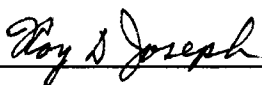


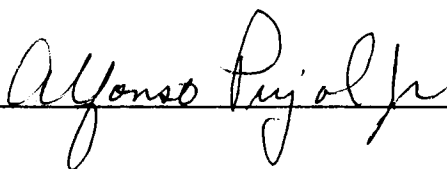
To the Graduate Council:

I am submitting herewith a thesis written by Andrew Eric Nelius entitled "A Digital Signal Processing Algorithm for Laser Particle Size and Velocity Measurement." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.


Bruce W. Bomar, Major Professor

We have read this dissertation
and recommend its acceptance:





Accepted for the Council:


Vice Provost
and Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Masters degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Inter-library Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Andrew E. Nelson

Date 18 APRIL 1990

A DIGITAL SIGNAL PROCESSING ALGORITHM FOR
LASER PARTICLE SIZE AND VELOCITY MEASUREMENT

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Andrew Eric Nelius

May 1990

ACKNOWLEDGMENTS

I want to express my gratitude to those who assisted me during the course of my research and preparation of this document. I gratefully acknowledge the guidance and suggestions of Dr. Bruce Bomar. I also wish to thank Chris Layne and Peter Sherrouse for sharing with me their expertise and support.

Most importantly, I wish to thank my parents for their encouragement and support throughout my entire graduate school career.

ABSTRACT

Size and velocity measurements of particles in a flow field are often required in industrial applications. The signal produced by a nonimaging laser optical measurement system contains the information to obtain particle size and velocity. However, the signal contains noise generated primarily by the photomultiplier tube in the measurement system.

A method for processing the digitized electrical signal to reduce the noise in the signal and to extract particle size and velocity is developed. The effectiveness of this processing algorithm is determined with simulated signals with pulses of known width and height. The signals correspond to three particle sizes ranging from the smallest to the largest size of interest. For each size, 100 simulated waveforms are generated using a Monte Carlo simulation of the photodetector.

With simple averaging, the processed waveform contains too much noise to reliably yield pulse data without bias or errors. Therefore, a ninth-order Finite Impulse Response filter with specified frequency characteristics is implemented to smooth the raw data. This gives good pulse height measurements and unbiased but imprecise pulse width measurements. Individual baseline calculations and linear interpolation of the pulse endpoints are added to the algorithm to improve the precision of the pulse width measurements. A parabolic interpolation of the pulse minimum did not improve the pulse height precision or accuracy appreciably.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION	1
II. SIMULATION OF SIGNALS	8
Modeling of the Ribbon Beam Pulse	9
Modeling of the Cone Beam Pulse	14
Modeling of Noise	14
III. ALGORITHM DEVELOPMENT	23
Algorithm with Neighborhood Average	28
Algorithm with FIR Filter	36
Algorithm with FIR Filter and Individual Baseline Calculation	42
Algorithm with FIR Filters, Individual Baseline Calculations, and Endpoint Interpolations	43
Algorithm - Final Version	47
IV. PROPOSED HARDWARE IMPLEMENTATION	58
V. SUMMARY	61
BIBLIOGRAPHY	63
VITA	66

LIST OF FIGURES

FIGURE	PAGE
I-1. Block Diagram of Multibeam System	3
I-2. Multibeam System Beam Positions in Test Section	4
I-3. Multibeam Signals as a Function of Particle Position and Size	6
II-1. Area of a Beam Blocked by an Obstruction	10
II-2. Mathematical Representations of Particle and Ribbon Beam	11
II-3. Intensity of Ribbon Beam as Particle Crosses	13
II-4. Block Diagram of Multibeam System Electronics	15
II-5. Simulated Ribbon Beam Signal for Smallest Particle	19
II-6. Simulated Ribbon Beam Signal for Mid-Sized Particle	19
II-7. Simulated Ribbon Beam Signal for Largest Particle	20
II-8. Simulated Cone Beam Signal for Smallest Particle	20
II-9. Simulated Cone Beam Signal for Mid-Sized Particle	21
II-10. Simulated Cone Beam Signal for Largest Particle	21
II-11. Actual Ribbon Beam Signal for a Mid-Sized Particle	22
II-12. Actual Ribbon Beam Signal for a Large Particle	22
III-1. Baseline Value, Trigger Level, and Threshold Level	27
III-2. Simulated Multibeam Signal Smoothed with 5-Point Neighborhood Average	29
III-3. Simulated Multibeam Signal Smoothed with 9-Point Neighborhood Average	29

FIGURE	PAGE
III-4. Simulated Multibeam Signal Smoothed with 21-Point Neighborhood Average	30
III-5. Frequency Response of 5-Point Neighborhood Average	30
III-6. Frequency Response of 9-Point Neighborhood Average	31
III-7. Frequency Response of 21-Point Neighborhood Average ...	31
III-8. Frequency Content of Narrowest Ideal Pulse	32
III-9. Frequency Content of Noisy Waveform	33
III-10. Signal for Smallest Particle Smoothed with 9-Point Neighborhood Average	33
III-11. Frequency Response of 9th Order FIR Filter	37
III-12. Simulated Multibeam Signal Filtered with 9th-Order FIR	38
III-13. Frequency Content of FIR-Smoothed Waveform	38
III-14. Illustration of Endpoint Interpolation Parameters	46
III-15. Illustration of Minimum-Point Interpolation Parameters	48
III-16. Flowchart of Processing Algorithm for Ribbon Beam Waveforms	49
III-17. Flowchart of Processing Algorithm for Cone Beam Waveforms	55

LIST OF TABLES

TABLE	PAGE
III-1. True Values of Pulse Parameters	24
III-2. Results of Monte Carlo Simulation Using Neighborhood Average in Algorithm	35
III-3. Coefficients of FIR Filter	37
III-4. Results of Monte Carlo Simulation Using FIR in Algorithm	40
III-5. Results of Monte Carlo Simulation Using FIR Filter and Individually Computed Baselines in Algorithm	44
III-6. Results of Monte Carlo Simulation for Cone Beam	57
IV-1. Various Functions of Algorithm and Their Required Executive Cycles	60

CHAPTER I

INTRODUCTION

The need for size and velocity measurements of small particles has become very common. Previously important in primarily research-oriented projects, such measurements are now needed in commercial industrial efforts to obtain a higher quality product. Such commercial and industrial requirements include fuel spray combustion process optimization, agricultural spray analysis, nuclear reactor safety, cloud simulation for aircraft icing studies, and rocket plume analysis [1].¹ In response to this increasing need, the number of methods proposed for measuring the particle size and velocity has grown significantly. Most methods employ some type of optical technique to obtain the desired data.

Of the optical techniques used in particle analysis, a choice of either imaging or nonimaging methods is available [2]. Imaging methods use some type of illumination device to radiate a test section containing the particles under analysis, and some type of imaging sensor and data reduction system. This imaging system could be as simple as a collimated strobe light and synchronized Polaroid camera with the data reduction performed by a human, or it could be a pulsed laser imaging the test section onto a video camera, from which the video signal is analyzed by a digital image processor to yield size information [3]. Velocity data can be obtained from an imaging system by

¹Numbers in brackets refer to similarly numbered references in the Bibliography.

double-exposing the image receptor and measuring the distance between the twin images. A distinct advantage of imaging versus nonimaging techniques is the ability to detect nonspherical particles [2]: if the profile of the particle is desired, it must be imaged onto some type of permanent medium. However, if spherically-shaped particles are assumed, a less complicated non-imaging system will suffice.

Nonimaging particle sizing instruments generally consist of an illumination source focused to a very small probe volume relative to the mean distance between particles. This small probe volume permits the analysis of a single particle within the volume; at points along the beam other than the focal point, particles have negligible effect on beam attenuation. Beyond the focal point, the beam expands and falls onto some type of photodetector to yield an electronic signal varying in amplitude as particles interrupt the beam. In another approach, the receptor may be angularly removed from the beam's line of sight in order to get scatter signals from beam/particle interactions, as in Laser-Doppler velocimetry. In either case, a statistically significant sample of single particle measurements is obtained to estimate particle population characteristics.

One particular type of nonimaging particle size and velocity measuring instrument is the multiple laser beam (or simply multibeam) system which has been developed and utilized to collect data [4]. In the multibeam system, two parallel laser beams oriented in a direction perpendicular to a particle's travel are detected separately on two photomultiplier tubes as shown in Figure I-1. The first beam encountered by the particle, hereafter called the cone beam, acts as a triggering element to start a time-of-flight timer. The cone

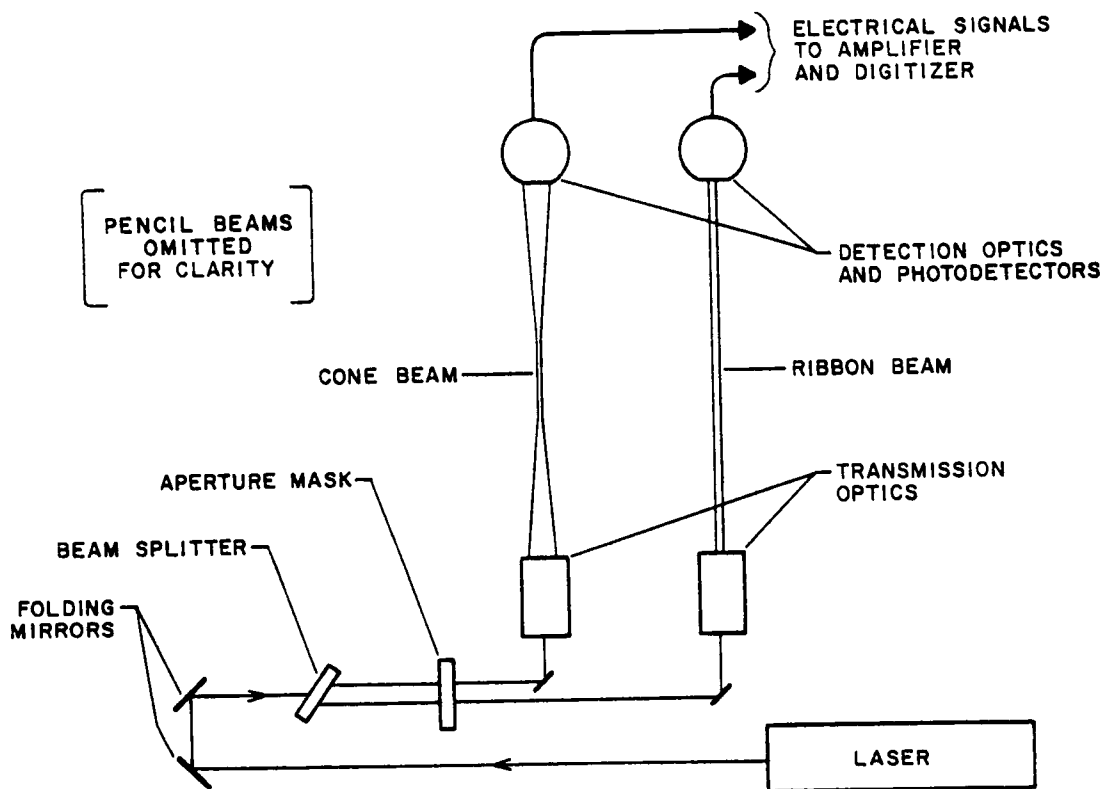


Figure I-1. Block Diagram of Multibeam System

beam is shaped like two concentric cones with points connected by a short right cylinder and bases located at the transmission and detection optics. This geometry allows triggering to occur only when a particle interrupts the probe volume, which is the cylindrical volume connecting the cone points. The second beam encountered by the particle, called the ribbon beam, acts as both a triggering element to terminate the time-of-flight measurement and a size measurement indicator. Size is determined from the reduction in intensity caused by the particle's blockage of a portion of the beam. The size so determined is the one in a direction (z) orthogonal to both the ribbon beam and

the particle's direction of travel (x). The ribbon beam's rectangular profile allows this type of measurement to be performed since its major axis is longer than any expected particle's diameter (see Figure I-2). In addition to velocity and the aforementioned particle dimension, the particle size dimension in the direction of the particle's travel (x) can be estimated by using the calculated velocity information and the time required for the particle to traverse the ribbon beam's width [4].

Two laser pencil beams located just in front of and above and below the cone beam are also present as shown in Figure I.2. A signal is sent to the data acquisition system to reject the data for a certain particle if the particle interrupts either of these pencil beams. This process prevents data errors that

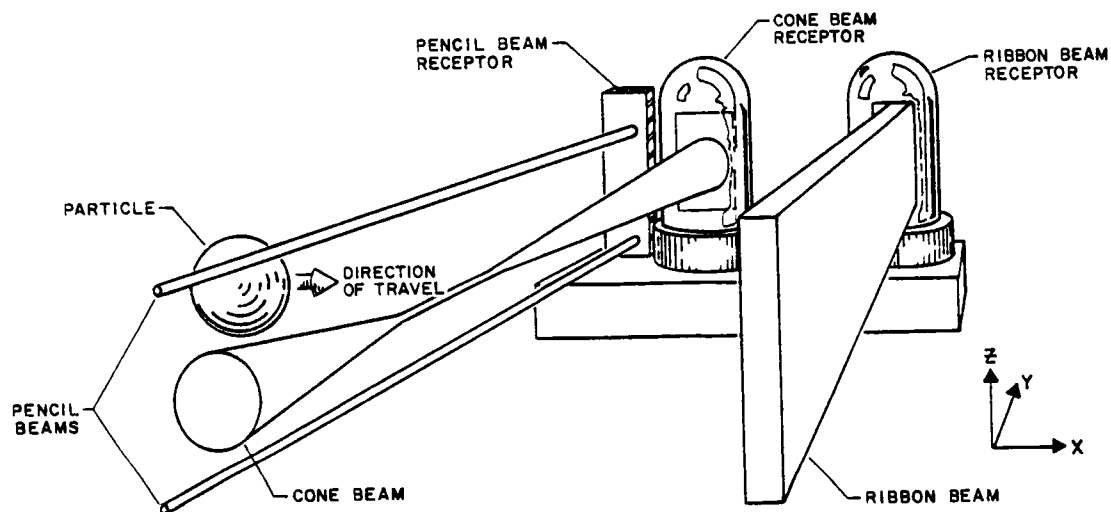


Figure I-2. Multibeam System Beam Positions in Test Section

would result from particles that are larger than the ribbon beam or particles that only partially intercept the ribbon beam.

Figure I-3 illustrates resultant photomultiplier tube signals upon particle/beam interaction. The calculations required to obtain size and velocity data from these signals will now be examined. If the start and finish points of the cone beam signal pulses are denoted by S_C and F_C , respectively, and the start and finish points of the ribbon beam signal pulses are similarly denoted by S_R and F_R , and the height of the ribbon beam pulse is designated by H_R , then the equations for the particle velocity and size dimensions are [4]

$$V_P = \frac{x_R - x_C}{S_R - S_C} \quad (1)$$

$$Z_P = Z_R \frac{H_R}{H_{R(max)}} \quad (2)$$

$$X_P \approx (F_R - S_R)V_P - W_R \quad (3)$$

where

x_R = Ribbon beam spatial coordinate

x_C = Cone beam spatial coordinate

Z_R = Ribbon beam height

W_R = Ribbon beam width

$H_{R(max)}$ = Photomultiplier signal with no particle present

V_P = Particle velocity

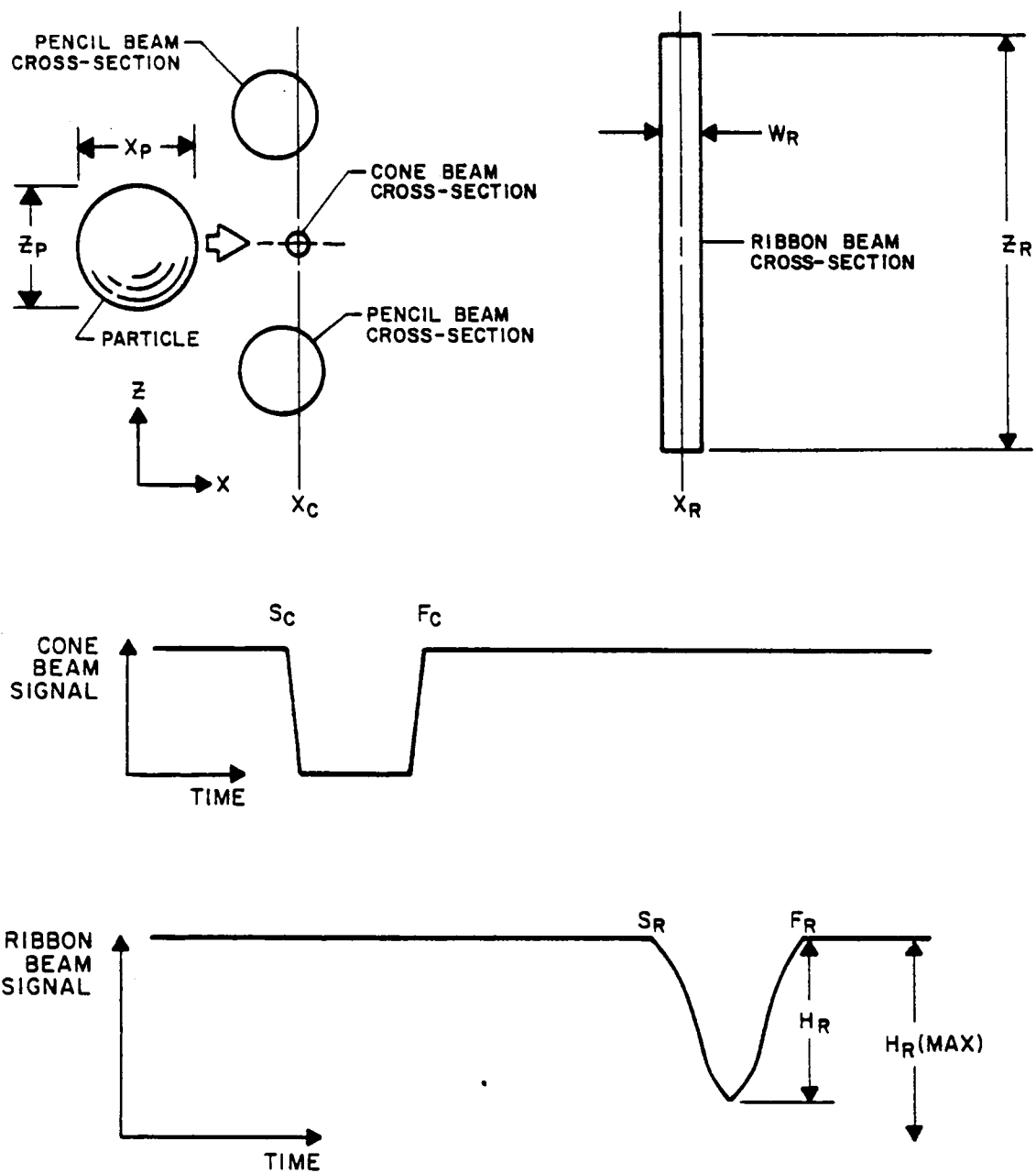


Figure I-3. Multibeam Signals as a Function of Particle Position and Size

X_P = Particle horizontal size
 Z_P = Particle vertical size

Clearly the parameters required to acquire the velocity and size measurements are S_C , S_R , F_R , and H_R . Because noise is generated by various sources in the multibeam system, the multibeam signal must be processed to reliably give the parameter values. A method for extracting S_C , S_R , F_R , and H_R from noisy multibeam signals rapidly is the subject of this thesis.

The performance of the algorithm which extracts the pulse parameters will be evaluated by processing 100 simulated waveforms containing a pulse with known width and height and realistic levels of noise. These simulated waveforms will be generated as described in Chapter III. Three such sets of 100 256-point waveforms resulting from each laser beam will be processed for a total of 600 waveforms. The first set of waveforms will contain small pulses, the second set will contain mid-sized pulses, and the third set will contain large pulses. This will allow evaluation of the algorithm's performance over the entire range of particle sizes upon which the multibeam system is expected to operate.

CHAPTER II

SIMULATION OF SIGNALS

In order to evaluate the effectiveness of the algorithm to be developed for determining the pulse parameters (S_R , F_R , H_R , and S_C), a standard waveform with known parameters must be used. Previous methods of calibration have included launching measured glass beads through the multibeam test section, which gives a rough idea of the system's performance. Unfortunately, this method often gives poor results since the beads fragment and ablate pieces of the propulsion nozzle giving particles of other sizes. Another method is to use a calibration reticle with various sizes of opaque circular disks photoetched onto a motor-driven glass plate [3]. Inserting this plate into the test section would overcome the problem of particle size uncertainty. However, signal durations would be much different than those encountered with actual particles and errors originating from uncertainty in the optical parameters (i.e., beam widths, etc.) would still exist. Differences in signal duration are especially serious since this will alter the way in which noise effects the measurements. What is needed is a test to evaluate the algorithm only, exclusive of the associated hardware that produces the signal.

Algorithm evaluation will be done by accurately modeling the photo-multiplier signals resulting from the interaction between a simulated calibration particle and simulated multibeam system. This chapter explains how these simulation data are produced. The simulation data have known pulse widths and heights and noise that simulates the noise produced within the multibeam system. This simulation process will produce a realistic waveform

with known pulse parameters that can be analyzed by the processing algorithm.

Finally, a statistical analysis of the extracted pulse parameters will be performed using a large number of simulated waveforms, all with statistically independent noise to simulate many particles of the same size and velocity recorded at different times. This data will be compared to the known pulse parameter data and the data from other algorithms to give a figure of accuracy and precision of the processing algorithms.

Modeling of the Ribbon Beam Pulse

To numerically simulate a photomultiplier's electronic output when the intensity of the radiation on the photocathode is temporarily decreased due to a particle's transit through the beam, the actual cause of the signal must be known. Before embarking on this effort, the following assumptions are made to simplify ensuing calculations:

- 1) The intensity of the beam is constant in the three spatial directions x , y , and z within the probe volume [4],
- 2) The intensity outside of the beams is zero,
- 3) The particle is spherical.

The particle will go through the ribbon beam completely in the z -direction if proper operation of the pencil beam rejection circuit is assumed.

Since the output from the photomultiplier is proportional to the intensity of the incident radiation in units of power/area times the area illuminated, a change in the output will occur if either the area illuminated or the power changes. Since the power of the ribbon beam remains constant, the area illuminated is decreased as shown in Figure II-1. Therefore, the signal output

is reduced by an amount proportional to the area shadowed by the particle. Although the simple shadowing model is not exact, it correctly predicts the reduction in photodetector output due to the presence of a particle. Since the particles to be measured are large relative to the light wavelength, a more accurate analysis could be performed by using Fraunhofer diffraction [6], [7], [8]. While this analysis would show a different distribution of intensity on the photocathode (i.e., an Airy disk at the center of the shadow), the total beam power blocked would still be the power absorbed by the particle.

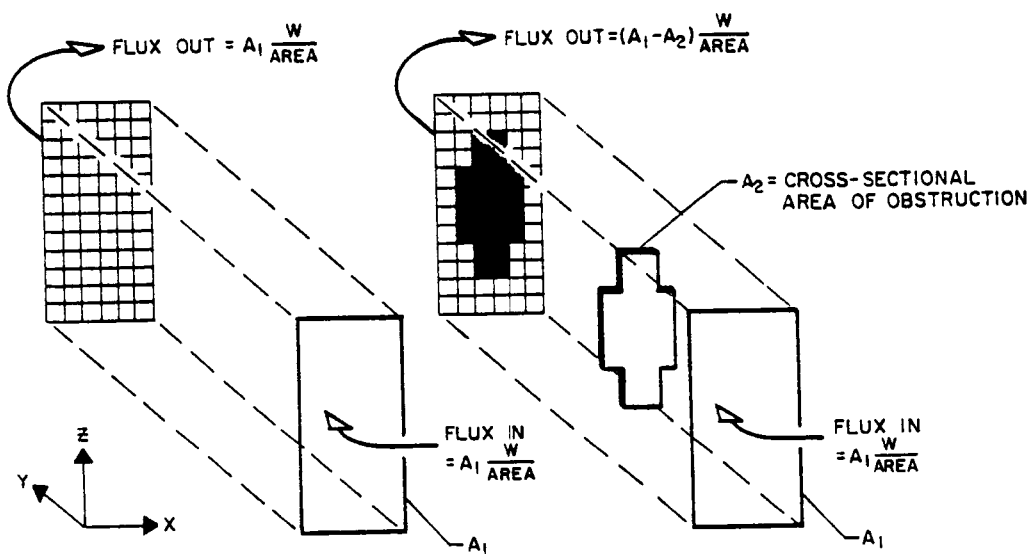


Figure II-1. Area of a Beam Blocked by an Obstruction

Only particles with diameters greater than the beam width will be considered below. Particles smaller than the beam width will not be considered since their signal level is too low to overcome the noise levels in the photomultiplier.

Figure II-2 illustrates the interaction of the particle and the ribbon beam. The rectangular pulse represents the intensity distribution of the ribbon beam in space; its height must be greater than the radius of the semi-circle, so the particle goes completely through the ribbon beam in the z-direction. The semi-circle represents one-half the cross-sectional area of the particle that blocks the beam.

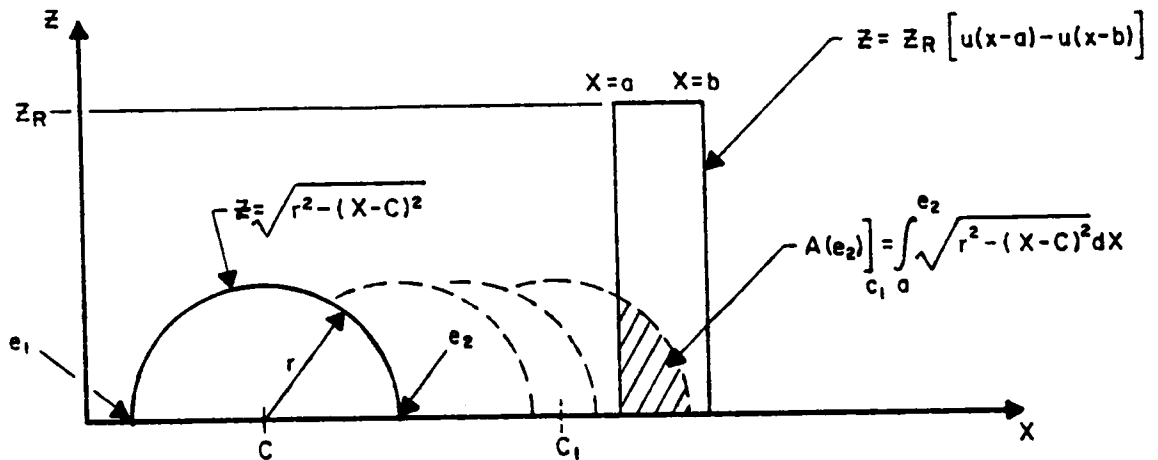


Figure II-2. Mathematical Representations of Particle and Ribbon Beam

As the leading edge e_2 of the particle with radius r , trailing edge e_1 , and center c travels along the x -axis into the beam, the area $A(e_2)$, common to both shapes is given by

$$A(e_2) = \int_a^{e_2} \sqrt{r^2 - [x - (e_2 - r)]^2} dx \text{ for } a \leq e_2 \leq b. \quad (4)$$

By substituting the dummy variable u for $x - (e_2 - r)$ and integrating (4), the expression for $A(e_2)$ becomes

$$A(e_2) = 1/2 \left[u \sqrt{r^2 - u^2} + r^2 \sin^{-1} \left(\frac{u}{r} \right) \right]_{u=a+r-e_2}^{u=r}. \quad (5)$$

By incrementally varying e_2 from a to b , one can calculate the area of the beam that is blocked by the particle over the interval $a \leq e_2 \leq b$.

When e_2 reaches a value greater than b which is when the front of the particle exits the beam, an adjustment in the evaluation limits is required. The new expression is

$$A(e_2) = 1/2 \left[u \sqrt{r^2 - u^2} + r^2 \sin^{-1} \left(\frac{u}{r} \right) \right]_{u=a+r-e_2}^{u=b+r-e_2} \quad (6)$$

which is valid over the interval $b \leq e_2 \leq a + 2r$.

The last case, when the particle's trailing edge e_1 (which is equal to $e_2 - 2r$) enters the beam once again warrants a change in the limits to yield

$$A(e_2) = 1/2 \left[u \sqrt{r^2 - u^2} + r^2 \sin^{-1} \left(\frac{u}{r} \right) \right]_{u=-r}^{u=b+r-e_2}. \quad (7)$$

This equation is used over the interval $a + 2r \leq e_2 \leq b + 2r$, that is, until the particle exits the beam.

Since the total area of the ribbon beam is $Z_R(b-a)$, the area, $Q(e_2)$, left after the particle blocks a portion of the beam is

$$Q(e_2) = Z_R(b-a) - 2A(e_2).$$

The photodetector signal is proportional to $Q(e_2)$ where the constant of proportionality depends on ribbon beam power and detector responsivity. The shape of the signal is shown in Figure II-3.

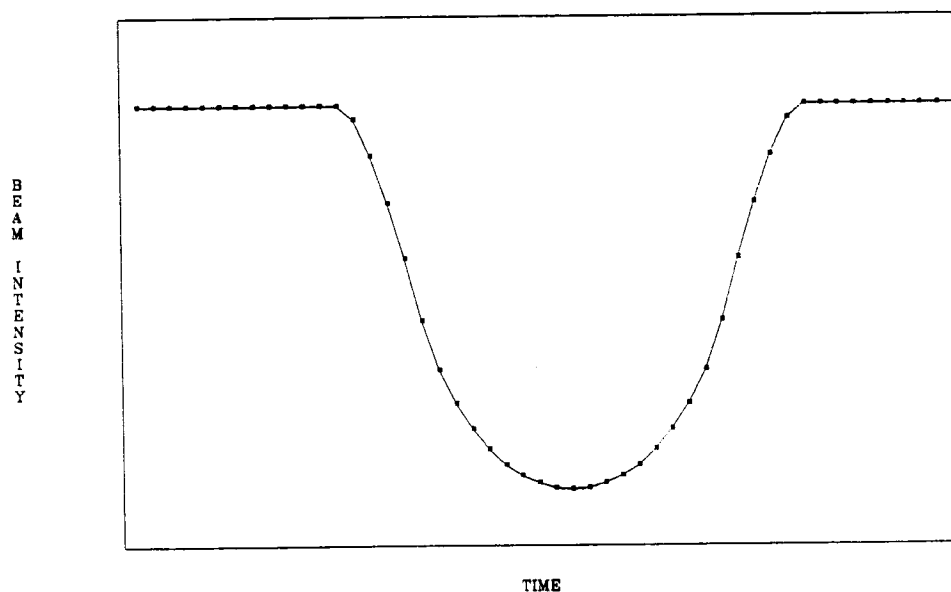


Figure II-3. Intensity of Ribbon Beam as Particle Crosses

Modeling of the Cone Beam Pulse

The shape of the cone beam pulse has a similar characteristic to the ribbon beam pulse. However at the minimum, a flattening occurs because the beam is completely blocked during the time required for the particle to transit the beam.

The derivation of the exact profile of the cone beam pulse could be mathematically found as before with the ribbon beam pulse. However, the fall and rise times of the cone beam pulse are much shorter than for the ribbon beam pulse because of the rapid cutoff of the entire cone beam. Typical rise and fall times are one to two samples long for particles of nominal size and velocity analyzed by the multibeam system at typical sampling rates. Therefore, with such a steep slope on the leading and trailing edges of the cone beam pulse, the cone beam pulse can be modeled as a step function scaled to give realistic signal levels. The duration of the step function will be calculated from the time the cone beam is completely blocked.

Modeling of Noise

Now that the ideal signals from the particle's interaction with the beams have been developed, the non-ideal behavior of real-world electronics must be modeled to produce realistic signal simulation. This is necessary to evaluate the general application of the pulse-parameter location algorithms under actual measurement conditions.

Figure II-4 shows a block diagram of the electronics required to obtain intensity readings of a light source. It will be assumed that all wiring is

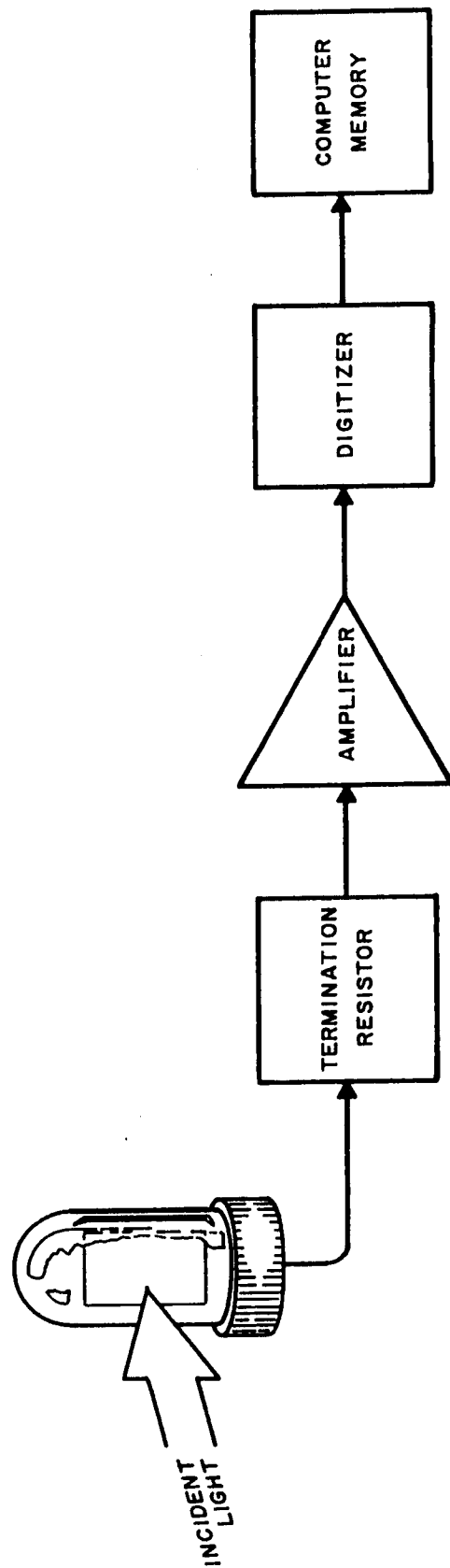


Figure II-4. Block Diagram of Multibeam System Electronics

properly grounded and shielded. This leaves the photomultiplier and the amplifier as possible sources of noise.

A common method for obtaining voltage signals from a photomultiplier is to drive a load resistor with the output current from the photomultiplier [9]. This gives a current-to-voltage transformation from the current signal of the photomultiplier, thus allowing a voltage measurement corresponding to light intensity to be made at the load resistor. The resistor value may be selected to be arbitrarily large to produce a high-level output voltage. However, high termination resistance causes the photomultiplier's frequency response to suffer due to the stray capacitance on the output circuit. But even for smaller load resistors, which do not affect the bandwidth appreciably, the output voltage can be made large enough to overcome the noise generated by a good-quality instrumentation amplifier, such as the Analog Devices Model AD524 which has peak-to-peak noise voltage in the microvolt range [10]. Amplifying the photomultiplier signal with a device which produces noise that is orders of magnitude less than the noise in the signal makes the amplifier noise virtually unnoticeable in the amplified signal. Therefore, amplifier noise will not be considered here. Also, lower values of the load resistor give very low resistor thermal noise so this noise.

The dominant noise generated in the circuit, then, is the noise from the photomultiplier tube itself. It has been demonstrated [5], [11] that the shot noise generated within a photomultiplier is a Poisson random process. This process can be simulated by loading a number representing the power of the light falling onto the photocathode of the photomultiplier tube into a random number generator that has a Poisson distribution. This simulates the number

of photoelectrons emitted from the photocathode during any time interval. For testing the algorithm, a typical value of beam power entering the photomultiplier and a typical digitizer sampling rate were used in a photomultiplier simulation computer program. This program performs a Monte Carlo simulation of a photomultiplier tube using Poisson statistics and random number generators. Simulated ideal signals resulting from three different sized particles spanning the range of sizes the multibeam system is expected to handle were provided as inputs to the program. The pulse widths and heights were scaled to yield a correct number of samples based on the programmed sampling rate and the amount of the beam's cross-section each sized particle would block. The three sizes were calculated to block a maximum of 88.7%, 44.1%, and 21.0% of the ribbon beam. Cone beam blockage was assumed to be 100% since all particles have greater diameters than the cone beam's, and therefore only pulse widths were changed for the different particles. The background radiation for both the ribbon beam and the cone beam signals was programmed as 15% of the unblocked beam intensity. This number was determined empirically from actual multibeam data. The ribbon beam pulse widths were normalized to a maximum width of 50.000 samples for the largest particle size based on a typical velocity and sample rate. This value of the pulse width for the largest particle yielded pulse widths of 27.437 and 16.155 samples for the other two particle sizes. Since the digital sampling algorithm can only yield an integral number of samples for the pulse length, the ends of the simulated pulses fall between samples.

With these guidelines for the production of simulated pulses, 100 pairs of waveforms for each of the three particle sizes were generated. Each of the 600,

256-sample waveforms was statistically independent of the others in order to maintain the validity of the simulation. Figures II-5 through II-10 show plots of one waveform of each beam signal. Figures II-11 and II-12 show actual multibeam waveforms of glass calibration beads for comparison. These actual waveforms contain a lower level of noise than the simulated waveforms which suggests that the noise level loaded into the simulation program was too pessimistic. However, this also suggests that the digital signal processing algorithm to be developed in Chapter III will produce measurements on actual data at least as good as the measurements obtained from the simulation data.

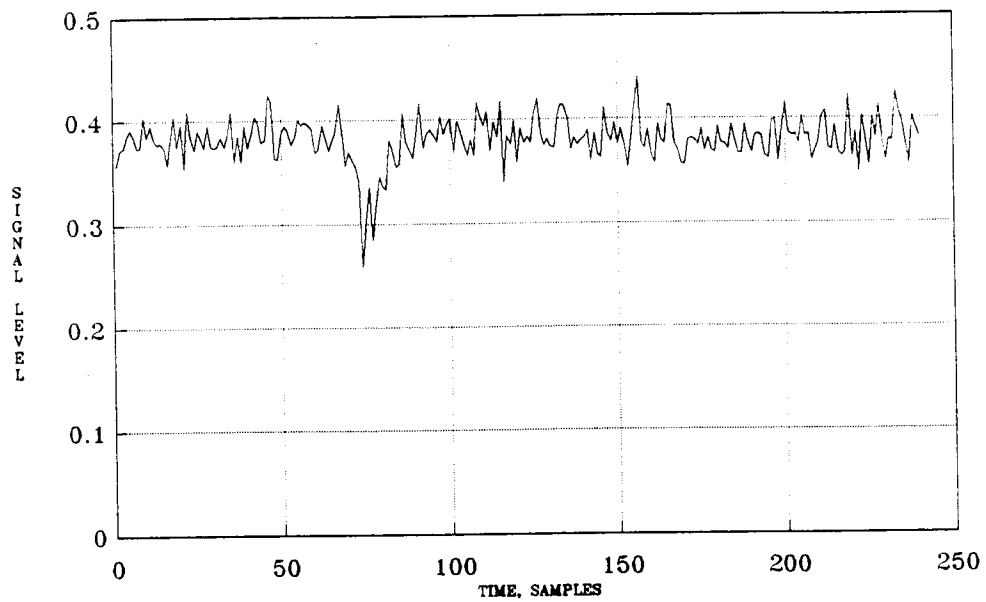


Figure II-5. Simulated Ribbon Beam Signal for Smallest Particle

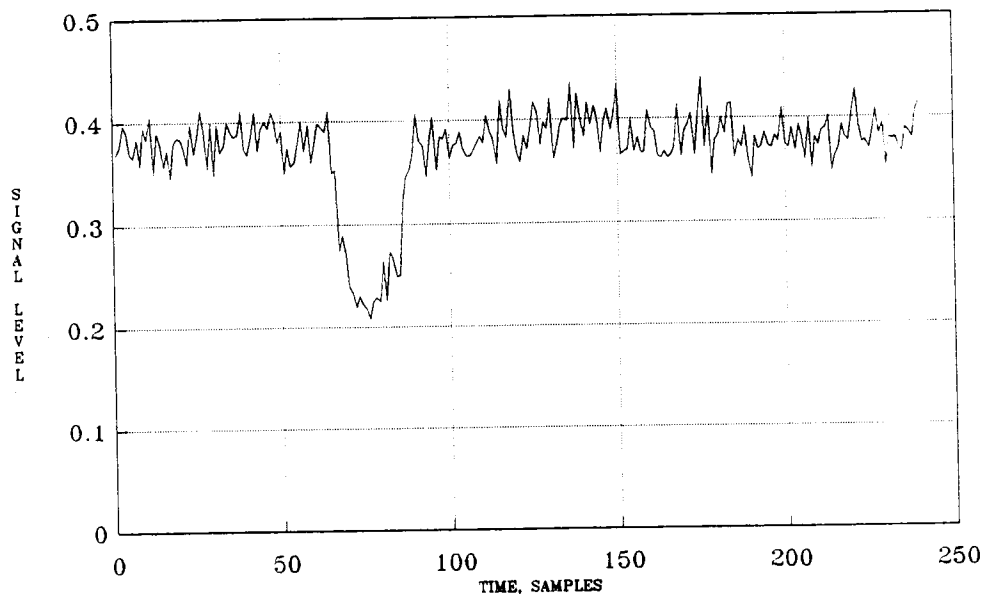


Figure II-6. Simulated Ribbon Beam Signal for Mid-Sized Particle

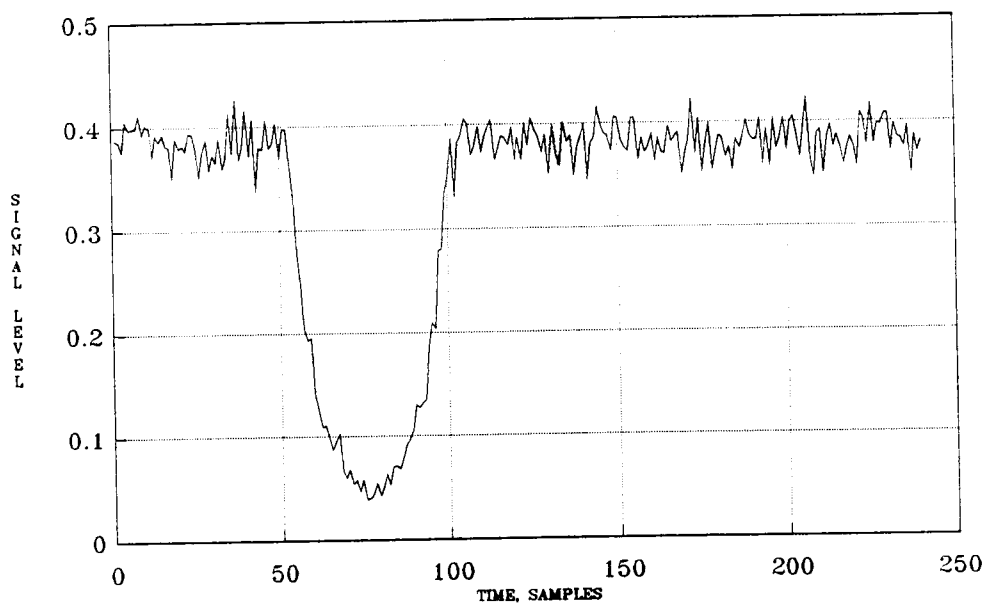


Figure II-7. Simulated Ribbon Beam Signal for Largest Particle

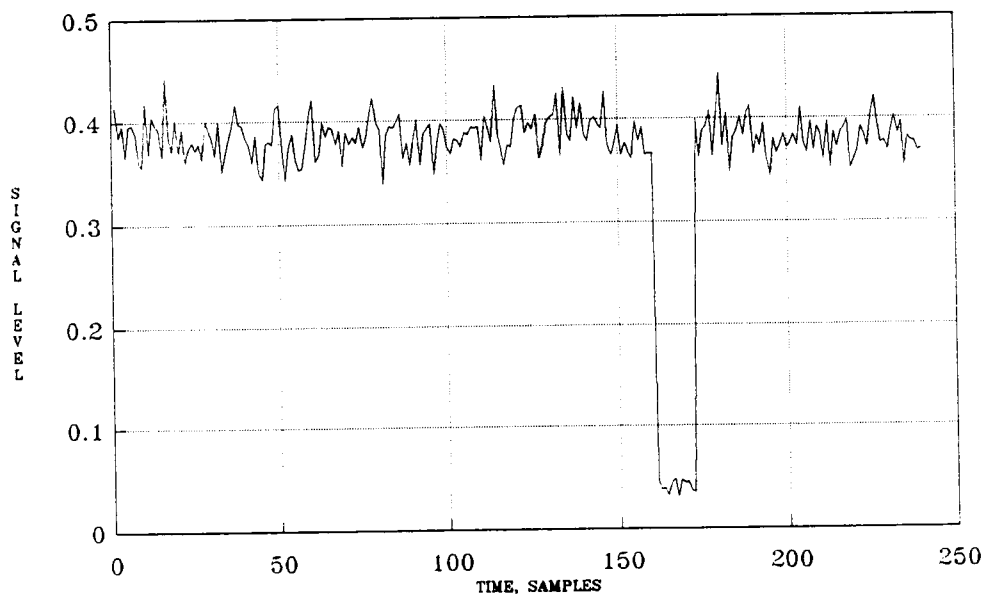


Figure II-8. Simulated Cone Beam Signal for Smallest Particle

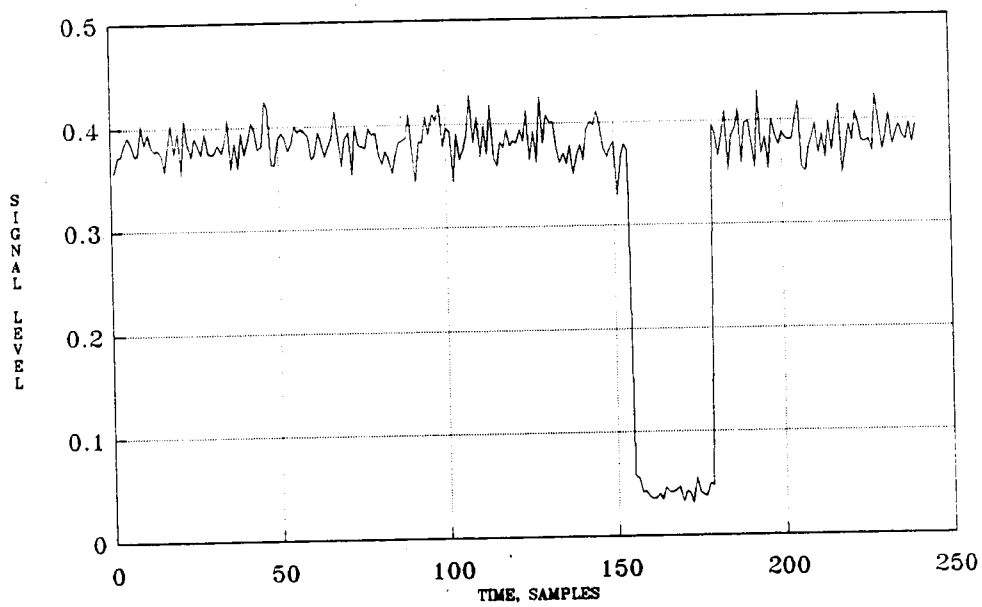


Figure II-9. Simulated Cone Beam Signal for Mid-Sized Particle

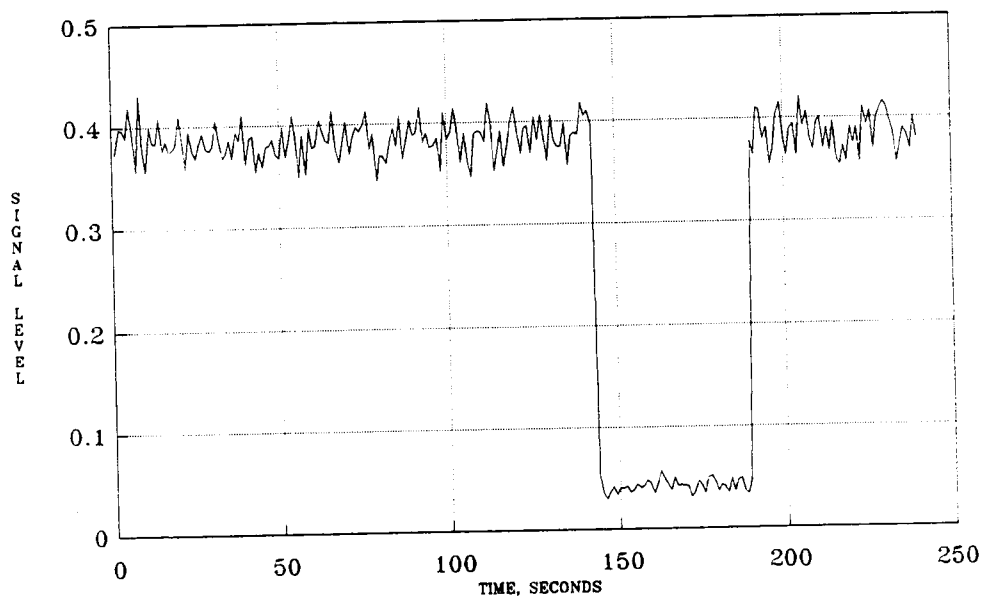


Figure II-10. Simulated Cone Beam Signal for Largest Particle

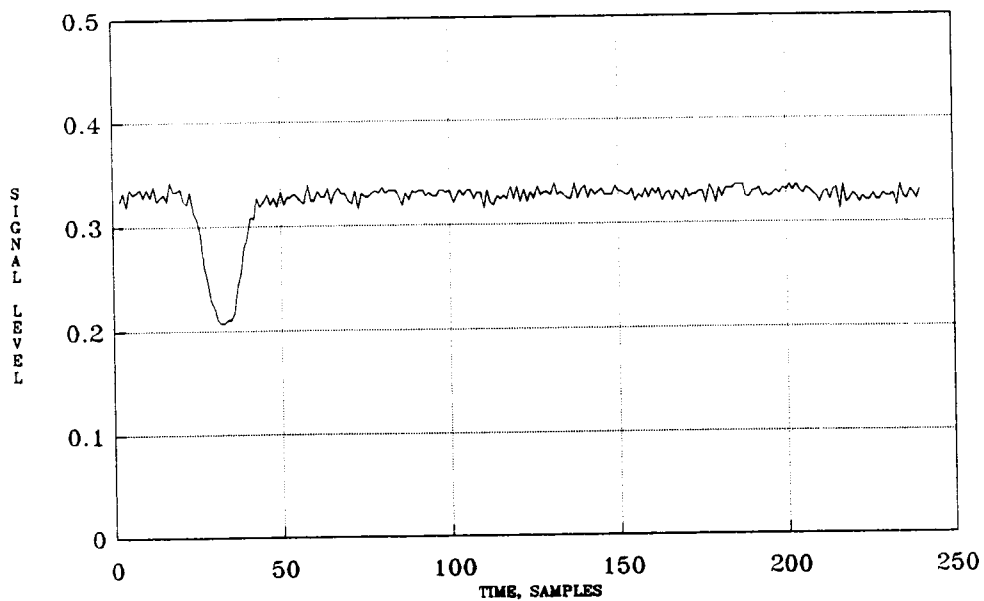


Figure II-11. Actual Ribbon Beam Signal for a Mid-Sized Particle

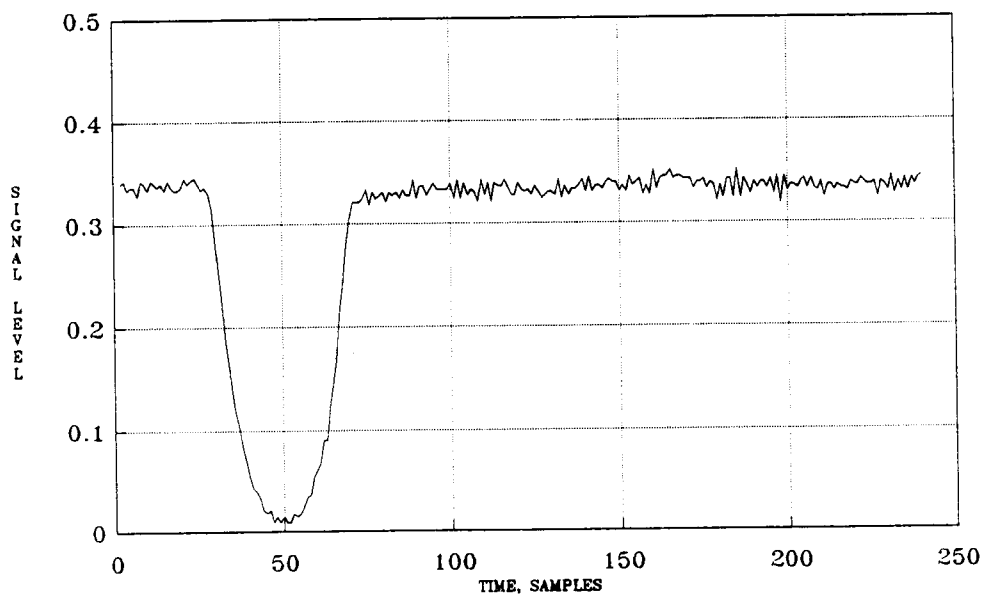


Figure II-12. Actual Ribbon Beam Signal for a Large Particle

CHAPTER III

ALGORITHM DEVELOPMENT

An algorithm for determining the pulse parameters S_C , S_R , F_R , and H_R will now be developed. To evaluate the accuracy of the algorithm, the data from a source of known parameters must be available. To achieve this, pulse width values are loaded into a pulse-generation computer program. The ribbon beam pulse heights, however, are determined indirectly because a definite value of this quantity doesn't appear in the pulse-generation program. The baseline values (waveform points exclusive of those contained in the pulses) of the 600 waveforms are averaged to obtain a mean baseline value. Then the percentage of beam blockage for the three different ribbon beam pulses is taken to yield the expected minimum value of each pulse. This method works well since the mean baseline value for any given sized pulse exhibits a high amount of agreement with the other five mean baseline values (99.93% agreement).

The true values of the important pulse parameters used by the computer simulation are shown in Table III-1. The pulse heights of the cone beam data are not shown since they are not required for particle size or velocity calculations.

The time-of-flight measurement, which yields the velocity of a particle, is obtained from the ribbon beam pulse and cone beam pulse starting points, S_R and S_C , respectively. This velocity data shall be assumed to have the same accuracy as the pulse width data since both the pulse width and the pulse velocity measurements rely on accurately determining these values.

Table III-1. True Values of Pulse Parameters

Parameter	True Value (samples)
R1 Width	16.1553
R1 Height	0.30531
R2 Width	27.4368
R2 Height	0.21642
R3 Width	50.000
R3 Height	0.0437
C1 Width	12
C2 Width	24
C3 Width	46

- R1 = smallest particle ribbon beam pulse
 R2 = mid-sized particle ribbon beam pulse
 R3 = largest particle ribbon beam pulse
 C1 = smallest particle cone beam pulse
 C2 = mid-sized particle core beam pulse
 C3 = largest particle cone beam pulse

Therefore, statistical analysis shall be restricted to the measurements of the ribbon beam pulse width, the cone beam pulse width, and the ribbon beam pulse height.

Before determining the parameters S_C , S_R , F_R , and H_R , it is necessary to obtain a value of the photomultiplier tube output when no particle interrupts the beam. This value, called the baseline value, is the reference point from which all other levels are defined. The baseline value used is the mean value calculated for the pulse height determination. This was done by averaging the values of the samples in all 600 waveforms exclusive of the samples in the pulses. In an actual working system, such a baseline estimate would be obtained by acquiring data in the absence of particles before initiating any particle measurements.

A second level called the trigger level is used to verify the presence of a pulse in a waveform and is determined as a percentage of the baseline value. All points in a waveform are compared to the trigger level to determine if a pulse is present in the waveform. If a number n of consecutive points in the waveform are found to be less than the trigger level, a pulse is declared to be present. The number n depends on the shortest expected pulse in the signal, which will be produced by the smallest and/or fastest particles analyzed.

Once a pulse has been determined to be present in the waveform, comparison of the remainder of the points in the waveform with the trigger level continues as a check for multiple pulses in a single waveform. If a second pulse is located in the waveform, an error is declared and the waveform is discarded. This is done to prevent the algorithm from having to match a certain cone beam pulse to a certain ribbon beam pulse in waveforms

containing multiple pulses. The large number of particles not entering the probe volume and the large number of waveforms available for analysis justifies this removal of waveforms containing multiple pulses. However, if only one group of at least n consecutive points is found to be less than the trigger level, the endpoints and minimum point of the pulse can be determined.

The endpoints of the pulse are found by comparing points in the waveform to a third level called the threshold level. The threshold level is set as a percentage of the baseline value. The start point of the pulse is the first point in the waveform with a value less than the threshold level. The finish point is the last point in the pulse with a value less than the threshold level.

Instead of comparing every sample in the waveform to the threshold level, the algorithm makes use of the pulse-determination information that was gathered earlier with the trigger level comparison. The first sample of the group containing n consecutive samples with values less than the trigger level is used as an approximate start point of the pulse. This way, only samples preceding this approximate start point need be compared to the threshold level.

Following the location of the start point, the algorithm locates the finish point of the pulse by comparing values of samples following the start point to the threshold level. The minimum point of the pulse is then determined by comparing each point between the endpoints to all others to obtain the one point with least value. A baseline value of a waveform is illustrated in Figure III-1. Also illustrated in Figure III-1 are a possible trigger level, used to

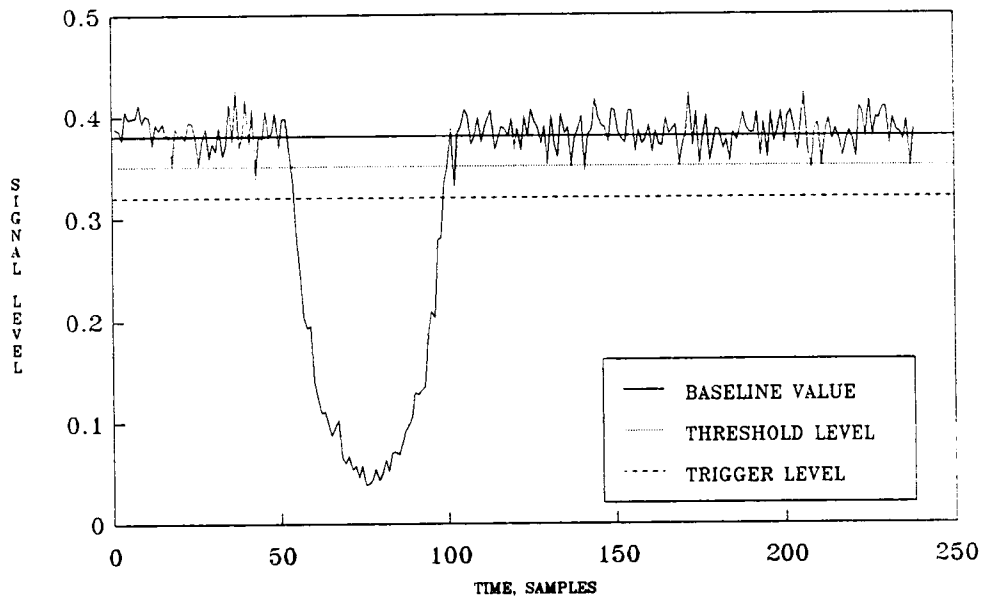


Figure III-1. Baseline Value, Trigger Level, and Threshold Level

determine if a pulse exists, and a possible threshold level, which is used to locate the endpoints of the pulse.

In the first processing algorithm attempted, it was decided for simplicity and efficiency to use equal trigger and threshold levels. In this algorithm, the first point in the waveform with a value less than the trigger/threshold level is assigned to be the start point of the pulse. Similarly, the finish point of the pulse is the last point (following a start point) of a pulse with a value less than the trigger/threshold level. With this approach it is important to set the trigger/threshold level far enough below the baseline to prevent noise from giving multiple pulse readings while at the same time maintaining a trigger/threshold level that is high enough to locate small pulses. Another reason not to set the trigger/threshold level too far below the baseline is to

avoid a negative bias in the pulse width measurement. This bias is manifest as the x-dimension particle size being less than its true value.

Algorithm with Neighborhood Average

Because of the measurement complications introduced by noise, a method for reducing such effects was next included in the algorithm. The first method considered was neighborhood averaging in which each point in the waveform is replaced by the arithmetic mean of the points in some specified neighborhood of the point [12], [13]. With neighborhood averaging, it is important to include the correct number of data points in the neighborhood. The greater the number of points included, the more significant is the smoothing that occurs.

Since it was desired to have the value of the sample being replaced included in the average, only an odd number of neighborhood points were considered. Therefore, the smallest number of points available to constitute an average is three. The greatest number of points available for averaging is dependent upon the position of the pulse and the number of samples in the waveform. For example, if the leading edge of a 25-sample pulse is located at the 75th sample of a 200-sample waveform, the longest neighborhood available is 75. However, a 75-point neighborhood average will hardly be desirable for pulses that are no longer than 50 samples long.

As Figures III-2 through III-4 illustrate, longer neighborhoods have a greater smoothing effect than neighborhoods of only a few samples. Figures III-5 through III-7 show the frequency response of neighborhood averaging with 5, 9, and 21 samples. While greater attenuation of the high-frequency components is achieved with longer neighborhoods, lower information-carrying frequency components are also attenuated, although to a lesser degree.

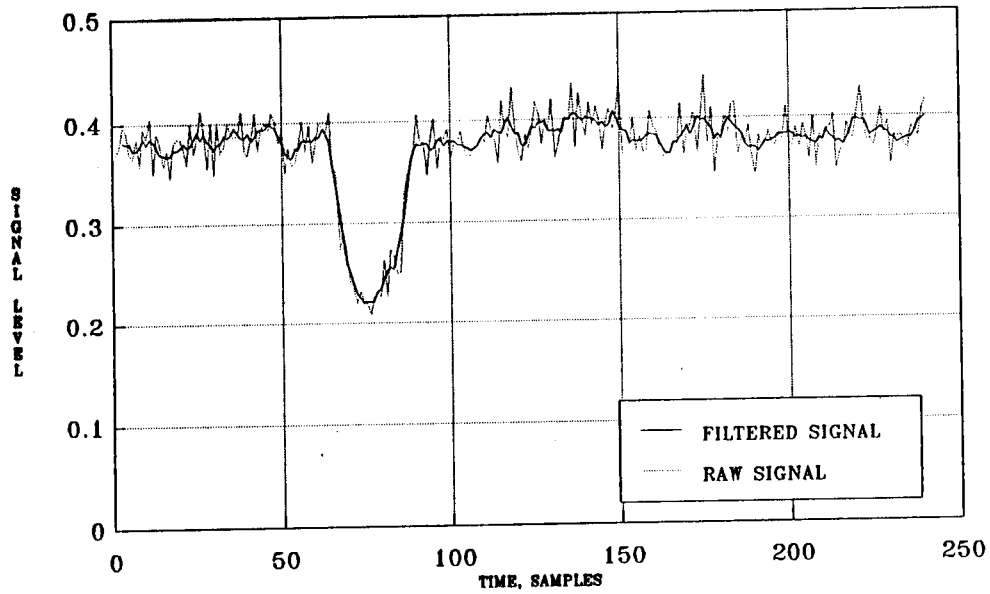


Figure III-2. Simulated Multibeam Signal Smoothed with 5-Point Neighborhood Average

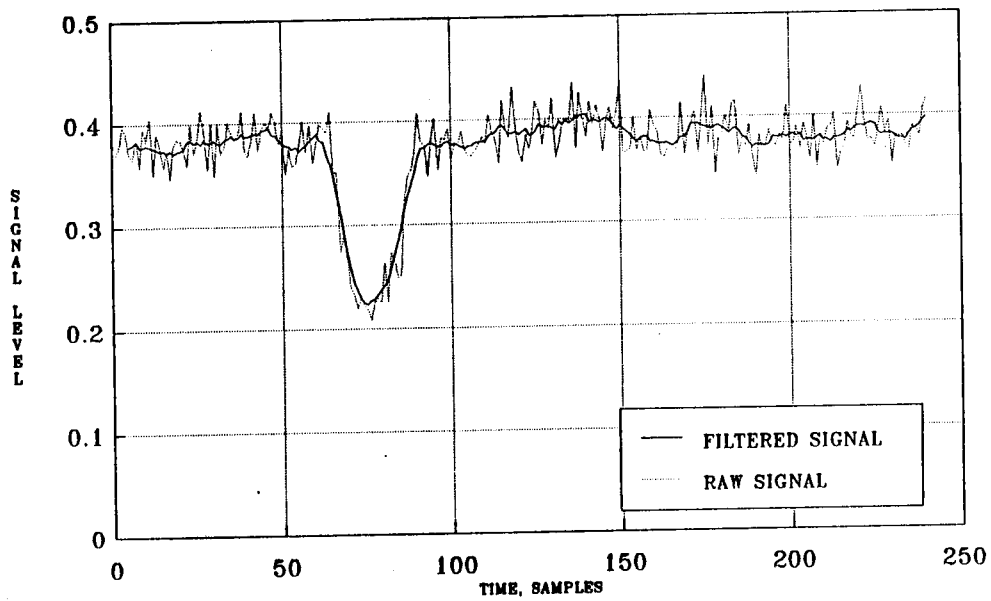


Figure III-3. Simulated Multibeam Signal Smoothed with 9-Point Neighborhood Average

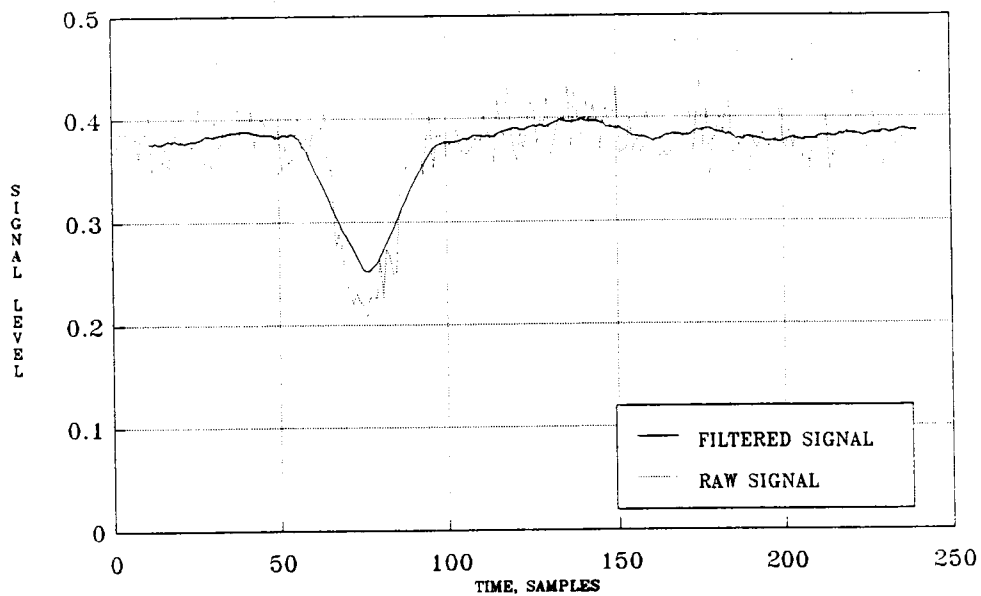


Figure III-4. Simulated Multibeam Signal Smoothed with 21-Point Neighborhood Average

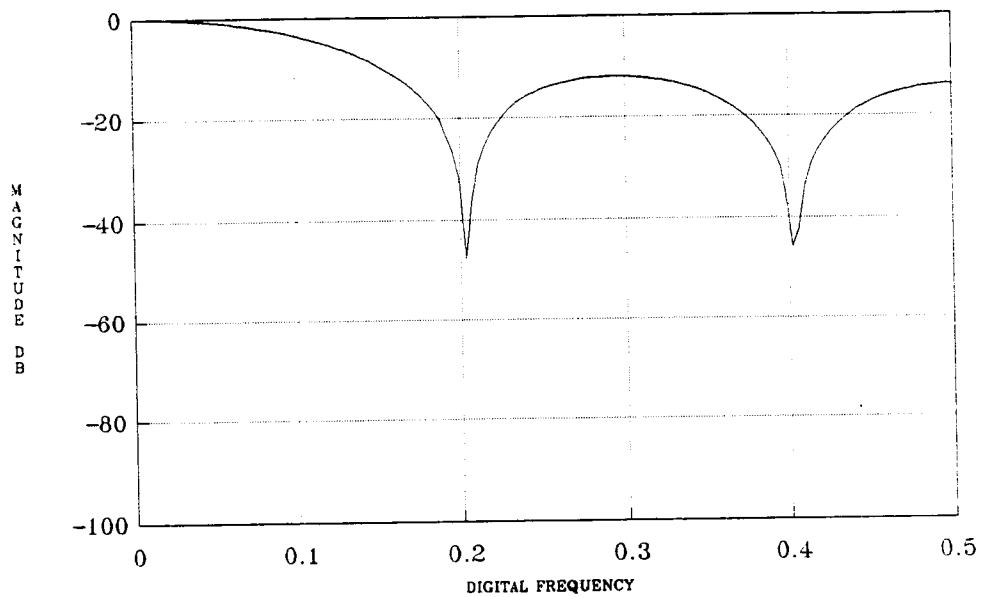


Figure III-5. Frequency Response of 5-Point Neighborhood Average

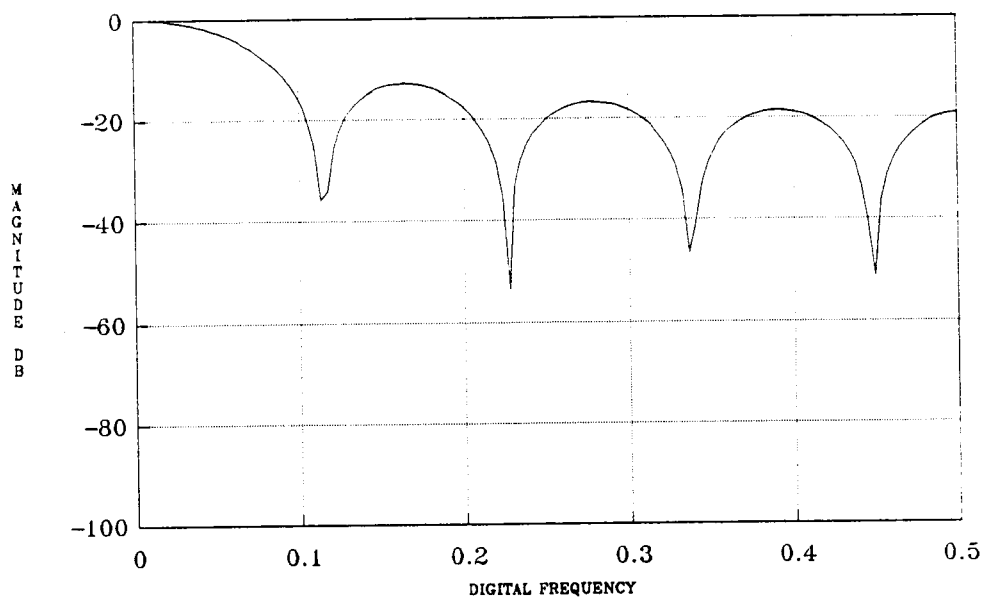


Figure III-6. Frequency Response of 9-Point Neighborhood Average

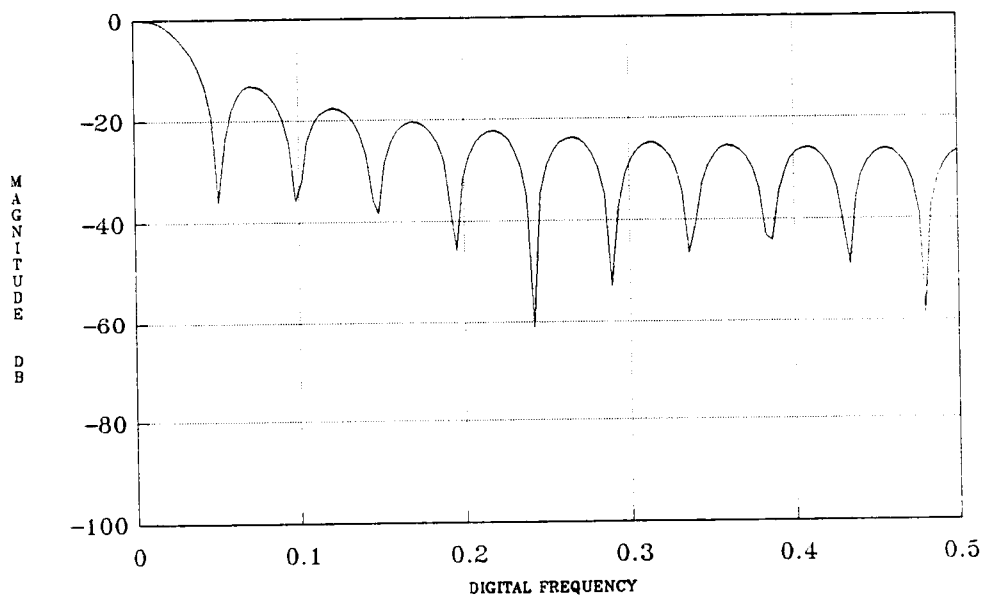


Figure III-7. Frequency Response of 21-Point Neighborhood Average

Figure III-8 shows the frequency spectrum of a typical pulse. As this figure shows, significant frequency components exist in the range being attenuated by the neighborhood average. Figure III-9 shows the frequency content of a simulated pulse with noise. The noise cannot be decreased sufficiently without reducing the signal spectrum by the filtering process. Figure III-10 shows the effects of a neighborhood average filter on a signal resulting from a small particle.

Three different attempts at reducing noise with a neighborhood average were made. The first was a 5-point neighborhood average with the common trigger/threshold level set at 97.00% of the baseline. The number n of consecutive points below the trigger level required to verify the presence of a

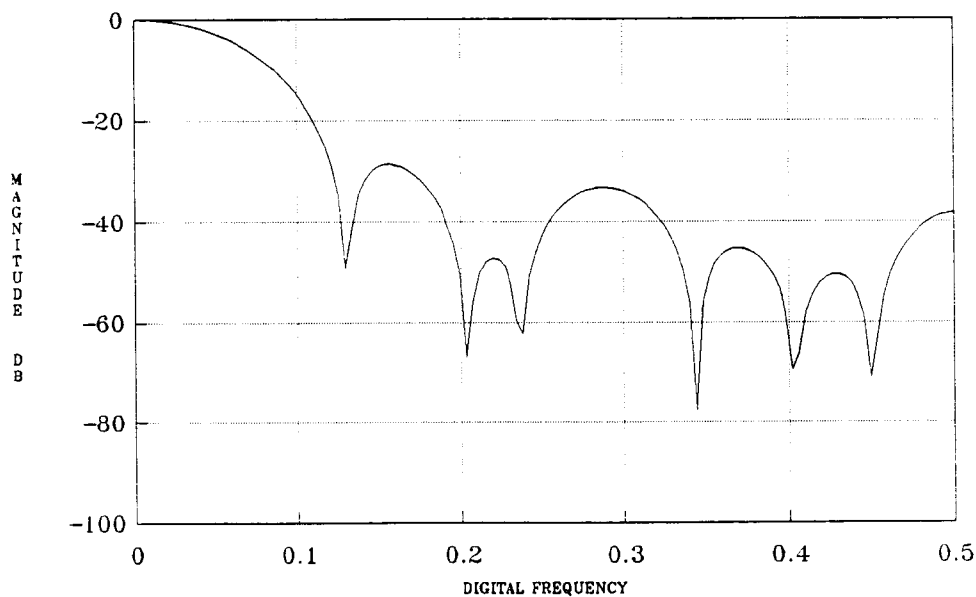


Figure III-8. Frequency Content of Narrowest Ideal Pulse

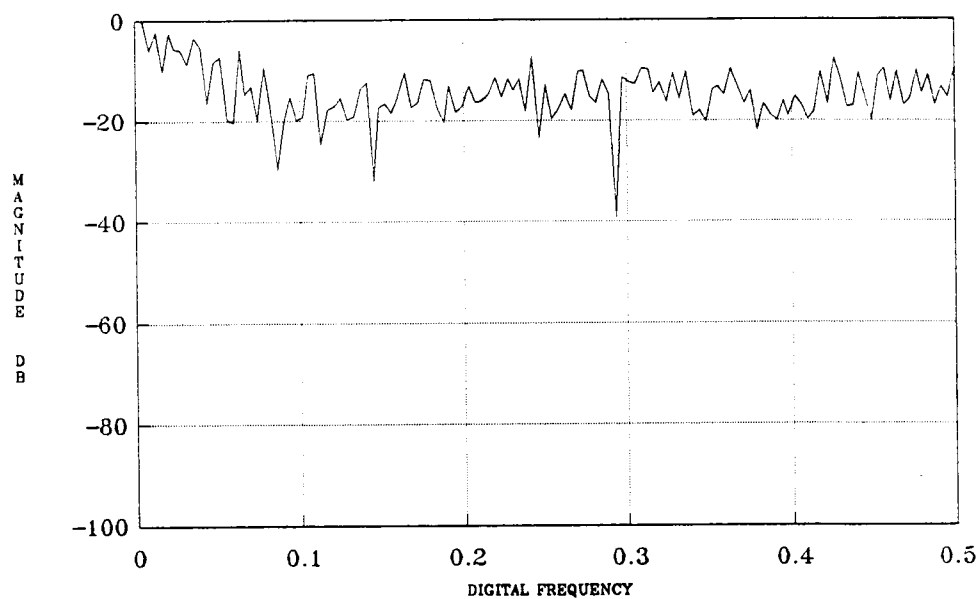


Figure III-9. Frequency Content of Noisy Waveform

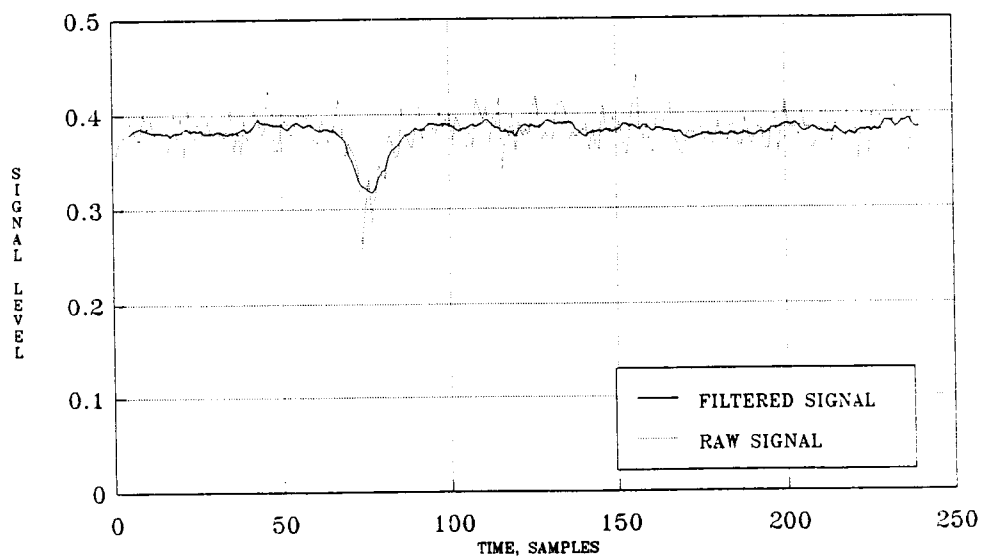


Figure III-10. Signal for Smallest Particle Smoothed with 9-Point Neighborhood Average

pulse was set to 13 since the true value of the shortest pulse width is 16.16 points. In an actual data processing application, n would be set to two or three samples less than the shortest expected pulse as determined by the sample rate, particle size and velocity estimates, and multibeam limitations. The 5-point neighborhood was used as a "first cut" and did indeed reduce the noise in the waveform substantially, as Figure III-2, page 29, shows. The mean pulse height measurements were very close to the true values. However, the mean pulse width measurements for the smaller two simulated pulses were found to be significantly shorter than the true values and raising the trigger/threshold level to correct this measurement error produced multiple-pulse errors. Greater reduction of the noise was needed to allow an increase in the threshold/trigger level without inducing more multiple-pulse errors. Therefore, the neighborhood was increased to 9 points.

The 9-point neighborhood average, a value of 13 for n , and a value of 97.08% of the baseline for the trigger/threshold level increased the mean pulse width measurements too much and also increased the mean pulse height measurements. The trigger/threshold level was subsequently decreased to 95.79% of the baseline, which did not change the mean pulse height measurements but decreased the mean pulse width measurement. However, the mean width measurement of the smallest pulse was decreased too much, as Table III-2 shows. It was therefore decided to try a better filter; one that has noise rejection comparable to a 9-point average while attenuating significant pulse frequency components like a 5- or smaller point average.

Table III-2. Results of Monte Carlo Simulation Using Neighborhood Average in Algorithm

		Condition		
		• 5-point neighborhood average • n = 13 • Threshold = 97% • Trigger = 97%	• 9-point neighborhood average • n = 13 • Threshold = 97.08% • Trigger = 97.08%	• 9-point neighborhood average • n = 12 • Threshold = 95.79% • Trigger = 95.79%
Parameter Measured	Results			
True Value				
R1 Width 16.1553	Mean Std Dev No. of Errors	14.37 2.42 6	16.36 1.79 2	14.505 1.472 None
R1 Height 0.30531	Mean Std Dev	0.3048 0.01371	0.31673 0.00674	0.31636 0.00487
R2 Width 27.4368	Mean Std Dev No. of Errors	26.4300 2.03 1	30.12 2.20 2	27.25253 1.076 None
R2 Height 0.21642	Mean Std Dev	0.21449 0.00637	0.22304 0.0044	0.22005 0.00490
R3 Width 50	Mean Std Dev No. of Errors	50.34 3.14 1	52.527 4.0694 4	50.9798 0.90987 None
R3 Height 0.0437	Mean Std Dev	0.04547 0.00069	0.04851 0.00231	0.04845 0.00200

Algorithm with FIR Filter

A weighted filter, often referred to as a finite impulse response (FIR) filter, can perform better than a neighborhood average which equally weights all the points in the neighborhood. The frequency response of a noiseless ideal pulse, shown in Figure III-8, page 32, was used to design the weighted filter. It was decided that a ninth-order FIR filter would be used to avoid lengthy processing times that a higher-order filter would require. This filter was designed to have a cutoff frequency at approximately 0.2 times the sampling frequency since very little information in the ideal pulses is carried by frequencies higher than this as Figure III-8 illustrates. Minimum ripple was to be present in the passband, with a zero db attenuation at zero frequency. The filter was designed using the standard Parks-McClellan design program for FIR filters [14]. The response of the resultant filter is plotted in Figure III-11 and the coefficients of the filter are listed in Table III-3. The filter attenuates the frequencies not contained in the actual signal without significantly altering the pulse parameters themselves; see a plot of a filtered waveform shown in Figure III-12. A comparison of the frequency content of this waveform, shown in Figure III-13, with the frequency content of a non-filtered waveform, shown in Figure III-9, page 33, further illustrates the effectiveness of the FIR filter. This filter does not have the processing efficiency of the equal-weight neighborhood average, but as shall be shown, the results are far superior to those obtained with the simpler filter.

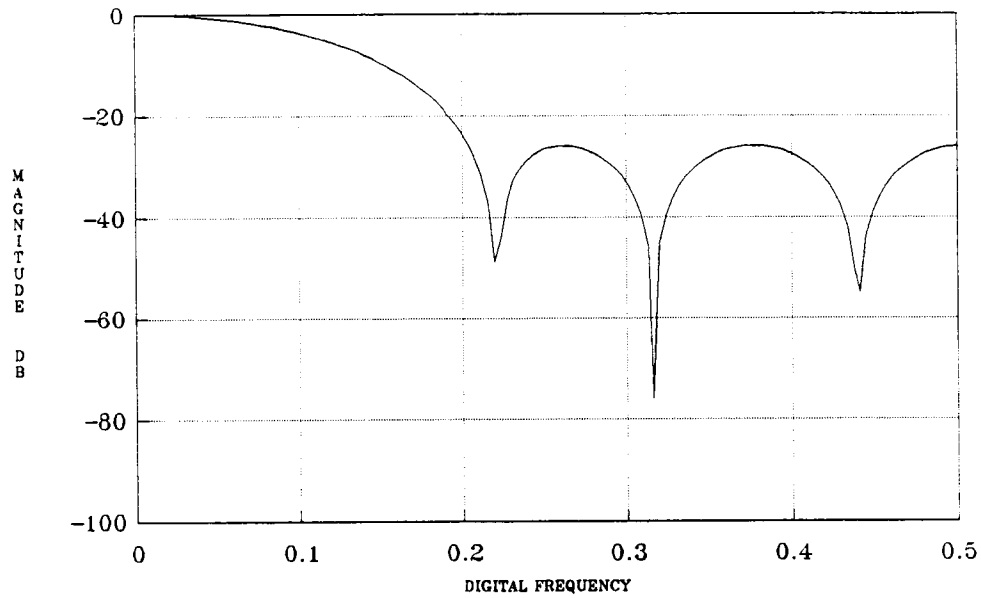


Figure III-11. Frequency Response of 9th-Order FIR Filter

Table III-3. Coefficients of FIR Filter

Tap Number	Value
H(1)	-0.0116848
H(2)	0.0575219
H(3)	0.130963
H(4)	0.205149
H(5)	0.236097
H(6)	0.205149
H(7)	0.130963
H(8)	0.0575219
H(9)	-0.0116848

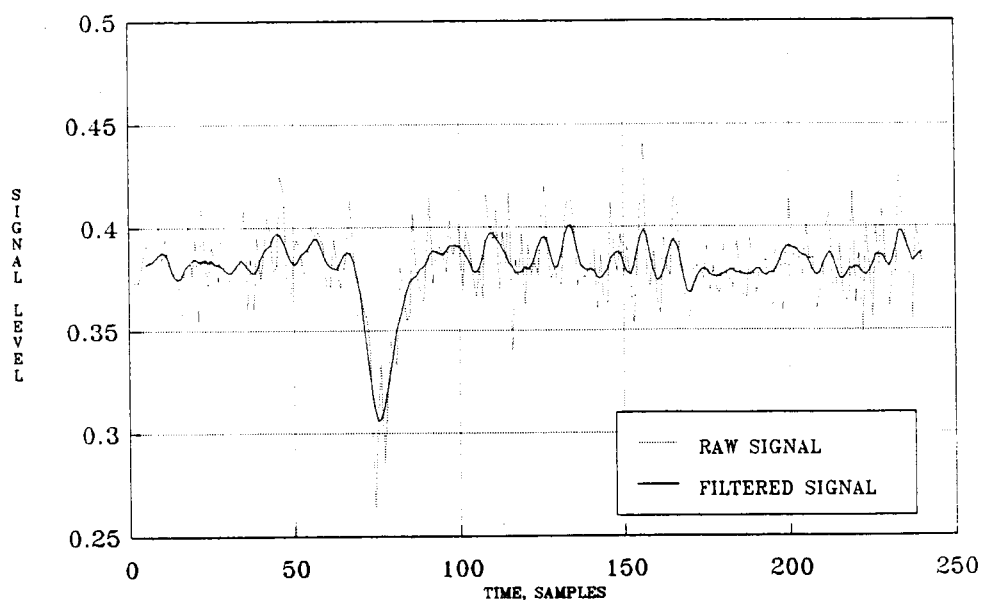


Figure III-12. Simulated Multibeam Signal Filtered with 9th-Order FIR

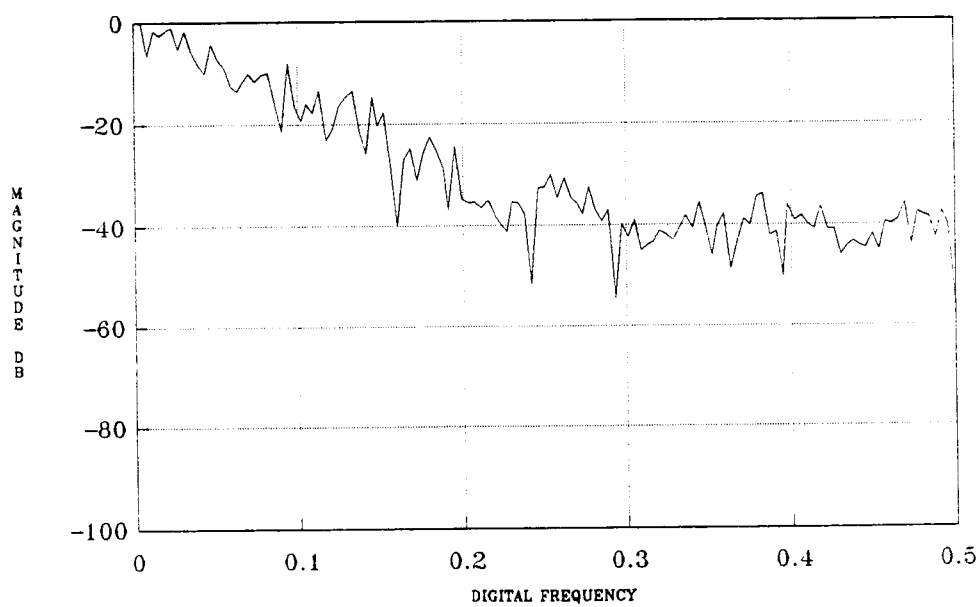


Figure III-13. Frequency Content of FIR-Smoothed Waveform

The improvement in the pulse height measurements, as shown in Table III-4, justifies the use of the FIR filter over the neighborhood average. However, little improvement in pulse-width measurement was realized. In fact, the trigger/threshold level of 97.86% of the baseline, which was required for the best pulse width measurements, produced an intolerable amount of multiple-pulse errors. From this evidence that multiple pulses are detected when the trigger/threshold level is set at a value yielding fairly accurate pulse-width measurements, it was decided that the trigger and threshold levels should be set separately with the trigger level set at a lower level than the threshold level. This would allow the algorithm to detect the existence of pulses with a trigger level set below the noisy baseline in order to avoid multiple pulse errors. At the same time, the threshold level could remain close enough to the baseline to allow the algorithm to yield accurate pulse width measurements.

The trigger level was lowered to 90.6% of the baseline value. Recall that the trigger level is used for comparison with waveform points to verify a pulse is present in the waveform. This was found to be small enough to prevent multiple pulse errors, as shown in Table III-4, but great enough to detect the smallest pulses. If the trigger level were decreased further, it would detect neither noise spikes nor pulses caused by small particles. The number n of consecutive samples required to determine the presence of a pulse was decreased to a value of three since the pulse width is somewhat shorter at 90.6% of the baseline than at, say, 97.86% of the baseline. Also, sporadic noise pulses that are three or more samples wide, which are fairly common at 97.86% of the baseline, are non-existent at the 90.6% level.

Table III-4. Results of Monte Carlo Simulation Using FIR Filter in Algorithm

		Conditions		
		9-point FIR - n = 11 - Threshold = 97.08% - Trigger = 97.08%	9-point FIR - n = 12 - Threshold = 97.86% - Trigger = 97.86%	9-point FIR - n = 3 - Threshold = 96.8% - Trigger = 90.6%
Parameter Measured	Results			
R1 Width 16.1553	Mean Std Dev No. of Errors	14.6566 2.2659 4	16.1515 3.02636 12	16.3030 2.0766 None
R1 Height 0.30531	Mean Std Dev	0.306775 0.00670	0.307272 0.00869	0.306775 0.00670
R2 Width 27.4368	Mean Std Dev No. of Errors	26.54546 2.9517 4	27.8384 4.09 10	28.36364 2.3675 None
R2 Height 0.21642	Mean Std Dev	0.21736 0.01542	0.21885 0.02119	0.21582 0.00616
R3 Width 50.0000	Mean Std Dev No. of Errors	50.03 5.7081 3	49.2828 7.845 13	51.7778 1.88319 None
R3 Height 0.4370	Mean Std Dev	0.04692	0.062977 0.0699	0.046829 0.00227

Table III-4. Concluded

		Conditions
Parameter Measured	Results	● 9-point FIR ● $n = 3$ ● Threshold = 97.86% ● Trigger = 90.6%
R1 Width 16.1553	Mean Std Dev No. of Errors	16.14142 3.0252 None
R1 Height 0.30531	Mean Std Dev	0.30677 0.00670
R2 Width 27.4368	Mean Std Dev No. of Errors	28.1111 3.5954 None
R2 Height 0.21642	Mean Std Dev	0.21581 0.00489
R3 Width 50.0000	Mean Std Dev No. of Errors	51.09091 2.55883 None
R3 Height 0.04370	Mean Std Dev	0.046829 0.00227

Table III-4 shows that the best results of the algorithm are obtained by the use of the FIR filter, the trigger level set at 90.6% of the baseline, and the threshold level set at 97.86% of the baseline. The threshold level is used for comparison with waveform points to determine pulse endpoints. This selection of algorithm parameters yields acceptable pulse height measurements, but maintains the skewed width measurements from one size of particle to the next. For example, if a certain threshold level was found to produce a mean pulse width measurement of the smallest particle exactly equal to the true value of that particle, the other two mean pulse width measurements would be somewhat greater than their true values. A solution to this problem was required.

Algorithm with FIR Filter and Individual Baseline Calculation

Since a change of the threshold level percentage will not produce an equal pulse-width measurement change for each size of particle, the quantity that must be adjusted is the baseline level from which the threshold level is determined. Instead of using a fixed baseline value obtained from the mean baseline values of all waveforms under analysis, it was decided to calculate the baseline value on a waveform-by-waveform basis. This is done by locating the approximate endpoints of the pulse in the waveform and calculating the baseline value of that waveform with the points exclusive of those of the pulse. This type of calculation is actually more suitable in practice than a single baseline measurement made prior to data acquisition because it accommodates slowly changing baseline values caused by contamination of optics, variations in laser power, and changing background light levels. Thus, the single

waveform being processed can produce the desired parameters without the need for any prior information.

The results in Table III-5 show that indeed the spread of the mean width measurements about the true values is less with the individual waveform baseline calculations than with the single mean baseline value. The threshold level had to be increased from 97.86% to 98.2% of the baseline to achieve close to the same mean width measurement as that obtained with the mean baseline value.

Further efforts to decrease the error of the mean pulse width measurements would not decrease the total error of the measurement appreciably as long as the precision error (the standard deviation of the measurement) remains high. Therefore, further efforts at improving the algorithm were directed at reducing the precision error of the measurements.

Algorithm with FIR Filter, Individual Baseline Calculations, and Endpoint Interpolations

Recall that the pulse width measurements are made from the number of points between the two endpoints of the pulse, which are the points with values just less than the threshold level. Since the true endpoints will not occur exactly at the instant when a sample is taken by the data acquisition system, it is possible that a pulse width measurement will be in error by nearly two samples, or about 12.5% of the smallest particle's pulse width, due solely to the location of the endpoints at discrete sample instants. In a similar manner, the point used to yield the minimum value of the pulse could conceivably be in error by up to one-half sample. Therefore, more exact methods of endpoint and

Table III-5. Results of Monte Carlo Simulation Using FIR Filter and Individually Computed Baselines in Algorithm

		Conditions	
		• 9-point FIR • $n = 3$ • Threshold = 98.2% • Trigger = 90.6% • Baseline values computed with individual waveforms	Final Version • 9-point FIR • $n = 3$ • Threshold = 97.63% • Trigger = 90.6% • Baseline values computed with individual waveforms • Linear interpolation of pulse endpoints • Parabolic interpolation of pulse minimum
Parameter Measured	Results		
R1 Width 16.1553	Mean Std Dev No. of Errors	16.2222 3.1158 None	16.1446 2.63142 None
R1 Height 0.30531	Mean Std Dev	0.30677 0.00670	0.30655 0.00673
R2 Width 27.43687	Mean Std Dev No. of Errors	27.69697 2.7134 None	27.7489 2.08485 None
R2 Height 0.21642	Mean Std Dev	0.21582 0.00616	0.21568 0.00617
R3 Width 0.21642	Mean Std Dev No. of Errors	50.74748 3.30394 None	50.88741 2.4896 None
R3 Height 0.04370	Mean Std Dev	0.046829 0.00227	0.04676 0.00228

minimum point location were tried in an attempt to increase the precision of the measurements.

To improve the pulse width measurements, linear interpolation was used to estimate the actual points at which the waveform crosses the threshold level. This was implemented according to the equations

$$S_{int} = S_1 + \frac{V_{S+} - V_T}{V_{S+} - V_{S-}} \quad (8)$$

and

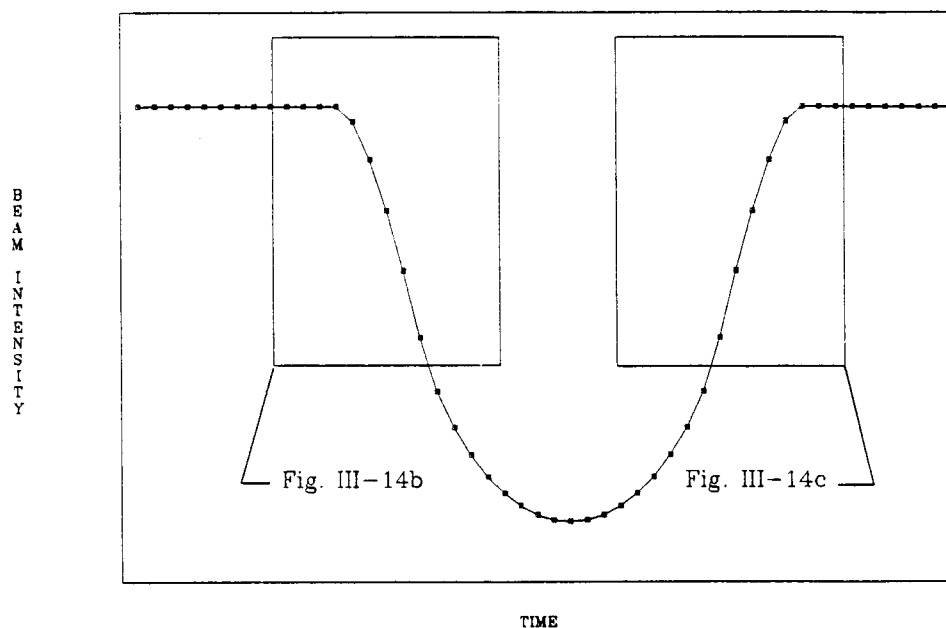
$$F_{int} = F_1 - \frac{V_{F+} - V_T}{V_{F+} - V_{F-}} \quad (9)$$

where

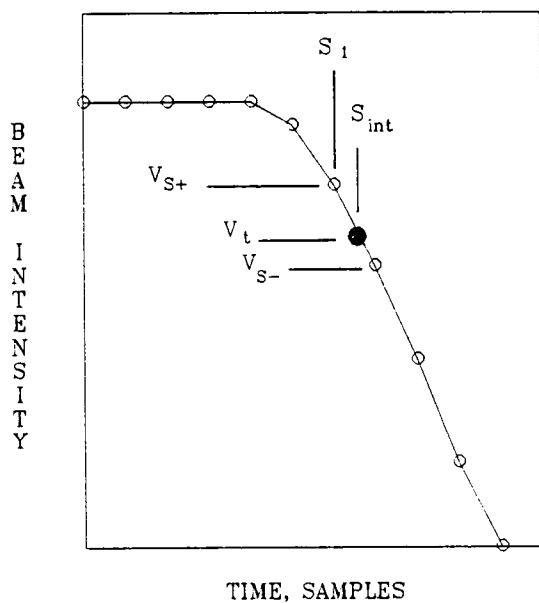
- S_{int} = The interpolated start point of the pulse
- F_{int} = The interpolated finish point of the pulse
- S_1 = The first sample preceding the pulse
- F_1 = The first sample following the pulse
- V_{S+} = The value of sample S_1
- V_{F+} = The value of sample F_1
- V_{S-} = The value of the first sample of the pulse
- V_{F-} = The value of the last sample of the pulse
- V_T = The threshold level.

Figure III-14 illustrates these parameters.

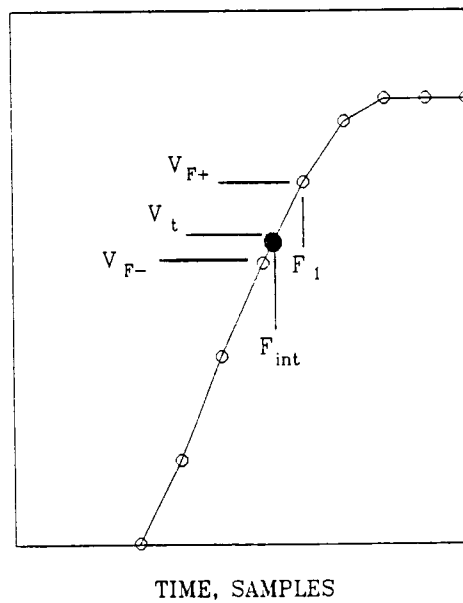
A similar approach was applied to estimate the pulse minimum. In this case, a parabolic interpolation was implemented according to the equation



a. Total Pulse



b. Descending Edge



c. Ascending Edge

Figure III-14. Illustration of Endpoint Interpolation Parameters

$$H_{int} = H_p + \frac{(H_{p+} - H_{p-})^2}{8(2H_p - H_{p+} - H_{p-})}$$

where

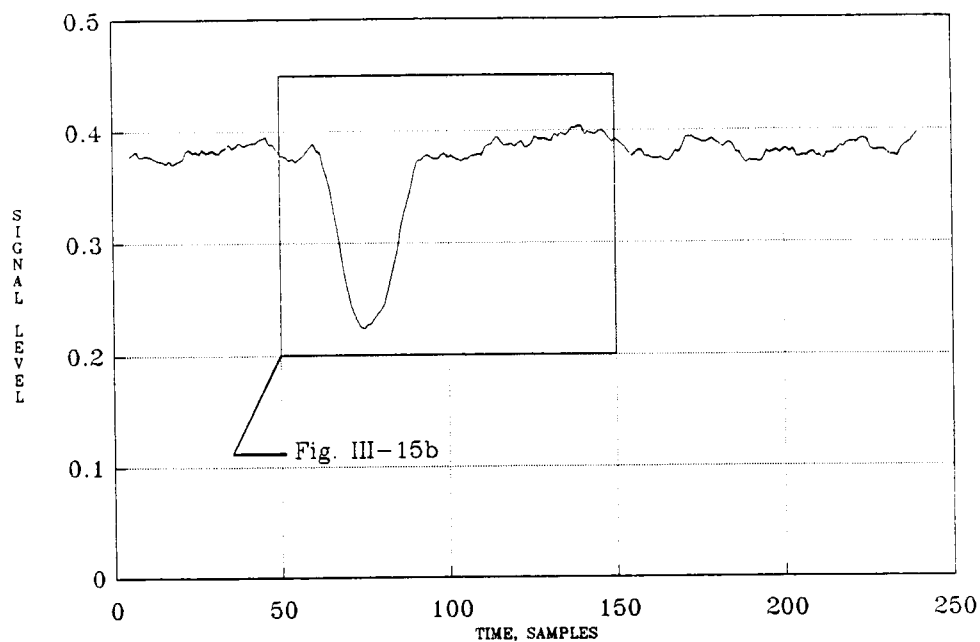
- H_{int} = The interpolated minimum value of the pulse
- H_p = The minimum value of the sampled waveform as determined by the point-to-point comparison
- H_{p+} = The value of the sample immediately preceding the sample of value H_p
- H_{p-} = The value of the sample immediately following the sample of value H_p .

Figure III-15 illustrates these parameters.

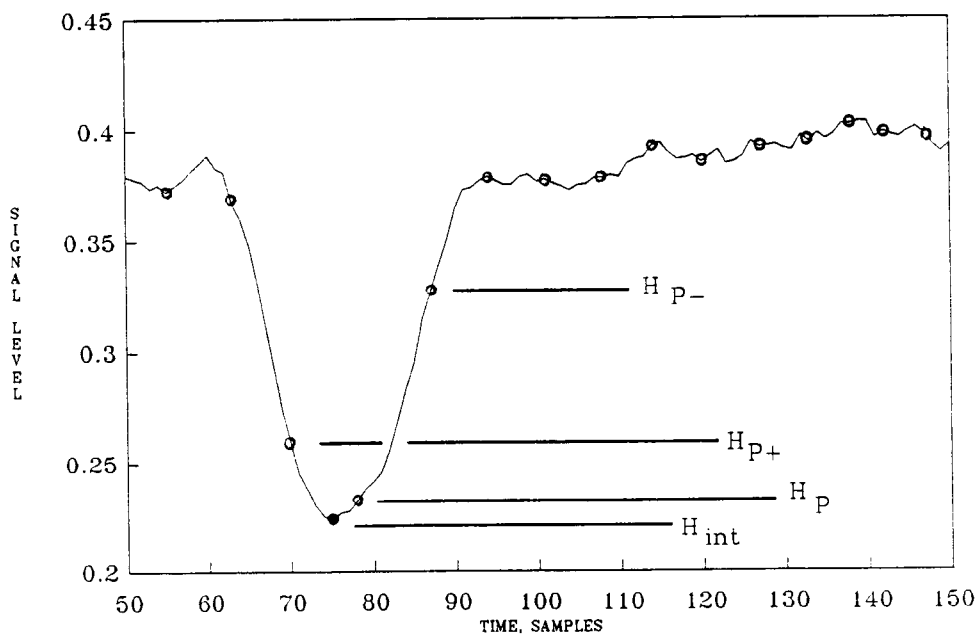
The results of the interpolation, though not as dramatic as expected, did improve the precision of the pulse width measurements, as shown in Table III-5, page 44. The threshold level had to be changed once again to optimize the accuracy of the pulse width measurements. The improvements in the precision of the pulse height measurements due to the parabolic interpolation proved to be too slight to warrant the additional computation and processing time it requires, so it was deleted from the algorithm.

Algorithm - Final Version

Figure III-16 illustrates the final algorithm for locating the parameters in the ribbon beam pulses. The first eight blocks of the flowchart basically account for filtering the raw data and computing the baseline, threshold, and



a. Total Pulse



b. Expanded View

Figure III-15. Illustration of Minimum Point Interpolation Parameters

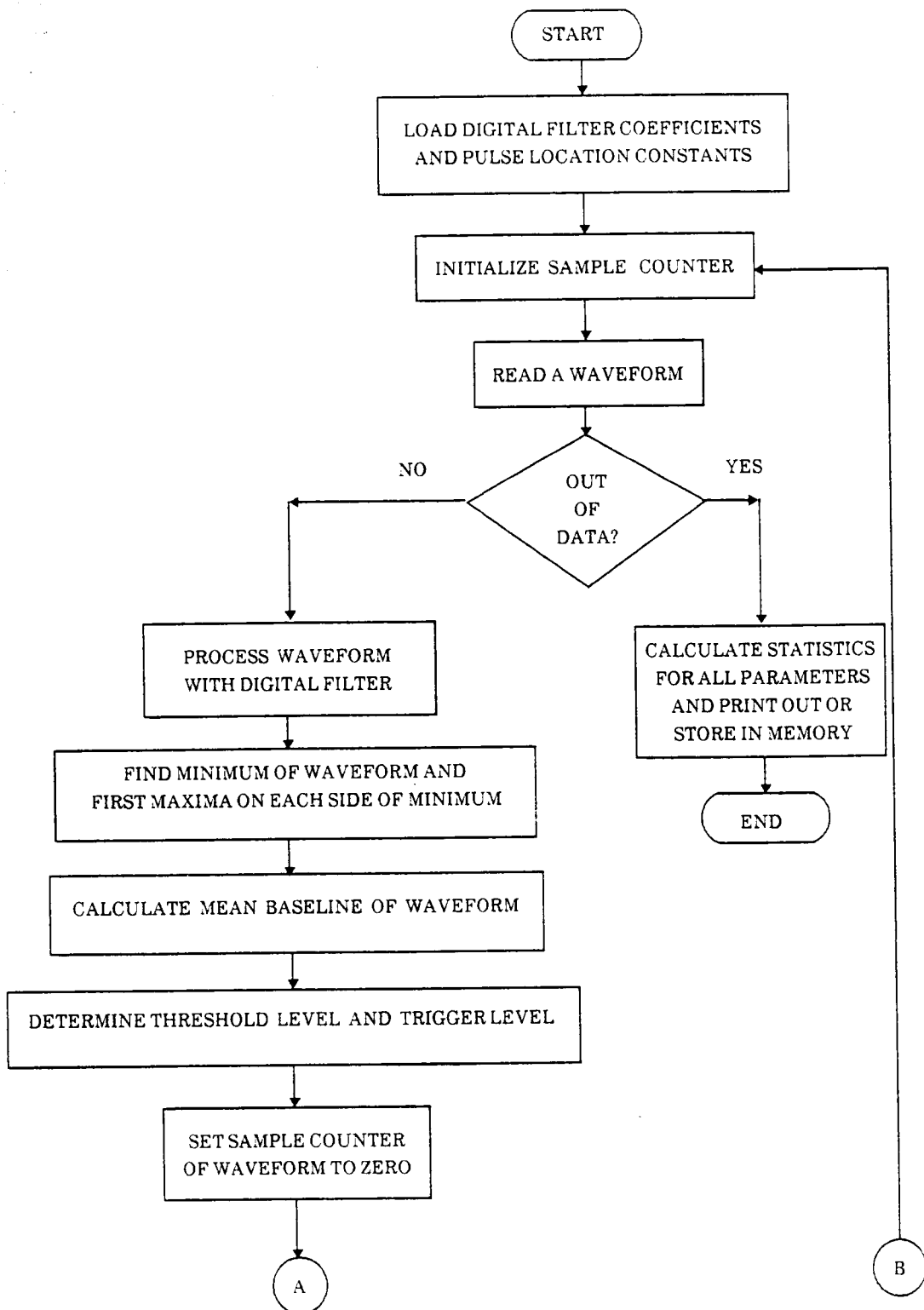


Figure III-16. Flowchart of Processing Algorithm for Ribbon Beam Waveforms

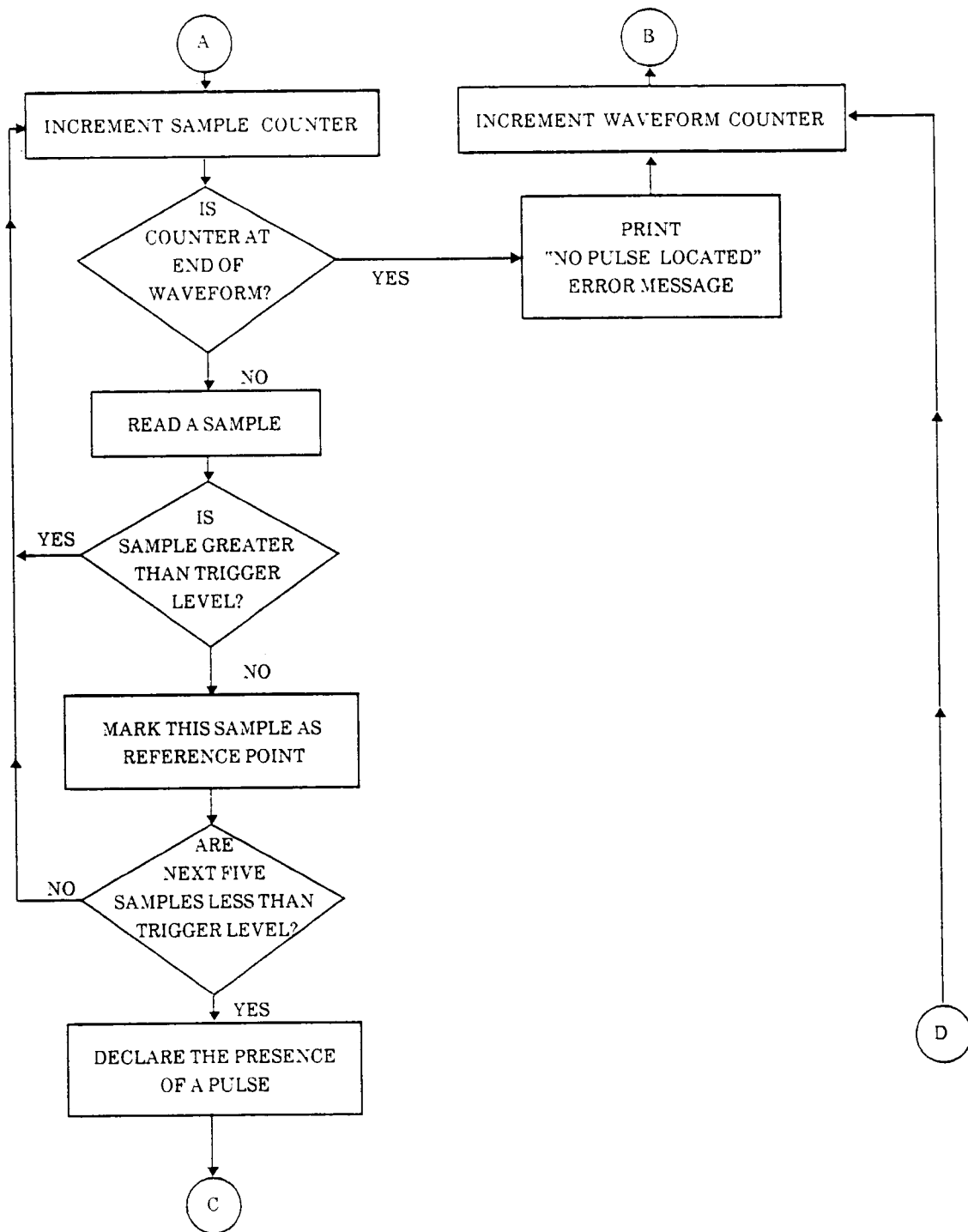


Figure III-16. Continued

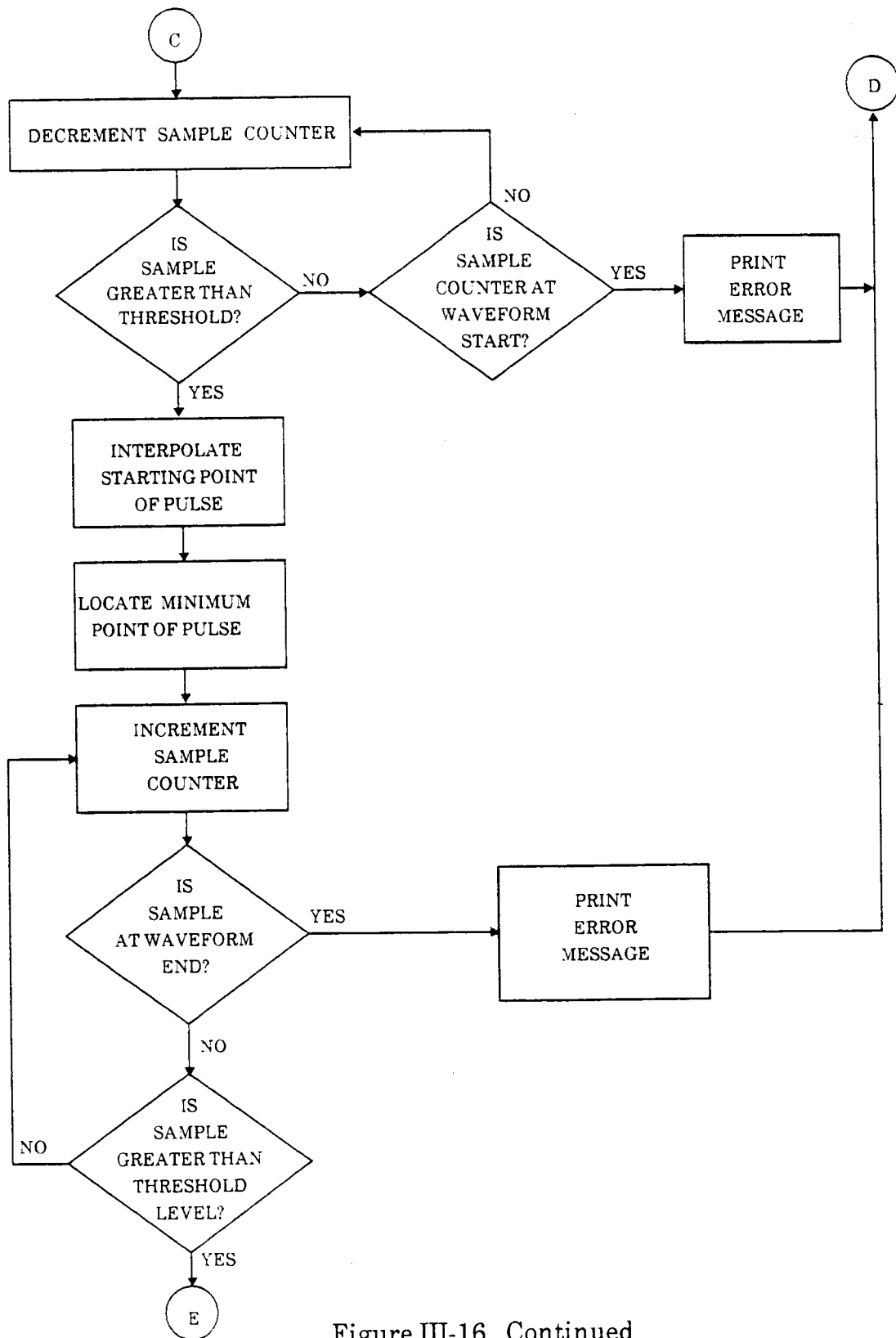


Figure III-16. Continued

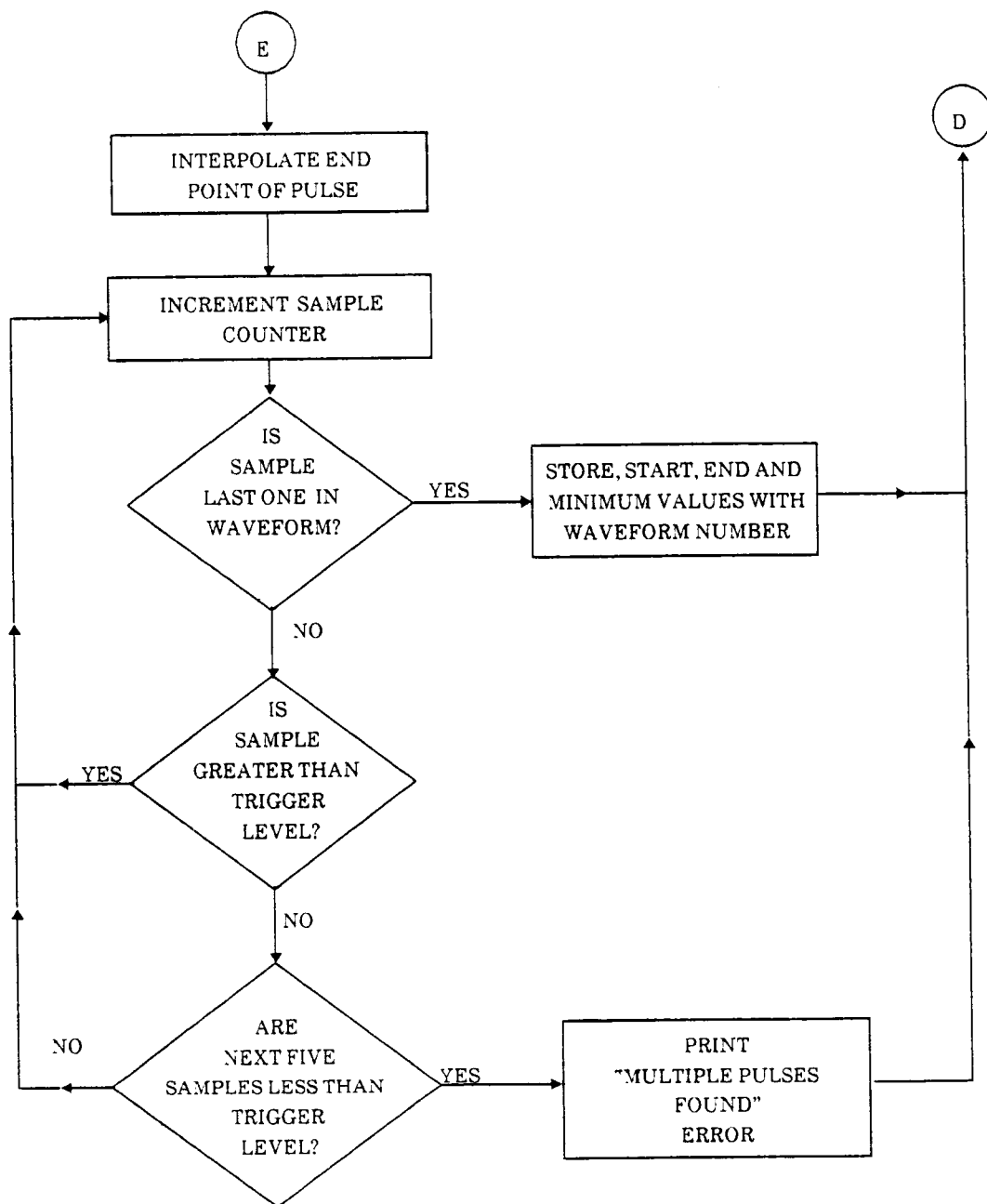


Figure III-16. Concluded

trigger levels. This is done by loading the desired threshold and trigger levels as a percentage of the baseline. Also, the values of the FIR filter coefficients are loaded in this part of the algorithm. The raw data from one waveform is then loaded into memory and is digitally filtered using the FIR filter. Following this, the minimum point of the waveform is found and the baseline of the waveform is calculated by averaging the points in the waveform exclusive of the points between the first maxima on either side of the minimum point. Then, the threshold level and trigger level are determined.

The next twelve blocks locate the pulse, interpolate the required parameters, and check for errors in the waveform. This is performed by resetting the sample counter to the first sample in the waveform and sequentially comparing each sample in the waveform to the trigger level to determine if there is a pulse in the waveform. If a pulse is not found, an error message is issued and the next waveform is loaded for analysis. If a pulse is located, the start point of the pulse is determined by comparison of the sample values with the threshold level and interpolation according to Equation (8). The minimum point of the pulse is next located by finding the one sample of least value between the start point and the maxima to the right of the start point. The end point is then located by comparison of the sample values with the threshold level and interpolation according to Equation (9).

The final blocks of the flowchart check for errors and store the pulse parameters in computer memory for future retrieval. This is done by comparing the sample values following the pulse to the trigger level to make sure that a group of n consecutive samples with values less than the trigger level does not exist. If another pulse is found, an error message is issued, the

pulse data are discarded, and the data from the next waveform are loaded for analysis. However, if no more pulses are located, the pulse start point, end point, and minimum are loaded into memory with a waveform index number for future retrieval.

Because of their pulses' large height and rapid transition characteristics, the cone beam data required no processing such as filtering or interpolation. Therefore, for these pulses the simplified processing algorithm shown in Figure III-17 was used. The results of this algorithm are tabulated in Table III-6. Comparison of Table III-6 with the bottom three rows of Table III-1, page 24, which contains the true values of the cone beam pulse widths, exhibits excellent agreement between the true values of the pulse widths and the values of the pulse widths as calculated by the algorithm illustrated in Figure III-17.

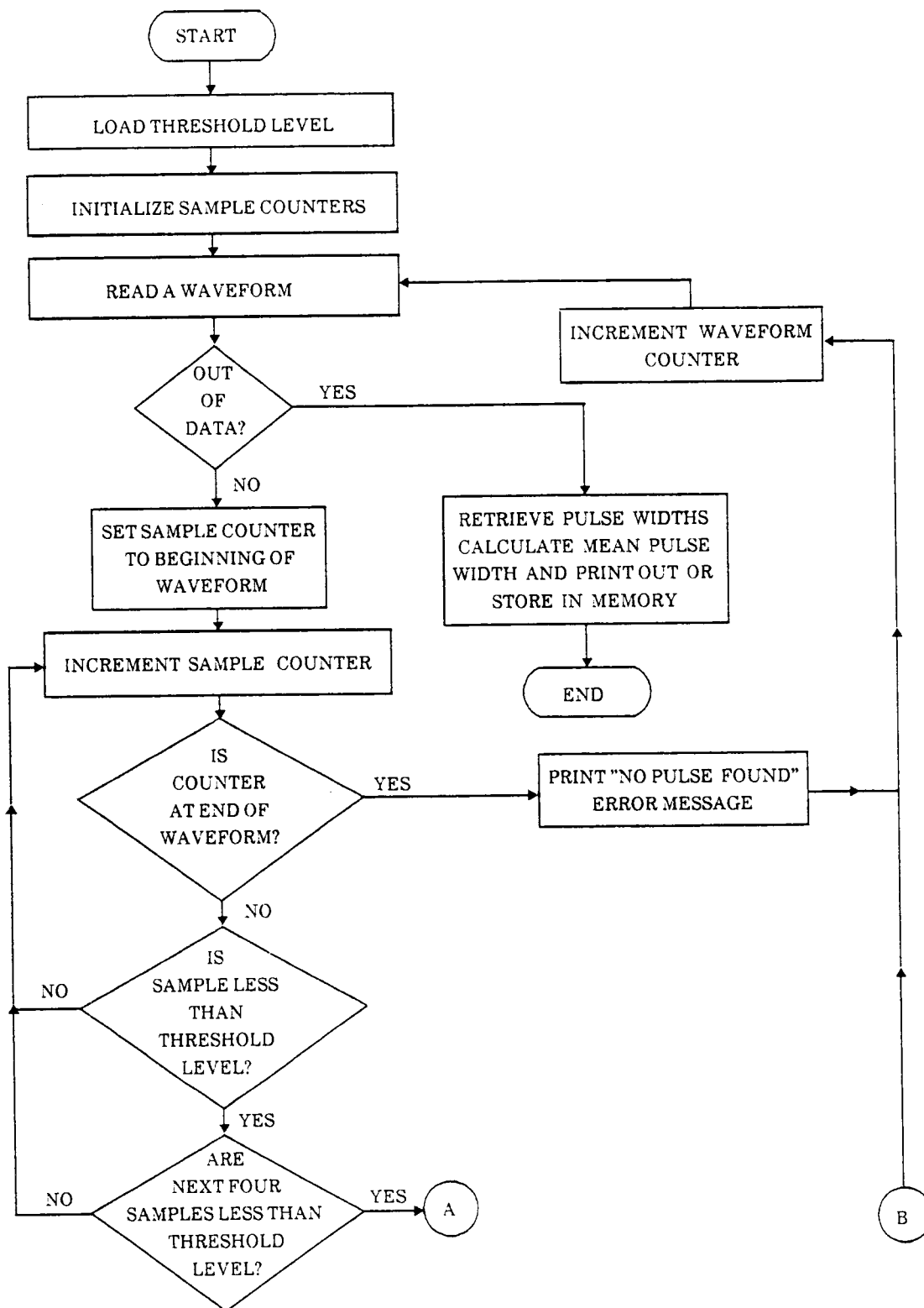


Figure III-17. Flowchart of Processing Algorithm for Cone Beam Waveforms

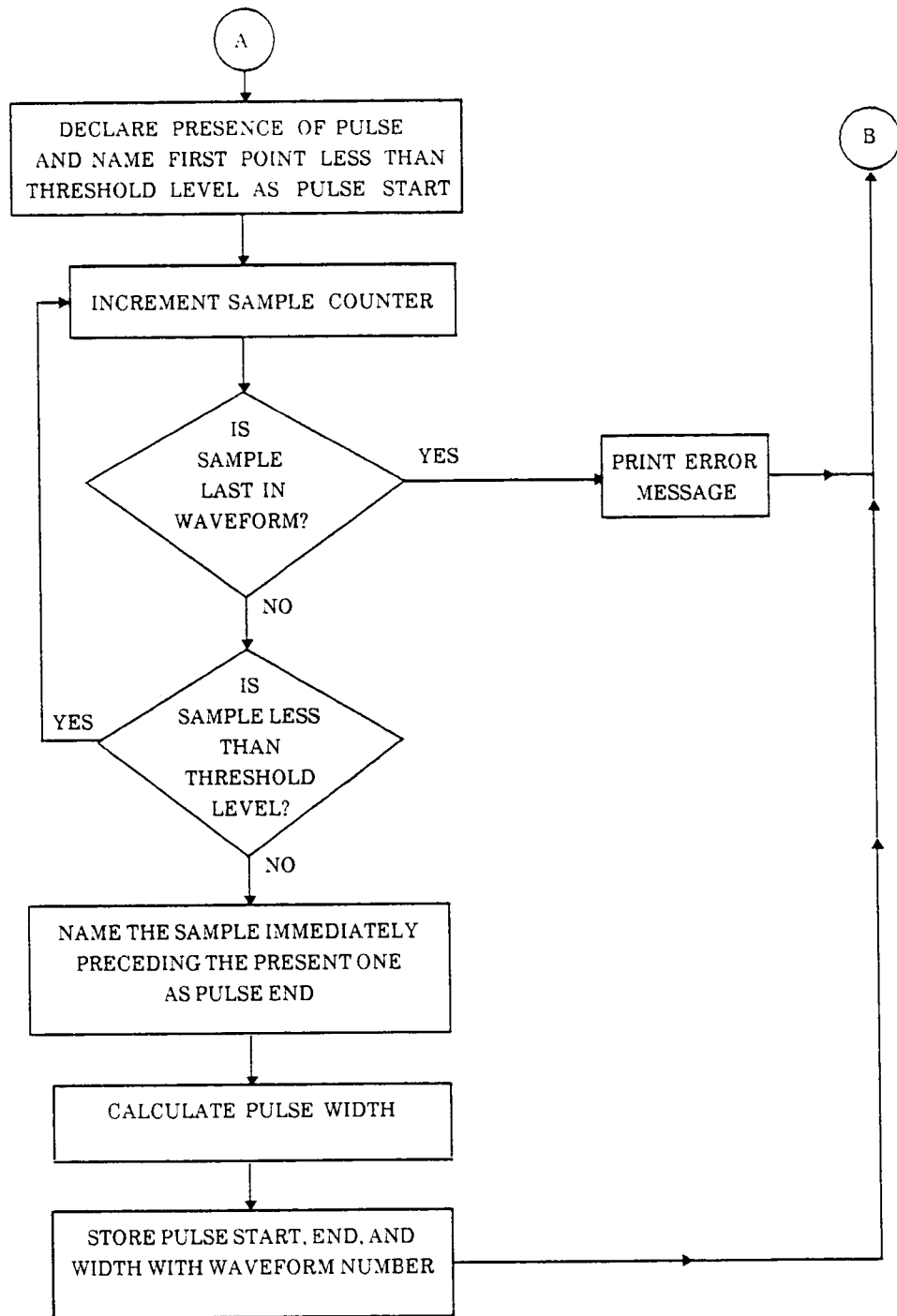


Figure III-17. Concluded

Table III-6. Results of Monte Carlo Simulation for Cone Beam

		Conditions
Parameter Measured True Value	Results	• No Processing • n = 8 • Threshold = 97.08% • Trigger = 97.08%
C1 Width 12	Mean Std Dev No. of Errors	12.000 1.27921 None
C2 Width 24	Mean Std Dev No. of Errors	24.040 0.95265 None
C3 Width 46	Mean Std Dev No. of Errors	45.9899 1.07779 None

CHAPTER IV

PROPOSED HARDWARE IMPLEMENTATION

The algorithm developed in Chapter III could be implemented by a commercial signal processing board to be used with a personal computer. Many companies now produce advanced processing systems designed for installation in PC/AT compatible personal computers. The Signatec DASP100 Data Acquisition and Signal Processing board, manufactured by Signatec, Inc., combines all of the features required to perform data acquisition, processing, and storage in P.C. memory [15].

The DASP100 board utilizes an 8-bit digitizer capable of sampling rates from 781 KHz to 100 MHz. One of the three available data acquisition modes activates data-taking when a triggering event is detected. Data-taking is suspended after a specified amount of time following the disappearance of the triggering event. This mode would be especially useful with the multi-beam system because the system would take data only upon the detection of a particle, thus saving memory during periods when no particles cross the cone and ribbon beams. Also pre-triggering, which allows the data existing prior to a triggering event to be recorded, is especially useful in recording transients such as multibeam pulses. Pre-triggering is performed by storing all data temporarily in a first-in, first-out memory to maintain a buffer of data that occurs before the triggering event. Additionally, the markers automatically inserted into the data by use of this mode would distinguish the data of each waveform from the others.

The TMS320C25 digital signal processor, manufactured by Texas Instruments, is the 16-bit processor that is used on the Signatec board. One feature of the TMS320C25 that is highly advantageous to the pulse processing algorithm is the single-cycle multiply/add instruction. This feature is very beneficial to any algorithm utilizing a digital filter, which by nature requires multiplications and additions for every point in the waveform times the number of taps in the digital filter.

In order for rapid processing to be realizable, the speed at which a waveform is processed must be great enough to prevent bottlenecks in this part of the system. Table IV-1 shows the major time-consuming functions in the final pulse parameter location algorithm with the approximate number of clock cycles required to execute each function [16]. Multiplication of the function execution cycles with the number of times each function is required for processing a 256-sample waveform yields the total estimated number of clock cycles required to extract the pulse width and height from a waveform. For the total estimated number of clock cycles of 5898 and a 100-ns cycle execution time [16], one waveform is estimated to require 590 μ -sec for processing, yielding a processing rate of approximately 1695 waveforms per second for a 256-sample waveform.

Table IV-1. Various Functions of Algorithm
and Their Required Executive Cycles

Function	Clock Cycles Required per Function	Repetitions Required per Waveform	Total Clock Cycles
Loading of Raw Sample into Processor	2	256	512
Digital Filtering	10	247	2470
Location of Approximate Endpoints of Pulse	3	256	768
Calculation of the Mean of the Waveform	1	256	256
Comparison of Processed Sample to Trigger Level for Endpoint Location	6	256	1536
Comparison of Processed Samples within Pulse Endpoints to Locate Peak	4	20	80
Interpolation of End- points	18	2	36
Overhead	240	1	240

Total

5898

CHAPTER V

SUMMARY

A digital signal processing algorithm was developed to process the noisy signals of a nonintrusive laser multibeam particle sizing and velocimetry instrument. The algorithm extracted the parameters required for determining two particle size dimensions and particle velocity.

The algorithm's accuracy was evaluated using simulated multibeam signals. The simulated signals were generated by finding the area blocked by a particle as it crosses the ribbon and cone beams. The beam areas blocked permitted the optical intensity at the photodetector to be determined. A simulation of the photodetector noise then was added to yield the simulated test signals producing waveforms corresponding to pulses of known widths and heights. One hundred such waveforms of three different particle sizes were generated for both the ribbon beam signal simulation and the cone beam signal simulation. These waveforms were then processed by the algorithm. The pulse widths and heights extracted by the algorithm were then compared with the known values to evaluate its performance.

Several versions of the algorithm were proposed, evaluated, and improved. The first algorithm included neighborhood averaging of the data and a pulse detection routine. This version gave pulse width measurements with unsatisfactory bias error and pulse height measurements that were adversely affected by the frequency response of the neighborhood average. In the next version, a nine-point FIR filter with a specified frequency response

was included to improve the pulse height measurement. The final version of the algorithm included the nine-point FIR filter, pulse confirmation and multiple-pulse error detection routines, and linear interpolation of the pulse endpoints to improve resolution.

A hardware system for executing the algorithm was also proposed. The system consists of a Signatec DASP-100 signal processing board installed in a PC/AT personal computer. The board has many features that can be taken advantage of when processing multibeam signals. These include pre-triggering and single-cycle multiply/add instructions, which are useful in implementing the digital filter.

The performance of the algorithm developed is felt to be satisfactory since it offers improved accuracy over previous methods and it provides a systematic and efficient method of obtaining desired particle information from noisy data. The algorithm could process approximately 1695 256-point waveforms per second using the Signatec signal processing board and an AT personal computer.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Bachalo, W. D. and Houser, M. J. "Phase/Doppler Spray Analyzer for Simultaneous Measurements of Drop Size and Velocity Distributions," Optical Engineering, Vol. 23, No. 5, September/October 1984.
2. Tishkoff, J. M., Ingebo, R. D., and Kennedy, J. B. Liquid Particle Size Measurement Techniques, Philadelphia, PA: American Society for Testing and Materials, 1984.
3. Ahlers, K. D. and Alexander, D. R. "Microcomputer Based Digital Image Processing System Developed to Count and Size Laser-Generated Small Particle Images," Optical Engineering, Vol. 24, No. 6, November/December 1985.
4. Brayton, D. B., Bomar, B. W., and Elrod, P. D. "New Particle Diagnostic Methods in Arc-Heated Ablation/Erosion Facilities," Arnold Engineering Development Center-TR-80-64, Arnold Air Force Station, Tennessee, August 1981.
5. Yariv, Amnon. Introduction to Optical Electronics, New York: Holt, Rinehart, and Winston, Inc.: 1971.
6. Roberds, D. W. "Electronic Instrumentation for Interferometric Particle Sizing," Ph.D Dissertation, The University of Tennessee, Knoxville. 1975.
7. Hecht, E. and Zajac, A. Optics, Fourth Printing, Reading, Massachusetts: Addison Wesley Publishing Company, 1979
8. Fishbach, F. A. and Bond, J. S. "Fraunhofer Diffraction Patterns of Microparticles," American Journal of Physics, Vol. 52, No. 6, June 1984.
9. Photomultiplier Tubes, Japan: Hamamatsu, 1988.
10. Linear Products Databook, Norwood, Massachusetts: Analog Devices, 1988.
11. Bennett, W. R. "Methods of Solving Noise Problems," Proceedings of the IRE, Vol. 44, 1956, Page 609.
12. Gonzalez, Raphael C. and Wintz, Paul. Digital Image Processing, Reading, Mass: Addison-Wesley, 1987.
13. AN-2018 Digital Signal Processing, Chestnut Ridge, NY: LeCroy Corporation, 1987.

14. Rabiner, Lawrence R. and Gold, Bernard. Theory and Application of Digital Signal Processing, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.
15. DASP100 Operator's Manual, Corona, California: Signatec, 1989.
16. Digital Signal Processing Applications with the TMS320 Family, Houston, Texas: Texas Instruments Incorporated, 1986.
17. Electronic Components, Harrison, NJ: RCA, 1973.
18. Schaub, S. A., Alexander, D. R., and Barton, J. P. "Theoretical Model for the Image Formed by a Spherical Particle in a Coherent Imaging System: Comparison Experiment," *Optical Engineering*, Vol. 28, No. 5, May 1989.
19. Weiss, B. A., Derov, P., DeBiase, D., and Simmons, H. C. "Fluid Particle Sizing Using a Fully Automated Optical Imaging System," *Optical Engineering*, Vol. 23, No. 5, September/October 1984, page 561.
20. Oppenheim, Alan V. and Schafer, Ronald W. Digital Signal Processing, Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1975.
21. Swokowski, Earl W. Calculus with Analytic Geometry, 2nd Edition, Boston, Massachusetts: Prindle, Weber, and Schmidt: 1979.

VITA

Andrew Eric Nelius was born in Tullahoma, Tennessee on May 26, 1962. He attended Tullahoma City Schools and graduated from Tullahoma High School in May 1980. The following Fall he entered the University of Tennessee at Knoxville. In June 1984 he received a Bachelor of Science degree in Electrical Engineering.

He began part-time graduate study in September 1984 at The University of Tennessee Space Institute. In May 1990, he received the Master of Science degree with a major in Electrical Engineering.

The author is a registered Professional Engineer in the State of Tennessee. He is presently employed by the Calspan Corporation at the Arnold Engineering Development Center as an instrumentation engineer.