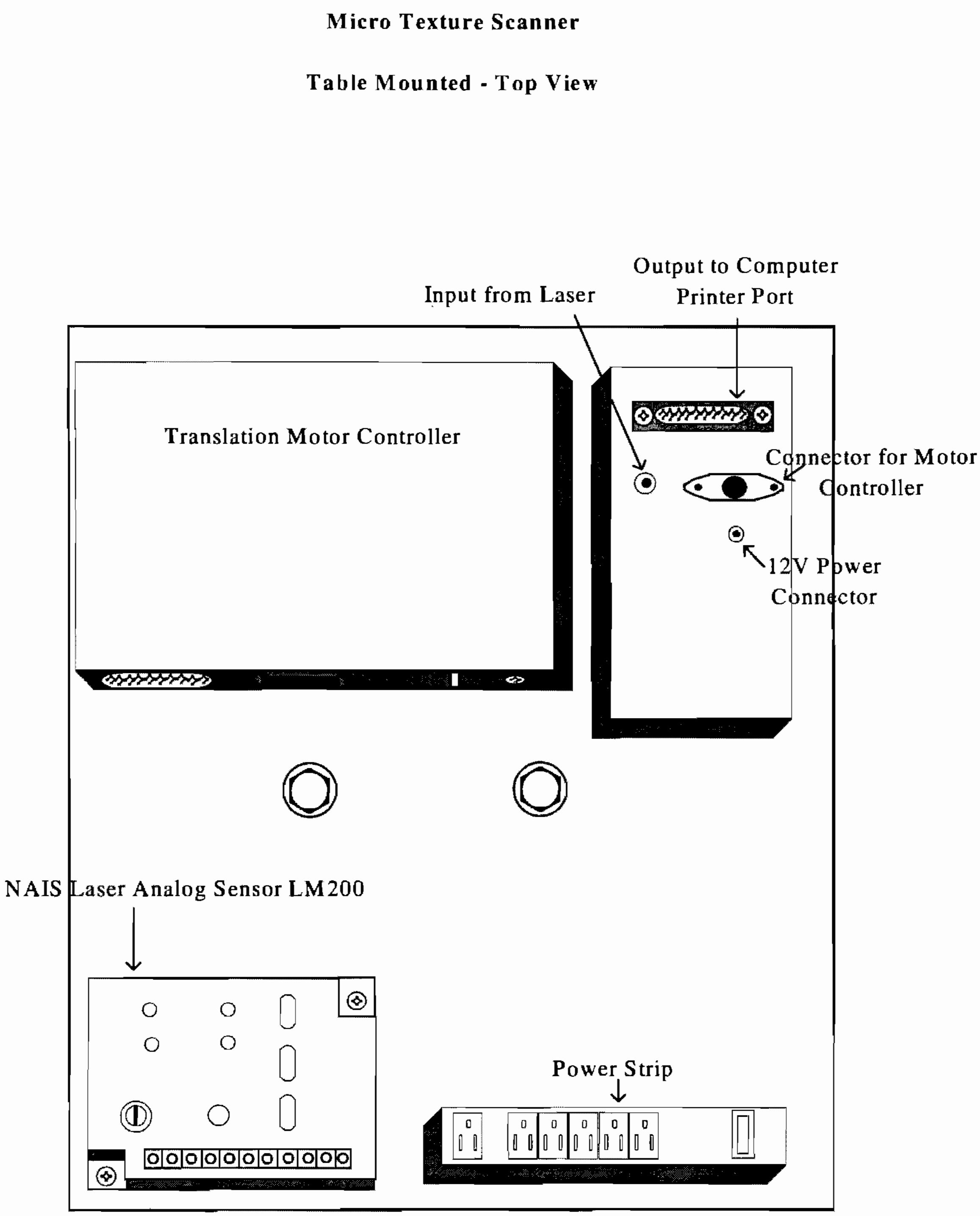


Figure 2.3 Table Mounted – Top View – Microtexture Scanner

Table Mounted - Bottom View

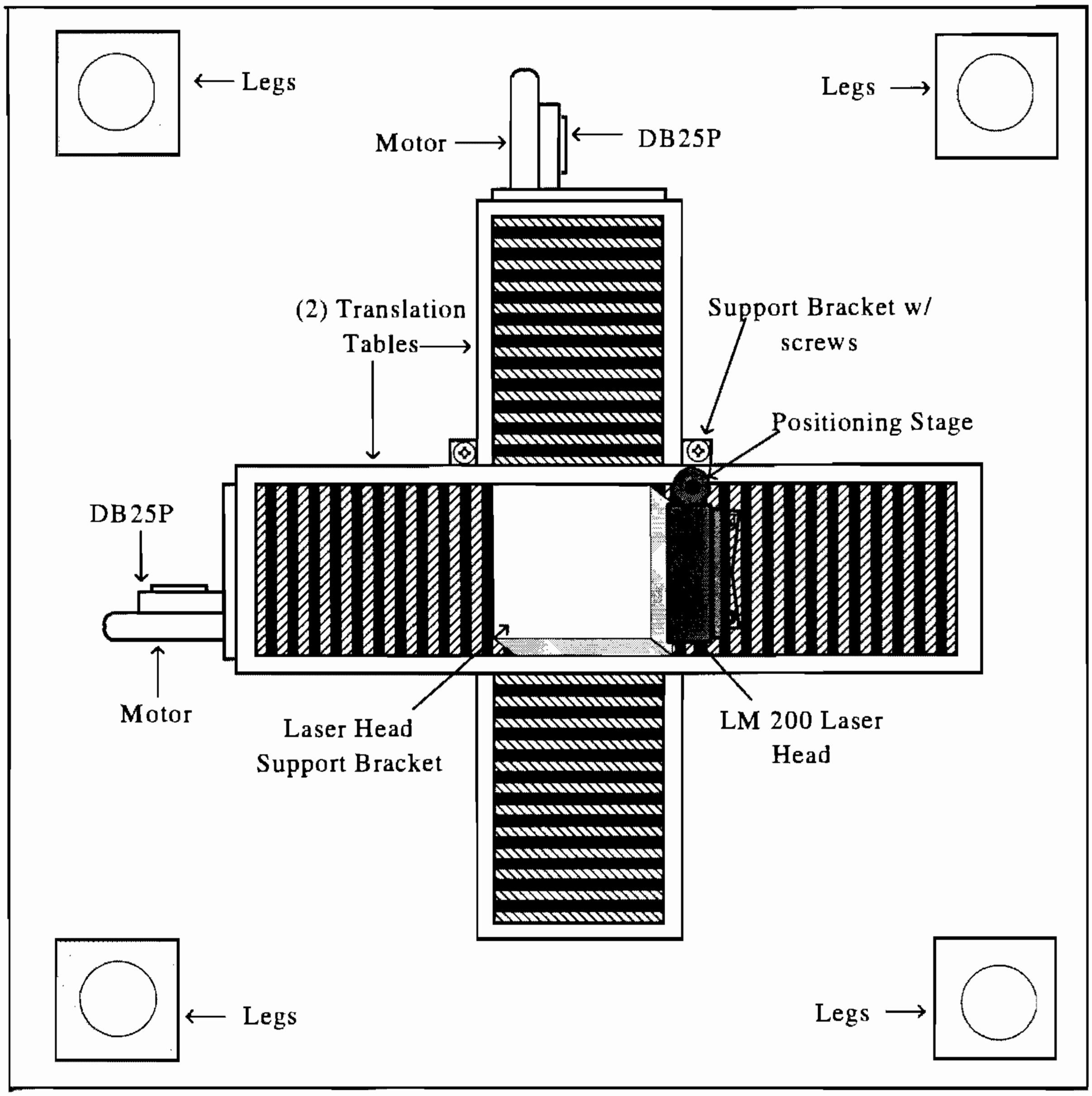


Figure 2.4 Laser Translation Assembly

Commercial Office Tile Joint Intersection

Figure 2.5A Commercial Office Tile Joint Intersection

3D Texture Scan of Two Tile Joints

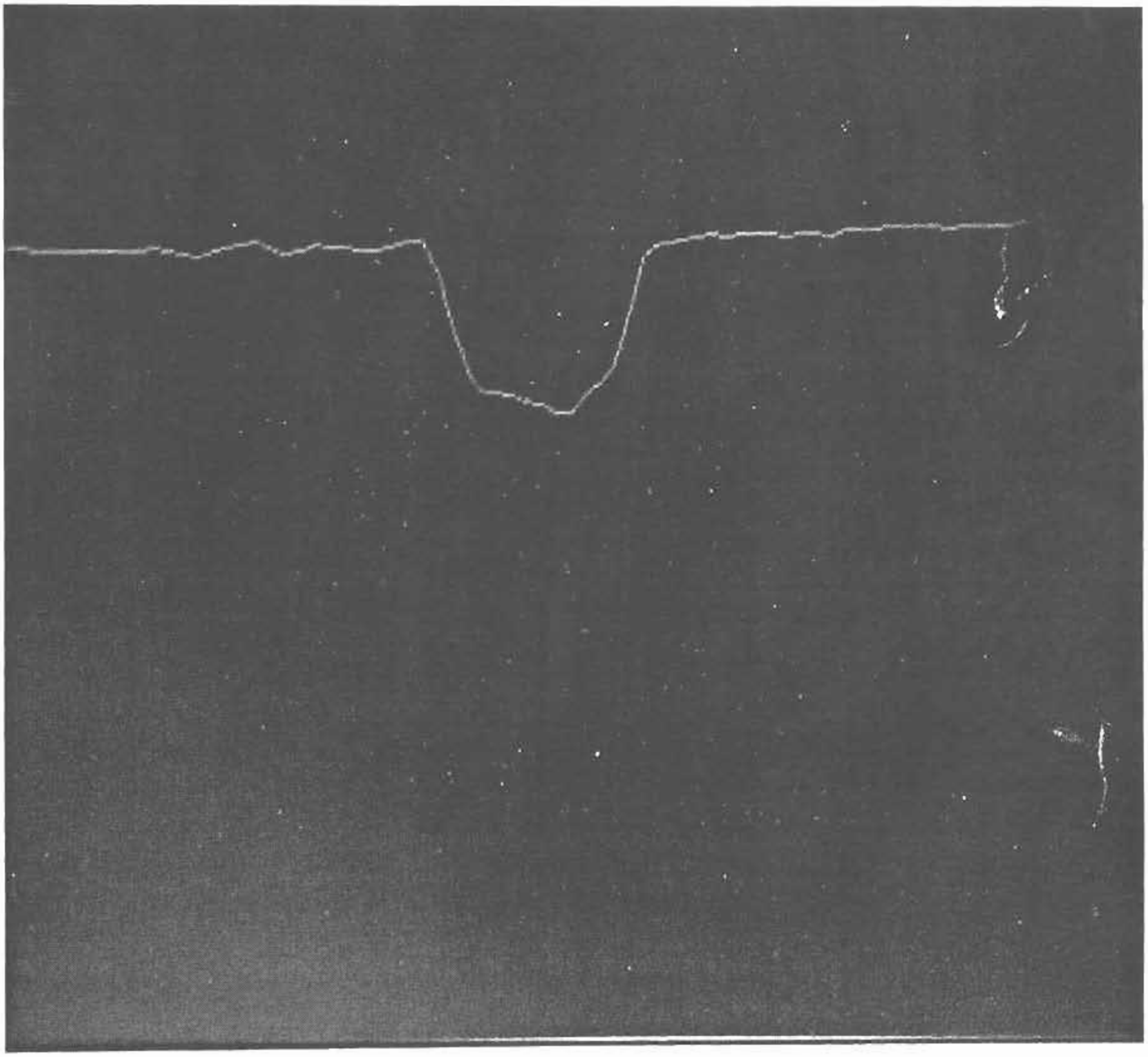


Figure 2.5B 3D Texture Scan of Two Joints

3D Scan of Commercial Office Tile Joint

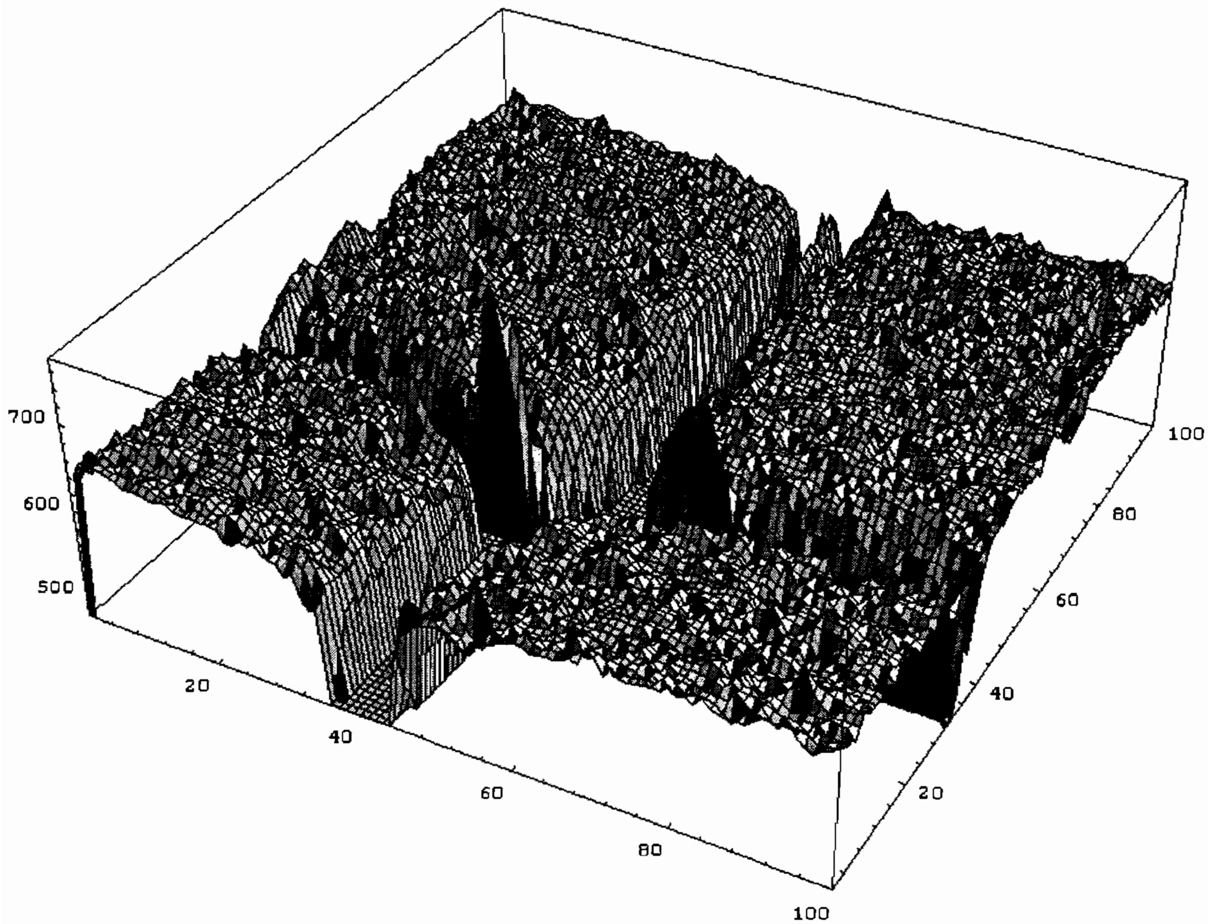


Figure 2.5C 3D Scan of Commercial Office Tile Joint

2.1 3D Micro-Texture Scanner

The Scanner system consists of two 5.9 in. (150 mm) travel translation tables with a precision distance measurement laser. These are illustrated in Figure 2.4 and Figure 2.6. The translation tables will step with a 0.004 mils (one micrometer) step size and at a maximum speed of 0.98 inches (2500 micrometers) per second. The laser will measure distances within a ± .118 in. (3 mm) distance from a 1.18 in. (30 mm) standoff from the target with a 0.007 mils (0.2 micrometer) resolution. The laser spot size will vary from 1.97 mils (.05mm) by 4.72 mils (.012 mm) at .118 in. (3 mm), 2.76 mils (.07mm) by 1.18 mils (0.03) mm at 0 mm, and 05.9 mils (.15 mm) by 2.76 mils (0.7) mm at 0.118 mils (3mm) from the 1.18 in. (30 mm) standoff distance.

For operations, the user specifies the number of X and Y samples and the sample spacing to the scanner program. The software scans the target in the X direction at the various Y positions. It samples the laser output at one micrometer steps in the X scan direction and stores the average sample distance within the specified X sample spacing. The results are stored as an ASCII file on disk.

2.2 Operation Procedures

Equipment Setup:

1. Using a 25-pin E508B1 cable, connect the female end to the 25-pin connector of the lower Translation Table (X axis) and the male end to the AXIS 1 connector of the Universal Interface Box. Using a 25-pin E508B1 cable, connect the female end to the 25-pin connector of the upper Translation Table (Y axis) and the male end to the AXIS 2 connector of the Universal Interface Box.

2. Using the Encoder cable, connect the 15-pin male end to the AUX connector of the Universal Interface Box and the 4-pin male end to the corresponding connector on the A/D Interface Box.

3. Using the shielded Analog Distance cable, connect the red lead to the DISTANCE screw connector and the black lead to the adjacent GND screw connector on the LM200 Laser Controller. Connect the BNC end of the cable to the corresponding connector on the A/D Interface Box.

4. Connect the LM200 Laser Head to the LM200 Laser Controller using the corresponding connectors.

5. Using the Motor Controller Cable, connect one male end to the MM 2000 connector of the Universal Interface Box and the other male end to the ISA Motor Controller Board that is installed in an ISA slot of the computer.

6. Using a DB25 Printer Cable, connect the female end to the corresponding connector on the A/D Interface Box and the male end to an LPT port connector on the computer.

Translation Table Assembly

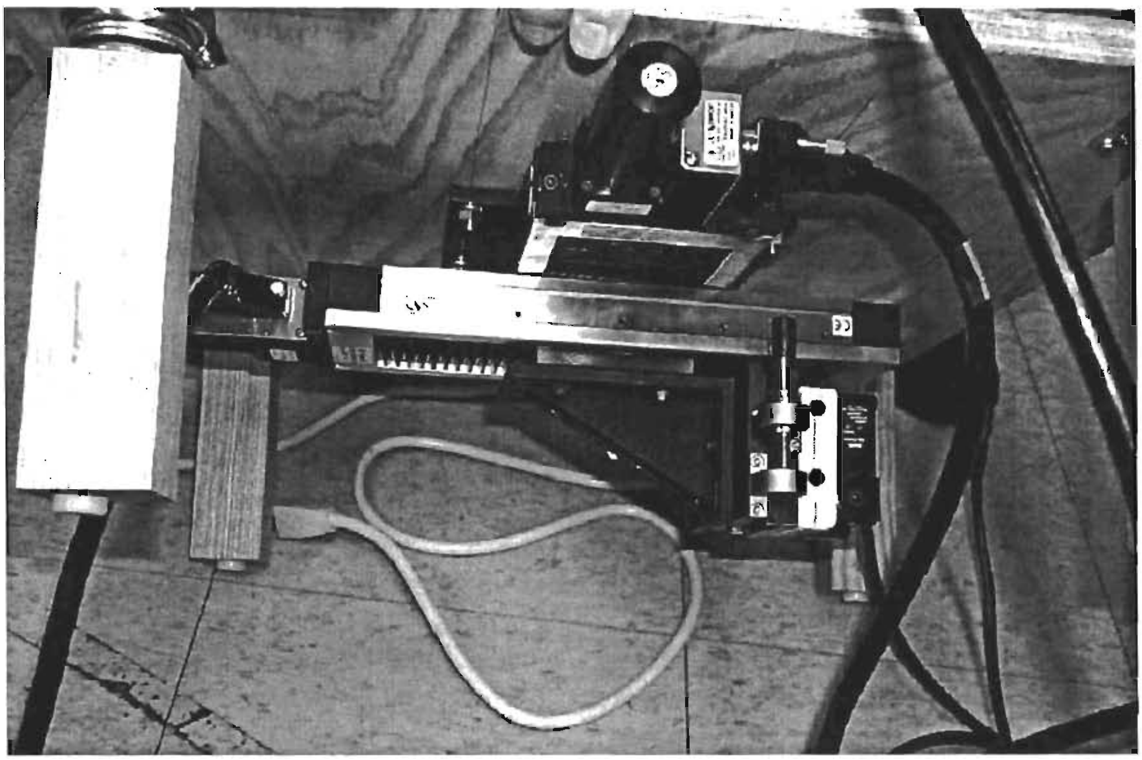


Figure 2.6 Table Mounted - Bottom View – Translation Table Assembly

7. Apply power to the PWR 1 connector of the Universal Interface Box using a Electromech Model 2950 Power Supply. Apply 115 VAC power to the 110-120V AC screw connectors on the LM200 Laser Controller. Using the A/D Interface Power cable, apply a DC voltage between 9 and 15 volts to the A/D Interface Box (center pin, red lead, positive).

Equipment Operation:

1) Place the 3-D Micro Texture Scanner on the horizontal surface that is to be digitized. The laser head must be approximately 30 mm from the surface to be measured (standoff distance). The laser head can be vertically adjusted 18 mm with the translation platform on which it is mounted.

2) Execute the LM200.EXE program on the computer. The program will first initialize the X & Y translation tables by locating their end positions and then positioning each table at their center (mid-point) positions. The program will respond "Motor Initialization Complete" when this task is complete.

3) On the LM200 Laser Controller, place the GAIN switch to AUTO, SPEED switch to 3KHZ, and OUTPUT switch to ±3V. Adjust the Laser Head Translation Platform until the RANGE light on the Laser Head or Laser Controller glows steady. This means that the Laser Head is within 0.5 mm of the 30 mm center standoff distance. A more accurate adjustment can be obtained by adjusting for zero volts at the DISTANCE output screw terminals of the Laser Controller.

4) The program will present the following 4 options:

   a) R = Read a Value: The program will report the distance from the center position in micrometers.

   b) C = Continuously Read Values: The program will continuously report the distance from the center position in μ meters until the Esc key is pressed.

   c) D = Data Collect: The data collection mode will request the following information and then scan the target forward and backward in the X direction in steps of the Y direction. The Z information is displayed on the computer screen during each scan in the X direction. Pressing the Esc key during the scan will terminate the operation. When the scanning is complete, the system will reset to the center position.

      i. Enter the output file name where the data is to be stored. A file name starting with a # will store to the Bit Bucket (no storing). If the data is stored, then the data is converted from binary to ASCII format before it is stored.

ii. Enter the number of X values and the spacing between the X values in μ meters. The X portion of the scan will be centered on the center position of the X Translation Table. The maximum scan length is 150,000 μ meters. Alternating scans in the X direction will be in opposite directions but the data is stored in one direction only.

iii. Enter the number of Y values and the spacing between the Y values in μ meters. The Y portion of the scan will be centered on the center position of the Y Translation Table. The maximum scan length is 150,000 μ meters.

iv. The program will display the scanning information and ask for conformation (Yes or No). A no answer will terminate the Data Collect mode. A yes answer will start the scanning which will continue until the scan is complete or the Esc key is pressed

d) E = End the Program: The program will terminate.

The format of the output file of the program will be:

- Record 1 will contain the number of samples (NX) and spacing (DX) in the X direction.

- Record 2 will contain the number of samples (NY) and spacing (DY) in the Y direction.

- Record 3 through (NX*NY+3) will contain the Z distance in μ meters (one sample per record). The samples are stored in row-major format (x varies within y).

2.3 3D Microtexture Scanner Controller

The Scanner Controller consists of a HC11 microprocessor, a 16-bit A/D converter, and various circuits for synchronization with the MM2000 Universal Interface Box and a LPT port of the PC. Figure 2.7 shows the circuit schematic of the controller which utilizes a Motorola M68HC11EVBU Universal Evaluation Board as the micro-processor system for the controller operation. It provides the A/D conversion of the LM200 analog signal. It places a 32-bit integer representation of the analog signal into a FIFO, which can be read, with the LPT port of the PC. The LPT port is used for sending values and commands to the controller. A bi-directional LPT port is not required to interface with the Scanner Controller.

The following modifications were made to the DB-15F connector on the Newport Motion Master 2000 MMD31 to provide the X-Axis encoder signals to the A/D Acquisition and Laser Interface Controller:

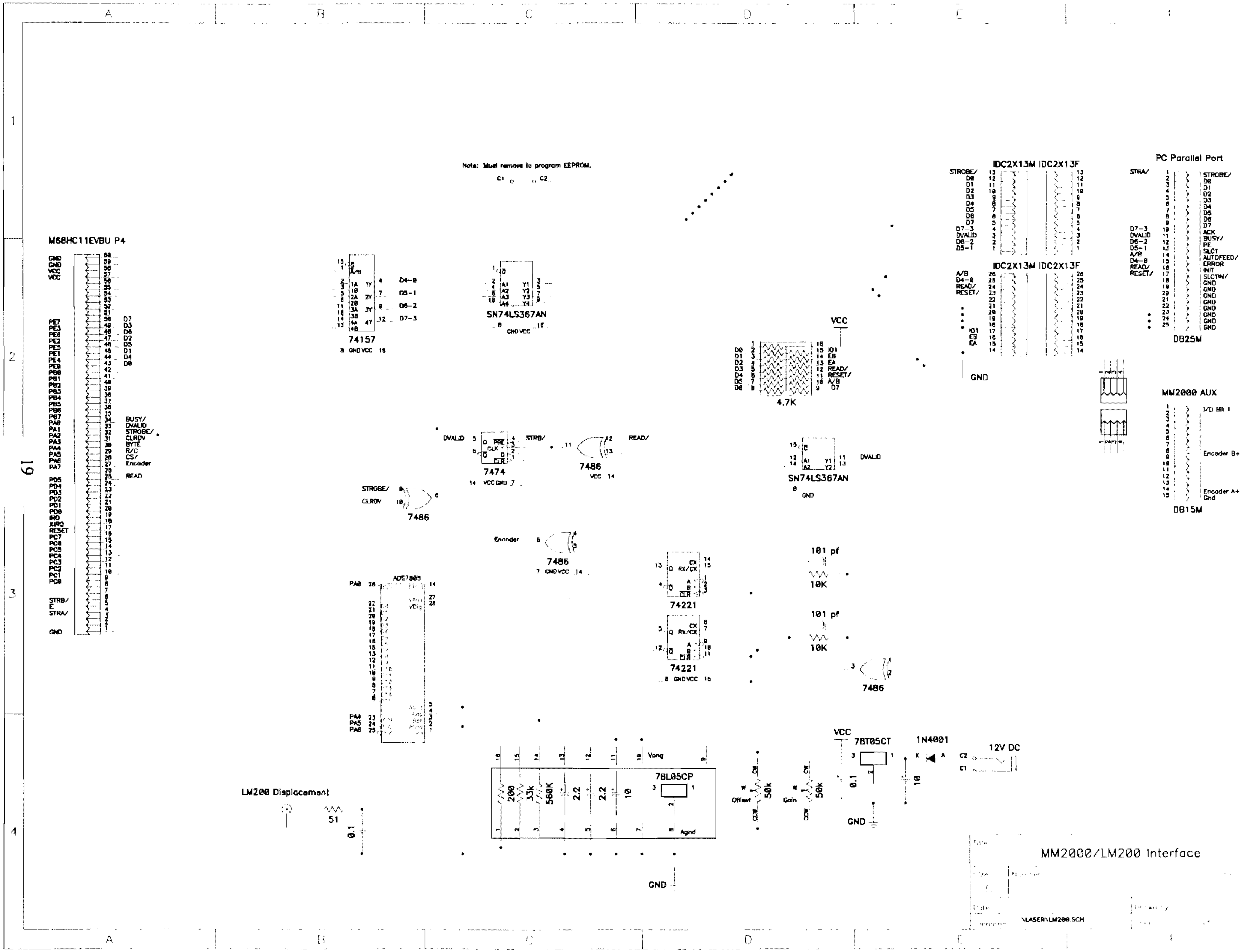


Figure 2.7 Schematic of MM2000/LM200 Interface

Pin 14 was connected to the AXIS 1 Encoder A+ output.

The trace Pin 8 (Reset) was cut and Pin 8 was connected to the AXIS 1 Encoder B+ output.

The assembly language program for the micro-processor in the controller is given on the project program disk. The C-language program for the PC which provides the user interface, sends commands to the controller, reads data from the controller, controls the two translation tables through the Motor Control System, and stores the 3-D data. This program is also on the project program disk.

Following are a list of the commands from the PC to the Controller:

Command 1:

Sample the LM200 Laser once and place the result into the output FIFO.

Command 2:

Start the Accumulation Mode. The controller samples the LM200 analog signal every time the X-axis moves 1 μ meter and sums the 16-bit value to a 32-bit sum. After the X-axis has moved the distance specified by the value read from the Command 4, the 32-bit sum is placed in the FIFO for output and the sum is reset to zero.

Command 3:

The Accumulation Mode is terminated and any motion of the X-axis is ignored.

Command 4:

Read the X sample spacing (in μ meters) from the LPT port.

2.4 3D Micro Texture Scanner Specifications

The 3-D Scanner consists of two 150 mm travel translation tables with a precision distance measurement laser. The translation tables will step with a 1 μ meter step size with a maximum speed of 2500 μ meters per second. The laser will measure distances within a ±3 mm distance from a 30 mm standoff from the target with a 0.2 μ meter resolution. The laser spot size will vary from (0.05 by 0.12 mm at -3 mm), (0.07 by 0.03 mm at 0 mm), and (0.15 by 0.07 mm at +3 mm) from the 30 mm standoff distance.

The user specifies the number of X and Y samples and the sample spacing to the scanner program. The software scans the target in the X direction at the various Y positions. It samples the laser output at 1 μ meter steps in the X scan direction and stores the average sample distance within the specified X sample spacing. The results are stored as an ASCII file on disk.

CHAPTER 3

High Speed Texture Laser Measurement System

3.0 Introduction

The primary sensor used by the texture system is the Selcom texture laser. Currently, Selcom lasers are used by the majority of road profiling measurement systems both in the United States and internationally. Selcom also offers a high resolution laser that is now being used for high speed texture measurements on other systems. The Selcom lasers have proven to be very reliable and effective devices for displacement measurements in the road measurement environment. Efforts were made in the early part of the project to find suitable candidates for texture measurements, but none were found to be significantly better than this laser. Because of the lack of other candidates systems, the pilot study described earlier, and the fact that the Selcom texture lasers are used elsewhere for texture measurement the Selcom laser 78KHz -16 micrometer laser was selected. This chapter will discuss the high-speed texture measurement system developed for the project. The discussions will include the system architecture, followed by details on the various modules used for texture measurements. The characteristics of the sensor data are also described. Finally, the laser interface system is discussed. The next chapter describes the program and procedures that were developed for collecting texture data.

3.1 System Overview

Figure 3.1 illustrates a block diagram of the measurement system. As noted, there are two main modules and three sensors. The two modules are the laser power and interface module, and the embedded PC. The laser power and interface unit has three functions. It provides power to the laser probe as the name implies. However, it also provides a direct connection from the laser to a specially built data acquisition board in the embedded PC. Finally, it provides the signal conditioning and interface between the distance and start sensors. The laser probe is used to measure the vehicle body and pavement surface displacements. The distance sensor is used for relating each measurement to a specific point or location on the pavement. The distance measurements are needed in order to determine the texture wavelength components. The optional infrared start sensor is used for detecting the start of test sections by detecting contrasting reflections between a start line, e.g., a painted white line or special tape, and the normal pavement color. This is useful when repeat measurements are desired.

A PC board was selected for the data acquisition and control module. The PC module provided the additional advantage for software development. When the requirements changed to a client-server data acquisition concept for use in the TxDOT Vehicle Network System (VNET), it also allowed the use of the MFC classes for implementing the Winsock requirements.

Texture Data Collection System Overview

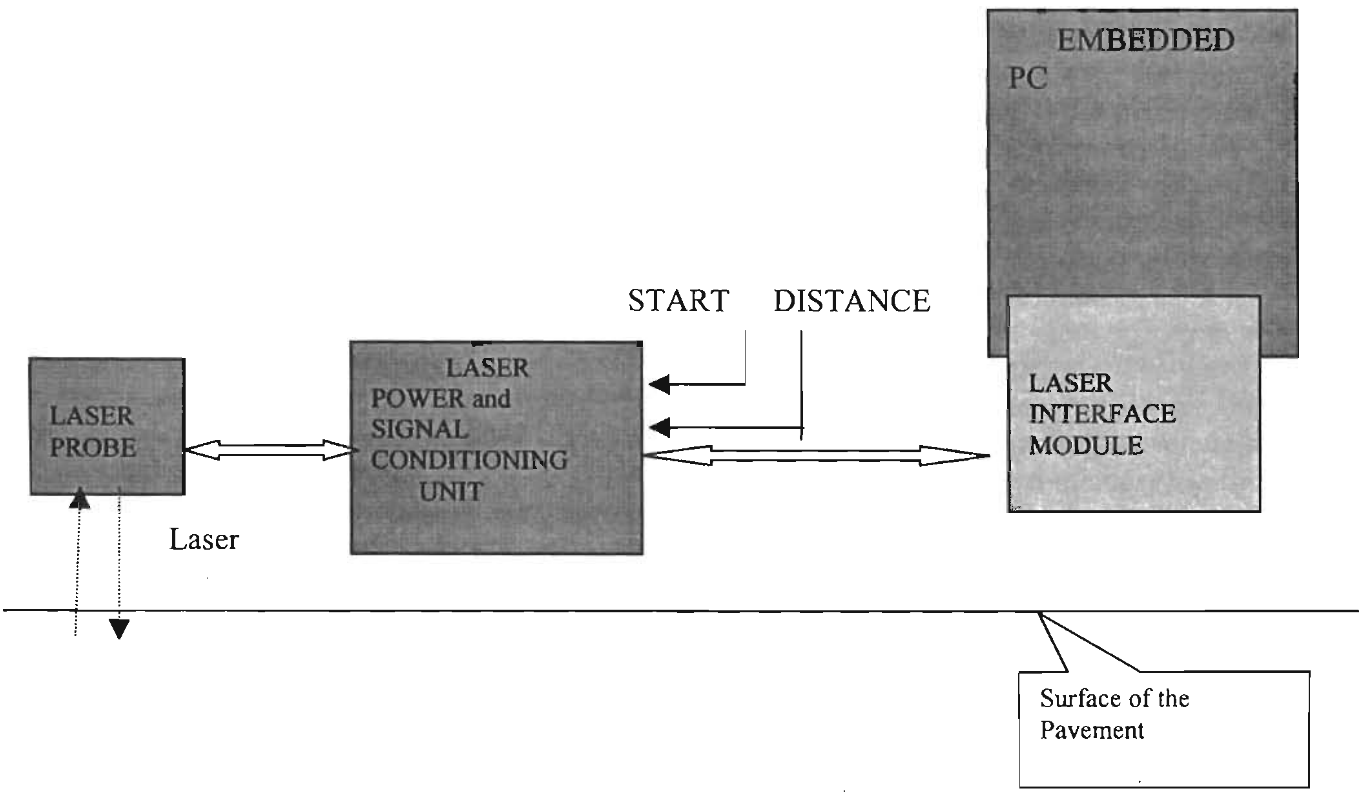


Figure 3.1 Texture Data Collection System Overview

As indicated, the laser, distance and start signals are interfaced to the PC via the PC Laser Interface Module or PCLIM. During the first year of the project, efforts were made to locate appropriate data acquisition equipment for both the microtexture and macrotexture measurements. Selcom offered a VME based system board for their lasers for this purpose. This board had been used in an earlier project with TxDOT, as well as in the Pilot study discussed in Chapter 1. It had been decided, however, that this technology was soon to be outdated and a PC based system was needed. Selcom did not offer such an interface board at the time. Initial efforts were directed toward reading the signal in analog form, and then digitizing the results by the data acquisition system so they could be recorded or processed.

This was how most profiler and texture measuring systems obtained laser data for the various surface measurements at the time the project began. After investigating the analog and averaging method used by Selcom for providing the analog laser readings, and because of the VME board requirements, it was decided to build a special board for this application. The requirements for the board design included a direct interface between the serial data stream sent by the laser probe and the PC via the system bus. Thus the method would provide a reduced noise advantage in that noise or errors introduced by taking the digital displacement readings, converting them to an analog form, and then once again back to digital would be avoided.

The Selcom system provided a selectable running average on a special averaging board that provided averaged displacement readings. It was decided to include the averaging function and allow the software to specify the amount of averaging. As will be later discussed, averaging the continuous laser readings is common and usually necessary in most laser systems. The Selcom averaging board discards readings that are detected as bad from the laser probe. That is, during measurements, if one of the continuous laser readings were found bad (a laser return signal was not received), it would be discarded and replaced by the next reading (assuming it was good). If readings are discarded, then the number of time-readings between two consecutive distance-readings could vary for the same speed, thus making it impossible to obtain accurate longitudinal wavelength information. The above considerations became requirements for the PCLIM board. The system would let the software determine how to average the laser readings, as well as maintaining wavelength integrity. The PCLIM provides a way to read the laser values directly into memory via the ISA or PC104 data bus (via an ISA to PC104 adapter). The PCLIM is discussed later in the chapter. The next section provides details on the architecture of the Laser Power and Signal Interface Unit.

3.2 Power and Signal Interface Unit

Although the Power and Signal Interface Unit, and the PCLIM (described next) were designed specially for this project, it was also used in the Calibration project, TxDOT Project 2984. The main components of the Power and Signal Interface unit are illustrated in the component layout of Figure 3.2. As noted in this figure, the unit consists of 6 main components. These include the optional laser interface module (LIM), the laser power module, the signal interface module, and three power supplies. The laser interface, laser power and signal interface modules are used in the Texas Profiler systems and are described in a separate TxDOT report. The optional LIM is only used to provide a status of the laser to insure it is providing valid data. The signal interface module conditions the start data collection signal. The laser power module provides the necessary voltages for the Selcom laser and the ± 15 volts for the LIM and signal interface modules. The signal flow in this subsystem is illustrated in Figure 3.3A and 3.3B. Figures 3.4A and 3.5A depict the front and rear of the subsystem where dimensions used for building the prototype systems are included in Figures 3.4B and 3.5B in the event TxDOT would like to construct additional units. The Selcom laser connects directly to this unit. The Selcom signals are then sent on to the PCLIM, located in the PC, where the laser displacement signals are obtained by the PC. The Selcom laser connects to the laser input. A 25 pin straight through cable is used to connect the DSUB connector on the serial connector to the DSUB connector on the PCLIM. The parts list for the Laser Power Unit is given in Table 3.1.

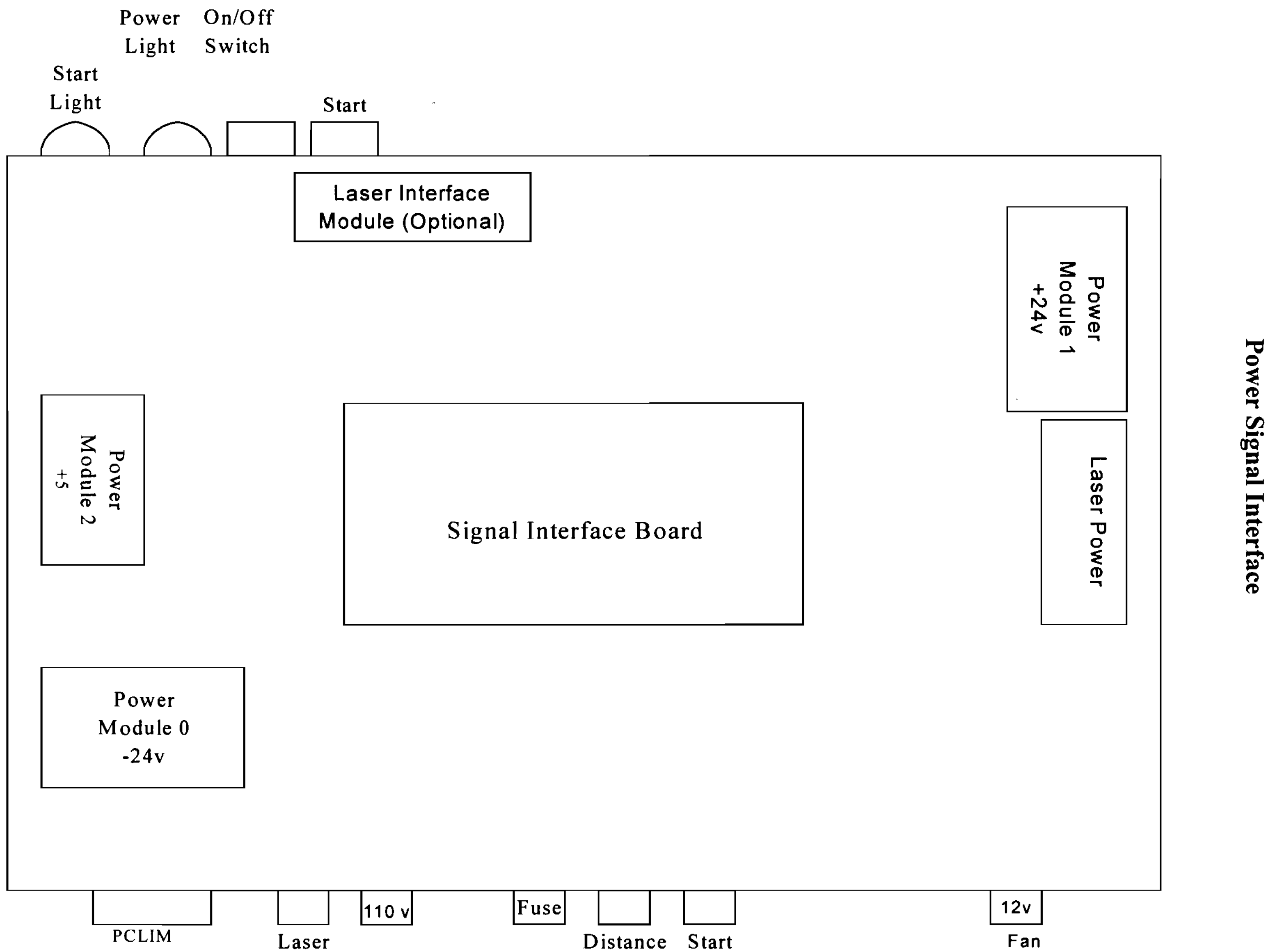


Figure 3.2 Power Signal Interface Unit

Laser Power Unit

Quantity	Description	Part #
1	Power Supply (Triple Output) : 5V@3A	HTTA-16W-A+ (CONDOR)
2	Regulated Power Supply : 24V@2.4A	SLS-24-024-TB (SOLA)
1	Brushless DS Fan	031158 (COMAIR)
1	Fan Guard	09250G (QUALTEK)
1	Circular Power Connector : 10 Connections	97-3102A-18S-1S (AMPHENOL)
1	Power Switch (On-Off Panel Mounted Toggle Switch)	No part number
1	Start Switch (Push Button)	AB15AP (NKK SWITCHES)
1	Power Signal Light (LED)	679-9911
1	Start Signal Light (LED)	679-9911
2	BNC Connector	31-221-RFX (AMPHENOL)
1	Fuse Holder	No part number
1	D-Sub Connector 25 Contacts Straight PC (Plug)	No part number
1	Power Receptacles	ZC-2821 (ALECTRON)

Table 3.1 Laser Power Unit Part List

PC Laser Interface Unit - Component Layout
Signal Flow

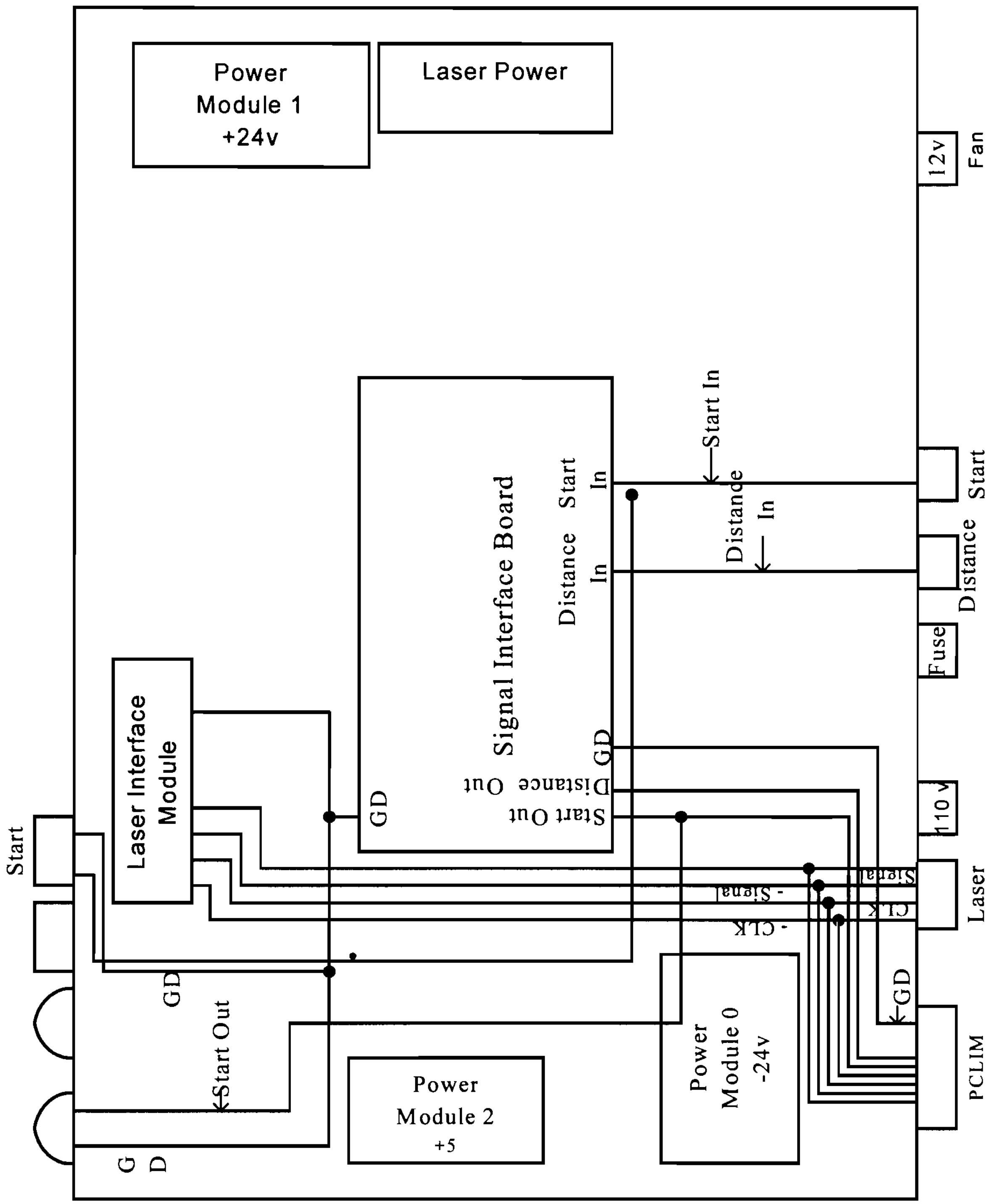


Figure 3.3A PC Laser Interface Unit-Component Layout Signal Flow

System Configuration

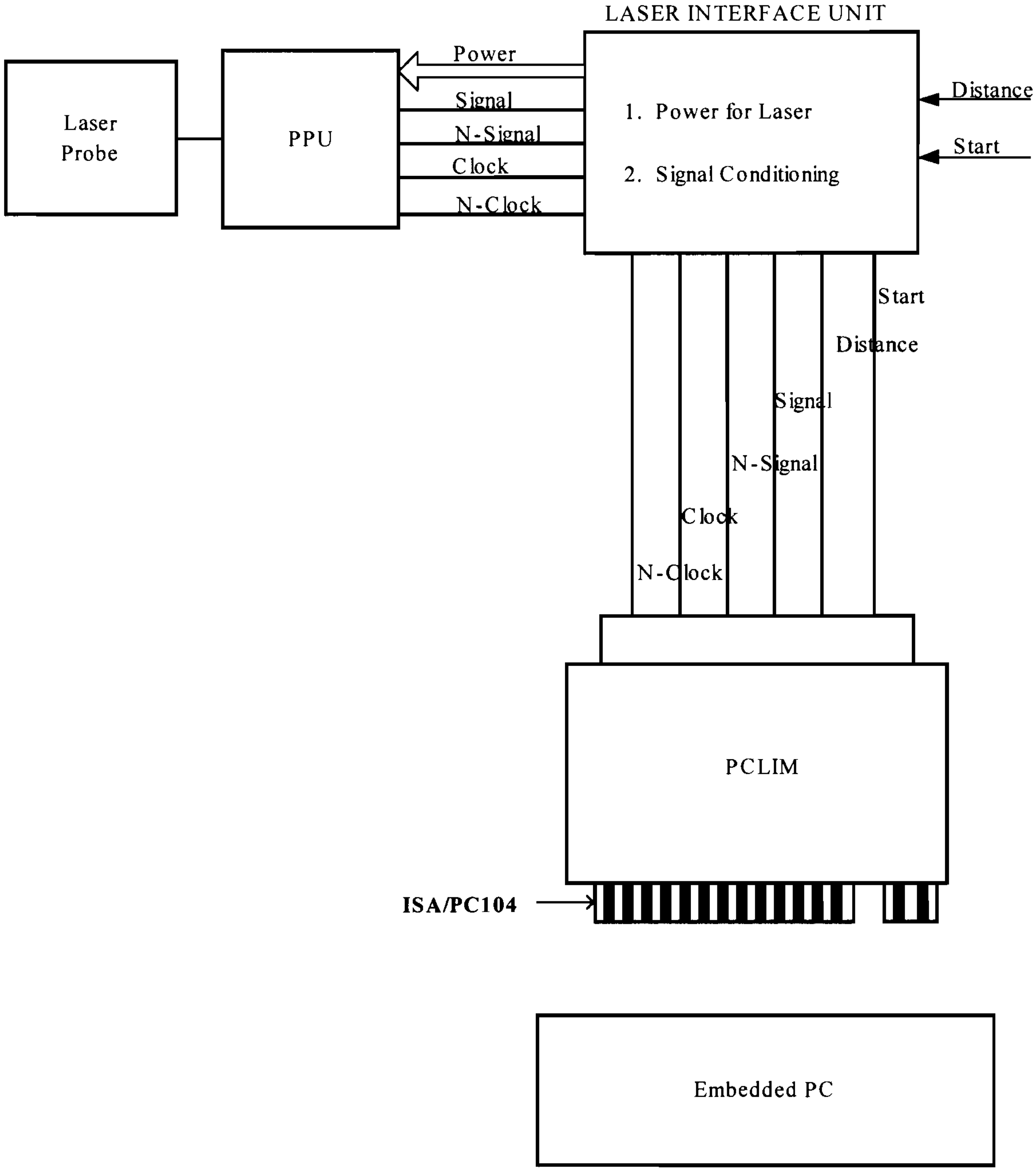


Figure 3.3B System Configuration

Laser Measurement System - Front Panel

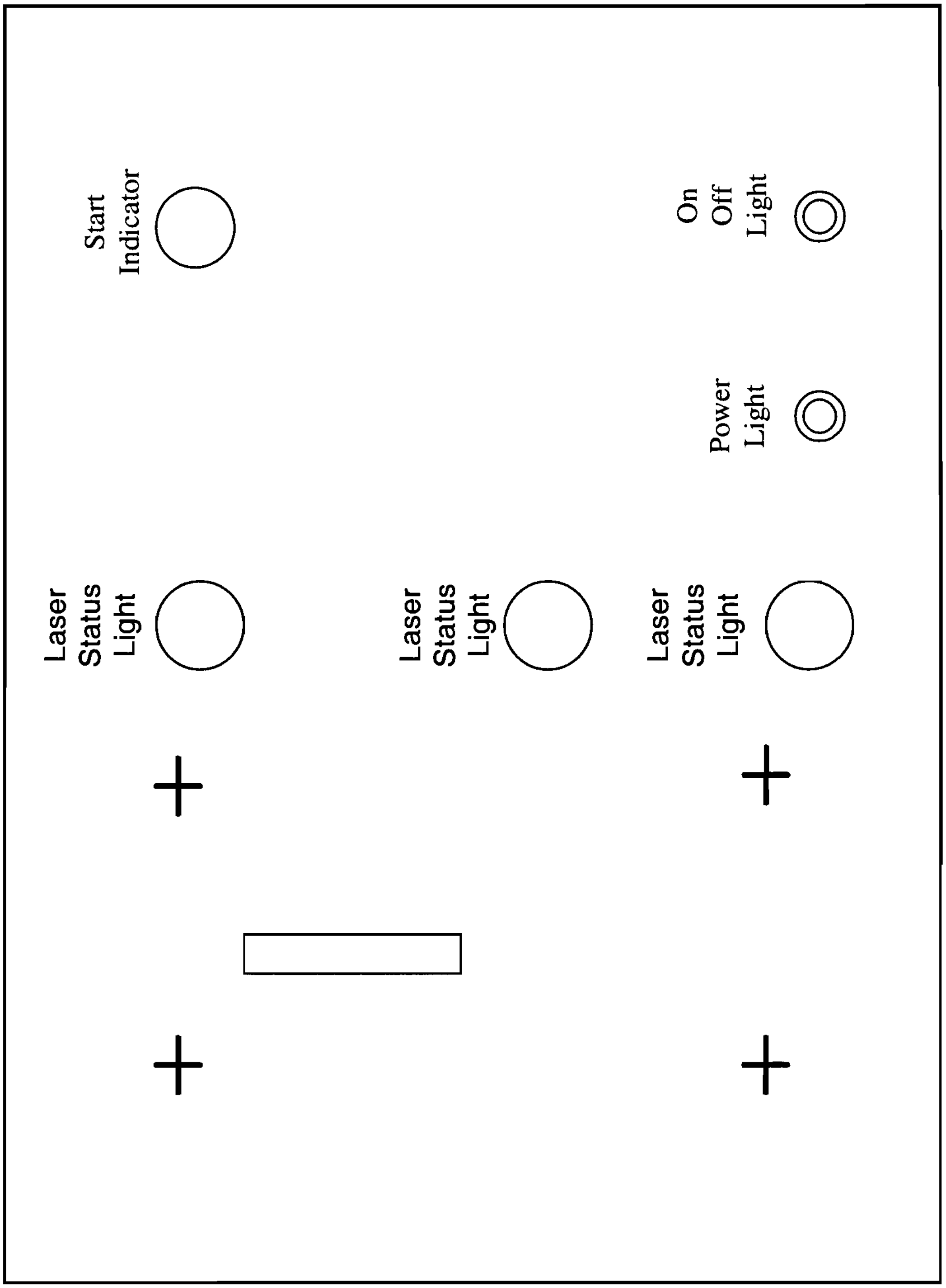


Figure 3.4-A   Laser Measurement System - Front Panel

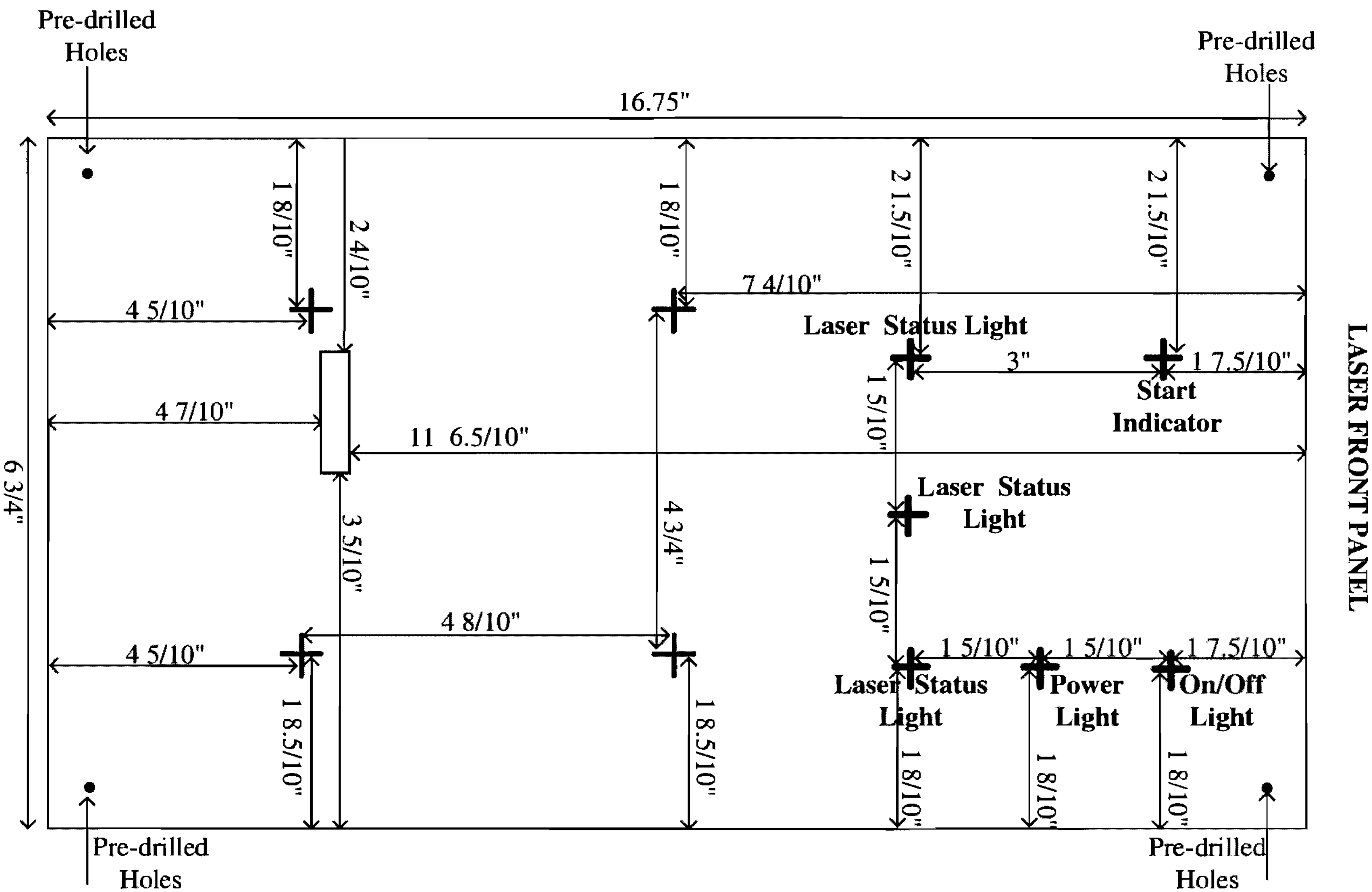


Figure 3.4-B Laser Measurement Front Dimensions

Laser Measurement System - Back Panel

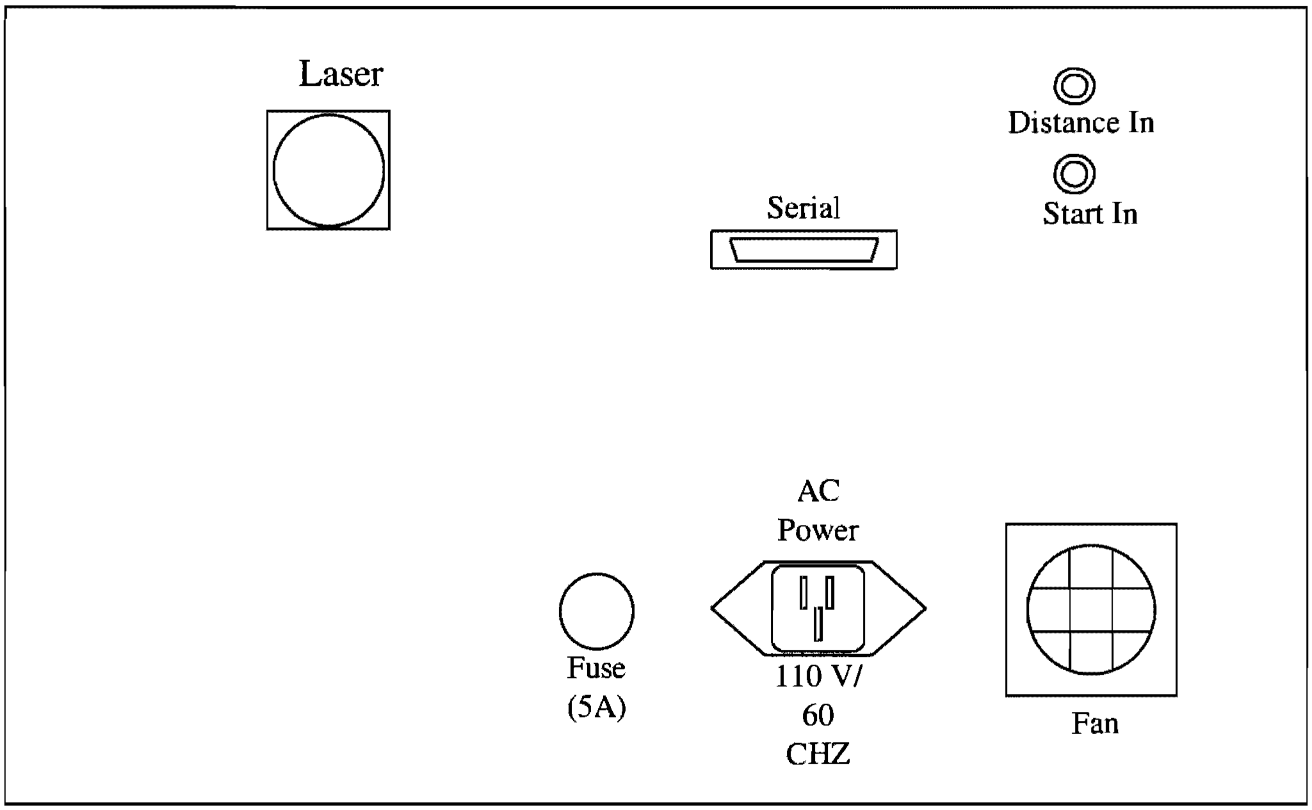


Figure 3.5-A Laser Measurement System - Back Panel

LASER BACK PANEL

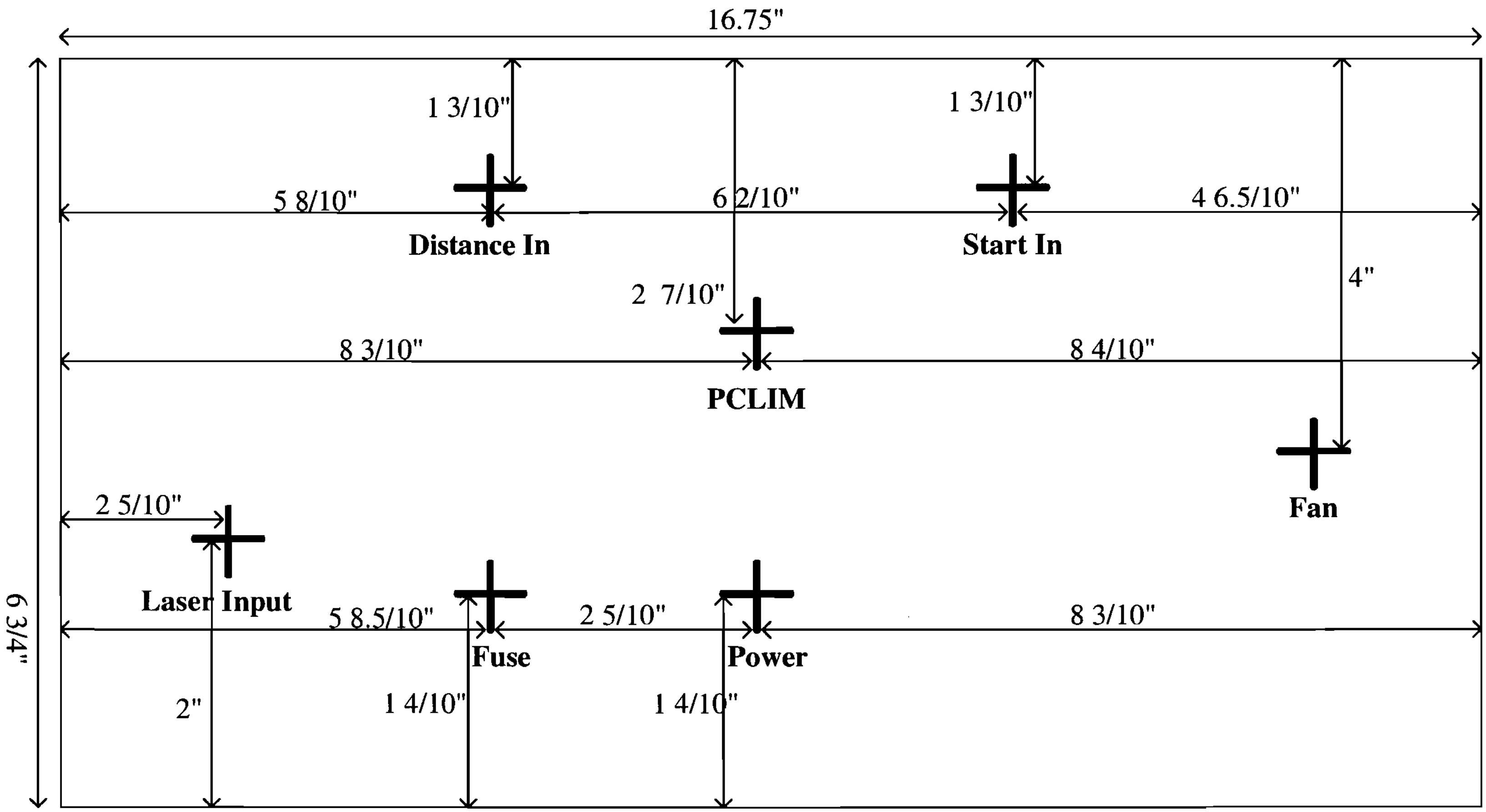


Figure 3.5-B Laser Back Panel Dimensions

3.3 PC Laser Interface Module

This section will describe the PC laser interface module (PCLIM). Before describing the PCLIM a few concepts that were considered in the design of this module are presented. Figure 3.6 provides the PCLIM component layout. An illustration of the PCLIM, a Selcom 32 KHz laser and the prototype laser interface unit, is shown in Figure 3.7.

3.3.1 PCLIM Design Considerations

As discussed above, one of the first design considerations for the PCLIM was to insure that the laser displacement data could be quickly and directly read from the Selcom laser. The serial data would be converted to parallel and then presented directly to the PC for access via the ISA bus. The method used by the Selcom laser for obtaining analog displacement readings is commonly referred to as optical triangulation. For this method, the laser light is emitted from a laser diode onto the surface of the object whose distance is to be measured. This light is then reflected back off of the surface. The reflected light is focused using a converging lens onto a linear position sensitive detector (PSD). The location of the focused light on the PSD is quantified by the laser detection circuit. This location can then be used to calculate the distance of the object. The location of the laser on the PSD is represented by an analog signal and processed by the laser detection circuit. This analog signal is digitized by an analog to digital converter and made available in digital form as a synchronous serial data stream.

The laser displacement data is sent to the PCLIM via the Laser Interface Module over four wires as illustrated in Figure 3.3B. These signals are included in the same cable as the laser power. Two of the signals provide the differential clock signal and the other two the differential data. This data is sent serial at 32*78000 bits per second from the laser probe to the PCLIM. The data format is illustrated in Figure 3.8. As can be noted in this figure, each 12 bit laser reading is included in a 32 bit data frame. The first 16 bits of the frame consists of the laser displacement value and 4 bits of status.

The laser 2.52 in. (64mm) displacement range is divided into 4000 equally spaced values, where each bit represents 0.63 mils (0.016mm). Only 4000 of the possible 4096 combinations are addressed as the data is linearized before it is sent. This serial data is sent on to the PCLIM where it converts the data using a shift register within the logic device located on the PCLIM board. The data is first received with the differential receivers and then placed into a buffer. The laser valid bit is included with the distance, sensor start, and 12 bit displacement readings that are stored in the PCLIM's 1024 word buffer.

The data format is illustrated as follows:

SDBxRRRRRRRRRRRR.
Where:
S – Start Signal
D – Distance Signal
B – Laser signal invalid
R – Reading of the laser displacement (0 to 3999)
The x bit is not used.

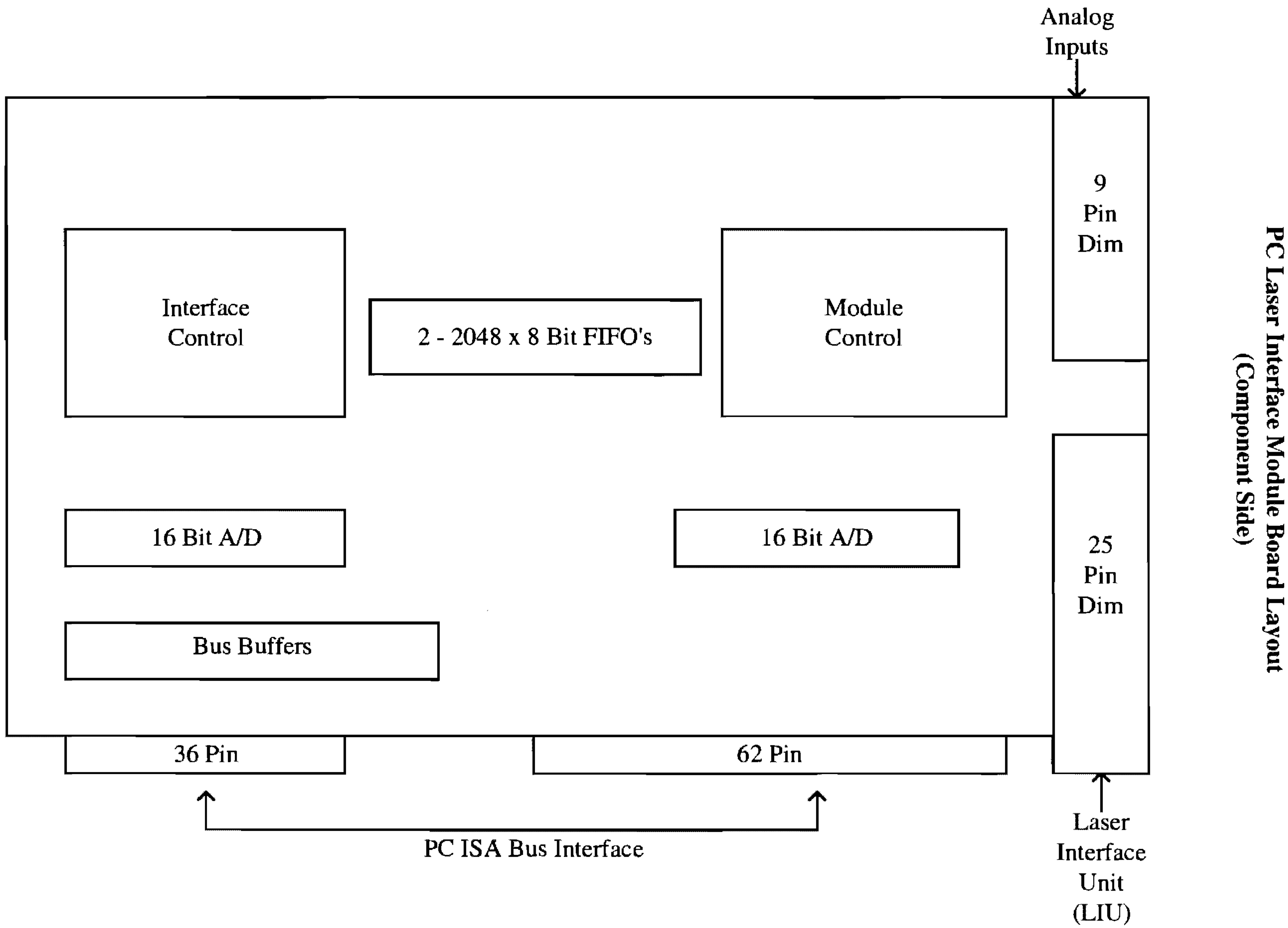


Figure 3.6 PC Laser Interface Module Board Layout - (Component Side)

PCLIM, 32KHz Selcom Laser and Laser Interface Unit

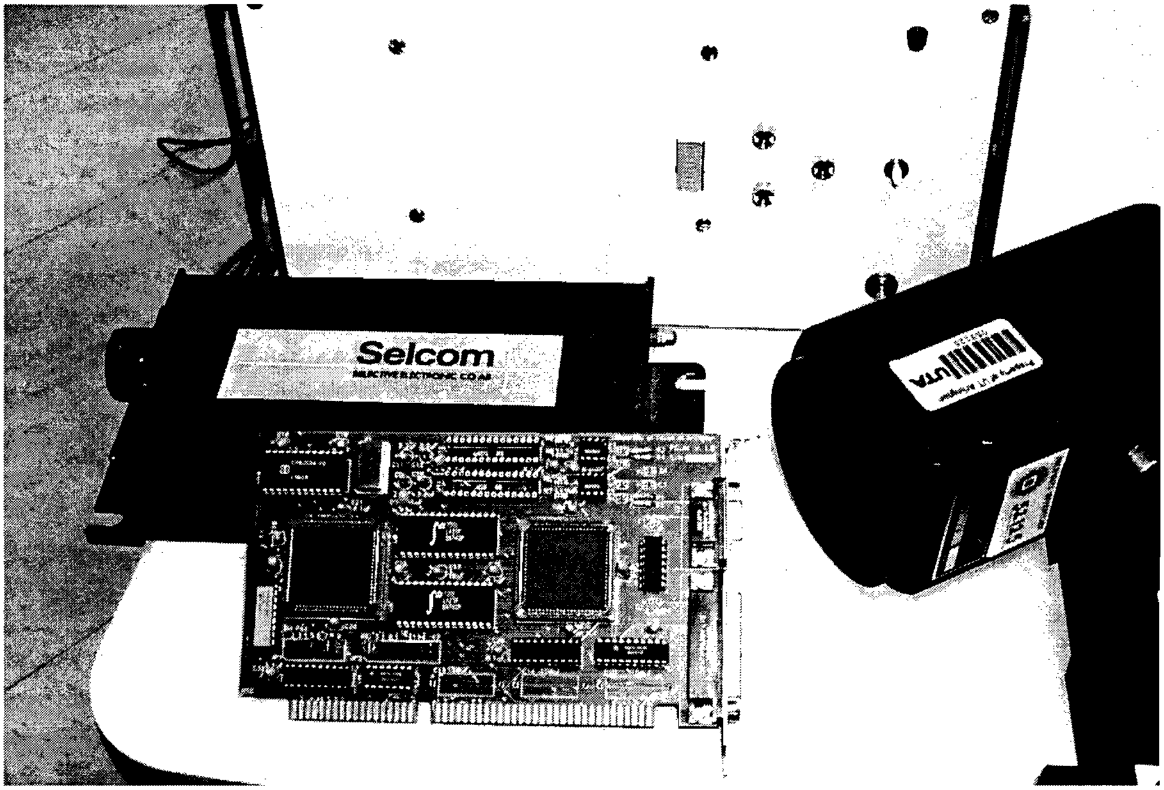


Figure 3.7 PCLIM, 32 KHz Selcom Laser and Interface Unit

Selcom Data Format

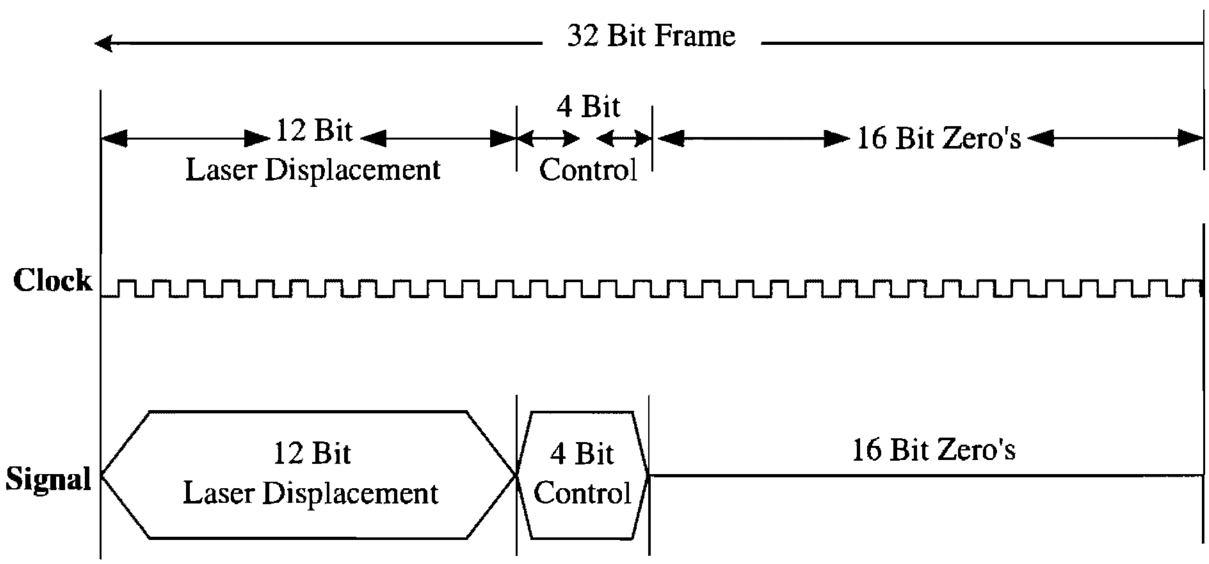


Figure 3.8 Selcom Data Format

The value of 'S' can be either 0 or 1. A value of 1 indicates the start by the infrared detector. 'D' stands for the distance transition occurrence. This signal toggles for every distance signal detected by the distance sensor. 'B' stands for the laser invalid bit. If the laser beam is not an adequate reflection from the target, this bit is set. The number of the distance readings detected in a data measurement, in combination with the number of displacement readings received allows both the speed traveled by the vehicle and the distance between two displacement readings to be computed. As will be discussed in the next chapter, the laser data is written into a file in a binary format, along with the header file record that indicates the section length, number of speed counts per kilometer, laser sampling rate, the number of displacement values read, the laser offset, the laser displacement units and the raw laser readings. Figures 3.9A, B, C, and D, provides the schematics of the PCLIM.

3.3.2 PCLIM Structure and Programming Model

The PCLIM is designed to interface the Selcom laser displacement unit to a Personal Computer (PC). This module is depicted in Figure 3.10. The unit allows the user to read the raw digital distance calculated by the laser. The PCLIM is comprised of the following functions:

- First In First Out (FIFO) Buffer – Allows software burst reads while preventing data loss.
- Optional Interrupt – Indicates to software that the FIFO is half full.
- External Trigger – May be used to start laser data collection.
- External Synchronization Signal – Allows the correlation of the laser data to an external event.
- Two (2), Independent, Sixteen (16) Bit, Analog to Digital (A/D) Converters.

3.3.3 Hardware Installation

This section describes the installation instructions for the PCLIM. The PCLIM may be installed in any sixteen (16) bit ISA slot. Before installing the board jumpers J1 and J2 must be configured.

Base Address Jumper

Jumper block 1 (JP1) selects the PC's I/O base address for the PCLIM. The base address determines the location of the board in the PC's I/O address range. All PC I/O addresses are comprised of ten (10) bits. The jumper block selects the upper eight (8) bits of the PCLIM's base address. Bit one (1) and bit zero (0) are hard-wired as logic zero (0). For this reason, the PCLIM requires four (4) consecutive I/O addresses. This also forces the board to reside at an even I/O address.

PCLIM Parts List

Quantity	Description	Part # (Allied #'s)	Reference Designator(s)
4	Capacitor : 2.2uF	881-7156	C9,C10, C11,C12
2	Capacitor : 10uF	881-7132	C16,C17
16	Capacitor :0.1uF	852-5685	C18,C19,C20,C21,C22, C23,C24,C25,C26,C27, C28,C29,C30,C31,C32, C33,C34
4	Resistor : 100K		R1,R2,R3,R4
2	Resistor : 200	870-0900	R5,R6
2	Resistor : 33.2K		R7,R8
1	Resistor Pack : 4.7K		RP1
2	Jumper Terminal	863-3160	JP1,JP2
2	MACH435-15	MACH435-15	U1,U2
2	74LS245	74LS245	U3,U4
1	74LS541	74LS541	U5
1	74LS05	74LS05	U6
1	75182	75182	U7
2	IDT7201LA50P	IDT7201LA50P	U8,U9
1	8254	8254	U10
2	PLCC Socket	978-2225	U1,U2
3	Socket : 20 Pin	512-4674	U3,U4,U5
2	Socket : 14 Pin	512-4671	U6,U7
2	Socket : 28 Pin	900-0068	U8, U9
1	Socket : 24 Pin	900-0066	U10
1	Crystal : 2.47MHz	B1200 (BOMAR)	X1
1	Connector : 25 Pin	617-A024S-Aji20 (AMPHENOL)	J1
1	Connector : 9 Pin	617-A009S-Aji20 (AMPHENOL)	J2

Table 3.2 PCLIM Parts List

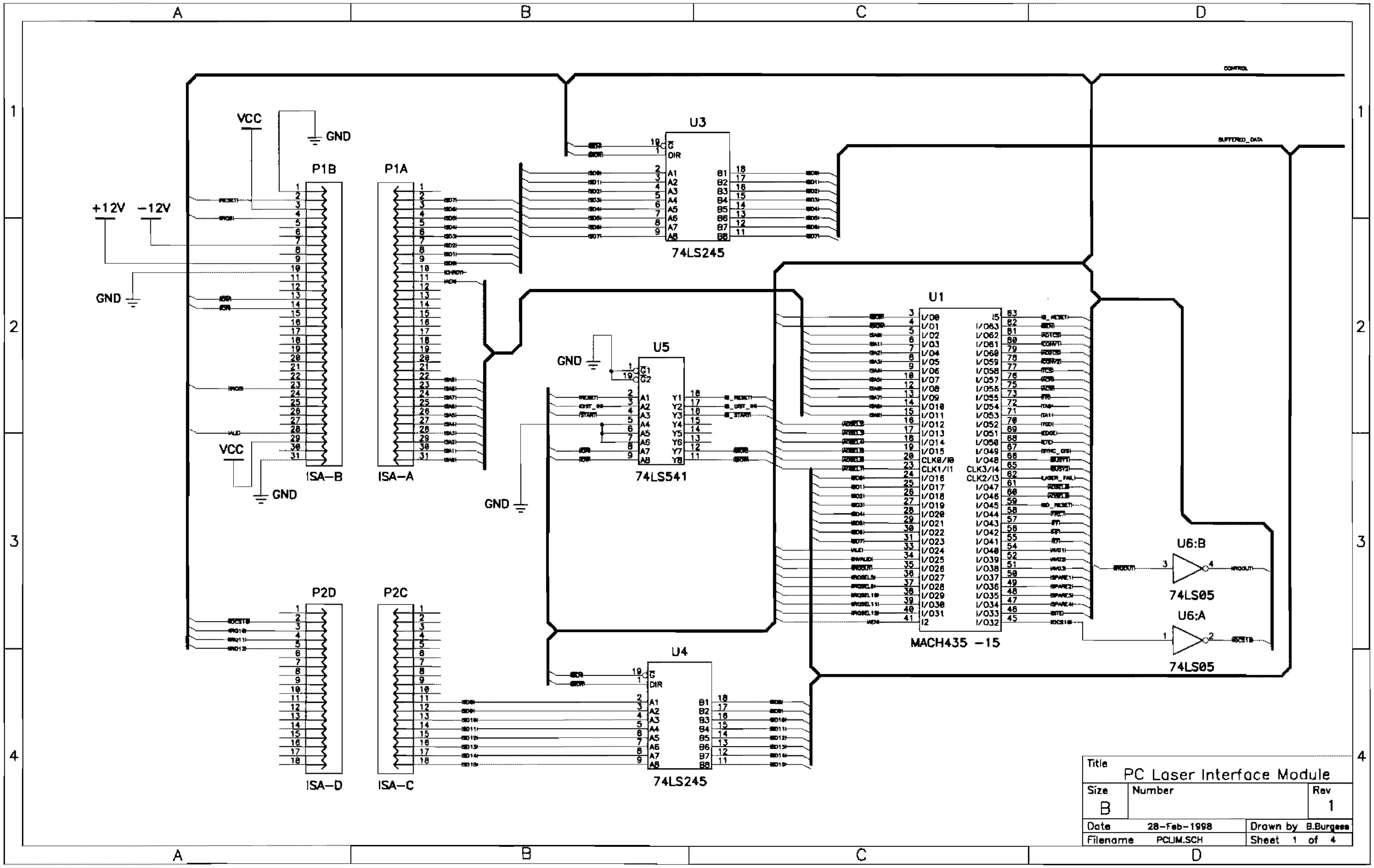


Figure 3.9A Schematic of PCLIM-page 1

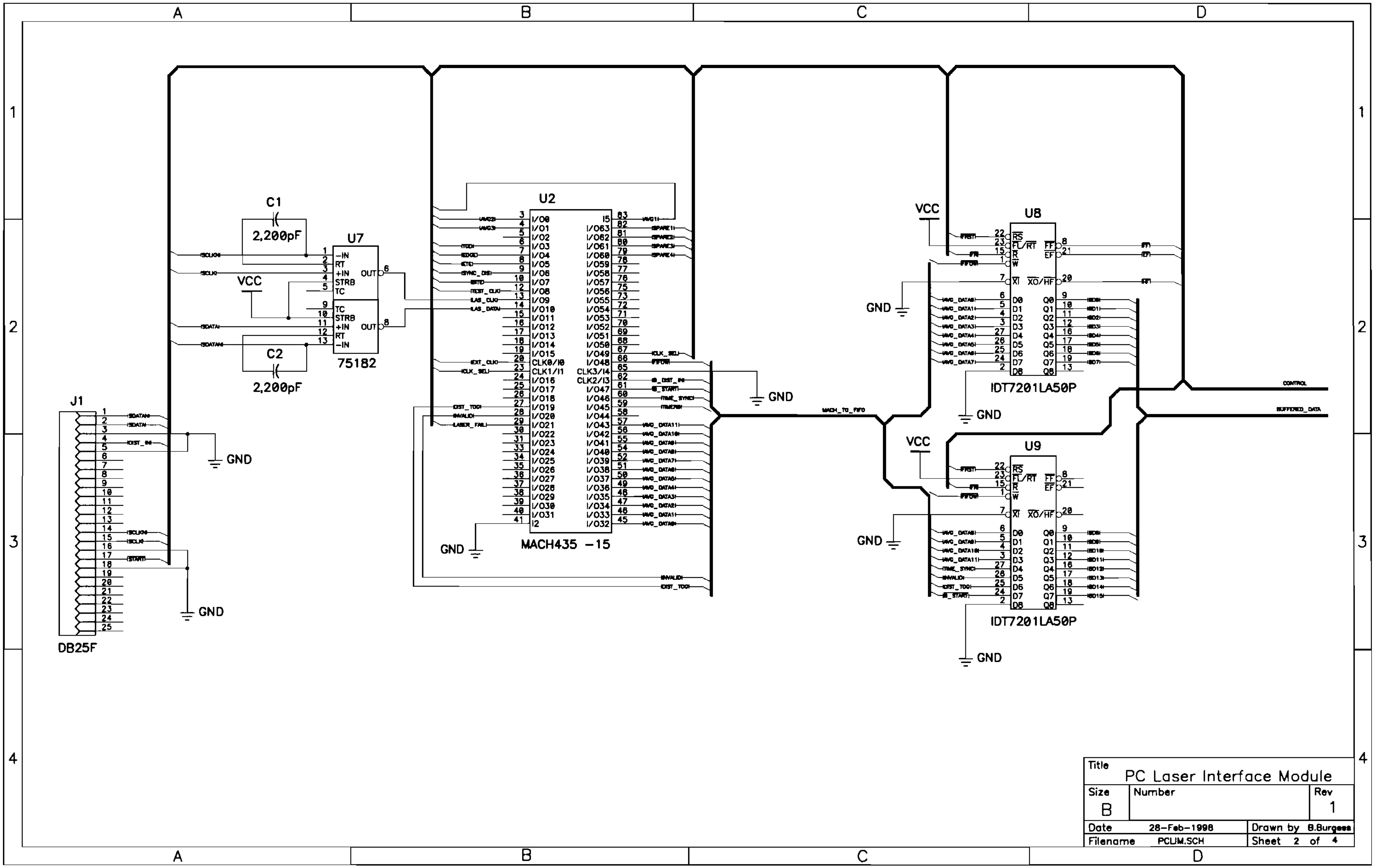


Figure 3.9B Schematic of PCLIM-page 2

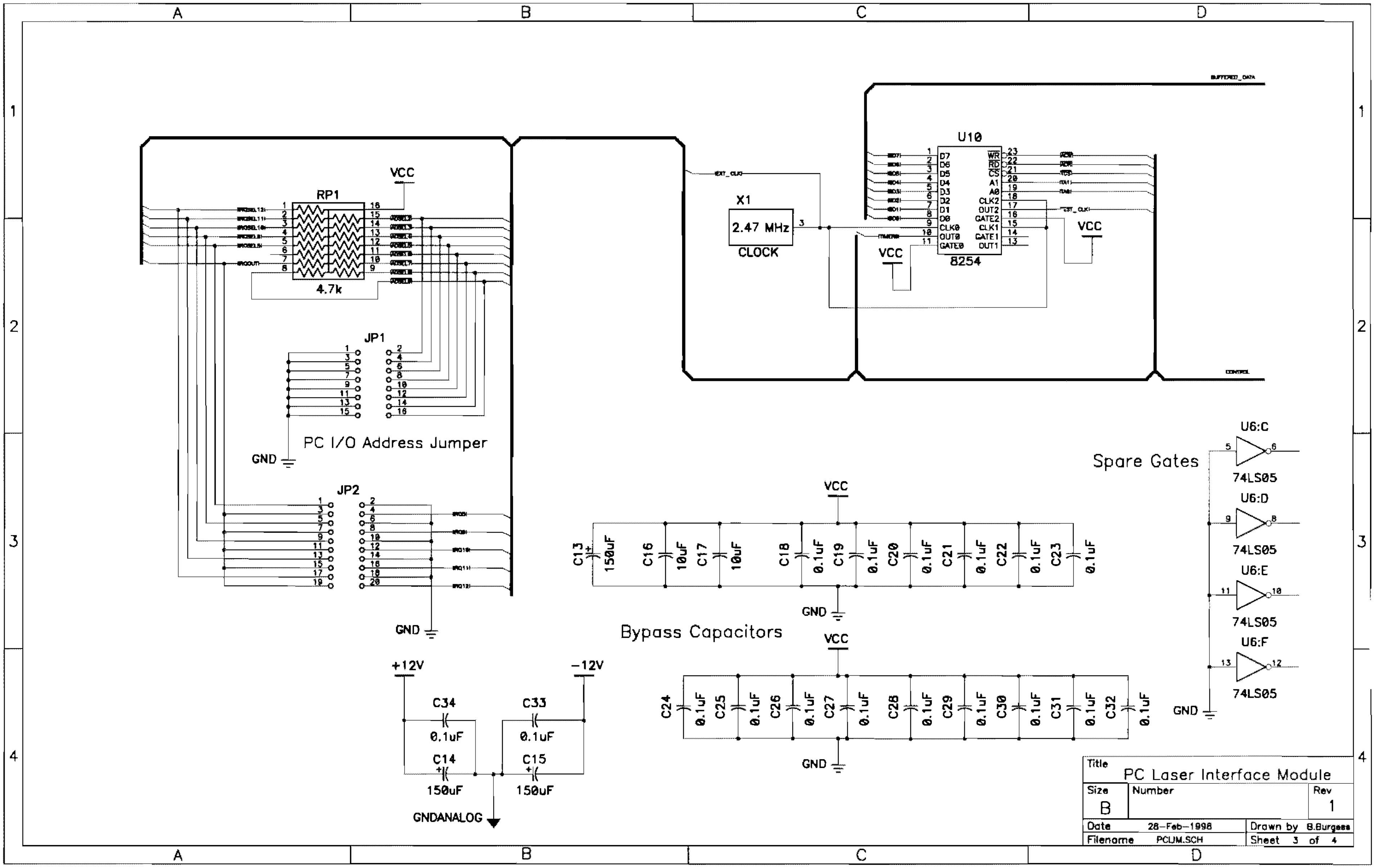


Figure 3.9C Schematic of PCLIM-page 3

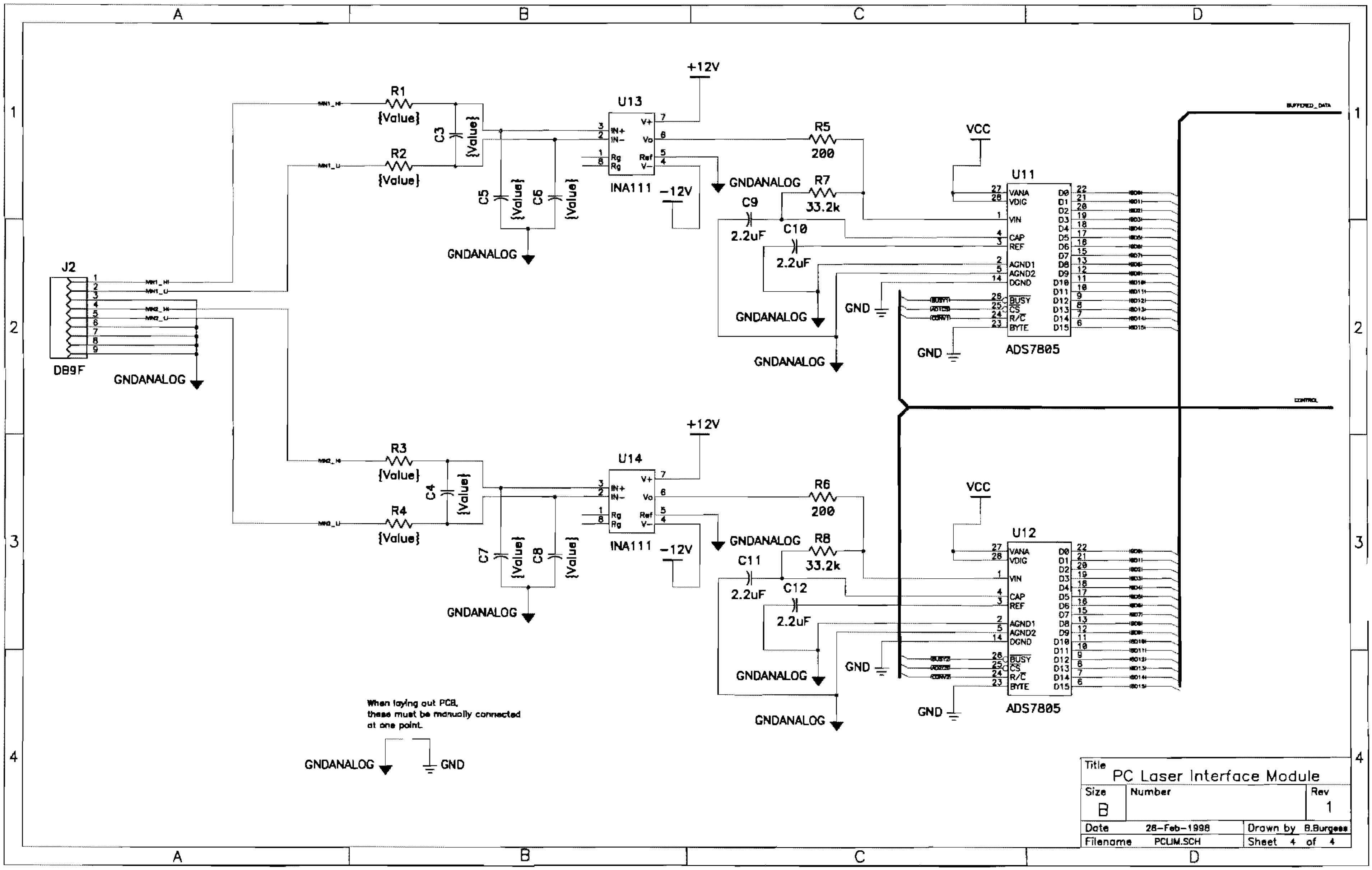


Figure 3.9D Schematic of PCLIM-page 4

Table 3.3 defines the relationship between the jumper block pins and the corresponding bits in the I/O base address. An installed jumper indicates that the associated bit in the address must have a value of one (1), while a jumper not installed indicates that the address bit must have a value of zero (0).

Jumper Pins	I/O Base Address Bit
15, 16	Bit 9
13, 14	Bit 8
11, 12	Bit 7
9, 10	Bit 6
7, 8	Bit 5
5, 6	Bit 4
3, 4	Bit 3
1, 2	Bit 2
NA	Bit 1 (Always 0)
NA	Bit 0 (Always 0)

Table 3.3 PCLIM Base Address Jumper (J1)

Base Address Selection Example :

Base Address = $1FC_{16}$: Install a jumper across pins 13 & 14, 11 & 12, 9 & 10, 7 & 8, 5 & 6, 3 & 4, and 1 & 2.
Base Address = 330_{16} : Install jumpers across pins 15 & 16, 13 & 14, 7 & 8, and 5 & 6.

Interrupt Level Jumper (JP2)

The interrupt level jumper provides for user selection of the interrupt level generated when the laser interrupt is enabled. Two jumpers are installed when selecting an interrupt. The value of the selected interrupt level may be determined by reading the status register.

IRQ Selection Example :

Disable all interrupts : Install no jumpers

Use interrupt request 10 (IRQ10) : Install a jumper across pins 9, 10 and 11, 1 2.

Table 3.4 defines the pins of the interrupt level jumper.

Jumper Pins	Interrupt Level - Bit Position
1, 2,3, 4	IRQ 5
5, 6,7, 8	IRQ 9
9, 10,11, 12	IRQ 10
13, 14,15, 16	IRQ 11
17, 18,19, 20	IRQ 12

Table 3.4 Interrupt Level Jumper (J2)

Programming Model Registers:

This section details the different board registers.

TRG	ES	INV	TS	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PC Address = PCLIM Base Address Mode : Read Only
Width: 16 Bits

Figure 3.10 Laser FIFO Data Word

The laser FIFO is read by performing a sixteen (16) bit I/O read at the PCLIM's base address. All laser data is read from the laser FIFO. The FIFO provides a buffer preventing the loss of data. To provide for fast data accesses, reads from the FIFO are not dependent on the value in the function address register.

D11 - D0 : Laser Data Word (12 Bits)

TS: Time Sync. Value of the 8254's timer one (1) clock output. If synchronization is disabled, this bit remains a logic zero (0).

INV: Invalid. When read as a logic one (1), this bit indicates that the laser data word is invalid. The Selcom Laser sets this bit. When averaging data values, by using the laser average register, this bit is set to a logic one (1) if any of the data values used to compute the average were invalid (as indicated by the Selcom Laser).

ES: External Synchronization. When external synchronization is disabled (see DS bit in the control register), this bit remains a logic zero (0). When the external synchronization function is not disabled, this bit toggles between logic one (1) and logic zero (0) on the rising edge of the external synchronization signal.

TRG: Trigger. Value of the external trigger input signal

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

PC Address = Base Address + 2 Mode : Read/Write
Width: 8 or 16 Bits

Figure 3.11 Data Register

For functions selected with the function address register, the data register is utilized to pass all data to and from those functions. This minimizes the number of required PC I/O addresses on the PCLIM. The data width is either sixteen (16) or eight (8) bits wide, depending on the function selected.

R4	R3	R2	R1	R0	AV2	AV1	AV0
7	6	5	4	3	2	1	0

Function Address = 1 Mode : Read/Write
Width : 8 Bits

Figure 3.12 Laser Averaging Register

This register allows the user to select the number of laser readings that are averaged before being written into the FIFO.

V2 - AV0: Average select bits.

AV2 - AV0: Average select bits.

Laser Averaging Selection

AV2	AV1	AV0	Samples Averaged (N)
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Table 3.5 Laser Averaging Selection

Following is example C code for resetting the PCLIM board, resetting the FIFO and reading data. The base address of the board is first set to hex 300.

```
#define PCLIM 0x300

/* The following C code illustrates the commands to initialize the PCLIM */
 outp(PCLIM+1,0) ;             /* Select the Control Register */
 outp(PCLIM+2,0x0080) ;               /* Set RST to 1 */
 outp(PCLIM+2,0) ;             /* Set RST to 0 */
```

The following C code illustrates the commands to reset the FIFO;

```
/* Reset the FIFO */
 outp(PCLIM+1,0) ;                     /* Select the Control Register */
 outp(PCLIM+2,inp(PCLIM+2)|0x0040) ;   /* Set FRST to 1 */
 outp(PCLIM+2,inp(PCLIM+2)&0x00BF) ; /* Set FRST to 0 */
 outp(PCLIM+1,3) ;                     /* Select the Status Resister 2 */

 /* The following C code illustrates the command to read the data from the PCLIM */

        /* Read the Laser Data From the FIFO
           data = inpw(PCLIM) ;  /* Read as 16-bit unsigned */
```

CHAPTER 4

Procedures for WinTxtr

4.0 Introduction

This chapter provides details on the texture measurement software used for operating the high speed texture measurement system. There are basically three measurement software systems that were developed during the course of the project. The first system was a DOS based system (referred to as Txtr) and was used for texture measurements until July, 1999. During the past two years of the project TxDOT began developing the TxDOT Vehicle Network System (VNET). The data collection activities for texture, as well as other measurement systems, are being required to follow this concept. A second version was developed that was designed to run with the VNET concept. In order to develop an application that would run on any of the current or future Windows based operating systems (eg., NT 95/98, 2000, and CE) this version was based around a third party device driver wizard for development of a Windows driver for the special board (PCLIM described in Chapter 3) used for interface to the texture laser. This version was provided to TxDOT as requested in January 2000. This second version was later replaced by a third version not requiring the third party software, but which only runs on Windows 95/98 systems. This is the version currently being used and that is described in this chapter.

As indicated the software system is designed to run with the TxDOT VNET system concept . A description of this concept can be obtained from Brian Michalk or Carl Bertrand, the developers of the TxDOT VNET system. Briefly however, for the method, data collection applications are divided into separate objects - a server object for interfacing either directly with the sensor or a sensor controller and a user interface object or client. The server is only communicated with via a network socket. For the current system, the server, WinTxtr, collects the texture data and stores it on a local drive for access by the client. The client provides the graphical user interface or GUI for the operator or user. The desired modules or objects should be able to run on the same computer with other applications or objects as long as the system can handle the work load. Or, the client could be on one processor and the server on a second. Currently the primary operating system for using WinTxtr for the VNET system is Microsoft Windows.

Figure's 4.1A and 4.1B illustrate the WinTxtr Server and Client Interface Design used when collecting data. The design accommodates both data collection and real-time processing. Also indicated in the figure are the Client/Server communications commands, e.g., START, SAVE, ENDP, etc. A header structure or tag is defined which provides various operational characteristics used by the texture measurement system. As each data set is computed, this structure is included with the data set. The design provides for including other information, such as the predicted skid number in this tag. The user, via the Client, could then simply indicate the file name to save the data and/or other information, such as mean profile depth or an estimated skid number.

Data Acquistion Entity Diagram

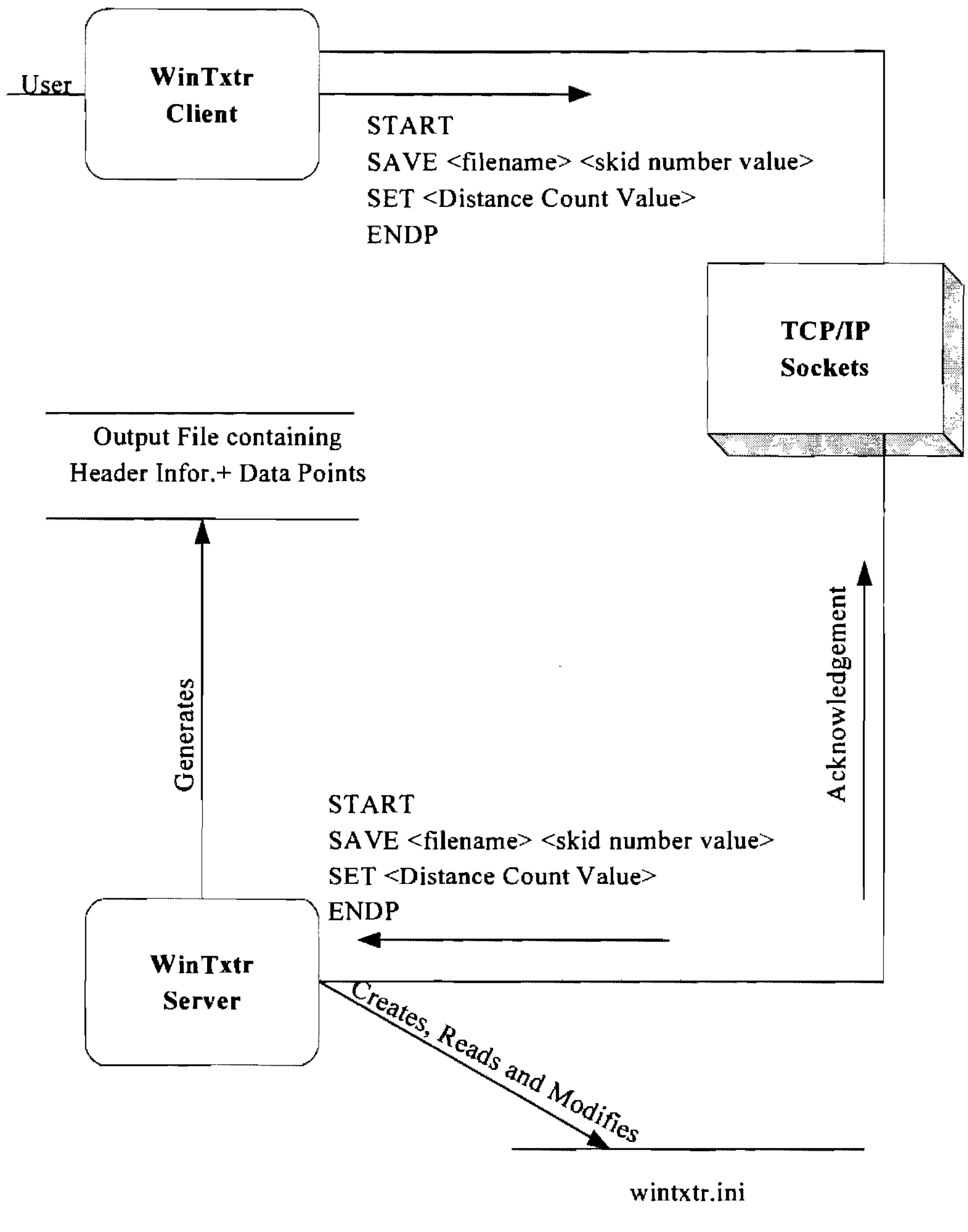


Figure 4.1A Data Acquisition Entity Diagram

Client-User-Server Interface

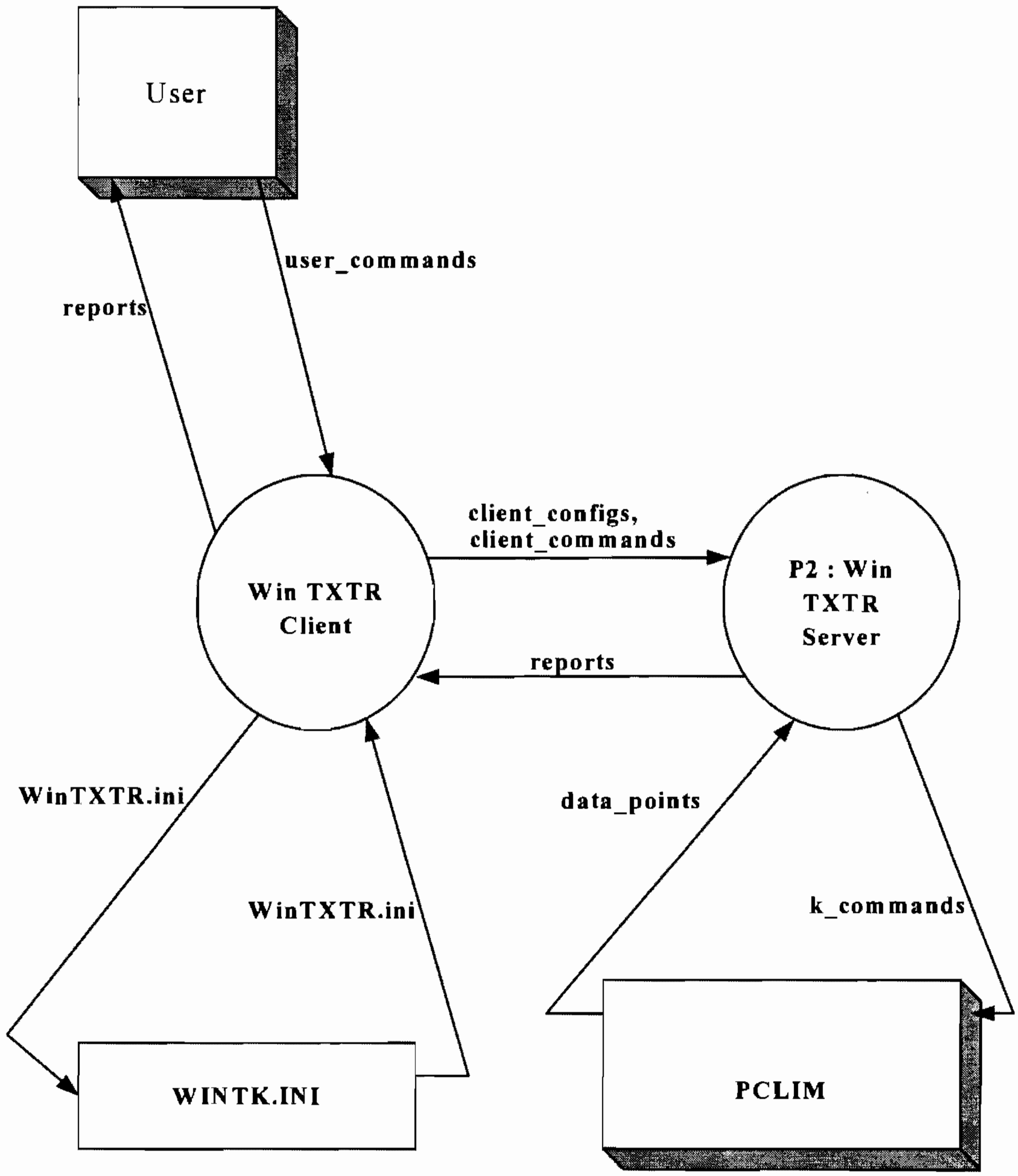


Figure 4.1B Client User Interface

Separate software applications can be developed that can be used for post processing to validate the data collected. For example, one such application known as Txtrp, was developed by researchers to quickly determine if the collected data is valid.

As indicated, the initial project proposal did not indicate a requirement for running under Windows and specifically using the VNET concept. Because TxDOT made the decision to switch to Windows and the VNET concept, software and procedures were modified to try and accommodate this concept as it was being developed. The first version used a third party driver package for the various Windows versions. The development of the second version of WinTxtr does not require a Windows driver for the PCLIM when using Windows 98.

4.1 WinTxtr Organization

The purpose of WinTxtr is to provide a software package that can be used by the texture system to collect the raw texture data. The detailed use cases and corresponding requirements for the design are given in Tables 4.1A and B. It was designed to also accommodate real-time computation of the various classification or statistical parameters of the texture data. One such characteristic is an estimate of the frictional characteristics of the pavement as measured by skid number. After data is collected it is saved according to specified name assigned by the client. The data includes a tag or header file that is used to maintain the various characteristics of the laser and measurement system that was used to collect the data. The header structure was defined early in the project using the DOS based program or Txtr and includes some parameters that are no longer used. To maintain compatibility with these initial versions, the header file is still used in the current WinTxtr version. Parameters that are no longer needed may be used for the newer statistics. For example, this was done in the current WinTxtr so that the skid number computed by a skid server could be attached directly to the corresponding texture data. The header structure does allow space for additional parameters as illustrated below. The header structure precedes each set of data points when raw texture data is desired. This is illustrated in Figure 4.2. Note that in this figure, the file may consist of only the header which contains the predicted skid number, mean profile depth, etc, or both the header and the raw data. This header structure is described as follows:

```
struct HeaderInfo
{
        double SectionLen;          //meters
        double SpdCnt;              //counts/km
        double SampRate;            //sampling rate
        double DisplayHz;           // this is where you add skid number
        double Disp1;               //mm
        double Disp2;               //mm
        double LaserScale;          //mm/count
        double LaserOffset;         //mm
        long   lMaxPoints;          //# max number of points allowed in memory
        long   lNumberofPoints;     //# of points to collect
```

```
        long    ADC1;                  //A/D reading of Disp1
        long    ADC2;                  //A/D reading of Disp2
        long    reserve (0)            //Predicted skid number
        long    reserved[19];          // reserved space kept in the structure for future
        expansion
        char    szDataFile[64];        //for datafile name
};
```

Following is a brief description of the main data collection parameters used:

SECTION LENGTH, SectionLen: The Section length represents the length of the section on the pavement for which the data is collected and/or statistics computed.

DISTANCE CALIBRATION count, SpdCnt: The distance calibration count determines the number of distance counts received by the distance measuring sensor for a specified distance unit. It is used by the data collection software to determine the vehicle speed, as well as the distance between two time pulses.

SAMPLING RATE, SampRate: The Sampling rate represents the data update rate of the particular laser sensor used for the texture measurements. The units represents the update rate in terms of the number of the samples it can collect for a time duration of a second. The current texture laser has a 78 KHz update rate.

DISPLAY HZ, DisplayHz: This parameter was used in the initial data collection software and was no longer needed. It was replaced with the actual skid number of the section as measured by the skid truck. The skid number is provided to WinTxtr via the Client that it obtained from the Skid Server.

MAX POINTS, MaxPoints: The MaxPoints represent the number of points which are to be collected for a given texture measurement.

ALLOWED POINTS, lMaxPoints: The Allowed points represent the number of points that are allowed in the memory. Because of the data collection rate, 156,000 bytes per second, a fixed number of data points (MaxPoints) is collected and stored in memory for each measurement. The data is then saved on disk as specified by the client following data collection for a pre-specified section length. The lMaxPoints statistic is used to indicate the maximum of data points that can be collected and stored in RAM in a particular embedded processor system and was useful for the DOS based system, Txtr for specifying MaxPoints.

DISP1 : This represents the first displacement reading.
DISP2 : This represents the second displacement reading.
ADC1 : This represents the first A to D converter reading.
ADC2 : This represents the second A to D converter reading.

WinTxtr: Detailed Use Cases

Action	Response
Connect to Server	
This use case begins with request to connect.	System checks whether a server is listening to cater the request to connect.
This use case ends when the Client gets the acknowledgement *"Connection Established"*.	If server is instantiated, it connects itself to the client.
Request Data Collection	
This use case begins with the Client sending a request message string 'START'.	The server checks if the initialization file '*wintxtr.ini*' exists. If the file does not exist, server creates the one with the default parameter values, and accesses the PCLIM to collect the data
Request to Save Data	
This use case begins with the client sending a message string"SAVE" <filename> <skid number>".	Server checks whether the data was collected before the arrival of the 'SAVE' request. Server creates a file with a name provided by the client. If the data was not collected, the server saves the parameter information with the skid number in the file.
Request to Set Speed Count	
This use case begins with the client sending a message string "SET <speed count>".	The server checks if the initialization file '*wintxtr.ini*' exits. If the file does not exist, server creates the one with the default parameter values.
This use case ends when the client gets the acknowledgement *"Distance per Km is changed"*.	The server modifies the speed count.
Request to End Server	
This use case begins with the client sending a message string "ENDP".	The server terminates.
This use case ends when the client gets the acknowledgement *"Server Terminated"*.	

Table 4.1A WinTxtr: Detailed Use Cases

WinTxtr: Requirements from Server

R1	Server must start listening to client as soon as it is instantiated.
R2	Server must start the data collection upon the request 'START'.
R3	Server must check before collecting the data, whether the initialization file named 'wintxtr.ini' containing the header information exists. If the file does not exist, the server should create one with the default set of initialization parameters.
R4	Server must save the data in the binary format file named using the name provided by the client with the 'SAVE' request. It should also save the Skid Number provided by the user in the same file. The server must also save the values of the parameters for which the data was collected.
R5	Server must modify the Speed Count parameter, one of the required system parameters, upon the 'SET' request from client, to the value provided with the request.
R6	Server must terminate upon the 'END_SERVER' request from the client.

Table 4.1B WinTxtr: Requirements From Server

WinTxtr File Structure

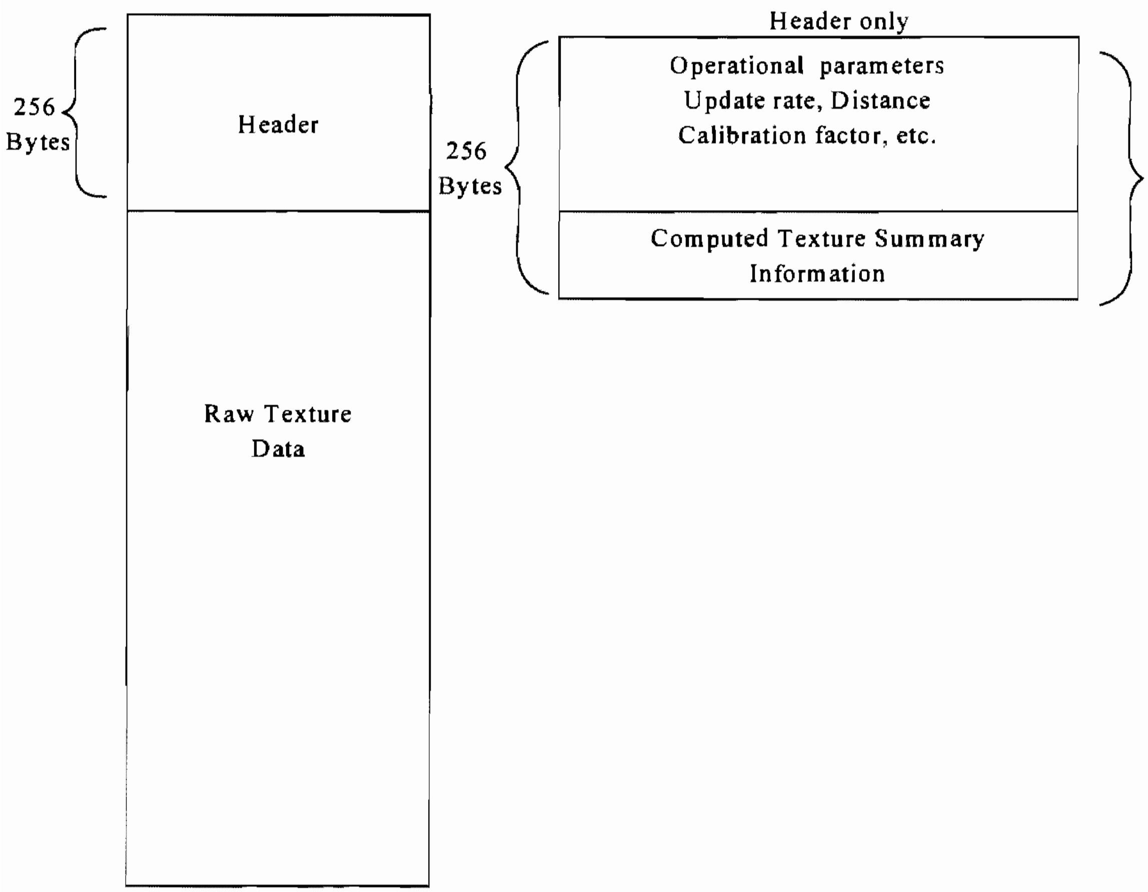


Figure 4.2 WinTxtr File Structure

These units are used for computing the actual displacement represented by a bit of the laser value. It is typically hard coded to 16 micrometers in the laser scale factor (next), the theoretical value specified by the texture laser.

LASER SCALE FACTOR : The Laser Scale Factor is computed using the formula Displacement 2 – Displacement 1) / (adc2 – adc1).

LASER OFFSET, LaserOffset: This reading represents the offset given by the laser used for the data collection.

The WinTxtr Server State diagram is illustrated in Figure 4.3. As indicated in this figure, data collection involves essentially two steps. First, a command to begin collecting data for a specific test section, START, and then saving this data to a file as specified by the Client, SAVE. The amount of data is fixed by the texture initialization file, WinTxtr.ini. The contents of this file is the header information described. This file is either generated when WinTxtr is invoked, or a preexisting file by this name is used. When data is collected, the header file is copied from the WinTxtr.ini file. After the specified number of points have been collected, the Client uses the SAVE command to specify the file name that the data set is to be saved under. It also allows for the Skid number to be embedded in the header file if available from the Client. The ENDP command will terminate WinTxtr. The Client may change the distance calibration factor using the SET command so that the number of distance pulses per unit distance can be set for the particular vehicle in which the texture system is installed. The other items are fixed as they typically will not vary unless a different laser is used. The main classes used by WinTxtr are described in the next section.

4.2 WinTxtr Server Classes

A description of the main classes and functions used by WinTxtr is provided in this section. Figure 4.4 provides the classes and functions used in the server dialog. After the server is instantiated, it begins listening for commands from the client that specify or direct the data collection operation.

The first class or Server Dialog implicitly calls the OnInitDialog() function. The OnInitDialog() function then does the following tasks:

1. BOOL CSockDlg::OnInitDialog() :This class initializes the header information. This is done by maintaining an initialization file 'wintxtr.ini'. The program first initializes the header information parameters to default values. Then it checks whether wintxtr.ini is already present on the hard disk. If yes, it reads the wintxtr.ini file and overwrites the default settings for header parameters by the corresponding values in the file. If wintxtr.ini is not present on the hard disk, the program creates one with the default values.

State Diagram

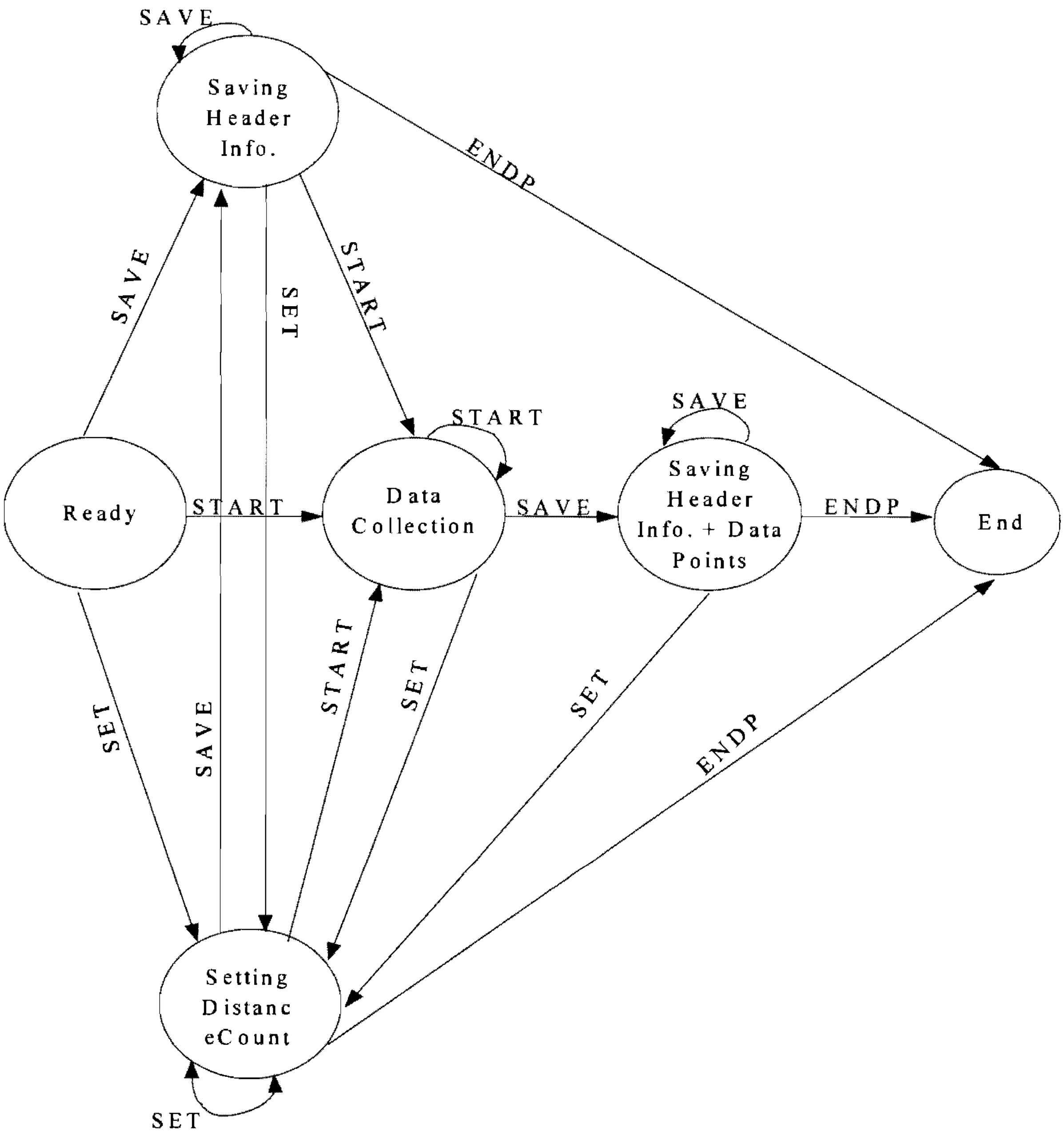


Figure 4.3 WinTxtr State Diagram

2. OnInitDialog() function calls 'GetValueBack()' function of the CMainFrame class. This function performs initial computation as required for Data Collection.

3. The OnInitDialog() function initializes the server name as 'loopback' and the server port as '4000'. This is the port number and can easily be changed in accordance with the VNET specifications.

4. In the CSockDlg class, two objects of CMySocket class have been defined in order to take care of communication between the Client and Server. They are m_sConnectSocket and m_sListenSocket. The OnInitDialog() function calls the SetParent(this) functions of these socket objects. This sets the Server Dialog as the parent for both the sockets.

5. The socket is then 'created' by command: m_sListenSocket.Create(m_iPort) and made to listen to the client requests by the command: m_sListenSocket.Listen(). Thus, as soon as the Server Dialog is initiated, it starts listening to Client requests.

Processing of Clients Requests is done by the function: CSockDlg::OnReceive(). All communication between client and server takes place when using Winsock.

In the CSockDlg class, two objects of CMySocket class have been defined in order to take care of communication between Client and Server. These are m_sConnectSocket; and m_sListenSocket. The CSockDlg has been set as the parent for these two objects. The sockets have a member function OnReceive(...). When the socket receives a request, it executes its parent function (note that Server Dialog has been set as the parent) as: ((CSockDlg*)m_pWnd)->OnReceive(). Here (CSockDlg*)m_pWnd is the server dialog. Thus, eventually, the OnReceive function of the server dialog is executed and the request is processed. There are 5 standard requests from the client that are processed by the OnReceive() function of the Server Dialog:

1) START: OnReceive() calls the startCollect() function of the Server Dialog and the Data is collected.
2) SAVE <file name> <skid number value> : OnReceive() calls saveSkid() to convert the skid number passed as string into a number ('double' data type) and then calls saveFile().
3) SET <speed count value>: OnReceive() calls saveDistance() to convert the skid number passed as string into a number ('double' data type) and then calls writeInitFile() function of the server dialog.
4) END_SERVER: OnReceive() closes the server socket by the command m_sConnectSocket.Close().

Figure 4.4 illustrates the organization and the relationship with these classes.

Server Dialog

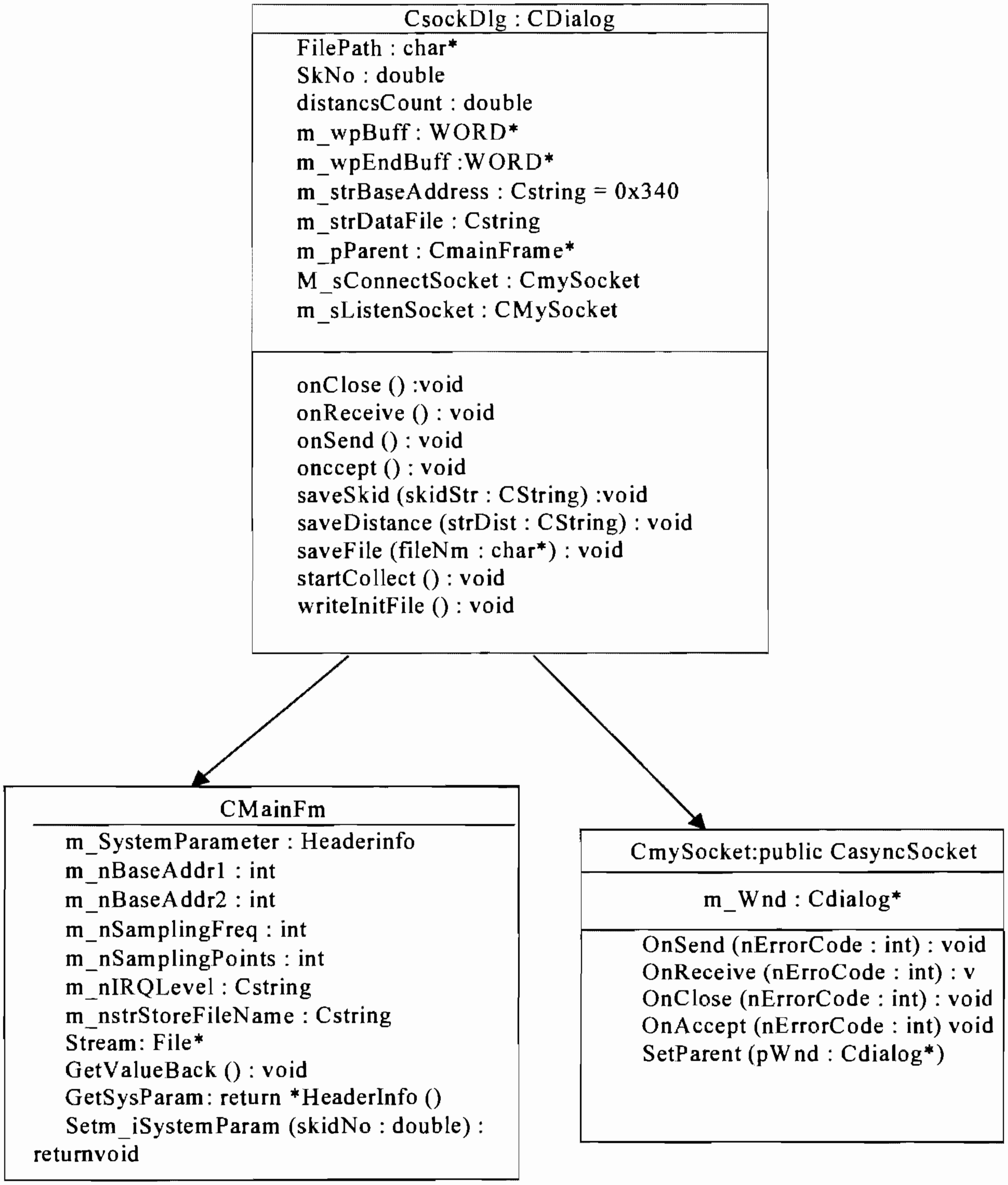


Figure 4.4 Server Dialog

4.3 WinTxtr Client

TxDOT will likely write their own Client. This was done for the data collection function used when capturing both skid number from the skid truck and texture profile. The following description is an example Client that provides the functions initially specified for this version of WinTxtr. This description can be used as guide for writing Client software, or can be used as is for data collection. The source for this, as well as the Server Software, is provided in the project program disk. The main screen capture of the WinTxtr Client is illustrated in Figure 4.5

WinTxtr Client Screen Display

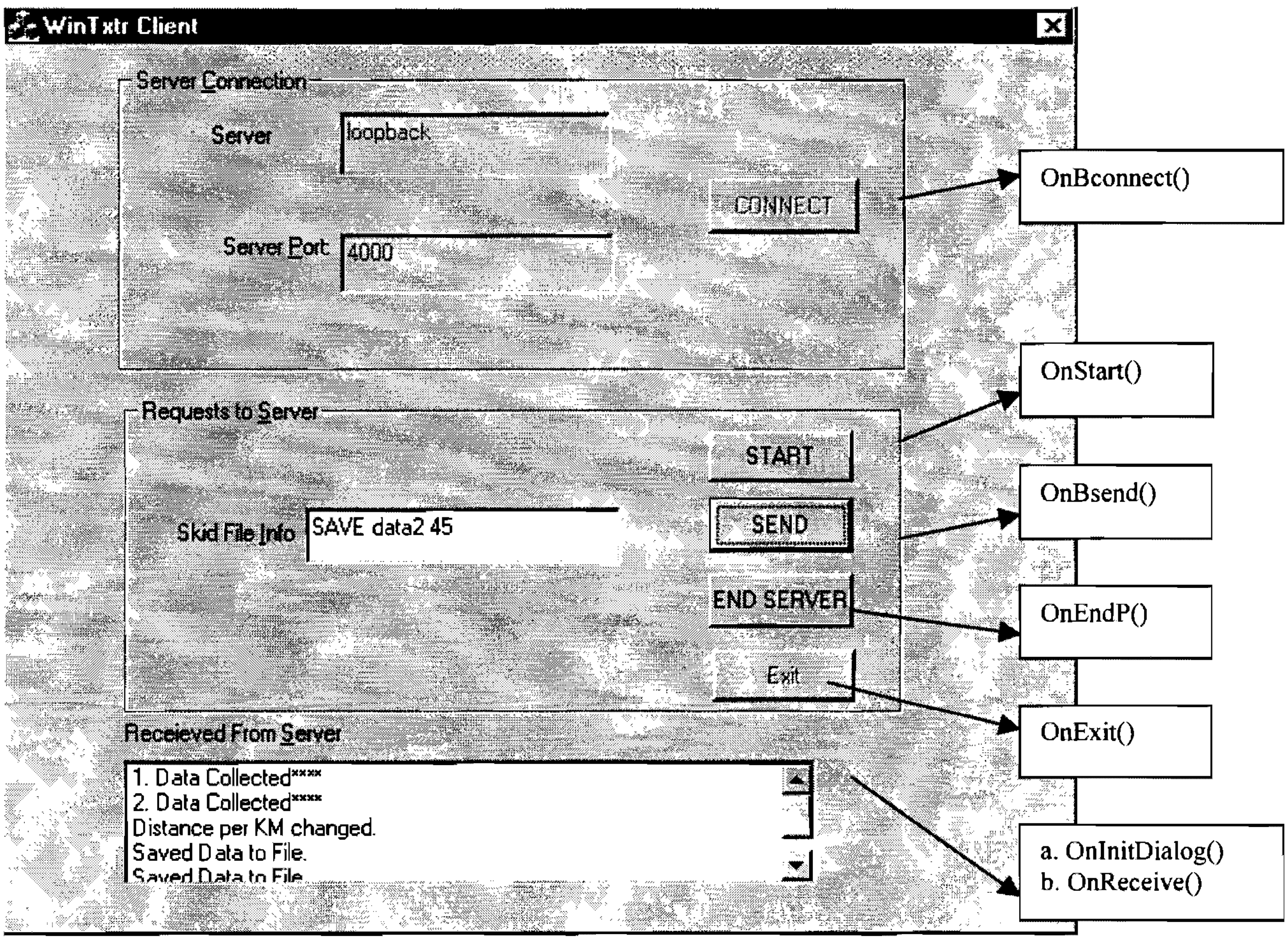


Figure 4.5 WinTxtr Client Screen Display

WinTxtr Client is User Interactive. The recommended sequence for user actions is as follows:

1. Press CONNECT Button:
   The Client will be connected to the Server.

2. Press START Button:
   The Server will start Data Collection. Once the Data Points are collected, Server will notify the Client by the message: Data Collected***

3. In the Skid File Info edit box, type SAVE <file name> <skid number value> and then hit the SEND Button:
   The Server will save the Data Points collected along with the header information in the file name specified.

4. The options 2 and 3 can be repeated as many times as the user needs.

5. For SET <speed count value>:

   In the Skid File Info edit box, type SET< speed count value >and then hit the SEND Button: The Server will set the speed count value in the wintxtr.ini file.

The SET command can be used any time, but it should be noted that it modifies the WinTxtr.ini file and not the file saved by the user using SAVE option. So, it is necessary to collect data points using START and then use SAVE to collect and save the data points for the new value of speed count.

4.4 Classes and Functions of the WinTxtr Client

The classes and functions used in the Client Dialog are illustrated in Figure 4.6. When the Client Dialog is instantiated, it implicitly calls OnInitDialog() function. The OnInitDialog() function then does following tasks:

1. BOOL CTclient2Dlg::OnInitDialog()
   a. The OnInitDialog() function initializes the Server name as 'loopback' and the server port as '4000'.
   In the CSockDlg class, two objects of CMySocket class have been defined in order to take care of communication between the Client and Server. They are m_sConnectSocket; and m_sListenSocket.. The OnInitDialog() function calls the SetParent() functions of these two objects.
   b. In the CTclient2Dlg class, an object of CMySocket class has been defined in order to take care of communication between Client and Server. It is named as m_sConnectSocket. OnInitDialog() function initializes the Server Name to 'loopback' and Port Number to '4000' and sets the parents of the CMySocket object to the Client Dialog.

Client Dialog

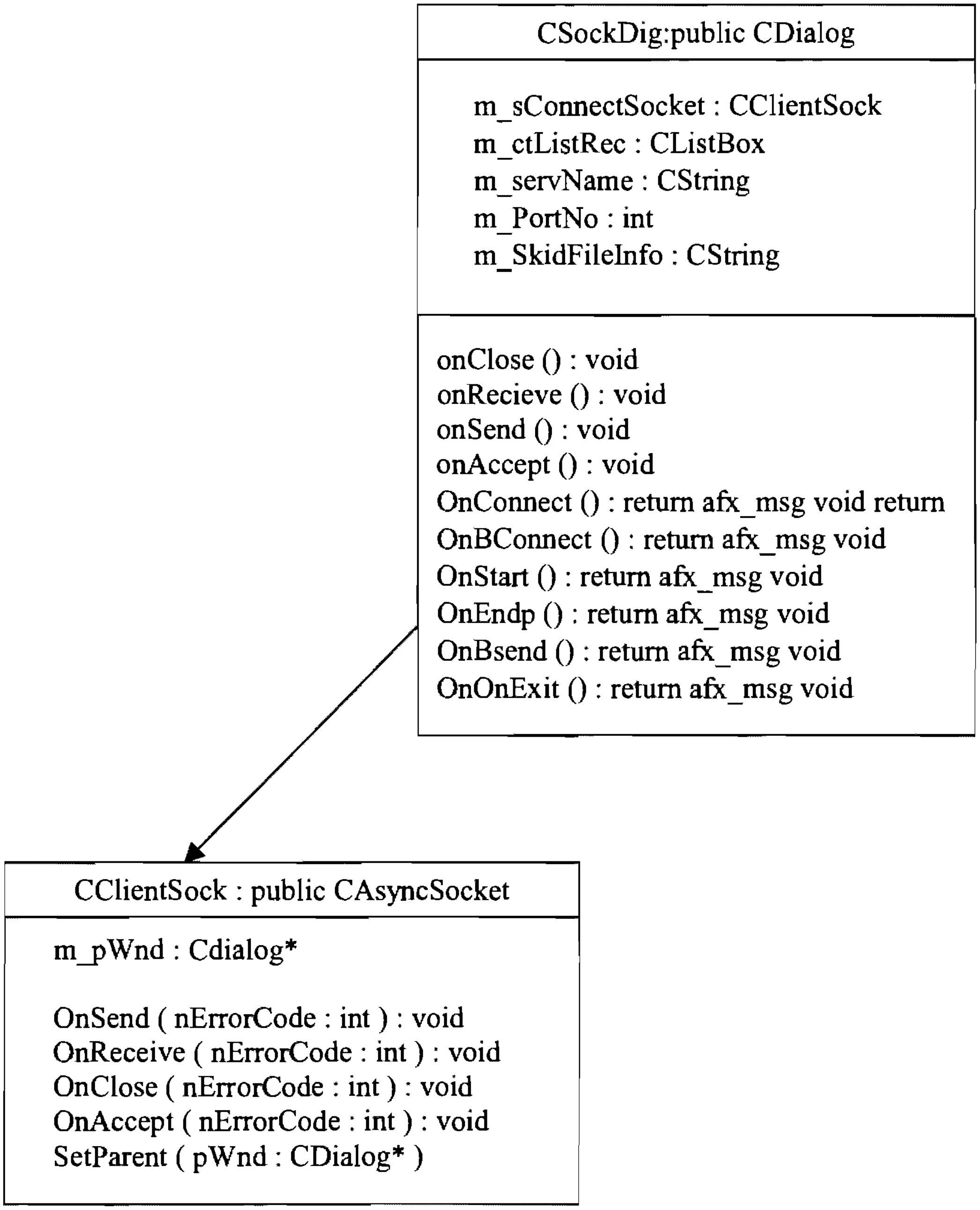


Figure 4.6 Client Dialog

2. void CTclient2Dlg::OnBconnect() : (Used to Connect the Server)

   This is the event handler for the CONNECT button. It creates the socket by calling the function m_sConnectSocket.Create() and then connects the Client to the Server by calling m_sConnectSocket.Connect(m_servName, m_PortNo);

3. void CTclient2Dlg::OnStart() : (Used to start data collection).

   This is the event handler for the START button. It creates a string 'START ' and sends it to the Server using the function m_sConnectSocket.Send(...).

4. void CTclient2Dlg::OnEndp() : (Used to end the Server connection).

   This is the event handler for the END_SERVER button. It creates a string 'ENDP' and sends it to the Server using the function m_sConnectSocket.Send(...).

5. void CTclient2Dlg::OnBsend() : (Used to request Saving of file or Setting the Speed Count).

   This function takes the string typed b the user in the Skid File Info edit box and sends it to Server using the function m_sConnectSocket.Send(...).

6. void CTclient2Dlg::OnExit() : (Used to close the Client Program).

7. void CTclient2Dlg::OnReceive() : (Used to receive acknowledgement from the server.)

   This function is called implicitly whenever the Server sends the messages to Client. Its' working is the same as that of the OnReceive() function of the Server.

4.5 WinTxtr User Manual

Operation of WinTxtr is very simple as there are only two required commands: (1) start data collection (START) and (2) save the data file of the collected data (SAVE). A SET command can be used for modifying the distance calibration parameter. This is the number of distance sensor pulses received per unit distance from the vehicle distance sensor. It is a function of the vehicle in which the texture system is installed, and thus will not often be changed. For example, the signal can come from the same sensor used by the vehicle to compute speed. This, typically, is never changed during the life of the vehicle. The set option is added so TxDOT can easily install the system on different vehicles.

As soon as the Server is instantiated, it begins listening to the CTclient requests. The screen shown in Figure 4.7 is displayed when WinTxtr is invoked. There are three frame boxes illustrated in the figure: (1) Server_Listen, (2) Commands, (3) WinTxtr Server-Client communications, and one Button labeled as Exit. These controls work as follows:

(1) Server_Listen :

This frame box gives the information about the name of the server (Server_Name) and the port number (Server_Port) on which the Server is listening to Client requests.

(2)Commands:

This frame box notifies the user about the Client request(s) (e.g. START or SAVE <filename> <skid number value> etc) as well as the messages sent by the Server after the command(s) are processed.

(3)WinTxtr:

This frame box gives the information about the version of the program user is running.

Note that user does not have to interact with the Server Dialog, except the forced exit using Exit button.

WinTxtr Server

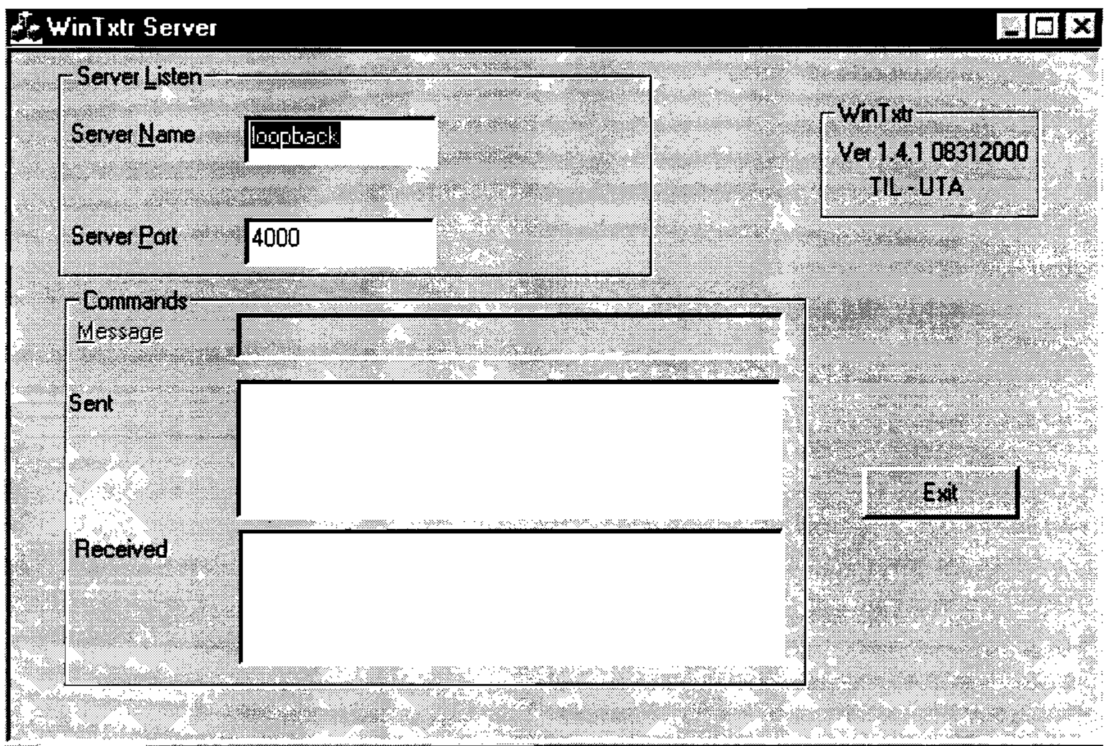


Figure 4.7 WinTxtr Server Windows Screen

Commands:

START: When 'START' request is received, a command is sent via the network to WinTxtr Server to begin data collection.

SAVE <file name> <skid number value> : After a preset number of points have been obtained from the Server, the name of the data file in which the data is to be stored is sent from the Client-Server Communication window.

SET <speed count value>: When this request is received OnReceive() calls saveDistance() to convert the skid number passed as string into a number ('double' data type) and then calls writeInitFile() function of the server dialog.

END_SERVER: OnReceive() closes the server socket by the command m_sConnectSocket.Close().

Figure 4.8 provides a typical screen capture of the server after a data collection process.

WinTK Server Screen Capture

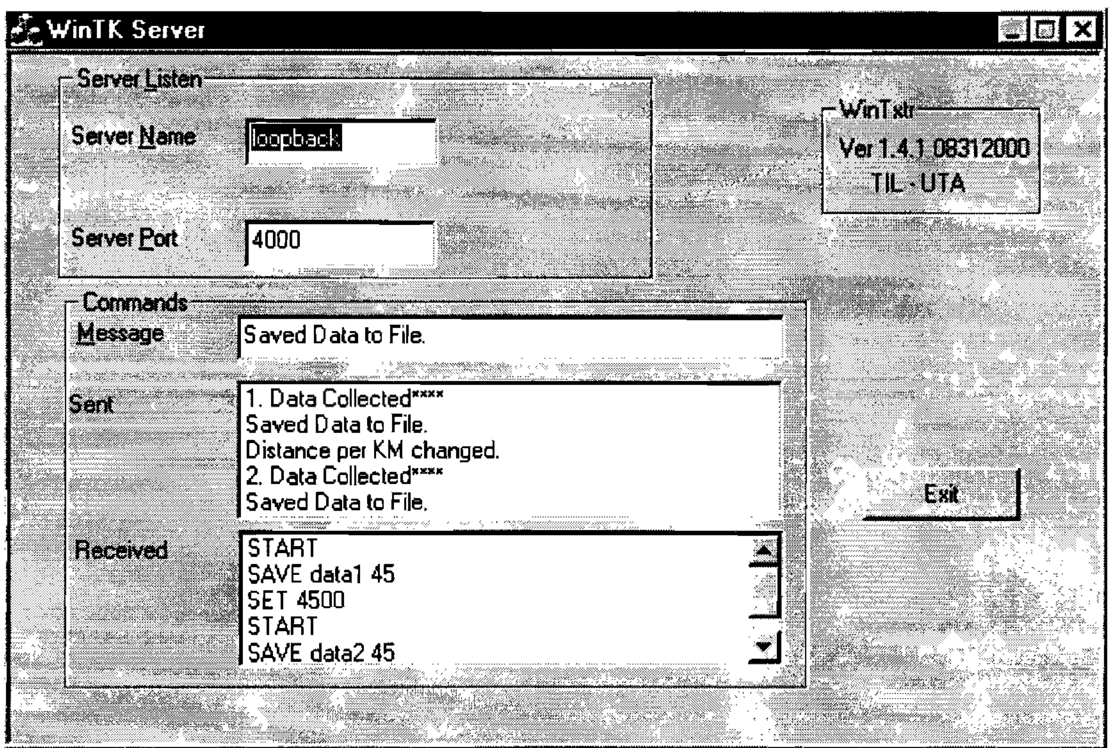


Figure 4.8 WinTK Server Screen Capture

4.5.1 Using the WinTxtr Client

The screen in Figure 4.9 illustrates the GUI provided by the Client when the Server is instantiated. The figure also shows the event handlers for different controls.

On the Client side there are 2 Frame Boxes 1) Server_Connection 2) Requests to Server, and one list box 'Received from Server'.

WinTxtr Screen Capture

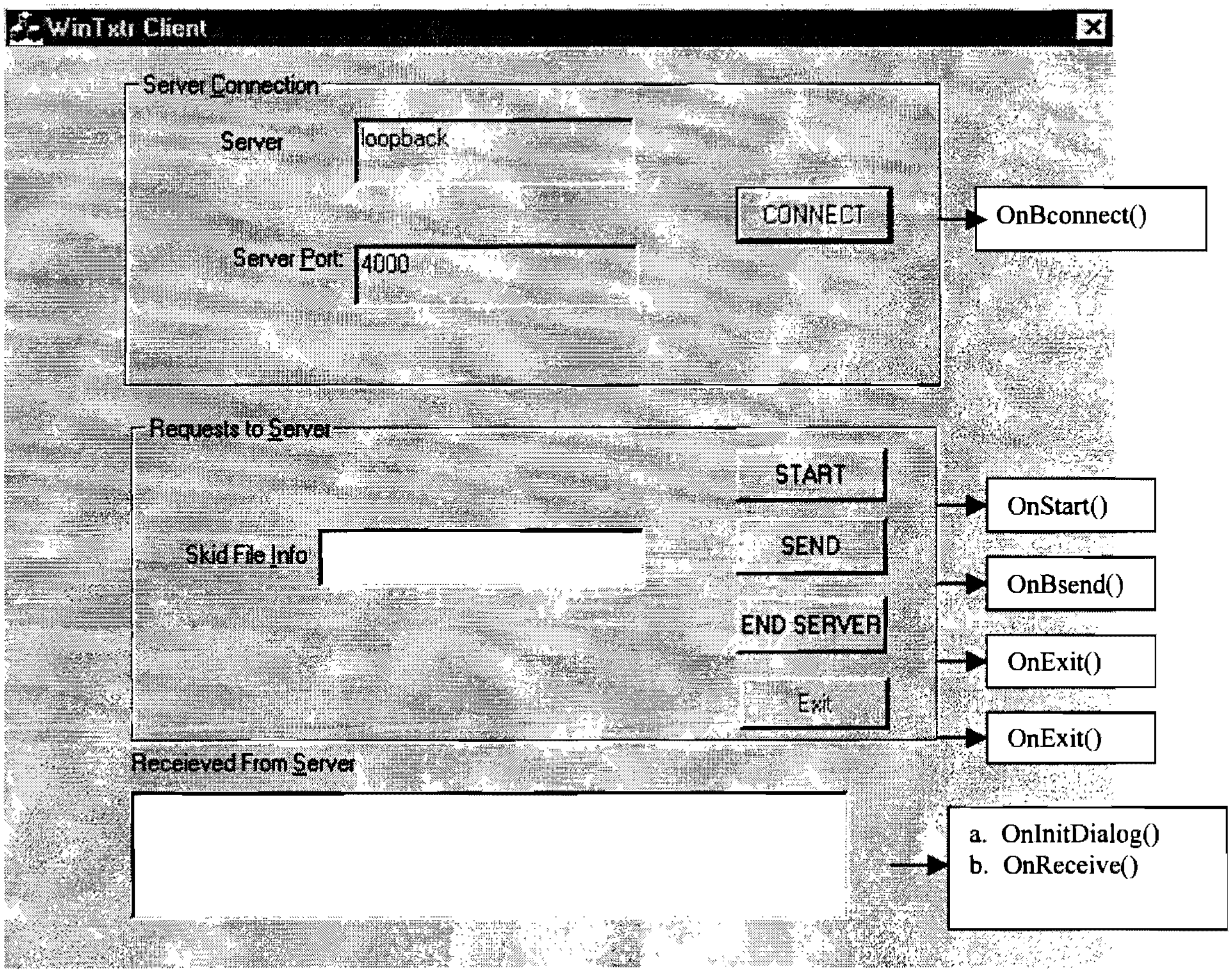


Figure 4.9 WinTxtr Client Screen Capture

1) **Server_Connection**: This box contains the edit boxes for Server Host Name and Server Port Address. In the example shown above, the Server host address was kept Loopback because it specifies the same machine address on which the program is running. In the example test shown in above snap shot, Client and Server were running on the same machine so the Server Address is Loopback but that does not have to be the case. Server Port edit box expects the port number at which server is listening which is 4000 in the above examples.

2) **Requests to Server**: This frame box has 3 buttons i.e., 1) START 2) SEND 3) END_SERVER and 4) EXIT plus one edit box (Skid_File_Info).

START button will send the command for starting data collection to the server. When the complete data collection is done by the server, the client is notified accordingly. In the screen capture shown below, client has received the notification from the server, "i Data Collected***". This signifies that the i^{th} request of Client for data collection has been processed.

SEND button will send the information written in the skid File Info edit box to the Server. Skid File Info edit box is used for sending the following two requests to the Server:

a) The name of the file in which the datapoints along with the header information that is to be saved on the Server side. Skid Number is to be obtained by Client and sent to the Server. The user should first type the SAVE as per the above format in the edit box and then hit 'SEND' button to tell the Server to process the command.

b) "SET (int SpdCnt)": The SET command provides the server with the new speed count value. The header information is updated with the new speed count specified in the above edit box. The user should first type the SET as per the above format in the edit box and then hit 'SEND' button to tell the server to process the command.

END_SERVER closes the Server.

1) **Received from Server:** This edit box displays the acknowledgement(s) from the Server after Server processes the request(s) from the Client. Table 4.2 gives the commands from the Clients and corresponding acknowledgements from the Server after processing the requests:

Request from the Client	Acknowledgement from the Server
START	$^{+}$i) Data Collected ***
SAVE <filename> <int skid number value>	Saved Data to File
SET <int speed count value>	Distance per KM changed.

Table 4.2 Client Request – Server Acknowledgement

+ : 'i' represents the number of times the request for the Data Collection was sent. 'i' ranges from 1 to 10 and gets reset to 1.

Figure 4.10 shows how the Client Screen Capture typically looks after sending several requests:

WinTxtr Client Capture Screen– Typical Requests

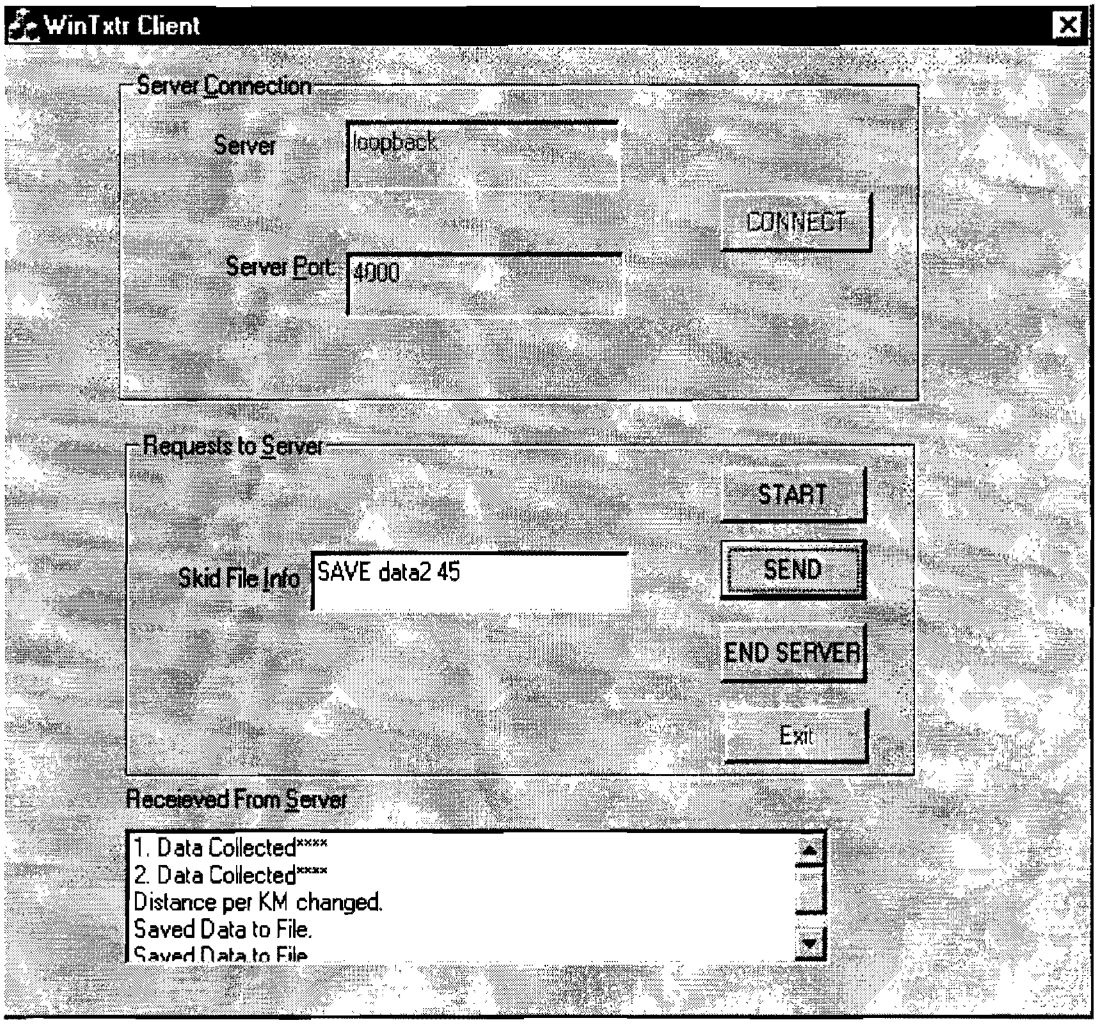


Figure 4.10 Wintxtr Client Capture Screen – Typical Requests

WinTxtr Client is User Interactive. The recommended sequence for user actions is:
CONNECT → START → SEND → END_SERVER

For SET <speed count value>:
In the Skid File Info edit box, type SET< speed count value >and then hit the SEND Button: The Server will set the speed count value in the wintxtr.ini file.

SET command can be used any time, but it should be noted that it modifies the wintxtr.ini file and not the file saved by the user using SAVE option. So, it is necessary to collect data points using START and then use SAVE to collect and save the datapoints for the new value of speed count.

4.6 Overview of the Texture Data Post Processing Module

A DOS based program, Txtrp, was written for a quick test of the validity of data collected by the DOS Data Collection Program, Txtr. A later Windows version was written but never implemented. Such quick test functions should typically be included as one of the functions of the client that is written by TxDOT. The functions normally performed which are included in the Windows based program are illustrated in the data flow diagram in Figures 4.11, - 4.14. This application analyses the actual collected data and displays the characteristics of the collected texture data in terms of the sampling information, distance and speed information, and also certain statistical information such as mean and the standard deviation, and generates the formatted output file consisting of the voltage values of the data and other formatted information related to the data. The functionality of the Post Processing Texture Data Analysis Application is as follows:

1. The user can select the input binary data file that has to be processed.

2. The header information will be displayed as the user selects the input file.

3. The user can process the data and can get all the analysis details.

4. The user can select the option to display the output file that is being generated.

The following are the analysis measurements that will be computed by the Post Processing Texture Data Analysis Application.

1. Header Information: Section Length, Speed Count, Sampling Rate, Display Hertz, Laser Scale Factor, Laser Offset, Displacement 1, Displacement 2, ADC1, ADC 2, Maxpoints and Allowed Points.

2. Sampling Information: Total number of samples that are collected, number of good samples and the number of bad samples.

3. Speed/Distance Information : Speed of the Vehicle in Kmph and Mph, Number of the Distance Pulses that are collected.

4. Statistical Information : Mean and Standard Deviation of the data in terms of mm, ADC and also in mv.

LEVEL 0 DATA FLOW DIAGRAM

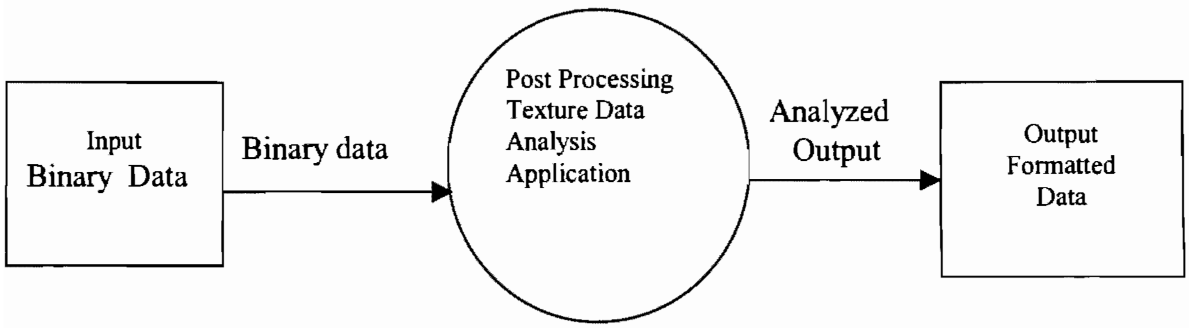


Figure 4.11 Level 0 Data Flow Diagram

Level 1 Data Flow Diagram

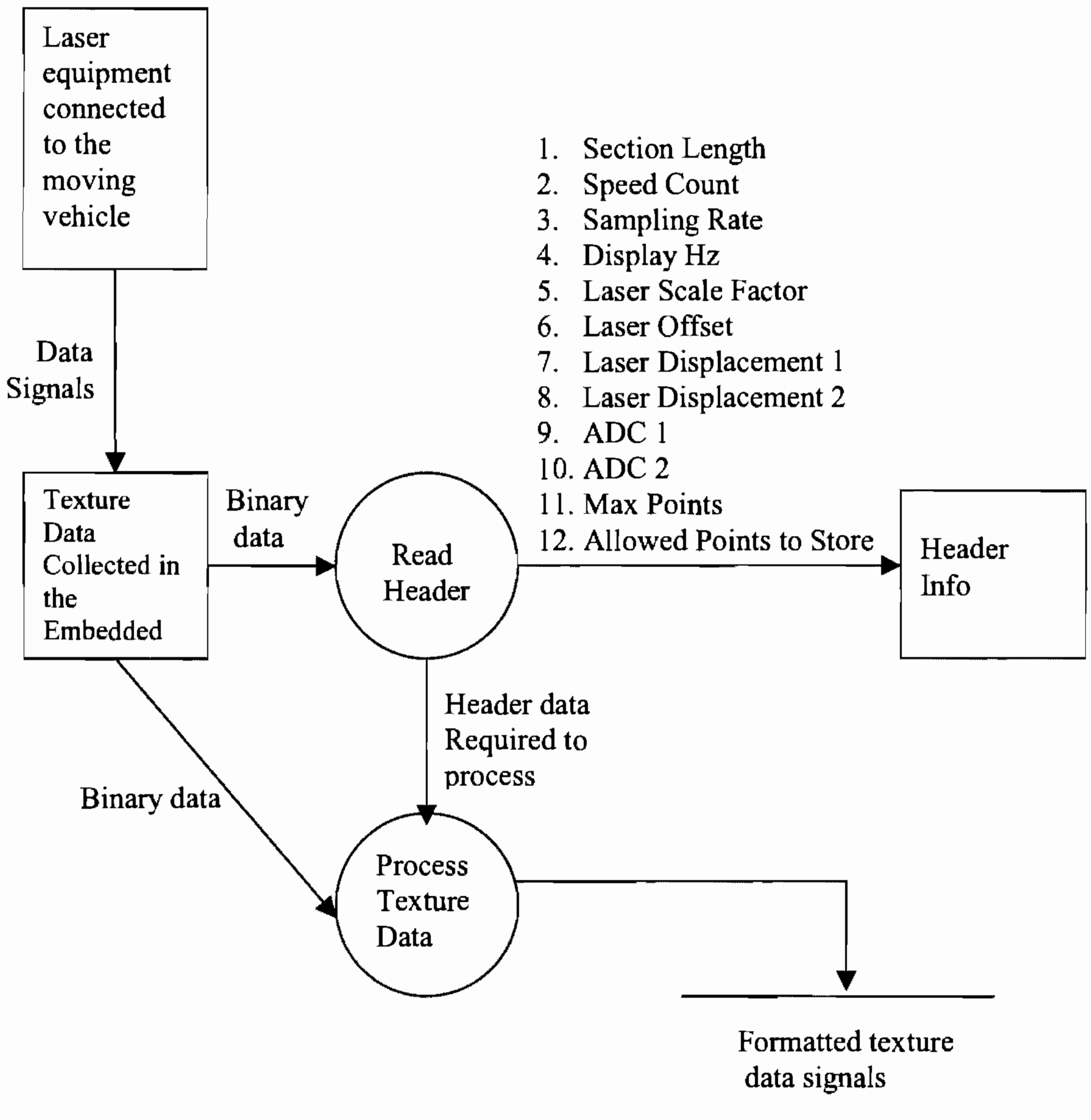


Figure 4.12 Level 1 Data Flow Diagram

Level 3 Data Flow Diagram

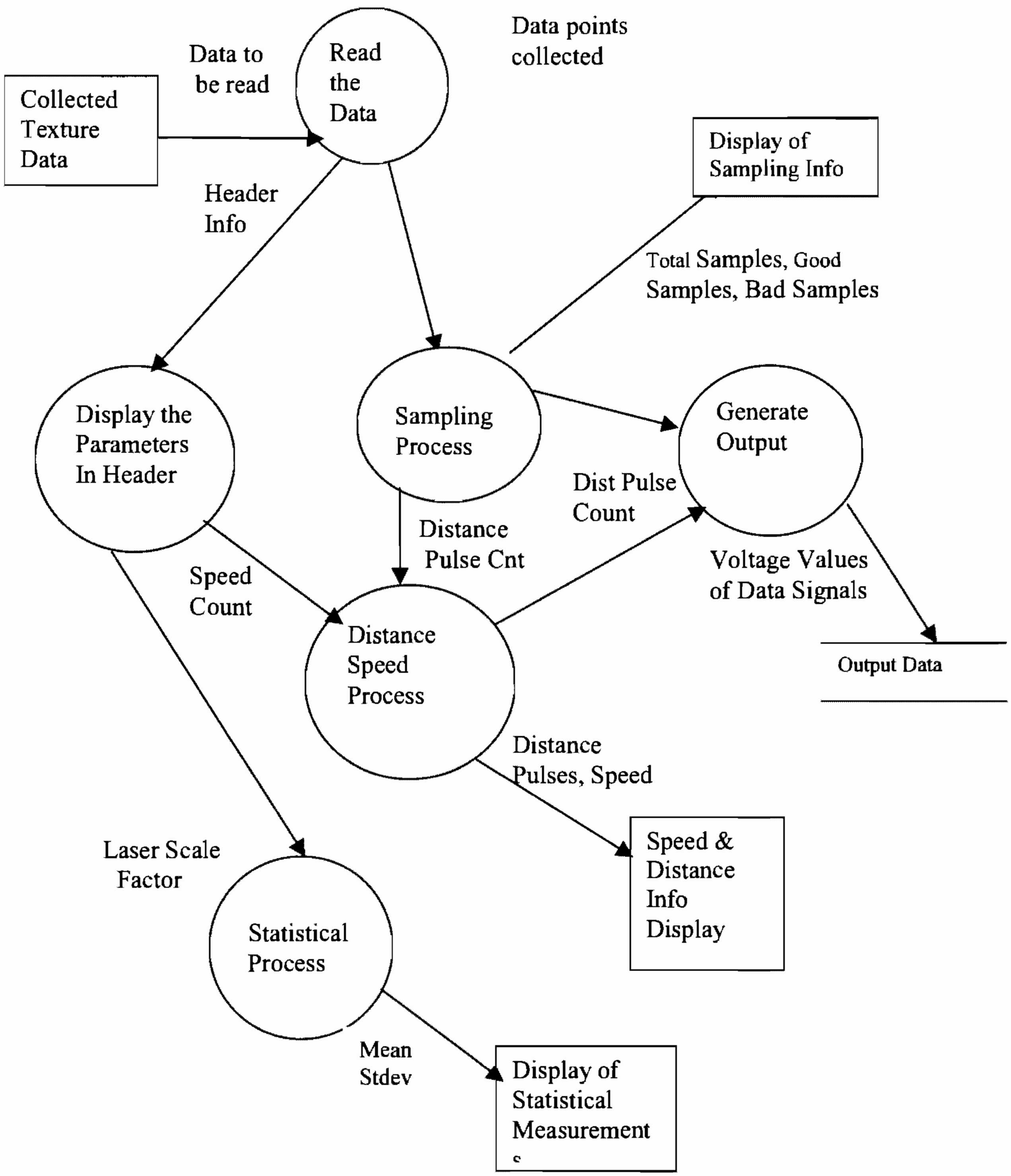
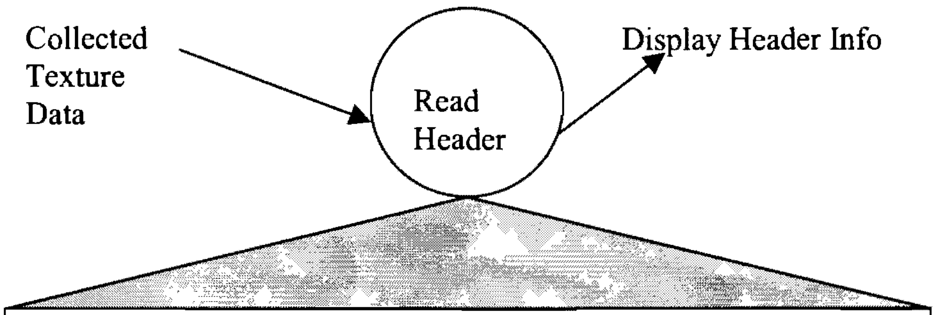


Figure 4.13 Level 3 Data Flow Diagram

Process Specifications



PSPEC for Displaying the Header Information

PROCEDURE

```
Open the binary input file
Instantiate an object of the class CsysParam
Read the object with the member function "Read(FILE *)"
If (Read(FILE *) != NULL)
    Display Section Length using GetSectionLen( ) function
    Display Speed Count using GetSpdCnt( ) function
    Display Sampling Rate using GetSampRate( ) function
    Display Hz using GetDisplayHz( ) function
    Display LaserDisplacement 1 using GetDisp1( ) function.
    Display LaserDisplacement 2 using GetDisp2( ) function
    Display Laser Scale Factor using GetLaserScale( ) function
    Display Laser Offset using GetLaserOffset( ) function
    Display MaxPoints using GetMaxPoints( ) function
    Display Allowed Points using GetAllowedPoints( ) function
    Display ADC 1 using GetADC1( ) function
    Display ADC 2 using GetADC2( ) function
Else
     Error Message
```

Figure 4.14 Process Specifications

CHAPTER 5

Texture Analysis Methods

5.0 Introduction

In this Chapter several analysis methods are described. One of the main objectives of the research effort was to develop a system that could measure raw texture at normal driving speeds. A high-speed system is discussed in the two preceding chapters that can be used for this purpose. This chapter will discuss several statistics that can then be used to further classify the raw texture data. The micro texture system discussed in Chapter 2 is much too slow for network level usage. Because of the measurement speed limitations of the microtexture system less data was acquired and analysis done.

The potential use of lasers for macro and microtexture measurements are directly affected by certain characteristics of the displacement measurement process. It is first appropriate to discuss these characteristics as they affect conclusions drawn and/or confidence levels in the data collected using the Lasers in the two acquisition systems.

5.1 Laser Characteristics that Affect Texture Measurements

The specifications generally given by the manufacturer regarding the resolution and accuracy of a laser displacement system are often misleading. The specifications quoted are the ones expected by averaging a set of samples for a particular target position. Figure 5.1 illustrates the effect of using different numbers of averaged readings. The readings in this figure were taken from a 78 KHz texture laser during a test conducted at UTA. This laser, which has a 0.63 mils (16 microns) resolution and 2.5 mils (64 microns) accuracy, is the one used by the high-speed texture system.

In Figure 5.1 successive readings were taken at a fixed distance on one particular target. First, an average of about 2000 readings were obtained to establish the current reference. Then the distance between the measured distance and this reference were taken for 1000 points and shown in the figure. The figure illustrates the resolution for the first point about 2 mils or 53 microns, then the average of the first two, followed by the average of first three, four, etc. That is, note that it took about an average of 66 readings before the 0.63 mils (16 microns) resolution was obtained. The number will vary depending on the sample type. Certain targets, for example, asphalt, will typically absorb more light than concrete, and thus will require more samples to meet the quoted resolution. For example, the surface of a floppy disk was used as the target. For this case the stated resolution was met after averaging seven points.

Resolution vs Number of Points Averaged

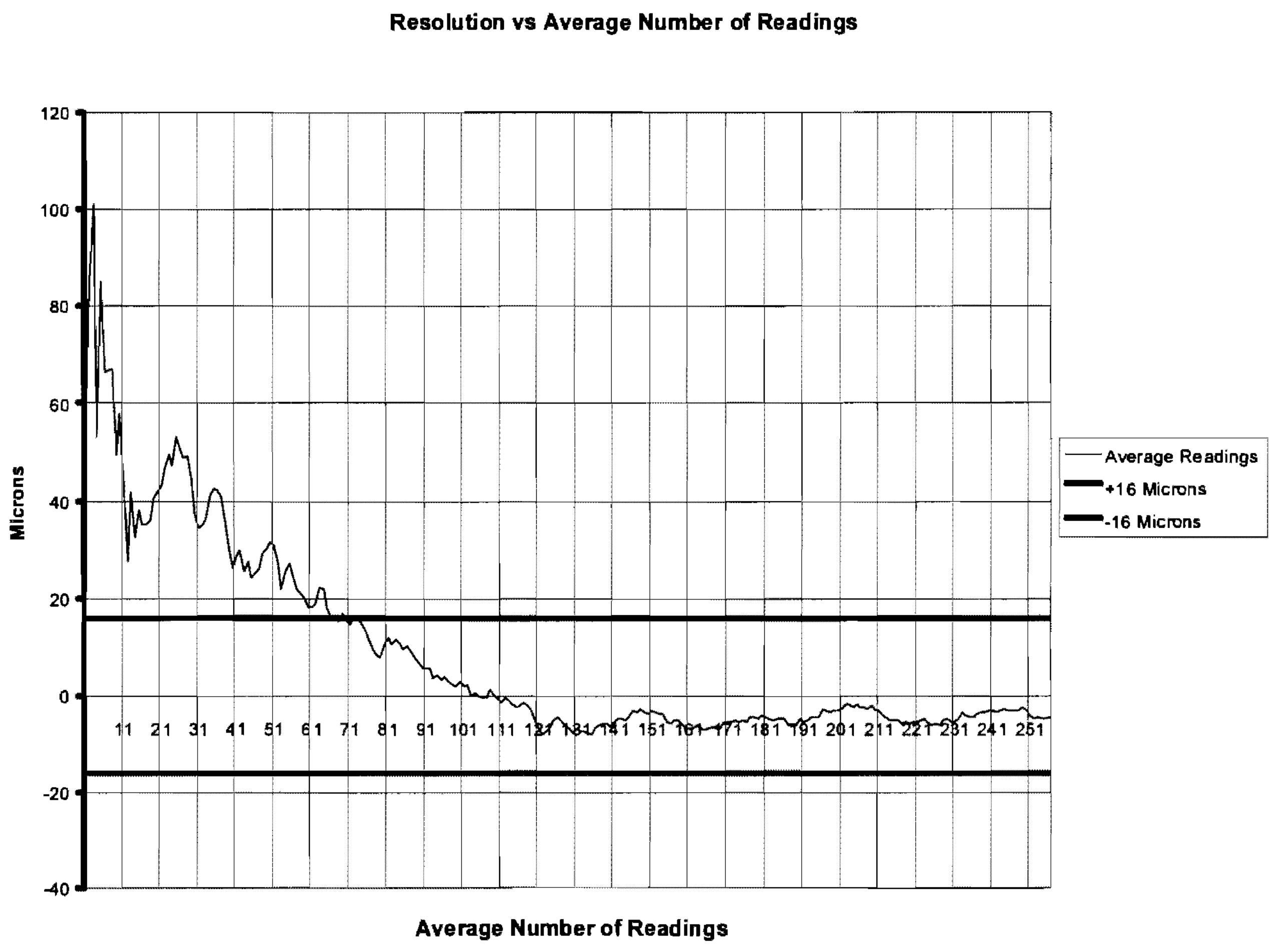


Figure 5.1 Resolution vs Number of Points Averaged

Besides the target material type, texture spot size and laser sampling or update rate also play an important role. The spot size will limit the resolution in the travel direction. For example, sampling at one sample per 9.1 mils (0.25mm) with a laser beam of 1mm won't result in 9.84 mils (0.25mm) resolution along the direction of travel.

Some laser manufacturers will give a true resolution and accuracy reading. However, for these, they perform enough averaging within the laser equipment before sending it to the data acquisition system and thus results in much lower sampling rates. That is, after the appropriate number of readings have been obtained and averaged as indicated in Figure 5.1. For example, for the Aromat system described in Chapter 2, resolution is directly related to the internal averaging rate that can be set to different values.

Since the example above is for the texture laser used in this project, it would seem to indicate that the data obtained is not adequate for texture readings. Or, at least a resolution of around 1.65mm (66*16) is the best that can be expected for vertical resolution. The horizontal resolution would typically be limited by the spot size, around 0.7mm. To date, significantly better lasers for the high-speed requirements are not commercially available so the Selcom Laser was selected. However, it could be possible that by continuously acquiring and averaging the laser readings, a 16-micron surface plane is obtained, and statistics from this plane indirectly provides a measure of surface texture.

During the project the laser systems were used for collecting data at several different sites. These are identified in the following section so that they can be referenced in the subsequent discussions in the chapter.

5.2 Data Acquisitions Sites

During the project data was collected at a number of sites for both system testing and analysis of the raw texture data. Those sites that are referenced in the chapter will be briefly discussed and provided a reference name. Exact locations will not be specified but can be obtained if necessary from TxDOT project personnel as they typically are used for skid measurements and testing. Any additional information about a site will be given, when referenced, if necessary. The first site, SW, is a road near the Pavement Design Section at TxDOT. For this site a history of skid data had been collected and was useful for determining relationships between several of the statistics below. It was also used for an experiment, discussed in the next section, to determine the effect of speed and laser type on the texture readings. This site consists of short sections in the inside wheel paths and on the shoulder usually next to a wheel path. For reference purposes SWS1 and SWW1 would represent both shoulder and wheel paths for section 1 respectively.

The second site, used on several occasions, was the skid calibration sections at the Central/Western Field Test and Evaluation Center at the Texas Transportation Institute in Bryan, Texas. Four test pads were used for both the micro and high-speed macrotexture measuring systems. This site is referenced as PADj where j indicates one of the four sites.

The next three references are made to three groups of sites that have been useful for determining if a relationship could be found between various texture statistics and SN. All three data collection activities were scheduled and data collected by TxDOT personnel who should be consulted for the exact locations. For the purpose of the discussions in this chapter each collection of data sites will be referred to as the month, or month and day as in the case of the 0309 reference that the data was collected. This reference will refer to the time the data was collected. For collecting data from the three sites TxDOT mounted the texture laser on their Skid Truck (ASTM 524 smooth tire). This was done so both texture and skid could be collected at approximately the same location. Figure 5.2 illustrates the texture laser mounted on the skid trailer that was used on these sections.

The 0309 data consists of a number of locations around the Austin area. These collections of sites were run using the skid-texture combination by TxDOT project personnel on March 9, 1999.

The 06 collections of data sites was actually run by one of the TxDOT District personnel as skid data was collected during normal data collection activities for that District. It was run primarily in June of 1999.

Finally, District personnel also ran the 07 data collections and although it included some data from June 1999, most of the data was taken in July 1999.

All data was collected using the original DOS based data acquisition program Txtr. The DOS program provided for an individual data collection mode, or an automated mode. For the individual mode, the user would specify the number of data points used, the type of laser, i.e., its sampling rate, resolution, etc., (described in the header file in the Chapter 4). Following each run, the user would then specify the file name the data was to be stored under. For the automated mode, each set of data was started by an electrical signal, which was connected to the skid computer. When a skid was to be made the texture data would be first collected, followed by the skid. Thus the skid software controlled the texture system. For the automated mode, the texture file was given a name associated with the date and time the data was collected. For example, the file, '08031433.19', is the data that was collected August 3 at 1433.19. The only problem that occurred with this method is that even though the skid and texture data was collected at approximately the same time, depending on the operating system, the timing could differ a second or so. The texture file names were then related to the appropriate SN by TxDOT project personnel and then later provided to UTA researchers for analysis. The process provided a lot of data but was

Skid System with Laser

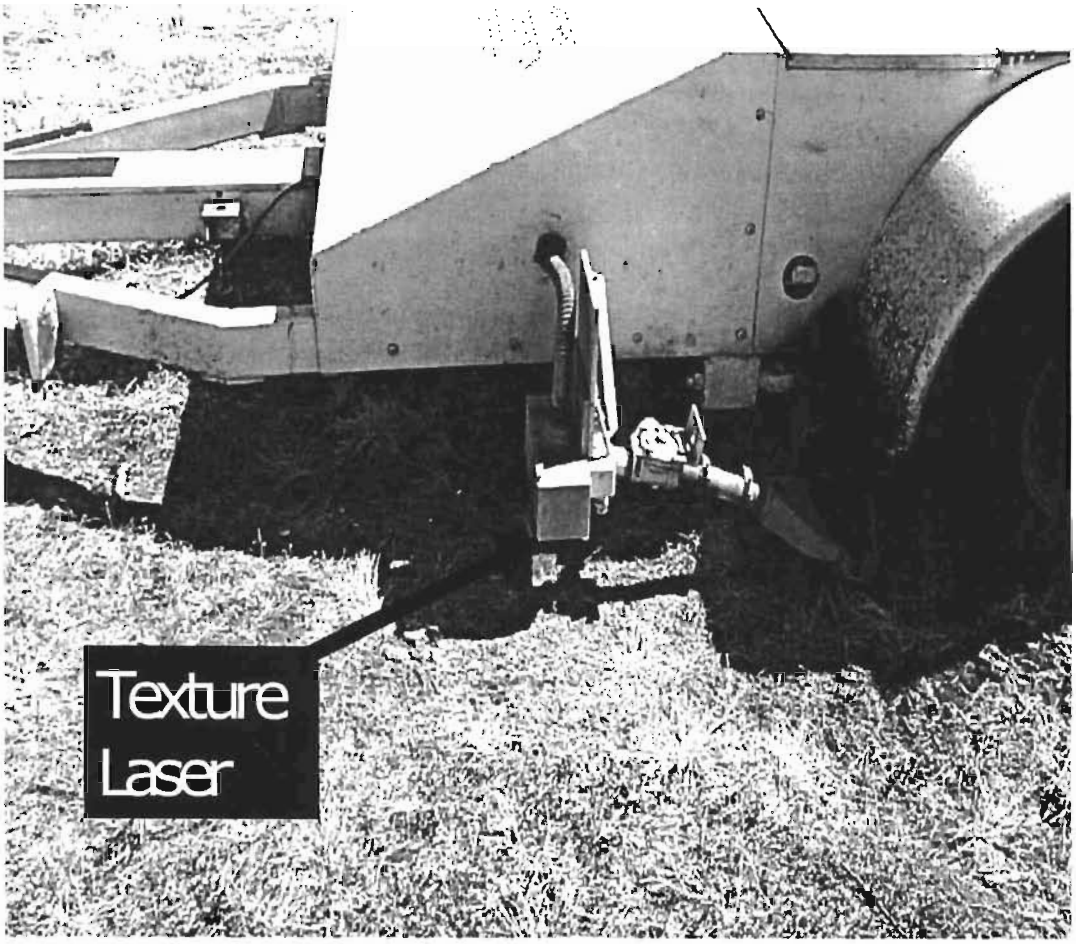


Figure 5.2 Skid System with Laser

difficult to insure that no errors occurred. It should be noted that the WinTxtr system discussed in Chapter 4 solves many of the operational problems as the skid software passes both the file name and skid number to the texture sever software so the appropriate data and its parameters are selected. As it turned out, the three data sets 0309, 06, 07, were all later found to have incorrect configuration information that affected the analysis. A program is currently being developed where this data can be corrected and thus it is planned to rerun the analysis performed and discussed in this chapter.

The next section discusses an experiment performed comparing the 32 KHz and 78 KHz lasers on the SW site.

5.3 Initial Evaluation of Selcom 78 KHz Laser for Texture Measurements

Early in the project, Selcom made a 78 KHz laser available for testing. TxDOT had been using the Selcom 32 KHz laser for profile and had begun initial tests for macro-texture measurements with this laser. An experiment was conducted where both the 78 and 32 KHz lasers were used to collect texture data over five of the SW sites, where three replications were made over each site at two different speeds. Subsequently, the corresponding mean profile depth or MPD was then compared using the Analysis of Variance. The Analysis of Variance was used to determine if MPD computed from the raw texture data was influenced by speed. Table 5.1 depicts the experiment and Table 5.2 the experiment design. From the results of these tests it was concluded that the data from 25 MPH and 50 MPH were not significantly different for the 78 KHz laser. As expected, and noted in Table 5.3, there was found a significant difference between texture types. It was noted, however, that the 32 KHz laser was affected by speed. Another interesting result was noted and that was in the number of invalid points. Recall, from Chapter 3, that when the laser does not get a return signal the invalid bit is set. For the 30 runs the percentage of invalid points noted from the 32 KHz laser was 6.6 percent, where as for the 78 KHz laser it was 0.000311 percent. In order to insure that the same sections of profile were measured, both systems were started by the infrared start sensor and the wheel path clearly marked. More data (78/32 or 2.4375) was collected with the 78 KHz laser than the 32 KHz laser to insure the same data length was covered. From these results, and using the mean profile depth statistic, the 78 KHz lasers provided improvements over the 32 KHz lasers for texture measurements.

There are, of course, other statistics that should be applied in a similar experiment to lend more creditability to these results; however, MPD is currently used rather extensively for a texture statistic by others and defined in both the ISO and ASTM. These and other statistics will be described in the sections that follow.

MPD from SW Runs

		MEAN Profile	**Profile**		**78 KHZ LASER**				
SEC 2 SHOULDER		**SEC 4 SHOULDER**		**SEC 1 WHEEL**		**SEC 6 WHEEL**		**SEC 5 WHEEL**	
sws2.1	0.924	sws4.1	1.076	sww1.1	1.164	sww6.1	0.835	sww5.1	1.115
sws2.2	0.987	sws4.2	1.061	sww1.2	1.14	sww6.2	0.881	sww5.2	1.272
sws2.3	0.97	sws4.3	1.151	sww1.3	1.079	sww6.3	0.852	sww5.2	1.272
sws2.4	0.965	sws4.4	1.036	sww1.4	0.979	sww6.4	0.91	sww5.4	1.177
sws2.5	0.949	sws4.5	1.104	sww1.5	0.97	sww6.5	0.933	sww5.5	1.101
sws2.6	0.936	sws4.6	1.079	sww1.6	1.023	sww6.6	0.936	sww5.6	1.115
ANOVA of Mean Texture Depth - Before Transformation for Reserving Homogeneous Variance									

Table 5.1 MPD from SW Runs

MTD Experiment Design for SW Runs

		SEC 2	SEC 4	SEC 1	SEC 6	SEC 5			
	SPD 25	0.924	1.076	1.164	0.835	1.115			
		0.987	1.061	1.14	0.881	1.272			
		0.97	1.151	1.079	0.852	1.272			
	SPD 50	0.965	1.036	0.979	0.91	1.177			
		0.949	1.104	0.97	0.933	1.101			
		0.936	1.079	1.023	0.936	1.115			

Table 5.2 MTD Experiment Design for SW Runs

ANOVA Results for 78 KHz Laser

ANOVA							
Source of Variation	**SS**	**df**	**MS**	**F**	**P-value**	**F crit**	
Sample	0.011	1	0.011	5.899	0.0247	8.096	**
Columns	0.299	4	0.075	41.242	2.17E-09	4.431	***
Interaction	0.038	4	0.009	5.199	0.004891	4.431	
Within	0.036	20	0.002				
Total	0.383	29					

** SPEED NOT SIGNIFICANT AT 99% LEVEL

*** TEXTURE TYPE (SECTION) IS SIGNIFICANT AT 99% LEVEL

Table 5.3 AOV Results for 78 KHz Laser

5.4 Mean Profile Depth

A statistic frequently used in describing texture is mean profile depth or MPD. The computation of this statistic has been well defined both by the ASTM and the ISO. The two dimensional surface profile for a small section of a pavement is illustrated in Figure 5.3. The profile is given in terms of amplitude and distance. Inertial reference profilers provide an estimate of the surface profile with long wavelength information removed. An estimate of the true profile can be measured by rod and level. The Texas profilers provide a band limited estimate because of the measurement method. There is considerable information defining profile frequency or wavelength, amplitude, as well as the measurements methods of road profilers and thus these concepts will not be repeated here. Texture Profilers provide measurements of the very short wavelength, related to the definitions given in Chapter 1 for micro and macrotexture. As indicated in Section 5.1, these systems must use a horizontal (sampling frequency) and vertical displacement measuring method such that the definitions of the bandwidth ranges for micro and macrotexture are met.

For longer wavelengths such as 0.5 feet to 500 feet, the combination of a displacement measuring laser and accelerometer are used to remove the influence of the vehicle suspension system in Road Profilers. For the much shorter wavelength range of surface texture, the vehicle influence is simply filtered out.

One simple statistic that might be used to characterize a surface texture profile is the surface texture variance or its square root, RMS.

$$\text{Var} = 1/n * \sum (p(i) - pm)^2 \text{ , } Rms = \sqrt{\text{var}}$$

Where p(i) represents the n profile samples in the section and pm provides the mean or first moment of the p(i) samples in the section. The mean, however, would not be useful as it should be zero, if the long wavelengths including DC are removed. The RMS value is often referred to as STD in the report when used in conjunction with the String Filter.

The Mean Profile Depth statistic is illustrated in Figure 5.4A. This statistic is simply the average value of the profile depth over a specified distance (window or referred to as the baseline). Estimated Texture Depth (ETD) is used to estimate mean texture depth and is defined by the equation:

$$\text{ETD} = 0.2 + 0.8 * \text{MPD}.$$

The baseline is typically 100 mm or 3.94 inches (related to the diameter of the spreader tool used in the volumetric method). It would seem that ETD is a primarily a measure of macrotexture and not microtexture, because of the much greater amplitude range of the longer wavelengths, although the effects of microtexture could perhaps be included if the horizontal sampling is sufficient.

A general method for computing MPD [12] is illustrated in Figure 5.4B. That is the profile wavelengths outside the areas of interest are usually filtered out. The program, Aetdv, was designed to provide the MPD and ETD. The program provides both a low and band pass filter prior to computing the two statistics. The program inputs include the name of the binary texture file, the output file the two statistics are to be written to, and the baseline length. The filter used for Aetdv is the simple averaging filter. There were two other filter types that were initially considered and tried; however, since many systems are using the averaging filter, the program was changed back to include this filter. The averaging filter, which is really an FIR filter, does not have very good frequency response characteristics; however, it does have linear phase and is easy to apply. Aetdv, as well as several other programs described in this chapter, are not documented nor supported. They were only for analysis purposes, such as described in the previous section. The results were used with third party software such as Excel and Matlab.

Texture Profile

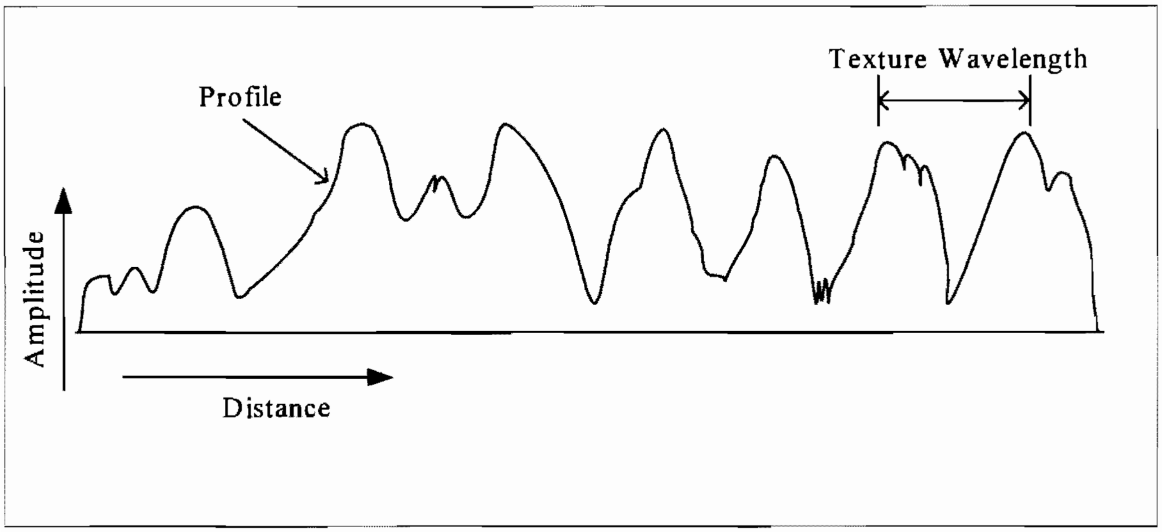


Figure 5.3 Texture Profile

Estimated Profile, Depth and Texture

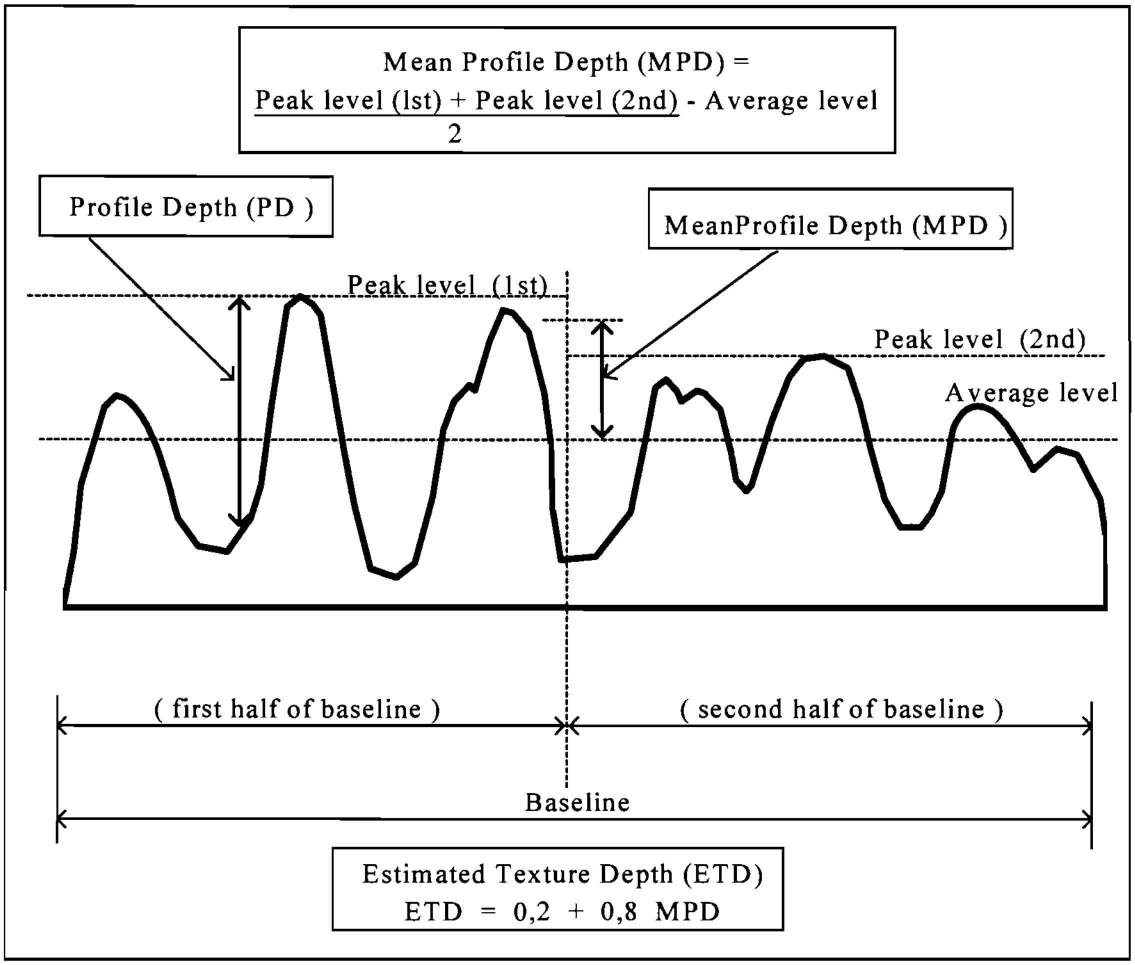


Figure 5.4A Estimate Profile, Depth and Texture

Illustration of the Measuring & Data Processing Procedure

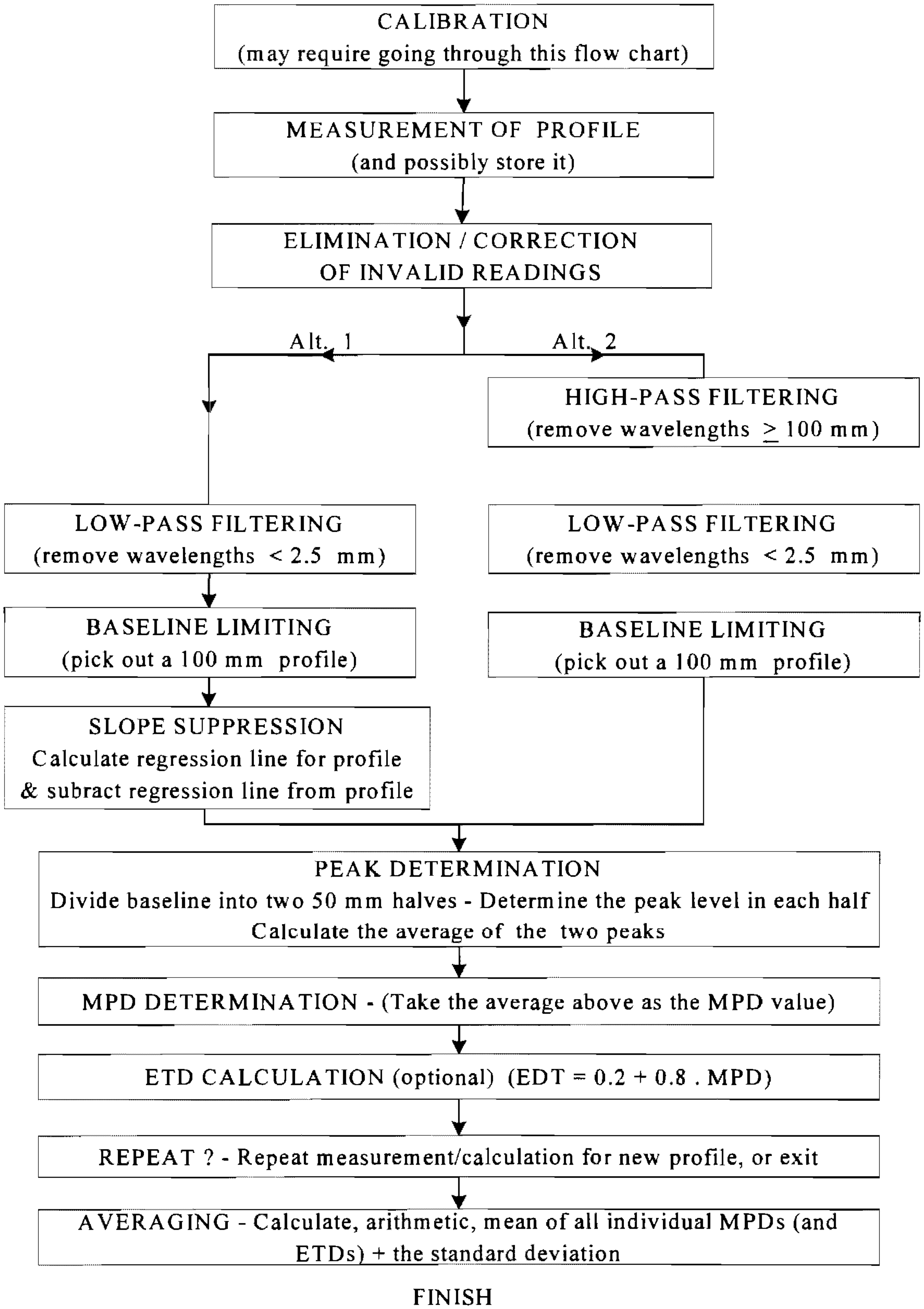


Figure 5.4B Measuring & Data Processing Procedure

5.5 Amplitude Distribution

A second analysis method applied is the use of the profile amplitude distribution [12]. This is simply a histogram of a texture sample. By normalizing this sample, an estimate of the sample probability density function, PDF of the texture profile can be computed. Both the MPD/ETD and PDF require that the data be filtered. Filtering of MPD/ETD was discussed in the previous section. PDF must be filtered because of both the bias of the location of the laser on the vehicle with respect to the target or pavement and the influence of the vehicle body. Figure 5.5 provides an excel plot of the four separate sections of test pavements from the SW site in Austin. The plots have been normalized to the 64000 points that were collected.

The first step in the procedure in computing the amplitude distribution is to determine the number and size of the bins to categorize the amplitude counts. The data is also filtered. Once the counts for the different inputs are found the results can be specified, constructing the histogram. A program was written that would provide such a histogram on a data set directly obtained from the texture data file. The histogram provides an estimate of the amplitude distribution. It can be normalized by dividing each count by the total number of counts, or size of the data file. Variations used for this method include the filter settings, and the number of bins or intervals, all of which comprise the parameter set. The characteristics of the amplitude distribution, such as the maximum height (bin with the maximum count) and shape of the distribution were then investigated to determine if these statistics would correlate well to SN as described in Section 5.7. The size and shape of the distribution can vary significantly with the parameters used to form the distribution. Figures 5.6 through 5.7 illustrate the use of the method on the 0209 data using the string filter, where the normalized amplitude is plotted vs SN.

5.6 String Filtering

One procedure found to better classify the raw texture readings, at least in relating the above statistics to SN, was noted by preprocessing by a filtering method, referred as 'the string filter'. That is, before computing each statistic described, e.g., the RMS, MPD, amplitude distribution statistics, etc., the string filter is applied in lieu of a low or high pass filter. This filtering process, derived from the string method used for computing rut, was used as a crude method of modeling the tire-surface interaction. Basically the string filter is applied as illustrated in Figure 5.8. The filter requires two parameters, base length or BL, and string length, SL. Similar to the European String Method used for rut, an imaginary string is stretched across a set of points, P1 to P16 as indicated in Figure 5.8. The set of points between SL are replaced with a new set, where all points making contact with the string have a zero value and those that fall below the string are replaced with the difference between the point and the imaginary string. The new set of 'string filtered' points are then further divided into one or more subsets, where the length of the subset is BL. In Figure 5.8, the set of filtered SL points consists of 3 subsets, BL_1, BL_2, and BL_3. A second variation of the string filter consists of including in each subset BL_i, only those points that do not make contact

Data Used for Figure 5.5

		SWS11	SWW1	SWW2	SWW3
Frequency	**Amplitude mm**	**Frequency**	**Frequency**	**Frequency**	**Frequency**
0	-32	0	0	0	0
0	-28.8	0	0.00011	0	0
0	-25.6	0	0.000156	0	0
0.00011	-22.4	0.00011	4.69E-05	0	0
9.39E-05	-19.2	9.39E-05	6.26E-05	0	0.000297
0.00025	-16	0.00025	4.69E-05	0.000344	0.000469
0.001158	-12.8	0.001158	0.00011	0.00097	0.000704
0.003380	-9.6	0.00338	0.001049	0.003787	0.002347
0.023192	-6.4	0.023192	0.018372	0.021753	0.021424
0.132801	-3.2	0.132801	0.137308	0.129718	0.122332
0.331737	0	0.331737	0.350282	0.342254	0.34867
0.347199	3.2	0.347199	0.327919	0.341315	0.34975
0.136322	6.4	0.136322	0.136526	0.136651	0.133036
0.021847	9.6	0.021847	0.025211	0.021721	0.019781
0.00169	12.8	0.00169	0.002676	0.001424	0.001111
0.000141	16	0.000141	0.000125	6.26E-05	7.82E-05
7.82E-05	19.2	7.82E-05	0	0	0
0	22.4	0	0	0	0
0	25.6	0	0	0	0
0	28.8	0	0	0	0
0	32	0	0	0	0

Table 5.4 Data Used for Figure 5.5

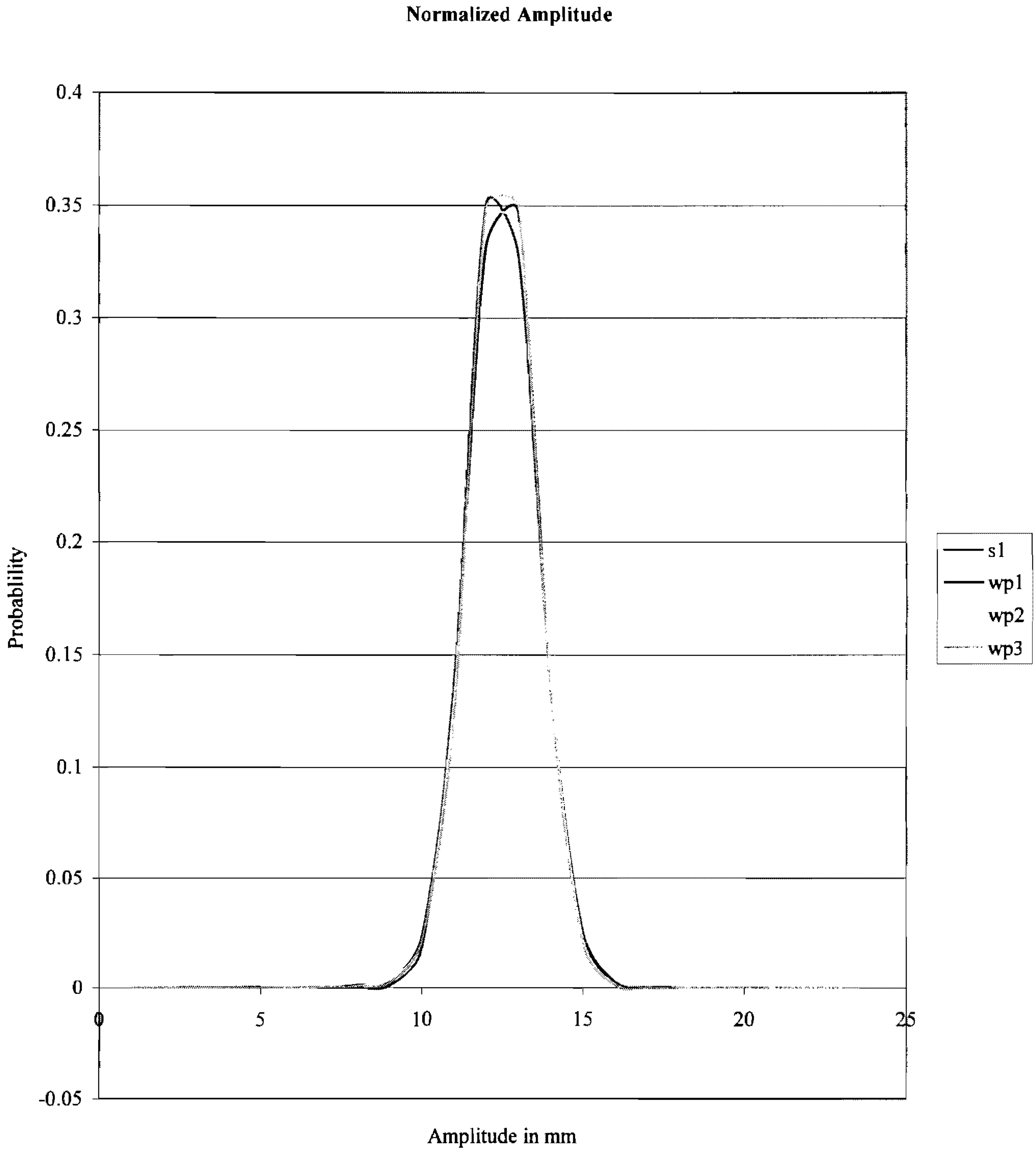


Figure 5.5 Normalized Amplitude for Four SW Sections (See Data in Table 5.4)

Amplitude Distribution 0-10

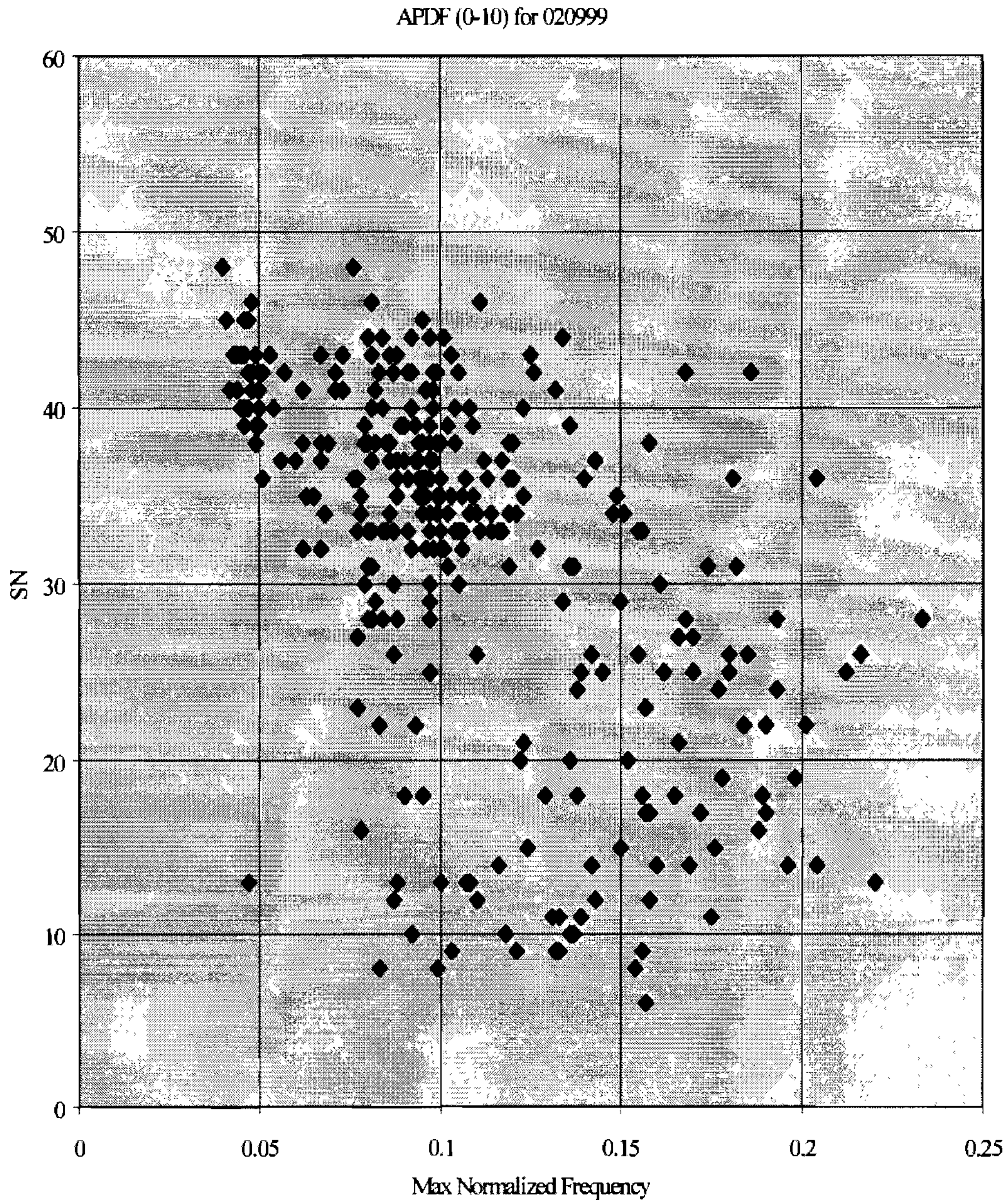


Figure 5.6 Amplitude Distribution 0-10

Amplitude Distribution 0-250

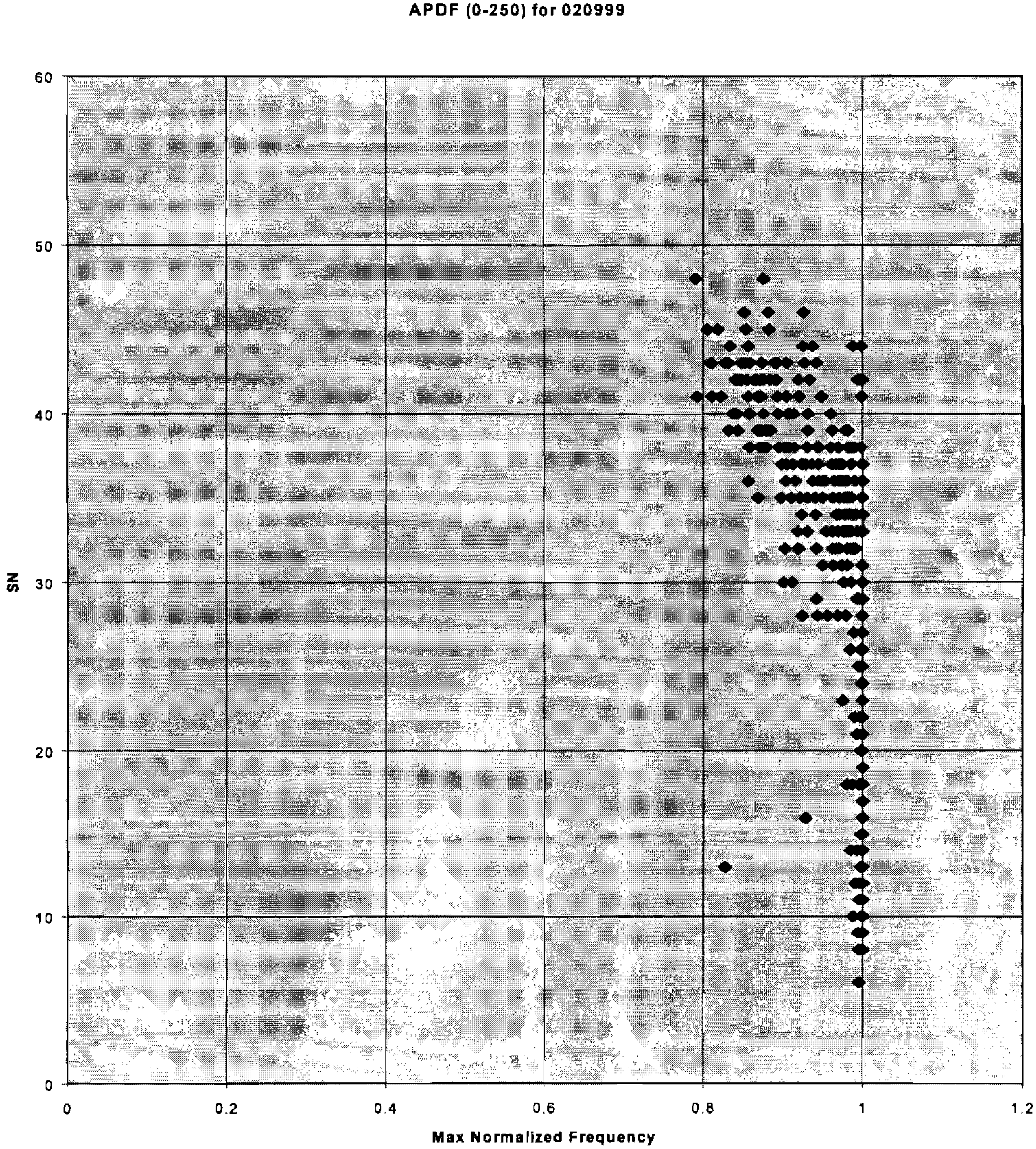


Figure 5.7 Amplitude Distribution 0-250

with the string. Although not necessary for use with each statistic, one or more of the above statistics can be computed for each BL subset and an average of the statistics for the SL set determined. For example, using the RMS statistic (also referred to as STD) and using the second variation we get for the first two subsets:

$$STD_1 = \left[\sum_{i=1}^{3} a_i^2 - \left(\sum_{1}^{3} a_i \right)^2 / 3 \right]^{1/2} \Big/ \sqrt{2} \qquad STD_2 = \left[\sum_{i=4}^{7} a_i^2 - \left(\sum_{i=4}^{7} a_i \right)^2 / 4 \right]^{1/2} \Big/ \sqrt{3}$$

$$BL_1 = (P_1, P_2, P_3, P_4, P_5, P_6)$$

$$BL_2 = (P_4, P_5, P_6, P_7),$$

$$SL = (P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_{14}, P_{15}),$$

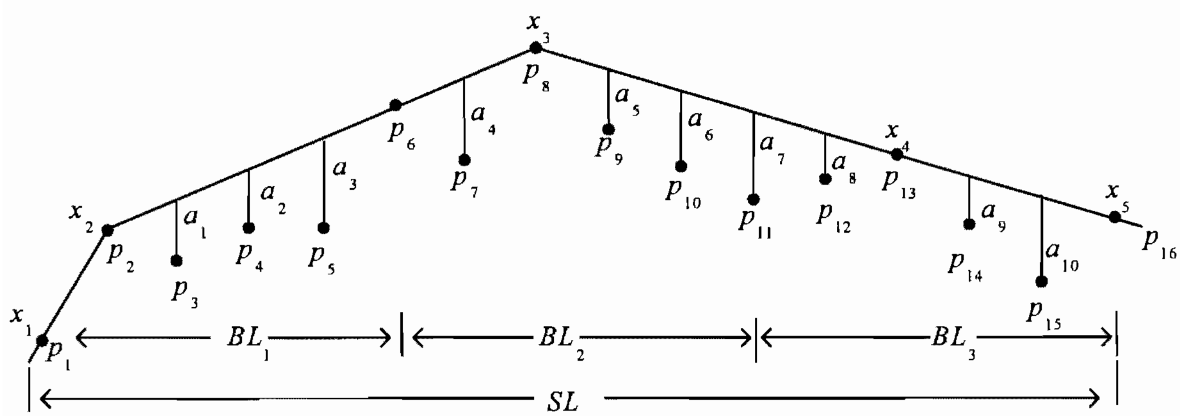
Where BL = Base length
SL = String length

The string parameters can be set the same as those used in MPD, STD, etc. That is, as noted above, the reporting interval or SL and string base length are set the same as the base length used for STD. Applying the string procedure on the different statistics described can result in many and very different results, depending on the string procedure and other various parameters used (parameters set). In the next section methods of applying the above statistics in order to develop a model for predicting SN are described.

5.7 Modeling Methods

As noted, one of the first objectives of the project after a texture measuring system was would be made to determine if one or more of these statistics could be found to relate to various frictional characteristics such as skid number. In the previous sections a number of statistics were identified, but all have various sets of operating parameters that have significant effects on the statistical values for texture classification. Thus, for a specific set of statistics, a set of operating parameters would need to found. The first approach was to try and find an optimal set of parameters for each statistic individually using SN as the objective function. For example, Figure 5.9 illustrates the relationship found by using MPD on the 0209 data. The MPD statistic was computed following the application of the string filter, and with a window length (SL) of 200mm and base interval (BL) of 5 mm. As indicated in Section 5.2, the 0209 data was later found to have an incorrect scaling factor, thus these readings would need to be adjusted appropriately. A search for an appropriate model was made that would account for the trend noted, as well as to minimize the scatter about the model.

String Filter



P_i = i^{th} texture profile point in segment less *th* SL_j

a_i = displacement from string to point

Figure 5.8 String Filter

A simple first order linear regression was first applied which resulted in an R square of 0.53. In a similar manner, applying a regression on the data and using the RMS (STD) for the same parameter set yielded slightly better results. After several tries the following transformation on the two variables was applied:

$$\log\left(60 - \mathrm{SN}\right) = \mathrm{a}\log\left(\mathrm{STD}\right) + \mathrm{B}$$

A first order regression resulted in the results provided in Table 5.5. The equation for this model is:

$$y = a - \left[x^{b} e^{\,c-d}\right]^{-1/d}$$

Applying this model in a similar manner to the parameters reporting interval 5mm, and string base length of 75mm yielded yet a better result of an R square of around 0.65. The parameter set was then varied and the corresponding R square used as the objective function. The combinations of several of the above variables were used in a multiple regression resulting in slightly better results, R square of around 0.7.

Clearly, there are many combinations of parameter sets, of the above statistics. One additional method was applied in which a combination of a genetic search procedure [18] and other expert selection processes resulted in even better results. Unfortunately, the procedure requires a number of processes yet to be automated, in conjunction with a number of different software packages and extensive computing. Because of this and the fact that the data sets used must be corrected these results have yet to be verified. Additional research is required at UTA before this new method can be verified and finalized.

Regression on Transformed Data

Regression Statistics	
Multiple R	0.776170857
R Square	0.602441199
Adjusted R Square	0.601141987
Standard Error	0.140392292
Observations	308

Table 5.5 Regression on Transformed Data

MPD VS SN W/STRING 5 - 200

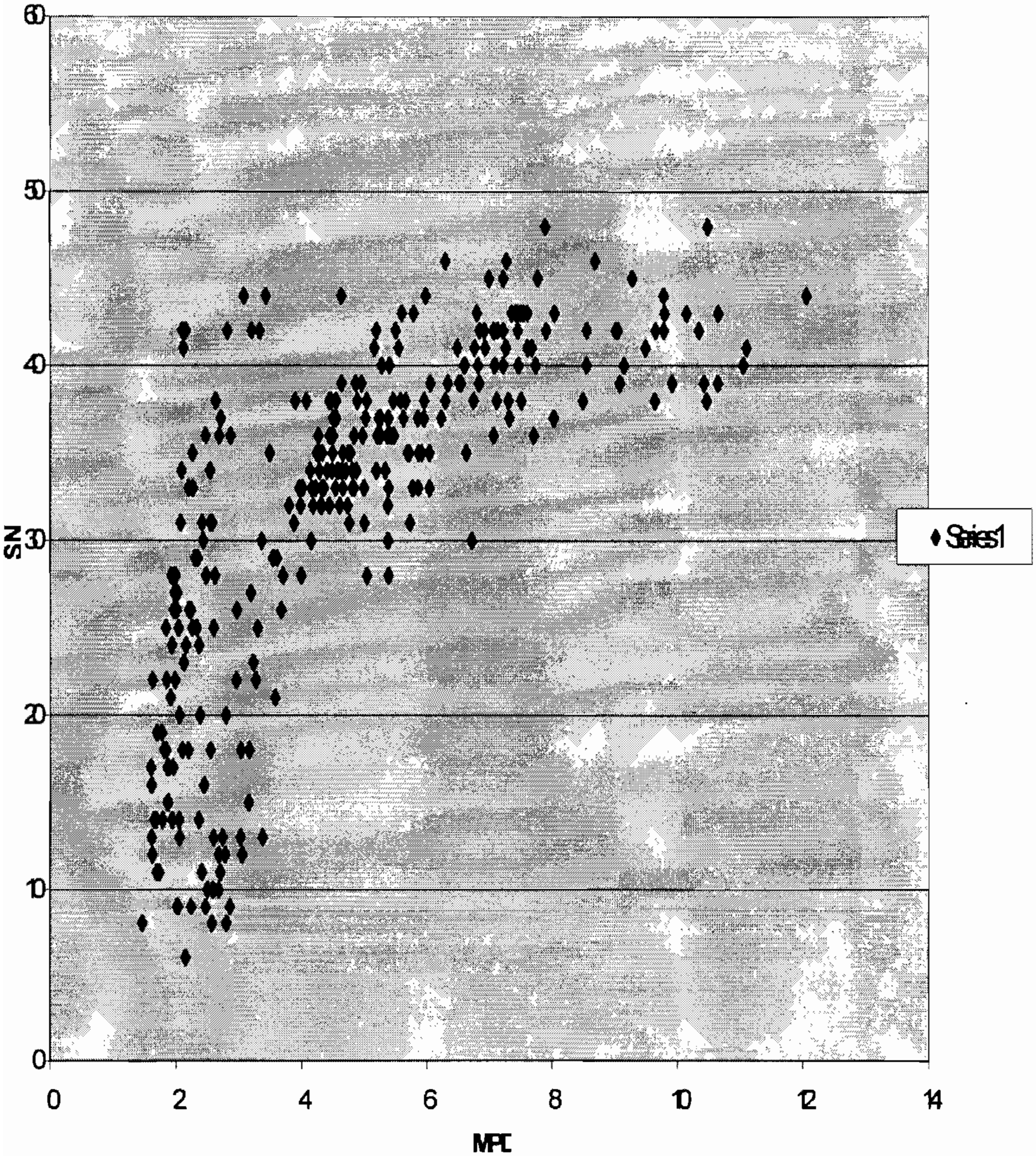


Figure 5.9 MPD vs SN w/String 5-200

CHAPTER 6

Summary and Further Studies

6.0 Introduction

This chapter provides a summary of the various topics discussed in the report, specifically methods for measuring surface profile for texture measurements. Two data acquisition systems are described for obtaining data from which pavement texture statistics can be computed. The first data acquisition system was designed primarily for microtexture measurements while the second system was for high-speed macrotexture measurements. Also discussed is current work in progress in search of a suitable model for predicting skid number (SN).

The main objective of the research effort was to develop a system that could measure texture at normal driving speeds. As was noted in Chapter 1, there are two levels of texture, microtexture and macrotexture. It was obvious from the outset that a high-speed microtexture system was not feasible because of the laser spot size, resolution, and update rate. However, it was plausible to develop a system for macrotexture for normal driving speeds that could be used for network level data collection and hopefully a system for microtexture at slow speeds.

6.1 Microtexture Measurements

Since it was not possible to measure microtexture at highway speeds, it was initially planned to use a system found at the start of the project that could be used in conjunction with other measurement systems such as the Falling Weight Deflectometer (FWD), that requires stop and go measurements. It was hoped that this system could be used for such measurements by placing it on an FWD or similar system and using it during FWD data collection activities. This would then provide project level measurements that could be used to further reinforce the high-speed macrotexture measurements. As noted in Chapter 1, it was generally believed by many researchers in this field that both micro and macrotexture measurements are required for accurate SN or frictional prediction. The microtexture would then be used to refine estimates of friction measurements obtained by the high-speed system where necessary. However, as indicated in Chapter 1, the company that researchers considered a strong candidate for the stop and go measurements could not deliver the system that was ordered during the first year of the project. Consequently, a much slower system was eventually ordered, but was found not to be practical for the original stop and go measurement. It does, however, provide a useful tool for micro and macromeasurements in a laboratory environment as described in Chapter 2.

6.2 Laser Operational Considerations for Texture Measurements

It was pointed out in Chapter 5 that the potential use of lasers for macro and micro texture measurements are directly affected by certain characteristics of the displacement measurement process. For example, the actual resolution and accuracy of a laser is often different from what is quoted by the manufacturer. The one quoted is the one expected by averaging a set of samples for a particular target position. The number for the 78KHz laser for one particular asphalt target sample was noted to require 66 samples. The number can vary depending on the sample type. Certain targets for example, asphalt will typically absorb more light than concrete, and thus will require more samples to meet the quoted resolution.

Other characteristics such as the sample rate, as well as the laser spot size, will limit the resolution in the travel direction. For example, sampling at one sample per 9.84 mils (0.25mm) with a laser beam of 39.4 mils (1mm) will not result in 9.84 mils (0.25mm) resolution along the direction of travel.

6.3 Implementation

Two types of measurement systems were developed and provided to TxDOT for implementation. The report includes necessary details where TxDOT can construct additional systems if needed. The microtexture system is limited to laboratory or low speed measurements; however, the macrotexture system can be used for typical highway speeds. Two high-speed macrotexture systems were developed and provided to TxDOT.

The data acquisition system software was redesigned during the last year of the project so that the high-speed macrotexture system could run under the new TxDOT VNET concept. For this concept, all communications with the data acquisition software is via an application or user client. Commands to begin sampling, stop sampling, save the texture data, or set required operational characteristics were defined. These commands, as well as responses, are sent from the Client to the Server via WinSock commands. Thus, the Client and Server can reside on the same or separate computers, communicating in the latter case via Ethernet cables. This software system is described in Chapter 4.

Chapter 5 discussed several statistics that can be used for classifying the texture measurements. Most systems use MPD, or mean profile depth, and ETD, estimated texture depth for measurement of texture. Relating the texture readings to frictional measurements such as skid number is more difficult. The MPD or ETD statistics were not individually found to be a good predictor of SN. However, a program for MPD and ETD was provided to TxDOT as these are the statistics primarily used by other systems for texture measurements. Other statistics discussed included the amplitude density function, periodogram or power spectral methods, and a texture profile variance using a string filter procedure. Various variables can be obtained from these general categories.

A model was described which could be used for predicting SN. It was not possible to fully determine the parameters or validity of the model until more data is obtained or much of the current data is corrected. As was noted in Chapter 5, three large sets of data were obtained but each had some particular error in the configuration setup.

A second method for model development was described for which some preliminary results have indicated much better prediction characteristics. This method, however, must undergo much more investigation, specifically with the new data before it will be ready for implementation.

6.4 Additional Studies

One of the problems, during the last year of the project, was not being able to obtain further skid data. The new skid and texture software system is now operational and it should be possible to easily acquire the raw texture data by continuing the process of mounting the texture laser on the skid truck and using the texture and skid measurement systems together. Large quantities of skid data can now easily be obtained. Additional studies are needed where the collected data can be used to refine the data collection process and specifically the statistics used to measure texture. Additional skid and texture data is needed so that appropriate adjustments to the current model can be made and it's validity determined. The PCLIM was designed to run on small embedded processors using DOS. The board should probably be redesigned to accommodate a larger buffer size for running with Windows. The capability of now being able to easily and quickly collect raw texture profile data could determine other uses. This report provides TxDOT the capabilities of building and maintaining many similar systems as the ones provided from the project.

REFERENCES

[1] Forster, S. W., "Pavement Microtexture and Its Relation to Skid Resistance", *Transportation Research Record,* No. 1215, Transportation Research Board, 1989, pp. 151-164.

[2] Henry, J. J., "Tire Wet-Pavement Traction Measurement: A State-of-the-Art Review", *The Tire Pavement Interface. STP 929.* M.G. Pottinger and T.J. Yager, Eds., American Society for Testing and Materials, Philadelphia, 1986, pp. 3-25.

[3] Henry, J. J., and Dahir, S. H., "Effects of Textures and the Aggregates That Produce Them on the Performance of Bituminous Surfaces", *Transportation Research Record,* No. 712, 1979, pp. 44-50.

[4] Kulakowski, B.T., "Mathematical Model of Skid Resistance as a Function of Speed", *Transportation Research Record,* No. 1311, 1991, pp. 26-32.

[5] Leu, M.C., and Henry, J.J., "Predictions of Skid Resistance as a Function of Speed from Pavement Texture", *Transportation Research Record,* No. 666, 1978, pp. 7-13.

[6] *1985 Annual Book of ASTM Standards,* Road and Paving Materials: Traveled Surface Characteristics, Vol. 04.03, American Society for Testing Materials, Philadelphia.

[7] Her, I., Henry, J.J., and Wambold, J.C., "Development of a Data Acquisition Method for Noncontact Pavement Macrotexture Measurement", *Transportation Research Record,* No. 1000, 1984, pp. 8-14.

[8] Selcom Technical Manual, Selective Electronic Co., (unpublished).

[9] Selcom Optocator Interface Module, Selective Electronic, Co., (unpublished).

[10] Payne, L.D., and Walker, R.S., "The Use of Lasers for Pavement Crack Detection", Research Report 1141-1, The University of Texas at Arlington, 1988.

[11] "The Use of the Selcom Laser for Pavement Texture and Skid Resistance Measurement, Technical Memorandum, Research Project #1290, Roger S. Walker, Ph.D., P.E. L. Donnell Payne, Ph.D. November 1994.

[12] "Second ISO/CD 13473-3, Characterization of Pavement Texture Utilising Surface Profiles – Part 3: Specifications and Classification of Profilometers", October 1997

[13] "Development of Portable Tester for Measuring Skid Resistance and It's Speed Dependency on Pavement Surfaces", Dr. Kazuo Saito et al.

[14] "Influence of Road Surface Texture on Traffic Characteristics Related to Environment, Economy and Safety", Ulf Sandberg, October 1998

[15] "Skid Resistance of Adjacent Tangent and Nontangent Sections of Roads", Bohdan T. Kulakowski and Wolfgang E. Meyer

[16] "Pavement Microtexture and Its Relation to Skid Resistance", Stephen W. Forster, *no date on paper*

[17] "Relation Between Tyre/Road Friction and Road Texture," Ulf Sandberg, Swedish Road and Traffic Research Institute, Linkoping, Sweden, 1990

[18] "Genetic Algorithms in Search, Optimization, and Machine Learning," Goldberg, Addison-Welsey.