APPENDIX 2

MEASUREMENT OF PARTICLE SPEED THROUGH OPTICAL REFLECTIVE SENSING

by

JOHN McCARDLE

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

1993

DISCLAIMER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

V_{mech}	Mechanical velocity estimate
m	Mass
ρ	Density
A	Area
χ	Particle shape factor
P	One pulse method pulse width
d	Detector separation distance
Δ	Two pulse method pulse delay
$\dot{m}$	Mass flow rate
A_i	Amplifier Gain at stage i
R1	Resistor 1
A_i	Voltage at stage i
V_{max}	Maximum velocity
r	Radius of pendulum
$\dot{\theta}$	Angular velocity of pendulum
ω	Radian frequency
L_{cm}	Length of center of mass of pendulum
I	Inertia of pendulum
g	Gravity constant
θ_o	Initial angle of pendulum
v	Velocity
$W(v)$	Shape correction factor
V_{meas}	Measured velocity
Vtrue	True velocity
P[X]	Probability of random variable X

$P[V_{meas}\|V_{true}]$	Probability of V_{meas} given V_{true}
$\tilde{V}$	Ratio of V_{meas} to V_{true}
$\hat{P}$	Continous approximation to discrete density function
V^*	Resultant vector from correlation of probabilities method

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

MEASUREMENT OF PARTICLE SPEED THROUGH OPTICAL REFLECTIVE SENSING

by

John McCardle

May 1993

Chairman: Jose Principe
Cochairman: Daniel M. Hanes
Major Department: Electrical Engineering

Two methods determine the speed of 3 mm glass spheres using optical reflective sensors embedded in a micro-processor system. The first method, which will be referred to as the one pulse method, is sensitive to particle size and shape. The pulse width of a detected particle is measured and normalized by a shape correction factor resulting in a speed estimate. Three models are developed to correct for effects due to particle shape and light scattering inhomogeneities. The second method, which will be referred to as the two pulse method, measures individual particle velocity components independent of size and shape with two detectors spaced a known distance apart. This distance is divided by the delay between the two detectors output pulses to determine speed. A by-product of both methods is a localized particle flux. The microprocessor subsystem automates the pulse detection, timing, velocity calculation and display which are accomplished by the micro-processor subsystem.

The detectors are calibrated and both methods tested using a 3 mm glass sphere attached to a disc rotating at a known velocity. The shape correction factor and particle light scattering characteristics are determined.

In the laboratory, a chute is used to generate particle flows with different characteristics. The detection system is tested in the chute for two different flows. A mechanical speed measurement is used for comparison to the one pulse method. The one pulse method is used for comparison to the two pulse method. A mechanical average mass flow rate is used for comparison to the flow rate measurements. Results obtained indicate that the one pulse method estimate is within 4% of of the mechanically measured speed. The two pulse method gives erroneous results, in this application, due to detector separation distance greater than 3 particle diameters. The mass flow rate measurement gives erroneous results due to detector head placement. Solutions are proposed to correct discrepancies.

CHAPTER 1
INTRODUCTION

1.1 Objective

The focus of this research is the design of a system that detects a single particle in a granular flow and determines its speed. A system such as this benefits researchers in the areas of granular flow and sediment transport because velocity is a key parameter in analyzing particle flow behavior. Mass flow rate, local solid fraction and linear concentration can be developed from velocity values[1]. For this reason many researchers have developed velocimetry devices. The basic components of these systems are a detection device and a processing element. The detector functions as a transducer, i.e., a physical event is transformed into an electrical pulse. The pulses generated by the detector are measured by the processing element to determine velocity. The most prevalent method of particle detection in an aerial medium is optical sensing. One of the first devices was developed by Savage[2] and a modified version is used by Ahn, Brennan and Sabersky[1] to determine particle velocity. This device consisted of two fiber-optic probes spaced a known distance apart. Each probe contains a bundle of fibers, half of which transmit light and the other half receive the reflected light. The received signals are digitized and stored for post processing. Velocity is determined by analyzing the cross-correlation of the two signals to achieve a time difference and dividing it into a known separation distance. The method of cross-correlation was also implemented by Shirai and Nishiwaki[3] using a more sophisticated probe that allowed better resolution of particle path. In either case, due to the intensive computational requirements of the cross-correlation technique, the information is post-processed by some form of computing element.

The velocimetry system for this research uses the Hewlett-Packard HBCS-1100 optical reflective sensor and a Motorola 68000 micro-processor to determine particle speed in real time using less computationally intensive techniques. Speed determination is achieved by two methods. The first, called the two pulse method, is analogous to the cross-correlation technique. The micro-processor measures the time difference between pulse occurrences from two detectors. Velocity is determined by dividing the time difference into a known separation distance. The second, called the one pulse method, measures the time duration of a pulse from a single detector. Speed is determined by dividing the time duration into a characteristic particle length. The characteristic length, which is found experimentally, combines particle size, reflectivity and geometry into a linear function. Note that directional information is not contained in the one pulse method thus the term speed is used rather than velocity as in the two pulse method.

The body of this thesis details the selection of the optical sensor, pulse amplification, analog to digital conversion, design of the micro-processor subsystem and construction of efficient machine code algorithms to achieve real time detection and velocity calculation.

1.2 Chute Description

Laboratory tests are conducted in a flow simulation system which consists of a 144" x 6" x 18" inclined chute with transparent side walls, a conveyer for particle recirculation, an upper hopper for granular storage, a lower hopper for guiding the discharge to the conveyer and an 8 inch diameter hose connecting the end of the chute with the lower hopper. The granular material consists of approximately 1 million clear glass spheres, 3 mm in diameter. Mass flow rate is controlled in the entrance chamber, located between the upper hopper and the beginning of the chute, by an adjustable gate. A mechanical means of velocity determination is arrived at by

$$V_{mech} = \frac{\mathsf{m}}{\rho \cdot \mathsf{A}} \tag{1}$$

where ρ and A are the local density and area and m is the mass flow rate. The mass flow rate is determined by measuring the time required for a given weight of particles to be discharged through the 8" hose. The local density is found using a system of guillotine-like traps spaced evenly above the chute. When released, a 12" x 6" area, A , of the flow is trapped. The particles inside the trap are weighed and a local density, ρ, is produced.

CHAPTER 2
METHODS OF GRANULE DETECTION

2.1 Method Selection

There are many methods to determine object speed, however, few are effective with small particles. The first consideration is interference of flow; the detection system must be unobtrusive. This implies a remote sensor detecting particles from a distance. However, the use of acoustics is rejected because of the difficulties of generating ultra high frequencies in an aerial medium. Also, any form of capacitive sensing cannot detect a single particle in a moderately dense flow without electrode interference. Optical reflective sensing is chosen because it has none of the drawbacks of the above methods. A significant benefit of a light sensing system is that a collimated, focused transmission signal can be generated which produces a return from a single particle within the diameter range of spheres used. This leads to speed measurement of an individual particle.

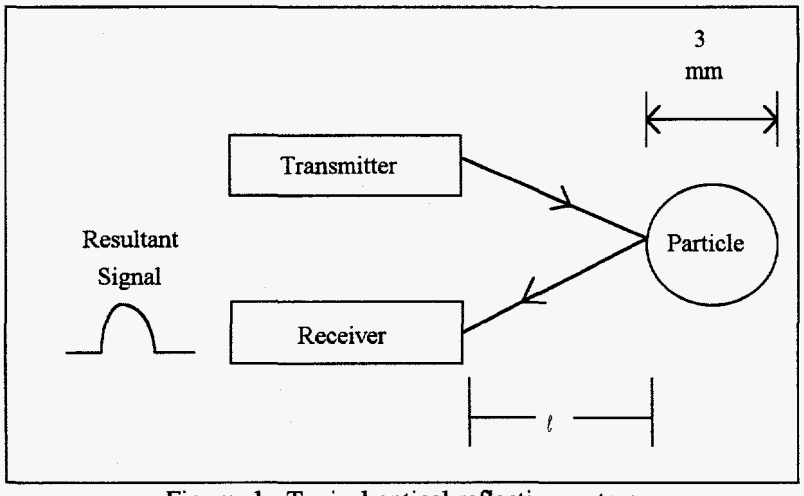

Figure 1. Typical optical reflective system

2.2 Pulse Detection

A simple optical reflective system is shown in figure 1. Light is emitted from a transmitter such as a diode laser, it travels a distance *d* and strikes the particle. A percentage of the light is reflected back in the general direction of the receiver which detects the intensity of the reflected light and produces a current proportional to the signal strength.

Three optical systems are evaluated. The first is based upon the Hewlett-Packard HBCS-1100 Optical Reflective Sensor. This device contains both a receiver and transmitter in one 8.22 mm diameter cylindrical package. A lens is used to focus 720 nm light into a spot 0.19 mm in diameter, at a distance of 4.27 mm from the transmitter. A second lens creates a directional receptive cone of the same focal length resulting in a narrow detection area. The most favorable aspects of this device are a highly focused transmission/reception system which reduces interference from background lighting and increased gain from an internal transistor amplifier.

The second system was temporarily on loan from the Department of the Army. It transmits visible light through two double stranded fiber optic probes and receives the reflected signal with a photo diode. The probes are angled such that the light beams intersect at the center point of the 4 mm wide diode. A major advantage of this device is a reduction in detector head size by a factor of two from the HP sensor. A small detector head reduces the separation distance between detector focal points which increases the accuracy of the two pulse method of velocity measurement.

The third system uses a through beam method. An unfocused multi-frequency light source is placed directly across from a 4 mm wide photo diode to produce a steady state intensity. The penumbra of a particle passing through the reception area reduces the light intensity at the receiver generating a pulse. This method has the advantage of increased focal range, up to 12 inches, but provides useful information only in sparse flows.

As stated previously, the received light intensity is converted into an electrical current by a photo diode. The current pulse produced from detection of a glass sphere is quite small due to the poor reflective properties of glass, thus signal amplification is required. A common emitter transistor amplifier is used with the HP and through beam

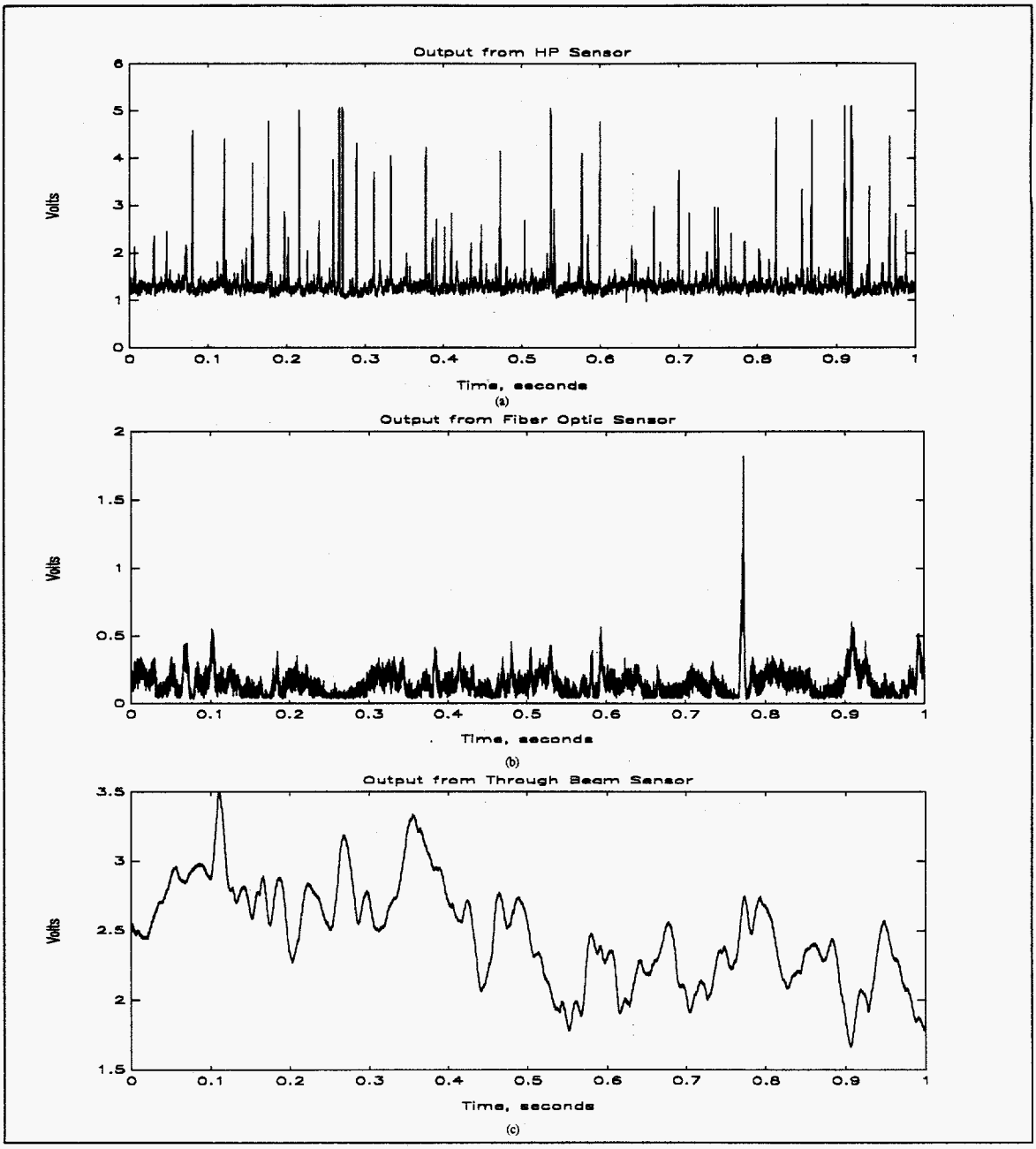

Figure 2. Comparison of detection methods in actual flow environment. a) HP sensor, b) Fiber Optic sensor, c) Through Beam sensor.

systems to increase the current gain of the received signal. The receiver circuitry of system two includes an amplifier thus no additional components are required. The amplified signal is sampled at 20,000 Hz using the Global Lab data acquisition system. Figure 2 shows examples of the signals output from the three types of sensors. A moderately dense flow is generated in the chute using 3 mm clear glass spheres. Each system is tested at the same position, approximately 18 in. from the entrance chamber. It is clear from figure 2c that discerning a detection with the through beam system is difficult for this type of flow. The fiber optic system has a maximum signal to noise ration of 45 dB, however, very few pulses are detected at this level. The Hewlett-Packard sensor provides a maximum signal to noise ratio of 62 dB with many more detection's per unit time than the fiber optic system.

Two factors are considered for selection of the optical sensor system; signal sensitivity and development time. The fiber optic system can be just as effective as the HP sensor if focusing lenses are employed to increase the number of detection's per unit time. The major drawback to this system is availability. Since the system was on loan, a considerable amount of time is required for reproduction and modification. The HP sensor is selected as the optical reflective system because of the high SNR, the large number of detection's per unit time, it is commercially available and relatively inexpensive.

Desirable properties of the reflected signal are large amplitude and wide pulse width. A high SNR facilitates pulse detection and a wide pulse width is desirable because greater resolution at higher speeds is possible with the one pulse method. Clear glass is not a highly reflective material thus these signal properties are difficult to achieve. The reflected signal from a glass sphere is reduced for two reasons. First, the law of reflection states that the angle of incidence is equal to the angle of reflection. For maximum reflection a zero degree angle of incidence is required, thus the transmission beam must be positioned along the equator at the center point of the sphere. The reflective area is reduced because at maximum reflection the 0.19 mm diameter focal spot corresponds to

an area which is 0.8% of the total reflective area of the particle. Experiment has shown that the return decreases rapidly as the beam moves away from the center point. Secondly, the intensity of the reflected light is reduced because the refractive index of silicate glass is 1.52 which corresponds to a transmission resistance 66% that of air. Thus most of the light is transmitted through the sphere at zero degree's incidence. The combination of these two difficulties result in very little return processed by the HBCS-1100 at the maximum angle of reflection.

Two options are explored to increase the reflectivity of the glass sphere. First, a glass sphere is coated with high gloss, red enamel finger nail polish. The second option is a commercially available glass sphere coated with blue colored, transparent thin film. A comparison of these two options with clear glass is shown in Figure 3. In order for the

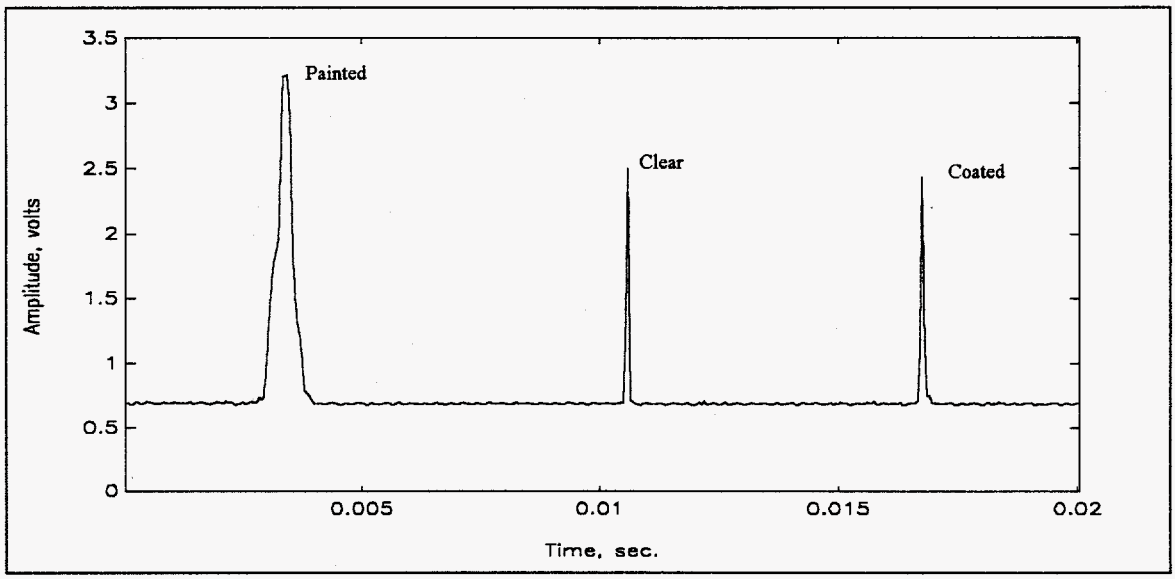

Figure 3. Comparison of sphere coatings

comparison to be meaningful the particles are of the same size, 3 mm, and measured at the same velocity. As can be seen in figure 3, the painted sphere produces a significant increase in the reflective area resulting in a signal with a larger pulse width. Although the results are desirable, this solution is impractical for three reasons: collisions will result in

chipping of the polish thus increasing the debris level in the chute, the coating will alter the coefficient of restitution and lastly, it is inconceivable to evenly coat nearly 10 million spheres with nail polish! The return signal from the coated sphere is nearly the same as clear glass. One option not explored is the use of opaque particles. A smooth opaque particle will have more reflected signal because of reduced transmissive losses. This option is not considered at the present due to a significant increase in cost. Since no practical solution is available to increase the reflectivity of a glass sphere, a wide band amplification system is required to boost the signal to noise ratio, increase the pulse width and account for fast rise times.

2.3 Velocity Calculation

Velocity is calculated by measuring either the time it takes for a particle to travel a known distance or the distance traveled in a known time. The former is more useful than the latter in optical sensing. Two methods are proposed to measure speed in real time. For convenience they are called the One Pulse Method and the Two Pulse Method.

2.3.1 One Pulse Method

The one pulse method relies upon measurement of the pulse width and a known particle size. The equation for this method is

$$V_{1p} = \frac{\chi}{P} \tag{2}$$

where χ is the characteristic length scalar and P is the pulse width. The parameter χ is derived in chapter 4. For the moment it is sufficient to say that χ is a function of the form $a \cdot x + b$ which fits the data to a calibration curve. The one pulse method of speed measurement takes advantage of the uniform size, shape and composition of the granular material. However, every detection will not occur at the particle equator thus maximum

pulse width is not expected for every detection. Due to the geometry of the glass spheres, smaller pulse widths are measured as the position of the detector focal point increases in latitude away from the equator. The reduction of pulse width is due to the decreasing angle of incidence resulting in more light reflected into space and less to the detector. The

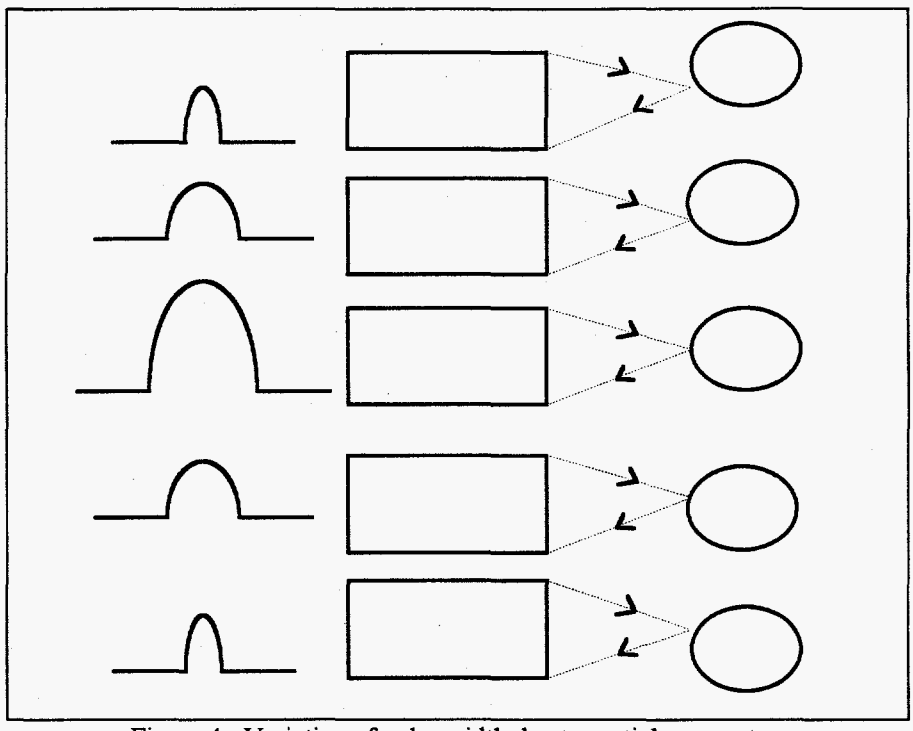

Figure 4. Variation of pulse width due to particle geometry.

probability of passage at a specific latitude versus another is equal. Consequently, for a random sample of detection's, a uniform distribution of pulse widths is expected with the mean equal to the true quantity(unbiased estimator). Thus individual speed measurements with the one pulse method are not reliable indicators to the true speed. However, for constant flow profiles, measurements averaged over time provide a meaningful estimate to the true speed.

2.3.2 Two Pulse Method

The Two Pulse Method is able to measure individual velocities because there is no dependence upon particle size, shape or position. This method uses a pair of detectors separated by a fixed distance. When a particle crosses the focal points of the detector pair, a time lag between the leading edges of the resultant pulses occurs. The velocity equation for this method is

$$V_{2p} = \frac{d}{\Delta} \tag{3}$$

where d is the separation distance of the detector pair focal points and Δ is the measured lag between pulses. This method is very straight forward and provides excellent results in a controlled environment. The one detracting factor is focal point spacing. The detector pair is mounted in a specially designed lexan housing such that the focal points are 4.27 mm from the housing face and separated by a distance of 9.5 mm(the theoretical minimum separation distance is 8.4 mm). This separation distance corresponds to more than 3 particle diameters which increases the possibility of a particle being sensed at one detector but not the other. An error such as this is nearly impossible to correct with software logic. Statistical algorithms can be applied to derive information from data sets containing these errors however this defeats the purpose of accurately measuring individual particle velocities. In summary, the two pulse method is a viable option for particles with a diameter greater than the focal point separation distance.

2.4 Flow Rate Calculation

Flow rate calculation is extremely straight forward. Essentially, the flow rate is determined by the number of detection's within a given time span. The unit for this measure is particles per second. A mass flow rate per unit area can be determined if the area in which a particle may be detected is known. The mass flow rate per unit area is

chosen because a mechanical mass flow rate measurement is available for reference. The reflective area is measured using an XYZ positioning mechanism mounted in an optical board and a glass sphere glued to the edge of a disc rotating at a constant velocity. The detector head is clamped into the positioning device and the focal point is initially adjusted to achieve maximum pulse width during particle detection. The detector head is incrementally moved in an X-Y plane(see figure 5) until no pulses are detected. The increment adjustment is 1/4 turn of the X and Y screws on the positioning device. A full 360 degree turn of an X or Y direction positioning screw corresponds to 1/80 inch, thus

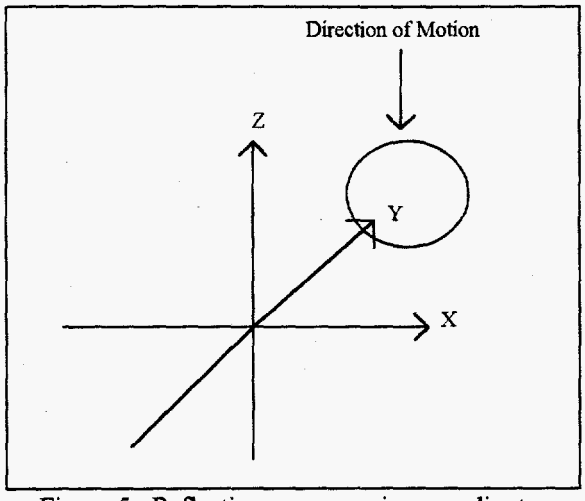

Figure 5. Reflective area mapping coordinates.

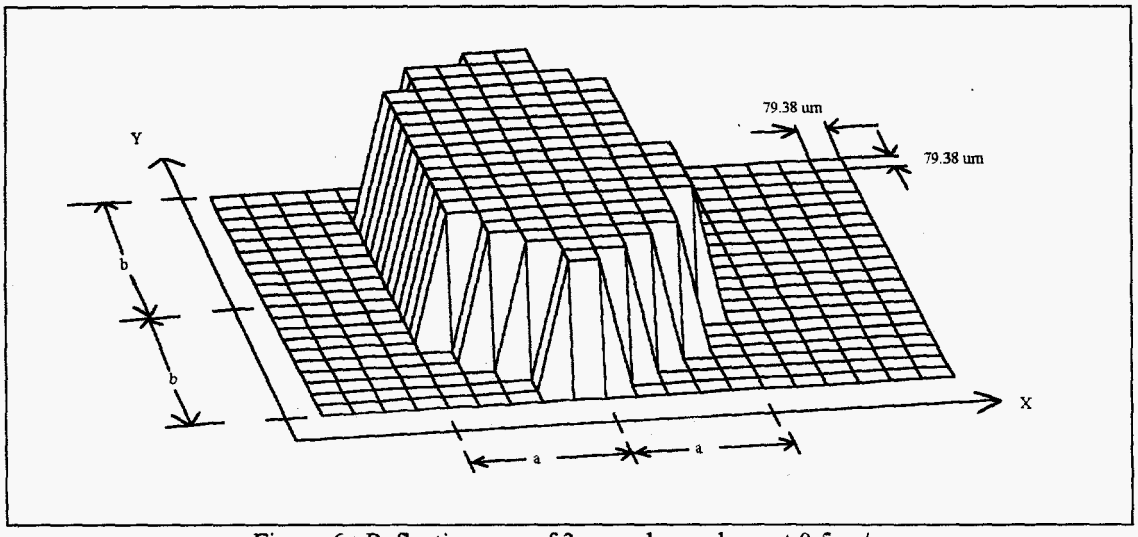

Figure 6. Reflective area of 3 mm glass sphere at 0.5 m/s.

1/4 turn is 1/320 inch or 79.38 μm. Figure 6 shows the measured cross-sectional reflective area of the particle versus the X-Y positioning coordinates. The perimeter of the reflected area resembles an ellipse thus a continuous approximation can be made using the equation $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$. Where a and b are the minor and major axes respectively. The flux area is obtained using the equation for the area of an ellipse, $A=\pi ab$. The mass flow rate per unit area is

$$\begin{aligned}\dot{m} &= \frac{(\#\,particles)(m_p)}{\pi ab} \\ &= \frac{(\#\,particles)(34.6\times 10^{-6})}{\pi(5\cdot 79.38\times 10^{-6})(10\cdot 79.38\times 10^{-6})} \\ &= (\#\,particles)(34.96)\ \frac{\text{kg}}{\text{m}\cdot\text{sec}}\end{aligned} \tag{4}$$

where *#particles* is the number of particles per second and m_p is the mass per particle. The average particle mass, 0.0346 grams, is determined experimentally by weighing 100 particles using a digital scale, then dividing the resultant value by 100.

CHAPTER 3
SYSTEM IMPLEMENTATION

3.1 Signal Conditioning

Figure 7 shows a schematic diagram of the analog signal processing system. There are three functional blocks within this circuit: detection, amplification and analog to digital conversion. The detection block consists of the HBCS-1100 and biasing components C1, R1, R2 and R3. The internal transistor is configured in a common emitter connection. The receiver photo diode, connected to pins 2 and 3, operates in reverse bias. When 720 nm light is detected 4.3 volts is applied to the transistor base causing saturation. Resistor R2 provides negative voltage feedback and along with the collector resistor R3 produces a voltage gain of

$$A_1 = \frac{R2}{R3} \cdot \Delta V_2 = \frac{10^7}{10^5} \cdot (5 - 4.3) = 70 \qquad (5)$$

where ΔV_2 is the change in voltage at pin 2. The output signal at pin 1 is connected to a

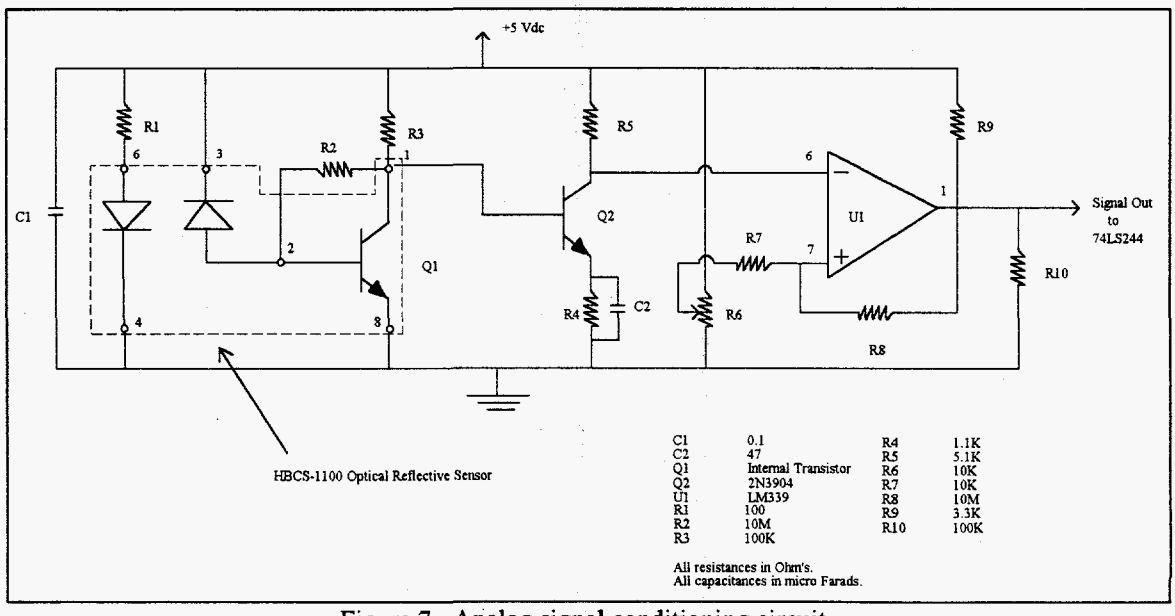

Figure 7. Analog signal conditioning circuit.

3 wire shielded transmission line which increases flexibility in detector head positioning. Capacitor C1 decouples power supply ripple and resistor R1 limits the transmitter forward current to 50 mA. The second functional block consists of a common emitter amplifier which is used to boost the signal voltage gain at the transmission line termination. Resistor R5 biases the collector and R4 biases the emitter base junction. Capacitor C2 coupled with R4 provides low frequency rejection. The voltage gain equations are

$$A_{f1} = \frac{R5}{R4} = \frac{5.1}{1.1} = 4.64 \tag{6}$$

for frequency $f_1 < 31$ Hz. A_{f1} increases linearly and stabilizes at

$$A_{f2} = g_{fe} \cdot R3 = 6.21 \cdot 10^{-3} \cdot 5.1 \cdot 10^{3} \tag{7}$$

for frequency $f_2 > 212$ Hz. The term g_{fe} is the characteristic transconductance of the transistor. The output of the amplifier circuit is connected to the inverting input of an operational amplifier. The OP amp circuit is configured as a Schmidt trigger comparator which outputs a 5 to 0 Vdc pulse when the threshold at the non-inverting input is exceeded by the amplified signal at the inverting input. The threshold level is controlled by the potentiometer R6 through the voltage divider network of R7 and R8. R8 also provides voltage feedback which inhibits unwanted oscillations and introduces a small amount of hysteresis. The hysteresis increases the pulse width slightly which is observed in figure 8 by comparing the voltage levels at the intersection of the leading and falling edges of the analog and digital signals. The resulting TTL signal from the comparator is connected to the input pins of a 74LS244 Octal Tri-State Buffer on the processor board. The buffer, when enabled, allows transfer of digital pulse information to the lower 8 bits of the 68000 micro-processor data bus. When not enabled the output is tri-stated, a standard procedure for bus connection. The threshold level is calibrated for a 12.9 msec. pulse width using 1.2 cm wide mat paper at a velocity of 1.0 m/s. This setting corresponds to

1.4 Vdc at the non-inverting input of the operational amplifier and is required to conform to the shape correction factor in the one pulse method.

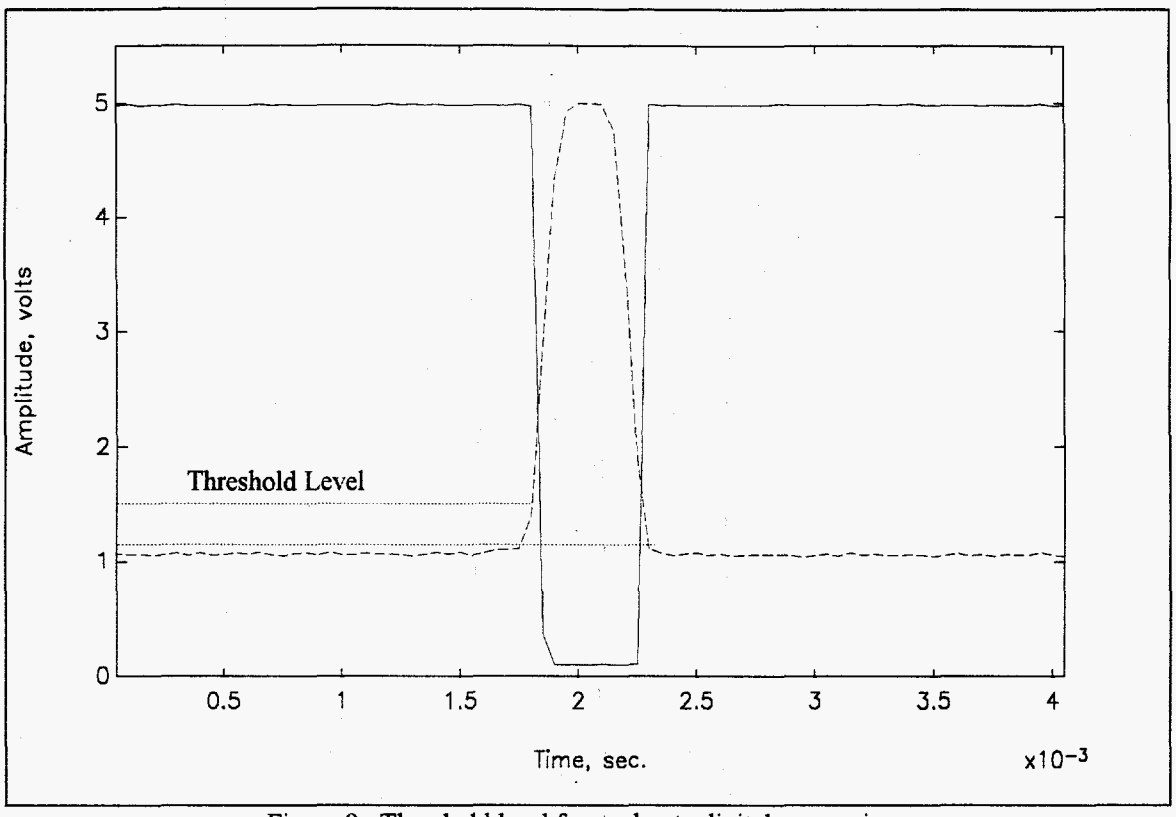

Figure 8. Threshold level for analog to digital conversion.

3.2 Processing Element

A functional block diagram of the velocimetry system is shown in figure 9. The system is composed of two subsystems: the detection subsystem discussed previously and the micro-processor subsystem. The focus of this chapter is the hardware and software

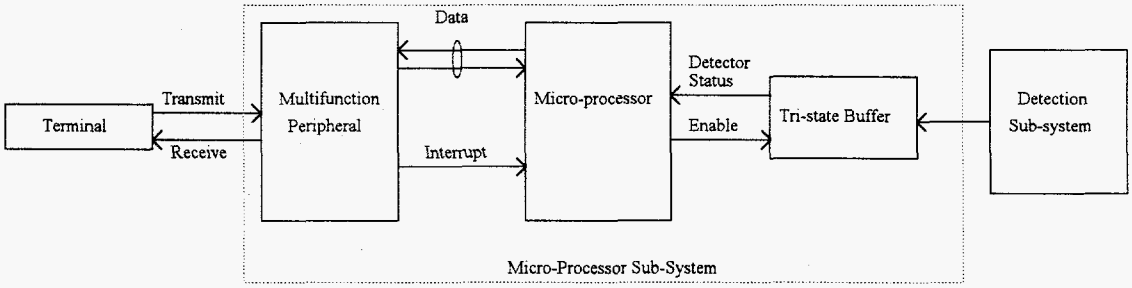

Figure 9. Velocimetry system diagram.

design of a micro-processor sub-system which translates electrical pulses into velocity values. The operation of the micro-processor sub-system is as follows:

- The method of measurement is selected at the terminal.
- An interrupt is sent to the micro-processor periodically to sample the detector port.
- For each interrupt the micro-processor suspends activity, enables the tri-state buffer and reads the detector status.
- The speed or flow rate is determined for each pulse occurrence.
- Results are sent to the terminal.

Commands are issued to the micro-processor through a Universal Asynchronous Receiver Transmitter(UART) in full duplex mode at 9600 bits per second(bps). The operating system interprets the command and transfers control to the corresponding executing program. In the case of a measurement command, the micro-processor programs the multi-function peripheral to generate interrupts every 16.28 μsec. At each interrupt the detector port is read to determine the status of 8 possible channels resulting in an effective sampling rate of 61.44 Khz per channel. Due to the relatively long period of time required for transmission of a single result, sampling occurs in intervals of 5 seconds. During the 5 second sampling interval flow count or velocity values are stored in RAM, at the end of the measurement interval results are downloaded to the terminal for display and/or storage in an ASCII file and another 5 second interval begins. The above process is repeated until the user intervenes by typing CTRL-C at the keyboard. Program control is then returned to the operating system.

3.2.1 Hardware

The schematic of the micro-processor sub-system developed for this application is plotted in figure 10. The sub-system is divided into 4 functional blocks: central processing unit, addressing and control, input/output and memory. The CPU block consists of the

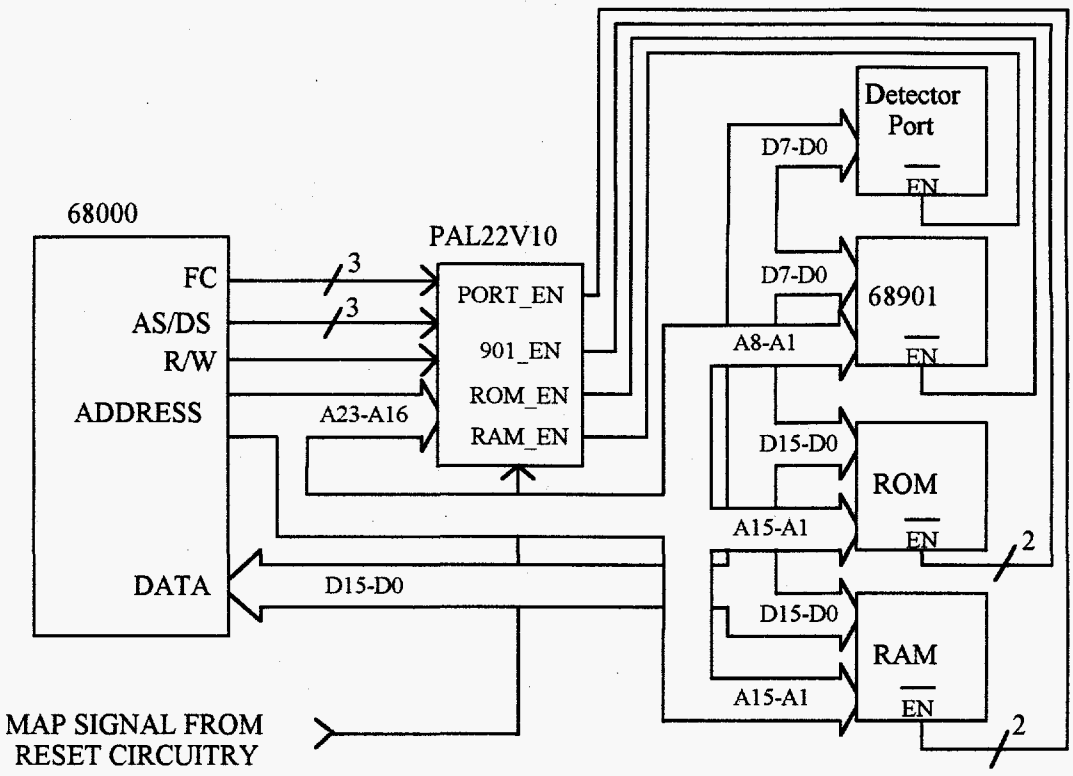

Figure 10. Micro-processor addressing and control.

Motorola 68HC000-16 micro-processor. The 68000 is a 16 bit processor operating at 16 MHz, which corresponds to a 250 *n*sec. machine cycle. It has eight 32-bit internal general purpose registers for data manipulation and eight 32-bit internal address registers which contain pointers to commonly accessed memory locations. The 68000 has a 16 bit data bus and a 23 bit address bus.

The addressing and control functional block is implemented in two PAL22V10's. Figure 10 shows the relationship between address/control and the CPU and memory blocks. The input of this functional block consists of the 68000 address and control lines which are combined into a set of Boolean equations to provide three functions: determine which memory modules may access the data bus, generate wait states in the CPU and indicate bus contention errors. The control signals used are:

- A23-A16 - partitions memory into 32 Kbyte blocks.
- MAP - control line used to relocate the ROM during reset.
- FC - selects supervisor/user data or program space.
- RW - indicates whether the current instruction is a peripheral read or write.
- DS - Data Strobe selects access to upper or lower byte of data.
- AS - indicates a peripheral access.

The logic equations are listed in two program files in appendix B. The file MEM_CS.PDS contains the equations for memory selection and DTKBERR.PDS contains the wait state and bus error equations.

The I/O block contains two devices: the 68901 Multifunction Peripheral and the 74LS244 tri-state buffer. The 68901 provides a 9600 bps full duplex 8 bit data link to a remote terminal and generates periodic interrupts to the CPU. The '244 or detector port provides 8 channels for detector inputs.

The memory block consists of two 32 Kbyte EEPROM's and two 32 Kbyte static RAM chips. The EEPROM's are grouped together to produce a 64 Kbyte program module. Likewise, the RAM is grouped to produce a 64 Kbyte data module. Each module has a 16 bit bandwidth although byte addressing is available.

Figure 11b shows the location of each peripheral in memory at runtime. Note that the 68901 MFP and detector port both occupy 64 Kbytes of logical memory even though the 68901 has 24 physical bytes of memory and the detector port has 1 byte of physical memory. For these peripherals, logical memory is larger than physical memory in order to reduce the number of address lines necessary for peripheral selection. Address line minimization allows one PAL22V10 to accomplish all address decoding. Figure 11a shows the memory map used in the reset sequence. A separate memory map is required for the reset sequence because the 68000 expects boot up information at memory location's 0000h and 0004h. However, during normal operation the interrupt vector

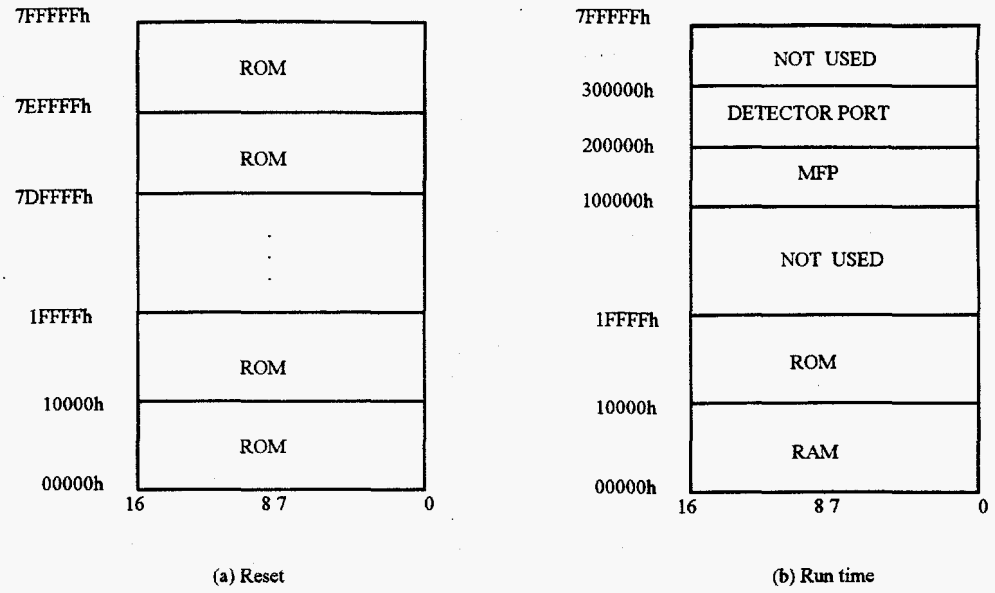

Figure 11. Micro-processor sub-system memory map. a) Reset memory map. b) Runtime memory map.

table's base address is at memory location 0008h. A runtime modifiable vector table is desired to increase system flexibility for future applications. Runtime modification is only possible if the vector table is stored in RAM. Thus, to accommodate reset sequence requirements and interrupt vector table flexibility, system ROM is mapped into all memory locations during reset by an active-low control signal called MAP. After completion of the reset sequence, a dummy read is performed to negate the MAP signal which maps system peripherals to the architecture of figure 11b. Figure 12 shows how mapping is accomplished using the system reset switch and the Read/Write line from the microprocessor. When the RESET switch on the front panel is pushed, the resulting active low signal is debounced and routed to the 68000 reset pin and the CLR input of a 74LS74 flip-flop. The flip-flop is cleared, forcing MAP true, enabling the ROM and disabling all other peripherals through combinational logic implemented in the PAL22V10. The first executable instruction after reset is a dummy read to memory. This instruction forces R/W high, activating the positive edge triggered CLK input of the D flip-flop. The constant true at the flip-flop input is clocked to the non-inverting output forcing MAP false which transforms the memory map to the runtime structure in figure 11b.

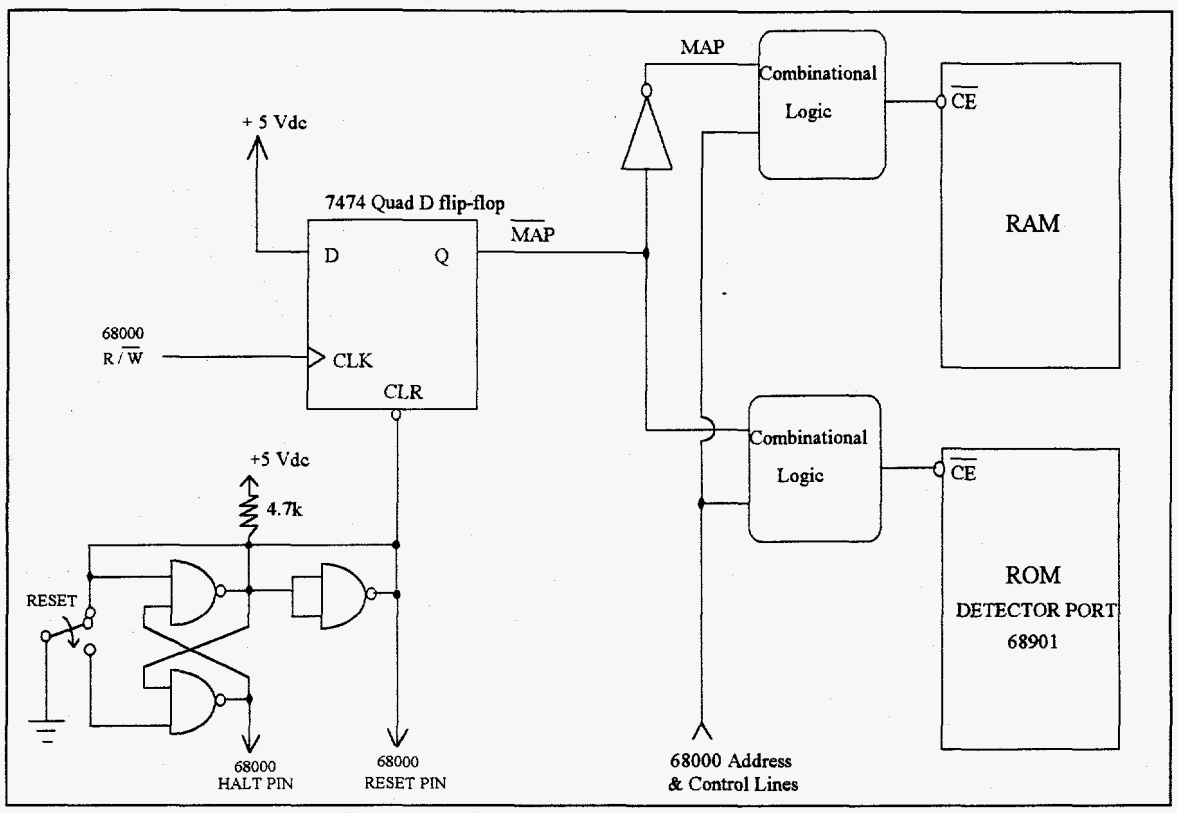

Figure 12. Reset and memory map circuitry.

One of the features of the 68000 is an asynchronous bus cycle. When the microprocessor makes a request to a device on the data bus, wait states are inserted into the machine cycle until the device is ready to proceed with the read or write operation. This feature allows communication with the slower peripheral's compatible with the 6800 series processor. The asynchronous bus operates by requiring all peripherals to generate a Data Acknowledge or DTACK signal to indicate its state of readiness. The 68901, like most Motorola peripherals, supplies its own DTACK, however the RAM, ROM and detector port do not. A Moore state machine is programmed into a second PAL22V10 to generate DTACK for these devices. The maximum setup time for a memory read cycle is 152.5 *n*sec., any device exceeding this time period will generate 125 *n*sec. wait states until DTACK is received. The RAM has an access time of 85 *n*sec. and the detector port has a setup time of 5-10 *n*sec. thus no wait states are required for these devices. The ROM has an access time of 150 *n*sec. which is too close to the machine cycle setup time for fault

free operation. Thus a wait state is introduced during ROM access by delaying the generation of DTACK for 125 *n*sec.

Another feature of the 68000 is bus error recovery. A bus error occurs when two devices try to use the data bus at the same time or addressing is attempted in an unused memory segment. The same state machine which generates DTACK also detects a bus error. The logic equations and state diagram for DTACK and Bus Error are located in appendix B under the filename DTKBERR.PDS.

The 68000 communicates to the outside world through the MC68901 Multi-function Peripheral(MFP) device. One of the features of the 68901 is a Universal Asynchronous Receiver Transmitter which allows full duplex serial communication. Since the '901 transmit and receive pins generate TTL levels, an RS232 driver(MAX232) is used to increase the voltage levels from 5 to 24 volt's peak-to-peak. The increase in voltage level reduces signal sensitivity to noise. The MAX232 transmit and receive pins are connected to a DB25 connector, the other end of which is connected to a communications port on a PC. Presently, the PC-PLUS communication software is used in terminal mode with parameters set at 9600 bps, 8 data bits, no parity, and 1 stop bit. These parameters are generated in the 68901 using an 8 bit timer and the UART control register.

The 68901 has 4 independent programmable timers A, B, C and D which operate at a 2.4576 MHz frequency produced by an external crystal oscillator. This frequency is selected such that when scaled by an integer, the desired communication rate is achieved. Each timer has an 8 bit control and data register. The control register selects the mode of operation and a divide by N pre-scalar where N = 4, 10, 16, 50, 64, 100 and 200. The data register contains the timer values. Timer's A and B operate in delay and pulse width mode while C and D operate in delay mode only. In delay mode, a pre-scalar is selected to generate a count frequency. A value loaded into the data register is decremented every count pulse until zero occurs, at the same moment a time out pulse is generated. The data

register is automatically reloaded allowing the process to continuously repeat itself resulting in a time out pulse frequency.

Two of the 68901 timers are used to generate the transmission rate and a periodic external interrupt. Timer C is used in delay mode to drive the transmit and receive clocks for the serial port. A divide by 4 pre-scalar is selected to produce a 614.4 Khz count frequency and 4 is loaded into the data register to produce a time out pulse frequency of 153.6 Khz. The UART Control Register(UCR) is loaded with 88h which selects a transmission syntax of 8 data bits, 1 stop bit, no parity and a divide-by-16 scalar for the time out pulse frequency producing the 9600 Hz transmit/receive clock frequency. Timer A is used in delay mode to generate periodic external interrupts to the 68000. A divide by ten pre-scalar is selected for a 245.8 Khz count frequency and the data register is loaded with 4 to produce an interrupt(time out pulse) frequency of 61.44 Khz. The timer A output pin is connected to all three interrupt input pins on the 68000 for highest interrupt priority.

The microprocessor sub-system is implemented on a perforated circuit card using chip sockets and wire wrap. A printed circuit card capable of handling four detectors is fabricated for the signal conditioning electronics. The signal conditioning and micro-

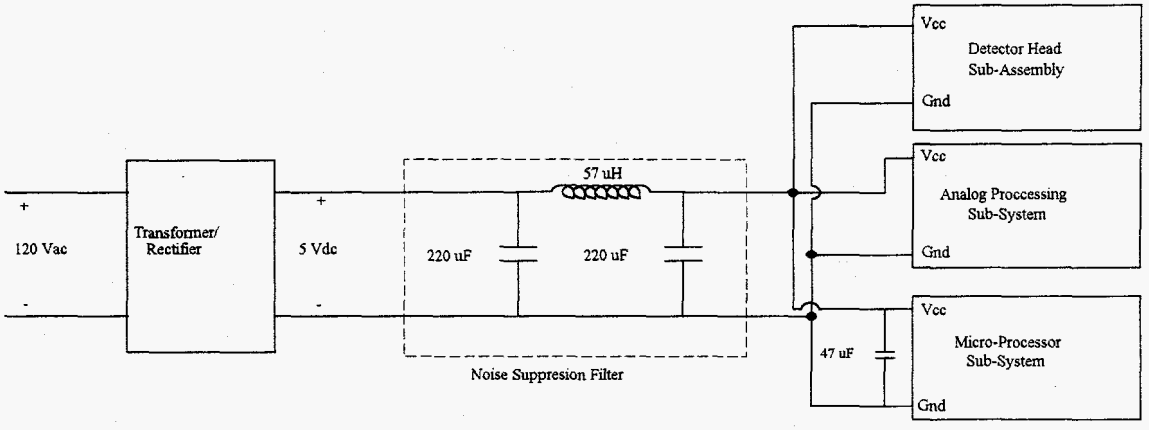

Figure 13. Detection system power supply and noise reduction.

processor sub-systems are mounted in a 3 1/16" x 8 1/4" x 6 1/8" metal box using plastic spacers. Power is supplied to the system by a RACAL-VADIC AC to DC converter. Several features have been added to reduce the amount of noise in the system(see figure 13). A PI filter connected to the +5Vdc and GND terminals on the power supply. This filter shunts to ground high frequency noise introduced into the analog circuitry by the micro-processor sub-system. A 47 μF electrolytic capacitor is placed across the +5Vdc and GND pins of the micro-processor sub-system for decoupling power supply noise. In addition, each chip socket has a 0.1 μF monolithic capacitor across the Vcc and ground pin for noise suppression.

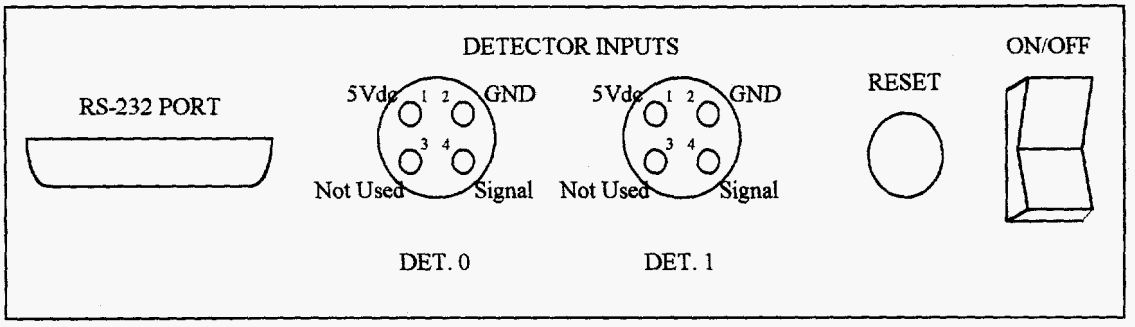

Figure 14. Detection system front panel.

The front panel is shown in figure 14. Features include an on/off switch, a push button reset switch, a DB-25 female RS-232 connector and two male 4 pin connectors for detector coupling. Presently the system is wired for two detectors, however, two more may be added using pin 3 of the detector input connectors as a signal path. The full schematic for the micro processor subsystem is plotted in figure 15.

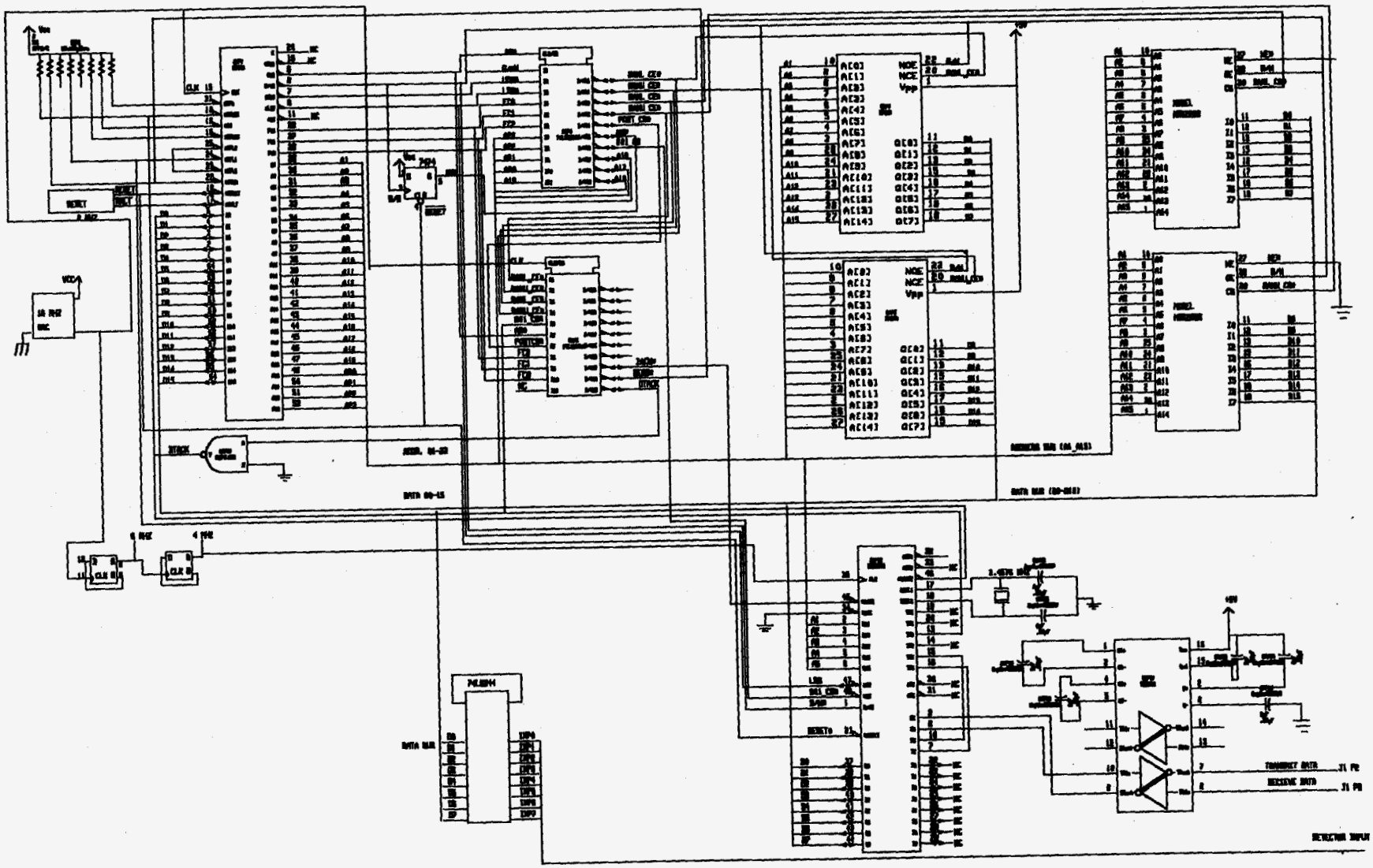

Figure 15. Micro-processor sub-system schematic.

3.2.2 System Software

Figure 16 shows a flow chart diagramming the overall operation of system software(the complete assembly level program is listed under filename VEL.ASM in appendix B). The boot up block is entered after power up or reset. The system stack pointer is loaded and the program counter is initialized with the reset vector. The

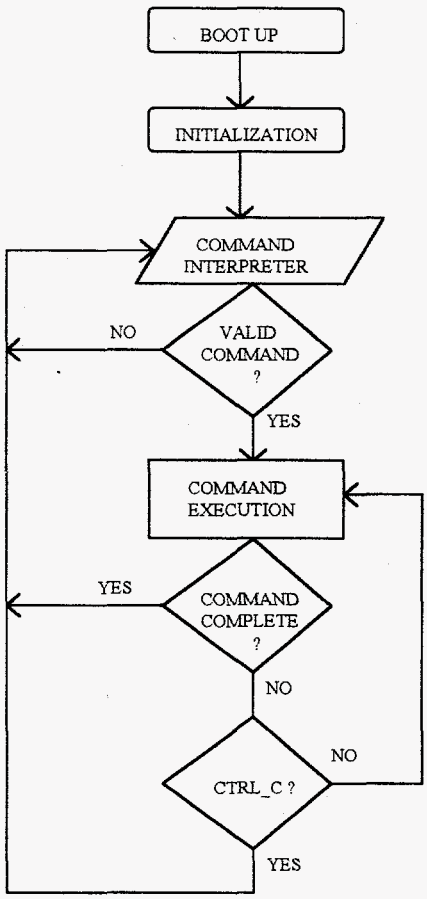

Figure 16. System software flow diagram.

initialization block performs a dummy read, sets up the interrupt vectors and initializes the 68901 for communication. The command interpreter reads upper or lower case ASCII characters from the keyboard, determines the command requested and transfers control to the corresponding execution unit. The execution unit completes the command or loops continuously until user intervention. When the execution cycle is complete program control returns to the command interpreter loop.

User commands are listed in table 1. They are divided into two groups: system and measurement instructions. System commands(H and CLS) execute once and return to the command interpreter. Measurement commands(1P, 2P and CAL) run in continuous

Table 1. System user commands.

COMMAND	FUNCTION
H	Display Help menu.
CLS	Clear Screen and Home Cursor
1P	1 Pulse Method Speed Calculation
2P	2 Pulse Method Velocity Calculation
CAL	Detector Calibration

loops. Flow charts detailing the execution of each command are listed in appendix A. The HELP and CLS functions are self explanatory. The three measurement commands are similar in operation. Inside a continuous loop they: read the detector port status using the same interrupt subroutine(ISR), make time measurements of a detected pulse(s) by capturing values from a single free running counter, calculate speed or velocity and display the results. The only differences between the three commands are the methods of time measurement, velocity calculation and display.

Software measurement commands require the input signal from the detector sub-system to be transformed from the physical to the logical domain, i.e., analog to digital. There are several techniques which can be implemented to accomplish this, the most notable of which is the analog to digital converter. In order to achieve good resolution for pulses produced from glass spheres, a relatively high sampling rate is required. For example, a 440 μsec. pulse is produced by a glass sphere at 1 m/s. Good resolution requires at least 10 sample points to represent the pulse. This corresponds to a sampling rate of 22.73 Khz per channel. If 4 channels are used then a 100 Khz A/D is required to meet the resolution criteria. Chips this speed are available, however, they increase power

consumption and introduce a significant amount of noise into the system. Also, more information is provided than necessary to accomplish the one and two pulse methods. The one and two pulse methods only need to detect signal transitions, information between the transitions is superfluous. In order to increase the efficiency, low power and high sampling rate, another technique is used to perform the transformation from the physical to logical domains. This method begins with the transformation from analog to digital in the signal conditioning circuitry, four channels can be converted with the LM339 comparator without sampling. The resulting logic signal, a 1 or 0, is routed to the microprocessor through a single pin on the 74LS244 Tri-State buffer. The micro-processor samples the buffer during every interrupt cycle producing a detector status byte. Treating each bit of the status byte as an individual flag accomplishes the transformation from a physical to a logical representation. The software determines the flag transitions and measures the time in between.

The interrupt subroutine is shared by all three measurement commands. The ISR samples the detector status and increments the free running timer at a periodic interval of 16.28 μs. This interval is determined experimentally by measuring the execution time of the longest segment of code. Upon interrupt detection, the processor suspends execution of the command main loop, executes the ISR, then resumes execution at the next instruction in the main loop. The ISR reads the detector port to obtain the present status byte. Since a pulse occurrence is active low, the status byte is complemented to obtain status flags before transfer to the least significant byte of data register D7. The free running counter, located in the 32 bit data register D5, is incremented by 1 and compared to $\frac{5}{16.28 \cdot 10^{-6}} = 307200\text{d}$ for 5 second time out. If the time out occurs, bit 16 in D7 is set to flag the 1P and 2P command main loop. When the time out flag is detected, the main loop suspends detector port sampling by disabling the interrupt. The stored velocity and

flow rates values are transmitted to the display. The calibration routine does not check the time out flag.

The one pulse method measures the width of the detected signal pulse by capturing a value from the free running timer at the leading and trailing edges of the pulse. Speed values are determined by using the difference of the timer capture values as an offset into a look up table containing pre-calculated velocities. Results are displayed every 5 seconds. The two pulse method measures the time between the two pulses by capturing a value from the free running timer at the leading edge of the each pulse. Velocity values are determined and results displayed by the same process as the one pulse method. The calibration function measures the velocity of a single 3 mm particle glued to a rotating disc, 10 cm in diameter. The free running timer measures the detection period(time between pulses) and divides this value into the distance traveled, $2\pi \cdot 51.5$ mm. The resulting velocity value is immediately sent to the screen since the time between detection's is much greater than the time required for transmission of the output characters. The 7 character result consists of: 1 character each for tens, ones, tenth's and one-hundredths digits, 1 character for the decimal point, and 2 characters for carriage return and line feed.

The algorithm for pulse detection is basically the same for each measurement command. A status flag mask is always present in the least significant byte of data register D6. The flag mask contains the previous value of the detector port status. Comparison of the present detector status with the flag mask indicates the occurrence of a rising or falling edge at one of the detectors. If no transition is detected, the receive buffer in the 68901 is checked for a CTRL-C. If CTRL-C is in the buffer, the program is terminated and control is returned to the command interpreter. If the termination command is not present, program control is redirected to the beginning of the command loop where the comparison process begins anew. If a transition is detected, the present detector status is compared to the flag mask on a bit by bit basis to determine which detector(s) changed state and whether the transition is a leading or trailing edge. In the calibration routine a leading

edge triggers velocity calculation and a falling edge is ignored. In the two pulse method a leading edge at the first sensor in a detector pair causes a timer capture and increments the detector flow counter. The leading edge at the second detector in the pair causes a timer capture, increments the detector flow counter and accomplishes velocity calculation using the difference between the initial and final timer captures as an offset into a look up table. Falling edges are ignored in the two pulse method. The one pulse method requires both leading and falling edge detection. A leading edge at a detector causes a timer capture and increments the flow counter. A falling edge at the same detector causes another timer capture, the difference between the timer capture values is used as an offset into a look up

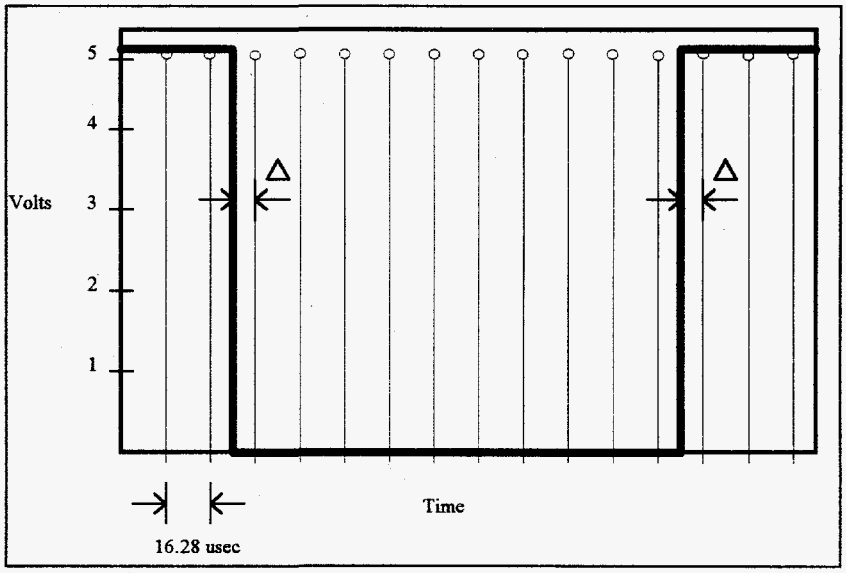

Figure 17. Sampling scheme for one pulse method.

table. A ten sample cut off is imposed on the timer capture difference value to reduce quantization errors. Figure 17 shows the quantization error Δ for a typical pulse. The probability of occurrence of a leading or trailing edge between the sample pulses is approximately equal thus a uniform distribution is assumed for the error. The expected value of the absolute error for a pulse is $E[\Delta] = \frac{16.28 \times 10^{-6}}{2} + \frac{16.28 \times 10^{-6}}{2} = 16.28 \times 10^{-6}$ seconds. The relative error in calculated speed, $\left| \frac{d}{PW} - \frac{d}{PW + PW \cdot E[\Delta]} \right|$, shows the

largest possible difference in speeds due to quantization error. The relative error, shown in figure 18, indicates that quantization error has a minimal effect upon the output speed values if the 10 sample cut off is employed.

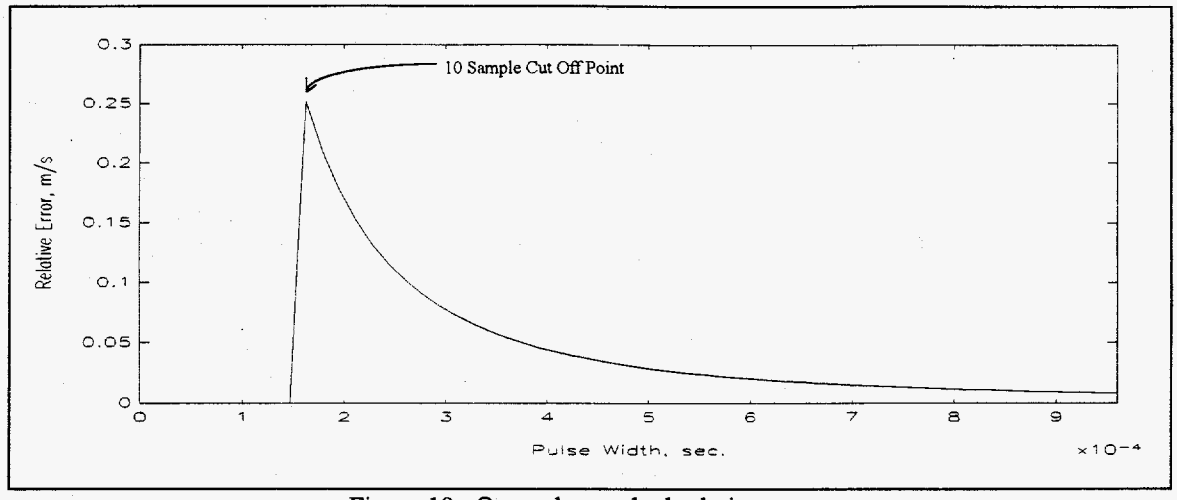

Figure 18. One pulse method relative error.

3.2.2.1 Code Optimization

Several optimizations are employed to increase the range of the system. Due to the relatively short pulse widths produced by glass spheres e.g., a speed of 1 m/s for a 3 mm sphere gives a pulse width of 440 μsec., the range of speeds is dependent upon the speed of the processing system. Since processor speed is fixed, code optimizations must be made in software logic routines to increase the throughput of the system, in particular the sampling rate at the detector port. The sampling rate is dependent upon the interrupt cycle. The frequency of interrupts is dependent upon the length of the interrupt subroutine(interrupt latency) and more importantly, the time required to execute the code. If the ISR is too frequent, the change in input parameters overwhelms the decision logic which causes erroneous values. Ideally, the interrupt period is of sufficient length for the longest contiguous segment of code to fully execute. Thus it is necessary to reduce the execution time of the decision logic and calculation routines to increase the sampling rate.

The first optimization uses one free running timer instead of separate counters for each detector. This reduces interrupt latency by removing the detection logic required for counter incrementation from the ISR. Also reduced is the additional overhead of maintaining more than one counter.

The second optimization employs look up tables rather than use the 68000 divide instruction for velocity calculation. A single velocity value requires 2-3 divide instructions to achieve its final format. The first divide is the velocity calculation described by equations [2] and [3]. Then depending upon the magnitude of the velocity value, 1 or 2 divisions are required to convert from hexadecimal to decimal. A 68000 divide instruction requires 150 machine cycles or 37.5 μs of execution time at 16 MHz. A total of 75-112.5 μs or 5-7 interrupt periods is required for each velocity value. The look up table for the 1 and 2 pulse method has a base address at C000h and extends to DFFFh. A table entry is accessed by adding an integer offset to the base address. The offset corresponds to the time difference between initial and final timer capture values. The table entries contain velocity values that are pre-calculated and converted to decimal. Eight machine cycles or 2 μs of execution time are required to add the offset to the base address and retrieve the table entry from memory. This corresponds to a factor of 50 reduction in runtime speed calculation.

Another time consuming task is sending results to the console. Presently the system is limited to a communication rate of 9600 bps. Since 7 ASCII characters are sent per speed value and each character is 8 bits wide the minimum time required for transmission of one value is 5.833 ms or 358 interrupt periods. During transmission time it is possible for pulses to go undetected resulting in conflicts between the present status word and the flag mask. For this reason, results are stored in a buffer and sent to the console every 5 seconds. Less significant optimizations include the use of the internal data registers for temporary data storage, internal address registers for pointers to frequently used memory locations and program counter relative branching.

The 68000 has several internal error checking features which alert the user of a system failure. The most common errors occurring at runtime are an illegal instruction, an illegal address and a bus error. Upon error detection, the 68000 generates an internal interrupt that vectors program flow to a user defined subroutine. The error handling subroutines perform a register dump for diagnostic purposes, then the current program is terminated and a software reset is performed.

3.2.2.2 Software Tests

Limited development equipment was available for use in debugging the software. Thus all testing was accomplished using an oscilloscope and application specific software. The error checking routines described above were written specifically for debugging. A register dump routine displays all of the data and address registers plus the current value in the program counter and the status register. This routine is different from the register dump in the error checking routines in that a call statement may be inserted into any section of code as a software test probe. Another invaluable feature is the 68000 trace exception. When the trace bit in the 68000 status register is set, a internal interrupt is executed. The trace ISR written for software testing performs a register dump then queries the programmer for single step operation or a return to the application program. This routine is useful for tracing the proper execution of code, however, it becomes ineffective when an ISR is active. Another diagnostic function is the 68000 CHK or check exception. This feature compares a specified value in an address or data register and generates an internal interrupt if the specified value is not present. The interrupt handler for the CHK function is application specific thus internal registers or memory locations may be displayed.

CHAPTER 4
SYSTEM TEST AND EVALUATION

4.1 Diagnostic Tools

The first problem encountered is forcing a glass sphere through the focal point of a detector at a known speed. Two methods are used to accomplish this. The first method is the suspension of a sphere at the end of a pendulum. The detector pair is positioned such that the focal points intersect the sphere at the lowest point of the pendulum trajectory. A compass is attached to the pendulum apparatus in order to measure the pendulum angle at the peak swing. The pendulum is started in motion from a known angle and subsequent oscillations are video taped. Slow motion play-back of the video allows angle measurements at the peak swing. Velocity is derived from angle measurements by the following equation

$$V_{\max} = r \cdot \dot{\theta}_{\max} = r \cdot \theta_o \cdot \omega \qquad (8)$$

where ω is the angular frequency, θ_0 is the initial angle and r is the length of the pendulum from pivot to end point(8.86 mm). The angular frequency is a function of the pendulum rod dimensions and is found to be

$$\omega = \sqrt{\frac{mgL_{cm}}{I}} = 13.6 \text{ sec}^{-1} \qquad (9)$$

where L_{cm} is the distance from the pivot to the center of mass of the rod and I is the moment of inertia. Insertion of ω and r into (9) gives

$$V_{\max} = 1.21 \cdot \theta_o \text{ m/s} \qquad (10)$$

As the pendulum passes back and forth through the detector focal spots the resulting pulse information from the detector pair is sampled at 20 Khz and stored in a data file using the Global Lab data acquisition system. The C program ANALYZ.CPP, listed in appendix B, determines the pulse width and time between pulses. The results are linked with the initial

angle information and processed using MATLAB. Developing results with this technique is a rather long, tedious process and is not recommended for extensive testing.

A better diagnostic tool is a rotating disc. A plastic disk 5 cm in radius is attached to the rotor of a DC motor mounted in an aluminum brace. One or more 3 mm glass spheres are glued to the edge of the disc. The detector pair is positioned such that the focal beam(s) intersect the sphere(s) as they rotate on the disc. The motor is controlled by a variable DC source which allows a range of velocities from 0.25 to over 10.0 m/s. Figure 19a shows plots of the velocity of a single sphere versus motor input voltage. The solid line represents the results produced by the CAL software routine, see figure A2 in appendix A for functional description, and the dashed line represents velocities developed using an oscilloscope to measure time differences. Figure 19b shows the calibration error.

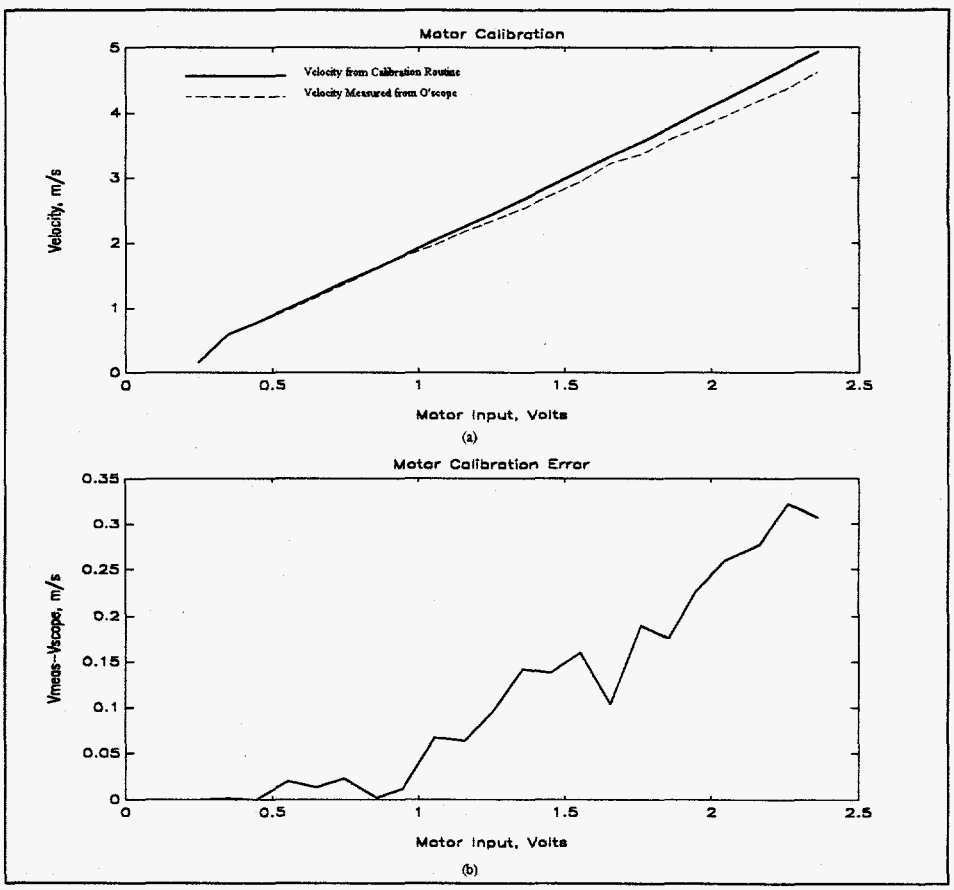

Figure 19. Motor calibration. a) Calibration velocity and measured scope velocity vs. input voltage. b) Error between calibration and measured scope velocities.

This system produces very accurate velocities below 2 m/s. Between 2 and 5 m/s the calibration velocities tend to increase because deformation of the plastic disc causes the detector focal spots to move away from the point of maximum reflection. Since the range of the detection system is typically less than 2 m/s, the rotating disc is the preferred diagnostic tool and is used in all phases of testing.

4.2 One Pulse Method Calibration

The first step in testing the one pulse method is finding the shape factor which represents the reflective area of a glass sphere. Experimentation has shown that a good approximation to the reflective area is 0.1 mm, about half the diameter of the detector focal spot. Several speed measurements are taken at increasing intervals using an assumed shape factor of 0.1 mm. These speed values are plotted along with the calibration speed in figure 20. The line traced by the measured velocity is fitted to the calibration curve by

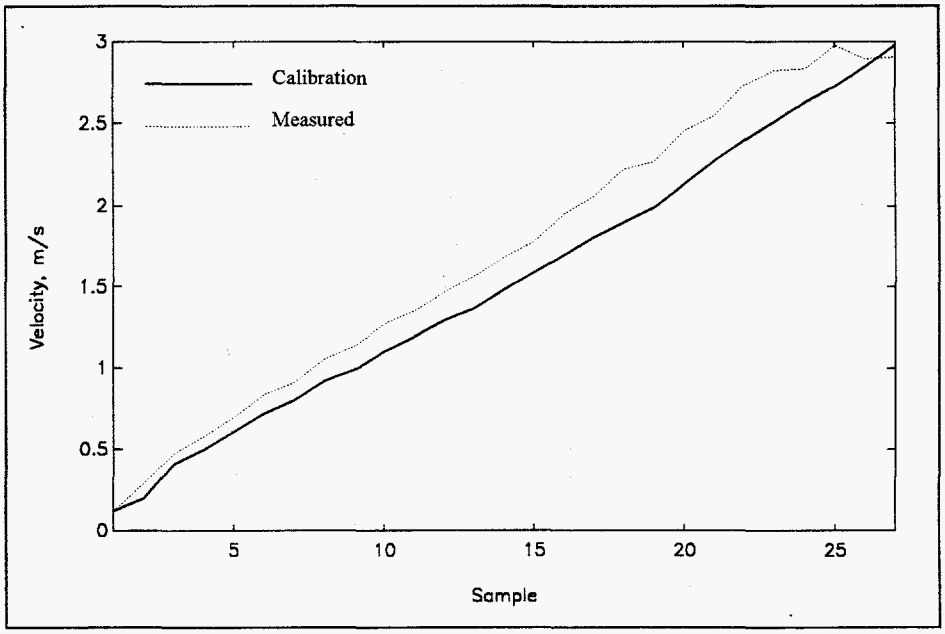

Figure 20. One pulse method results versus calibration velocity.

developing a new velocity equation W using a least squares approximation in MATLAB

$$W(v) = 0.9 \cdot v + 6 \tag{11}$$

where v is the measured data in figure 20. Equation 11 is used to modify the pre-calculated velocity values before storage in the one pulse method look up table. The system is tested again and the results are plotted in figure 21. Note that some error is still

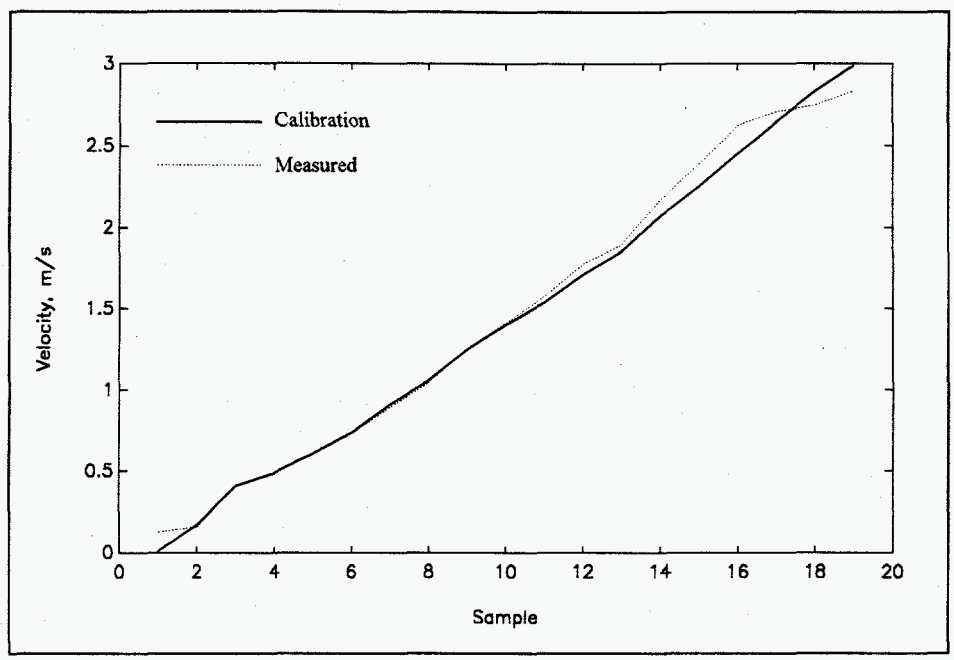

Figure 21. One pulse method results using curve fitting approximation.

encountered above 1.5 m/s. This error is due to the quantization effects discussed previously and may be reduced by performing a piece wise linear approximation. Since the highest speed expected in the chute is less than 1.5 m/s this technique is not performed.

4.2.1 Particle Flow Models

The experiments conducted to this point show that the speed of glass spheres can accurately be determined in a controlled environment in which the detector head is carefully positioned and the glass spheres travel in one trajectory at a constant, given speed. In the chute, particle trajectory and speed are not known apriori. As stated in chapter 2, the one pulse method does not accurately measure individual particle speeds. Thus in order to extract useful information from the measurements produced in a chute

flow, it is necessary to create a model which corrects for the errors inherent in this method.

The most difficult problem to be addressed is particle trajectory. Figure 4 in chapter 2 shows the variation in pulse width versus detector focal point position. A particular particle speed corresponds to a different pulse width. Thus, due to particle trajectory, a distribution of speeds is expected even though each particle travels at the same velocity. The relationship between calculated speed and particle trajectory is measured using the same process described in chapter 2.4 for mapping the reflective area. The results are shown in figure 22 for a fixed particle speed of 0.5 m/s. An alternative interpretation can be made if one considers the results in figure 22 to be the response of a linear system in the velocity domain to an input which is periodic at 0.5 m/s. This interpretation allows the use of system theory to model measured velocities.

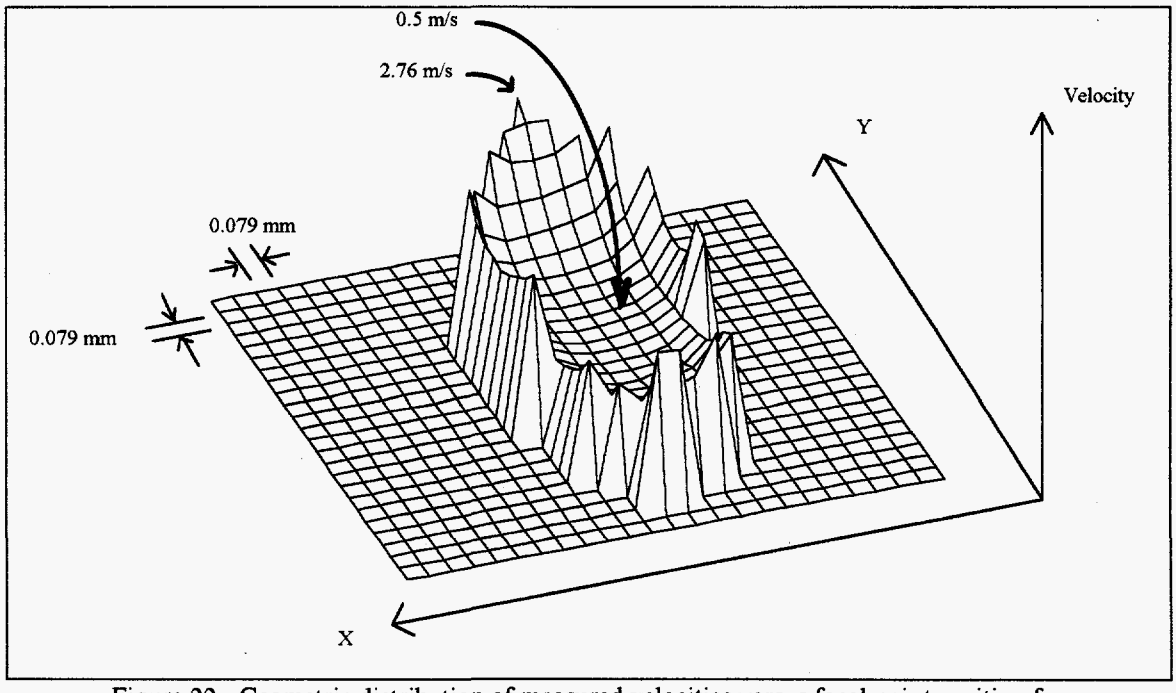

Figure 22. Geometric distribution of measured velocities versus focal point position for true velocity of 0.5 m/s.

Three methods are applied to correct for the distortion of velocities due to particle trajectory. They are: Baye's theorem, inverse system analysis and a statistical model

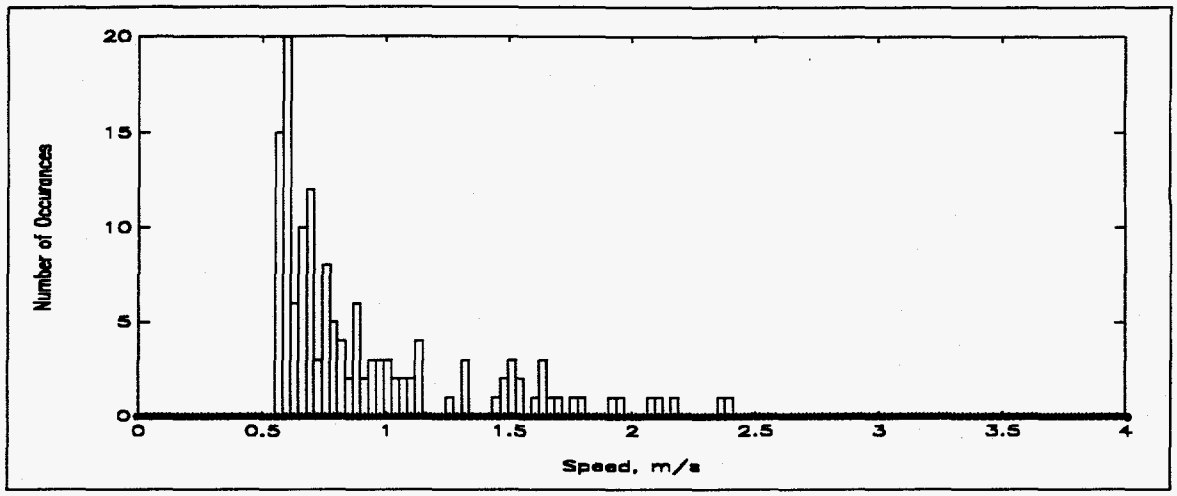

Figure 23. Statistical distribution of measured velocities for a true velocity of 0.5 m/s.

called the correlation of probabilities. The plot in figure 22 shows qualitatively the problem of velocity distortion, however, in order to perform a quantitative analysis it is necessary to transform the geometrical representation of figure 22 to a statistical representation. This is accomplished by separating the measured values from figure 22 into the 128 point histogram shown in figure 23. A question arises as to whether or not the distribution of figure 23 is continuous over the velocity range of the device. To answer this question, velocity distortion is measured for true velocities of 0.5, 0.75, 1.00, 1.25, 1.50, 1.75 and 2.00 m/s. The results, plotted in figure 24, show the distribution gradually changing over the velocity range tested. The gradual change is due mostly to decreased resolution in the upper velocity range. This accounts for the large spikes at the end of the 1.75 m/s and 2.00 m/s distributions. Another reason for the changing distributions is the 10 sample cut off which effectively truncates the distributions at 2.76 m/s. Consequently, less points are recorded in the upper velocity range than in the lower velocity range. Although the assumption of a continuos distribution across the velocity

range is shown here to be invalid it will be seen in chapter 4 that the function in figure 23 provides a good approximation to the response of the system.

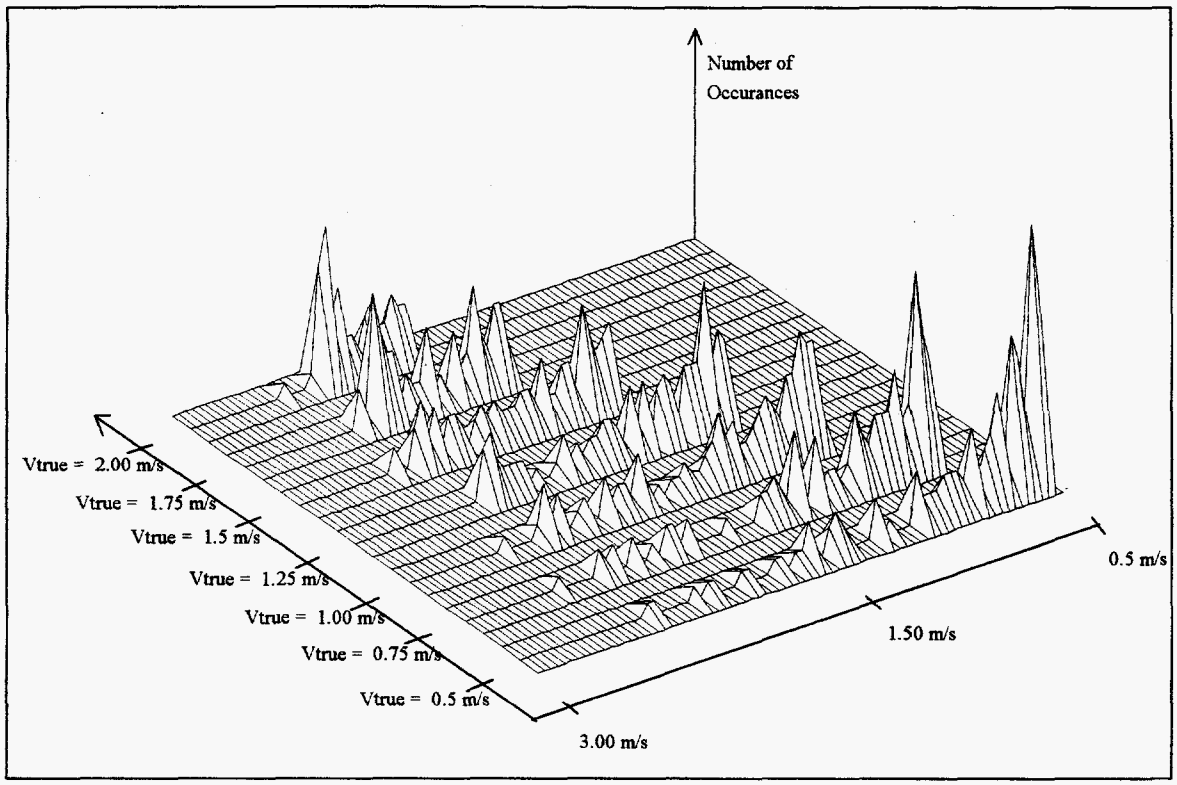

Figure 24. Distribution function or system response at different input velocities.

4.2.1.1 Bayes Theroem Analysis

Baye's theorem deals with conditional probability and is expressed as

$$P[Vtrue_k|Vmeas] = \frac{P[Vmeas|Vtrue_k]P[Vtrue_k]}{\sum_i P[Vmeas|Vtrue_i]P[Vtrue_i]} \qquad (12)$$

Where P[A|B] reads as the probability of A given B. If a uniform distribution of speeds is assumed apriori then $P[Vtrue_K]$ is equal to $P[Vtrue_i]$ and can be eliminated from equation 13 resulting in

$$P[Vtrue_k|Vmeas] = \frac{P[Vmeas|Vtrue_k]}{\sum_i P[Vmeas|Vtrue_i]} \qquad (13)$$

The file BAYES.M in appendix B contains the code to implement Baye's algorithm in MATLAB. Briefly, a measured velocity is pulled from the data set and equation 13 is performed for each distribution in figure 24. The maximum value for $P[Vtrue_k|Vmeas]$ corresponds to the most probable true velocity and is stored in a vector having an entry for each distribution. The process is repeated for each measured velocity in the data set. The final output is a corrected distribution displayed as a histogram. The expected value of the histogram is the average velocity.

4.2.1.2 Inverse System Analysis

This model uses an inverse system to counteract the distortion of velocities. The idea of using an inverse system in this application is to find a function which when convolved with the distribution of speed measured with the one pulse method will produce an impulse function at the true velocity. This method interprets the function in figure 23 as the system impulse response in the velocity domain and assumes that it is continuous over the range 0 to 3 m/s. Figure 25 shows the algorithm by which the inverse system is derived. Figure 26 shows the intermediate results from each step. The algorithm is as follows:

Step 1. The system response (B) is measured given a constant input (A) of 0.5 m/s.

Step 2. A FFT is performed of the system response (B) to produce (C).

Step 3. The point by point inverse, $\frac{1}{C}$, of the FFT of the system response is performed to produce (D).

Step 4. An inverse FFT of (D) is performed to produce the inverse system (G).

The MATLAB program file INVSYS.M, listed in appendix B, implements the inverse system. The input to INVSYS is a data file containing the measured velocities produced with the one pulse method. A 128 point histogram produced from the input is

convolved with the inverse system function (E). A search routine finds the location on the velocity axis of the maximum value of the function resulting from the convolution. This value is the true velocity.

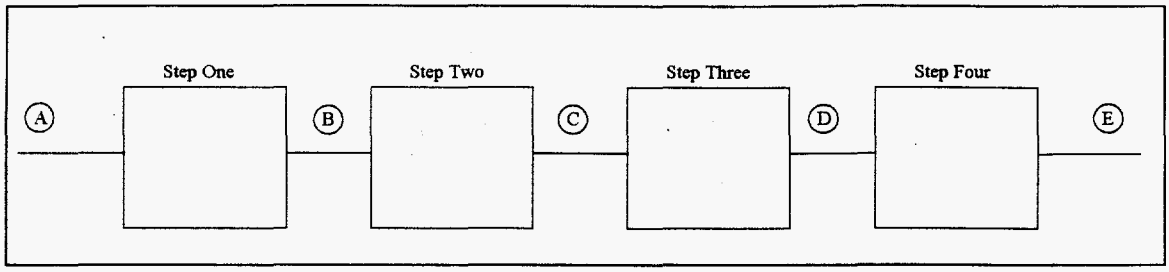

Figure 25. Functional diagram for inverse system algorithm.

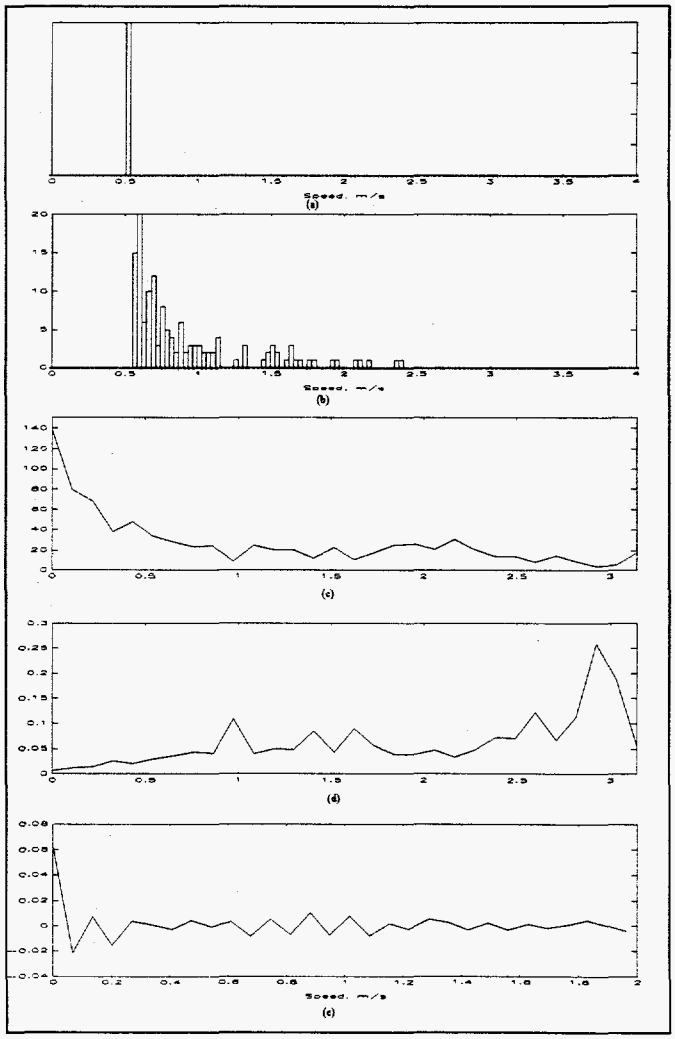

Figure 26. Inverse system intermediate results. (a) Impulse function in velocity domain, (b) Impulse response, (c) FFT of impulse response, (d) Inverse system of (c), (e) IFFT of inverse system.

4.2.1.3 Correlation of Probabilities

The last model is called correlation of probabilities. The plot in figure 23 is a function of the measured speed(Vmeas). Normalization by the true speed(Vtrue) and conversion to a probability density results in a new dimensionless function $P(\tilde{V})$ which describes the probability of a measured velocity being the true velocity. The following equation is used for normalization

$$\tilde{V} = \frac{Vmeas}{Vtrue} \tag{14}$$

Figure 27a shows a plot of $P(\tilde{V})$ along with a continuous approximation termed $\hat{P}(\tilde{V})$.

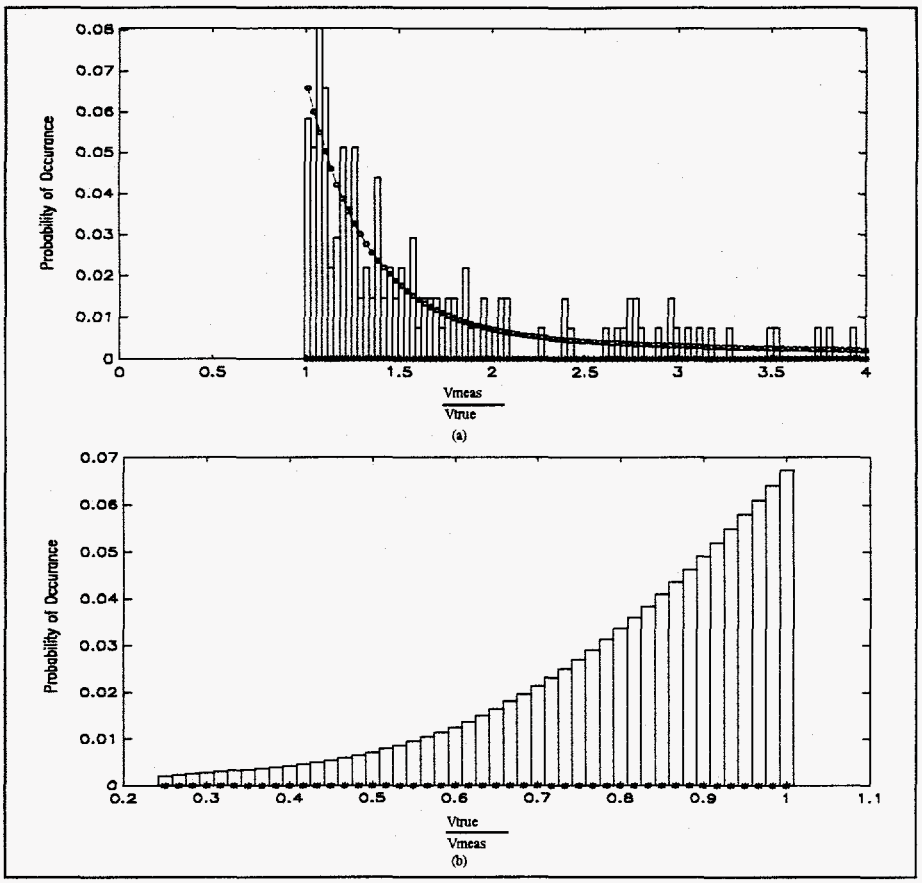

Figure 27. Correlation of probabilities method.. (a) Discrete density and continuous approximation. (b) Discrete function produced by inversion of axes.

The continuous approximation smoothes out the gaps resulting from the discrete positioning method discussed in chapter 2.4. The file FITEXP.M in appendix B is a

MATLAB program which utilizes the least squares method to fit a sum of two exponentials to a discrete function. The continuous approximation equation produced from FITEXP is

$$\hat{P}(\tilde{V}) = (0.01 \cdot e^{-0.397 \cdot \tilde{V}} + 1.431 \cdot e^{-3.163 \cdot \tilde{V}})(u(\tilde{V}-1) - u(\tilde{V}-4)) \qquad (15)$$

where $u(\tilde{V})$ is the unit step function. It is instructive to consider what a measured speed value represents. A single measured value can either be the true speed or a distorted version of the true speed. In the latter case, the measured value will always be greater than the true speed. Consequently, true speed can be written as a fraction of the measured value. This point of view suggests a $\frac{1}{v}$ transformation of equation 15 to determine the probability of a measured speed being a distorted version of a smaller valued true speed. The transformation is represented by the following equation

$$\hat{P}(\tfrac{1}{\tilde{V}}) = (0.01 \cdot e^{-0.397 \cdot \frac{1}{\tilde{V}}} + 1.431 \cdot e^{-3.163 \cdot \frac{1}{\tilde{V}}})(u(\tfrac{1}{\tilde{V}}-1) - u(\tfrac{1}{\tilde{V}}-4)) \qquad (16)$$

Figure 27b shows a plot of equation 16 evaluated at 46 intervals between 0.25 and 1 or $\tilde{V} = 1.0, \ldots, 4.0$. The intervals are chosen to satisfy the discrete density condition $\sum \hat{P}(\frac{1}{\tilde{V}}) = 1$. Equation 16 is used to determine the probability of speed distortion in the following manner: A measured speed from the data set is multiplied by the x-axis array of figure 27b.

$$\begin{aligned} V^*_i &= Vmeas \cdot \frac{1}{\tilde{V}} = Vmeas \cdot (0.25, \ldots, 1.0) \\ &= (\frac{Vmeas}{4}, \ldots, \frac{\text{Vmeas}}{1}) \end{aligned} \qquad (17)$$

The values of $\hat{P}(\frac{1}{V})_i$ have a one to one correspondence to V^*_i resulting in a vector which contains the probability of 46 possible values of Vtrue being a fraction of Vmeas. The resulting vector is added to a 128 point output array which ranges from 0 to 4 m/s. The

process is repeated for each value in the data set. The output array is a histogram which displays the corrected distribution. This algorithm is implemented in the MATLAB program file CORRP.M in appendix B.

4.2.2 Maximum Energy

Another technique used for distortion correction is maximum energy. The measured data set(output of the detection system) is separated into 128 equally spaced points over the velocity range 0 to 4.0 m/s. The resulting function is processed by a moving average filter with an 8 point window and squared. The true velocity is located at the largest concentration of energy. The file MAF.M in appendix B implements the preceding algorithm in MATLAB. This method may also be applied to the result of the inverse system analysis.

4.2.3 Model Test and Evaluation

All of the above methods are tested using the system response data shown in figure 24. The results are shown in table 2. The minimum speed of the data set is included for comparison. The inverse system provides the best estimation to the true velocities less than 1.5 m/s. Above 1.5 m/s the correlation of probabilities method gives good results.

Table 2. Test results for one pulse velocity method estimation schemes. All values are in meters per second.

method	Vtrue = 0.5	Vtrue = 0.75	Vtrue = 1.0	Vtrue = 1.25	Vtrue = 1.5	Vtrue = 1.75	Vtrue = 2.0
Min. of data set	0.5514	0.8109	1.080	1.2653	1.5365	1.7519	1.9657
Mean	0.9279	1.2241	1.5237	1.6925	1.9457	2.108	2.232
Bayes	0.9663	1.2477	1.5658	1.7853	1.937	1.9271	2.00
MAF	0.6929	0.9449	1.2283	1.3858	1.6693	2.2677	2.2677
Inv. Sys.	0.5654	0.8796	1.0994	1.3507	1.5392	2.3874	2.356
Energy of Inv. Sys.	0.6911	0.9424	1.1937	1.3822	1.6649	2.2617	2.2617
Corr. of Prob.	0.7532	0.9943	1.2378	1.3749	1.5806	1.7121	1.8128

All of the methods give approximations greater than the true velocity. This is to be expected since the minimum of the data set is closest to the true velocity, all other values are above this.

4.3 Two Pulse Method Calibration

The two pulse method produces speed measurements with greater accuracy and higher range than the one pulse method. Figure 28 shows experimental results displaying the accuracy and range of the two pulse method. In this experiment, the calibration velocity is determined by measuring the time difference between pulses from an oscilloscope. The small error encountered is due mostly to visual resolution of the oscilloscope display.

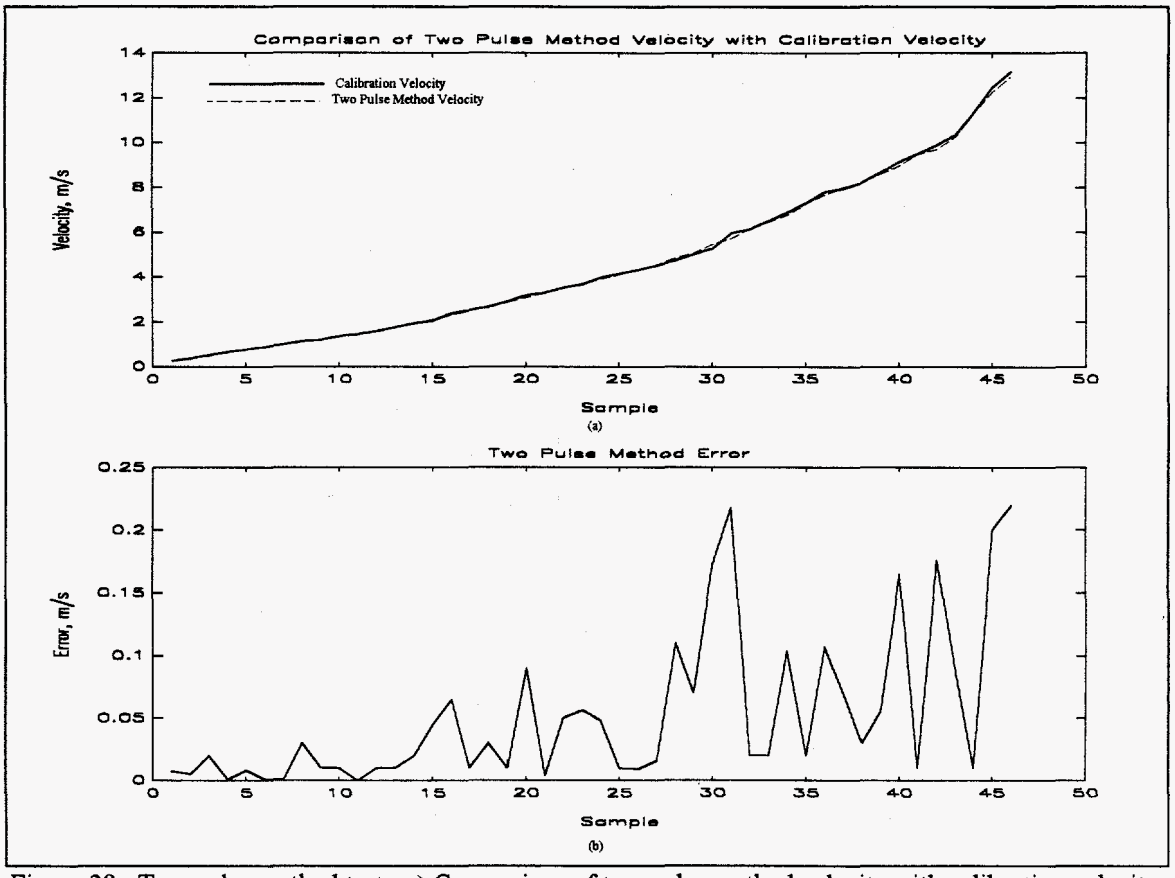

Figure 28. Two pulse method test. a) Comparison of two pulse method velocity with calibration velocity. b) Two pulse method error.

The one limitation of the two pulse method is the detector separation distance. When the size of the particle is smaller than the separation distance, tracking a particle from one detector to the next becomes difficult. The simplest example which details the logic complexity is a system consisting of a black box with a single input and output. Figure 29 shows the possible combinations of detection events which can occur for a set

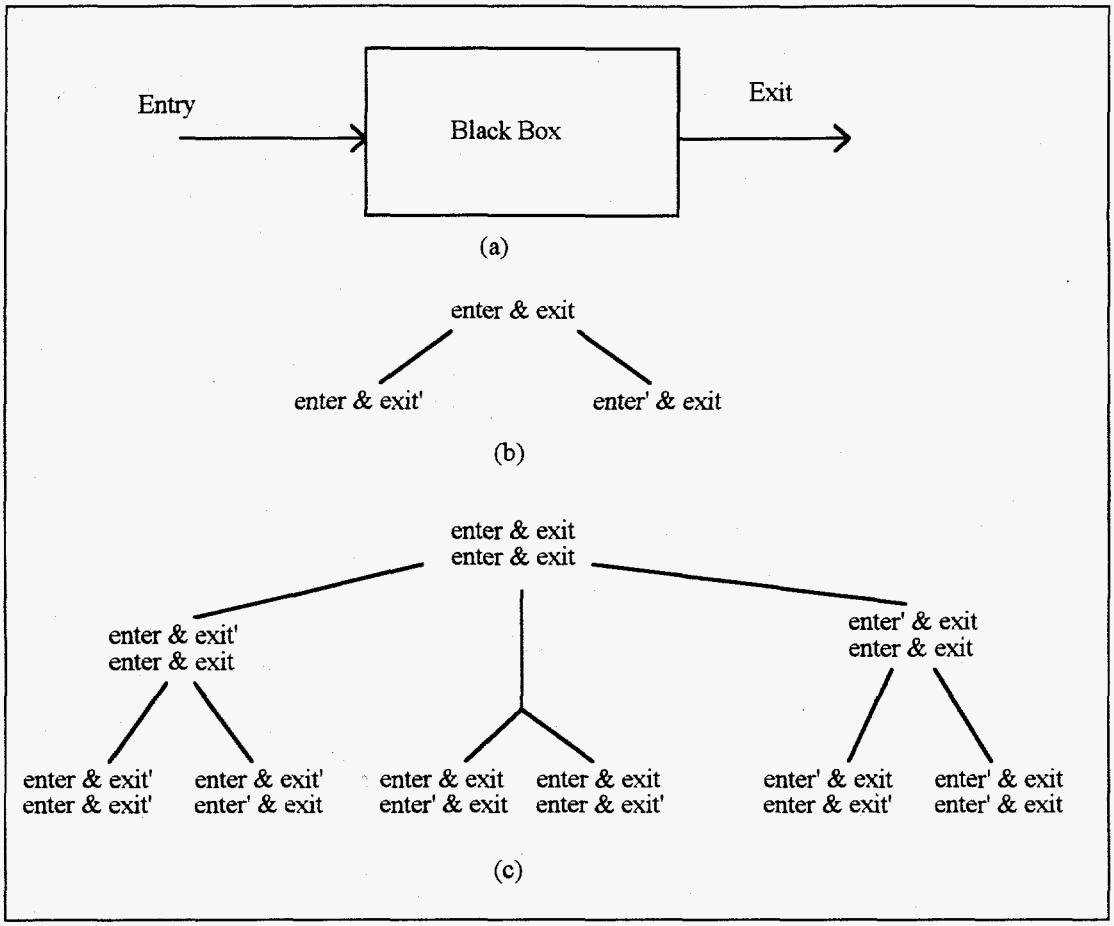

Figure 29. Logic complexity of two pulse method. (a) Black box system. (b) 1 diameter separation distance. (c) 2 diameter separation distance

of objects serially entering and exiting the black box. Figure 29b lists the possible events for the detection of at least one object and figure 29c lists the possible events given two detections. The & symbol represents the logical AND operation, X represents a true event and X' represents a false event. Each node in the tree represents a possible event, the number of nodes follows the progression 3^d where d is the number of detections. The black box is the simplest possible test case because the system is limited to one direction

for particle entry and exit. The two pulse system allows for multiple directions which increases the logic complexity exponentially.

In general tracking errors occur when a particle passes through one only detector then sometime later another particle passes through the other or both detectors. If there is only one occurrence per detector the time difference will usually be relatively large thus a maximum time difference is imposed. If there are 2 occurrences at one detector and 1 at the other, timer values will mis-align and erroneous values are produced. The alignment error propagates into future detections creating a permanent skew. An attempt to correct for these situations is made by sequentially storing the timer values into a memory buffer and implementing a search algorithm over the buffer. Recall that a rising edge from detector zero causes the current value of the free running timer to be transferred to memory. Timer values are stored in a memory address pointed to by the Lead Buffer Pointer. A rising edge at detector one uses another pointer, termed the Trail Buffer Pointer, to sequentially access past timer values stored in the buffer. The past timer values are subtracted from the present and compared to some preset value, such as the maximum value in the velocity look-up table. If the result exceeds the range, the next value is pulled from the buffer and the process is repeated until an acceptable result is achieved or the two pointers are equal, which ever occurs first.

Three experiments are performed to test the two pulse correction algorithm. Two beads are attached to the disc which is adjusted for a speed of 1 m/s. A detector pair is mounted in an XYZ positioner on the optical bench and is positioned to select the number of pulses present at each detector. The first experiment is the ideal case i.e., the same number of detection's at each sensor. The second experiment consists of one detection at detector zero and two detection's at detector one. The third experiment consists of two detection's at detector zero and one detection at detector one. The time series and results for each experiment are plotted in figure 30. A relationship exists between the number of

occurrences at each detector and the accuracy of the resulting velocity values. The relationship is:

CASE 1: # of occurrences at Det. 0 = # of occurrences at Det 1 => speed is accurate
CASE 2: # of occurrences at Det. 0 > # of occurrences at Det 1 => speed is probable
CASE 3: # of occurrences at Det. 0 < # of occurrences at Det 1 => speed is inaccurate

Where the number of occurrences is determined from the flow rate information generated every 5 seconds.

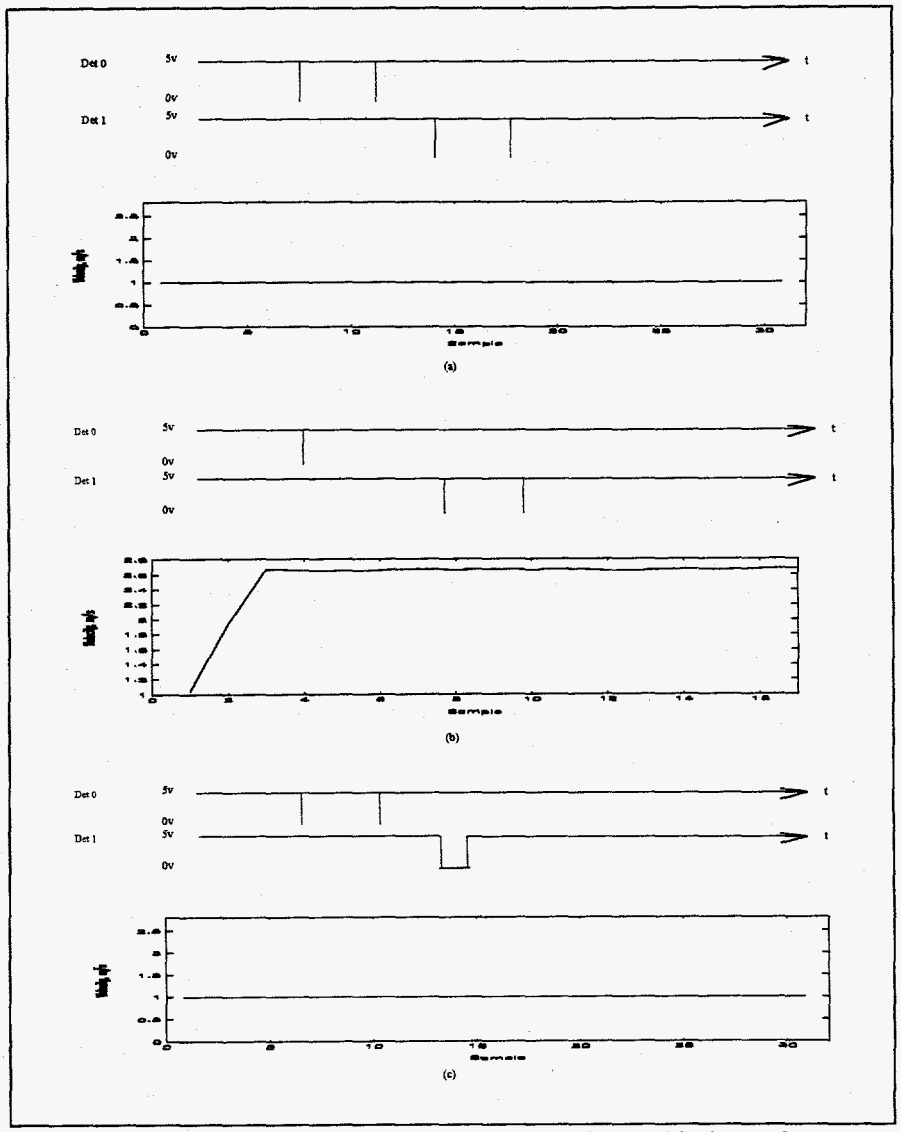

Figure 30. Two pulse method tracking error correction. a) ideal case. b) case 1. c) case 2.

4.4 Laboratory Tests and Results

Thus far the developed techniques have been tested in an environment in which the detector can be precisely adjusted for maximum pulse width. Also, the speed and number of particles is known apriori. None of these properties are available in flows generated in the chute. Two experiments are conducted in the chute to test the detection system. Each experiment models a particular flow profile: dense and moderately dense. Both profiles are achieved at a chute angle of 20 degree's by altering the height of the gate opening to change entry conditions. The gate heights selected are: fully open for the dense flow and one inch for the moderately dense. The detector housing is positioned at the midpoint of the chute, approximately one inch away from the side wall. The detector beams are perpendicular to the direction of flow with detector zero nearer to the chute entrance and detector one located 9.5 mm towards the chute termination. This is not the intended placement for the housing, eventually it will be mounted in a side wall or the chute bed to allow measurement of different flow strata. Thus, these two experiments measure only surface particles. A mechanical speed and mass flow rate measurement are conducted for each profile for comparison.

Figures 31 and 32 show the resulting distributions for the one pulse speed measurement method. Speed values for each technique are listed in table 3. The mechanical velocity, measured mean and Baye's algorithm produce results similar to each other. The inverse system does not produce valid results because the system function is sensitive to large spikes in the upper velocity range. For example, in the 1 inch gate height experiment, the maximum peak detected by the inverse system is at 1.4136 m/s which is above the reference value and away from the main energy concentration. The inverse system energy, the energy of the measured distribution and the correlation of probabilities method results are similar in shape and speed value estimation. The difference between the distributions produced by these techniques and that of the Baye's method is the

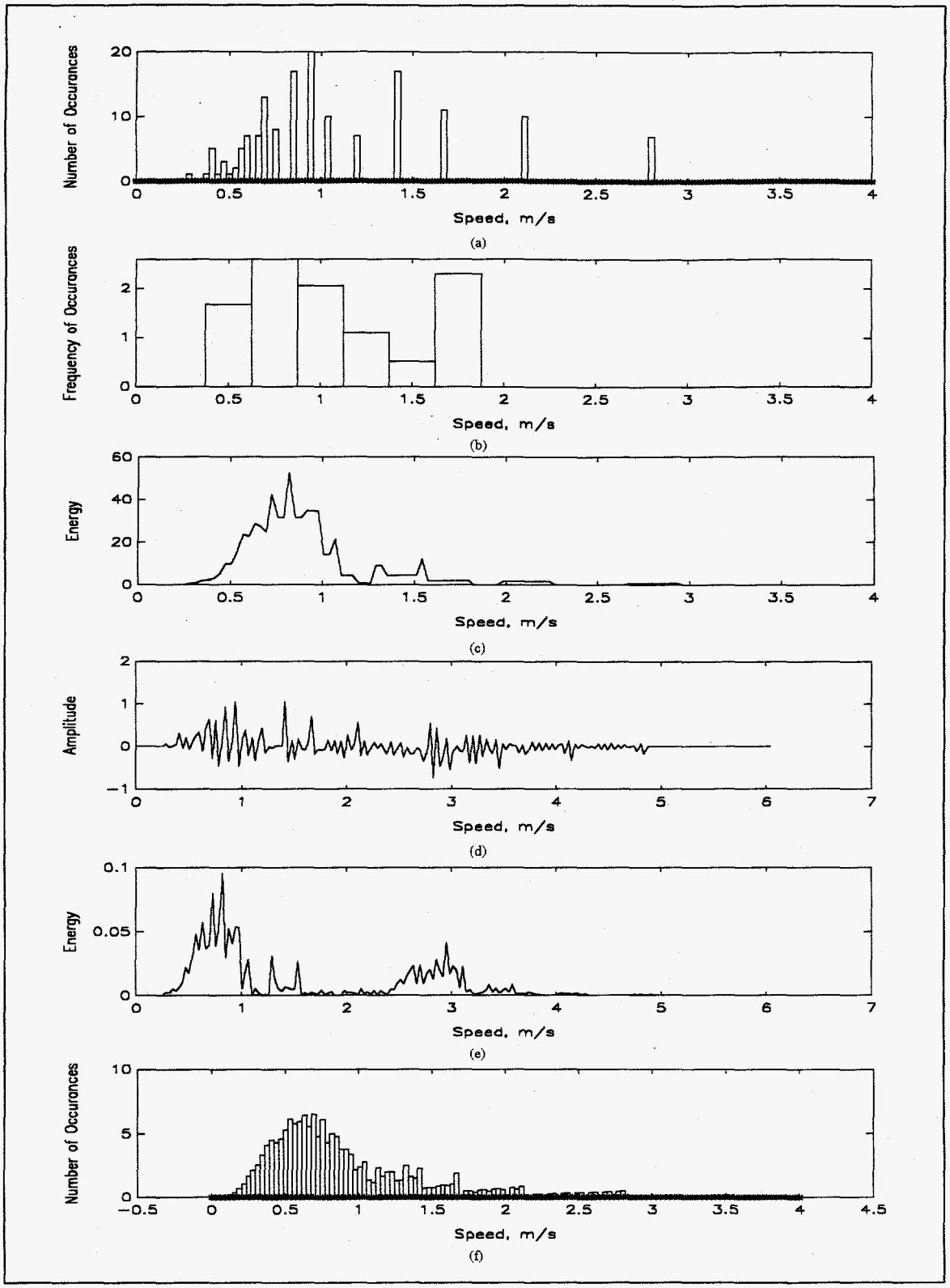

Figure 31. One pulse method results for gate height of 1 inch. Mechanically estimated speed is 1.14 m/s. (a) distribution of measured velocities. (b) Baye's method. (c) Maximum energy method. (d) Inverse system method. (e) maximum energy of inverse system method. (f) Correlation of probabilities method.

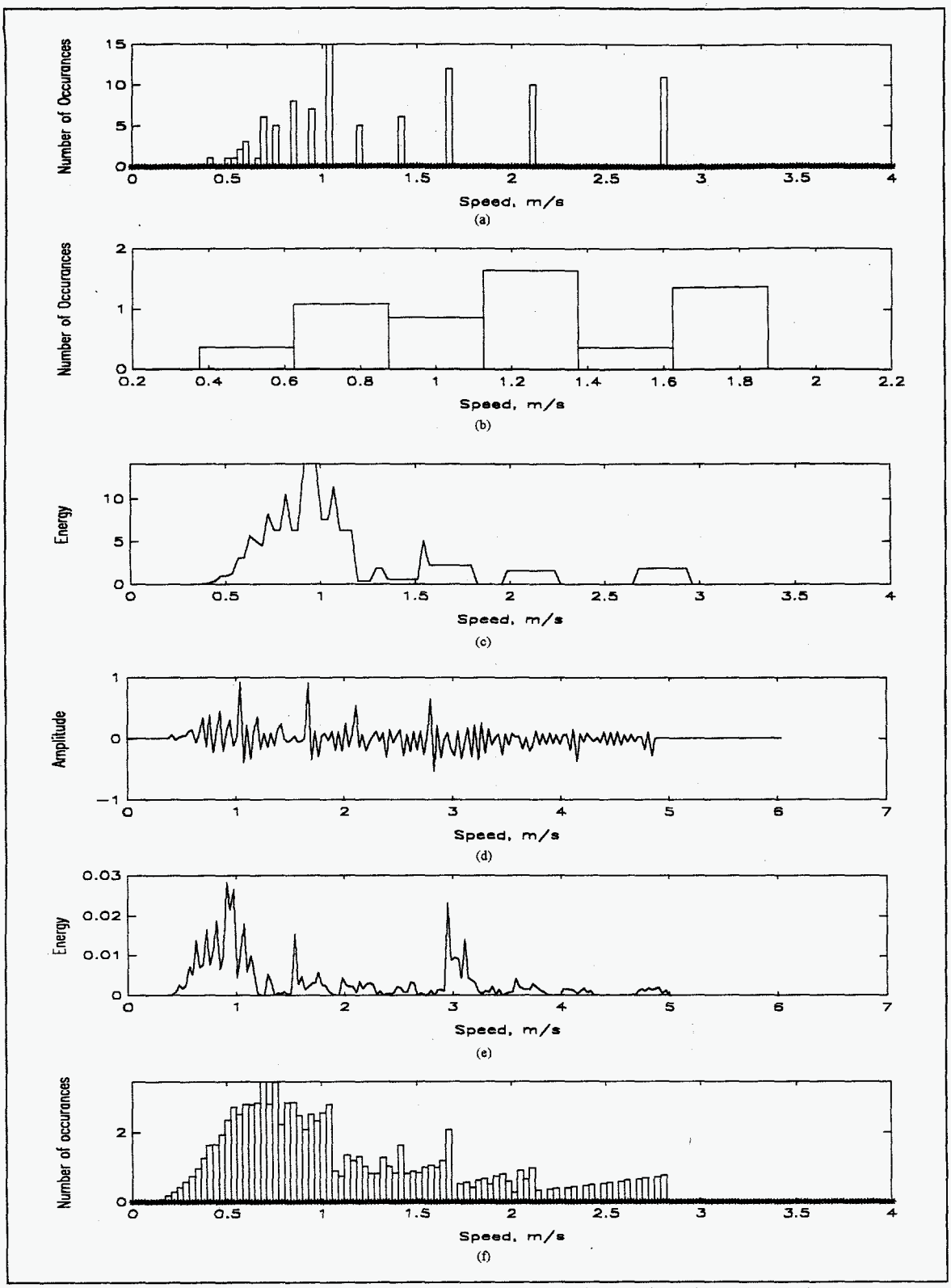

Figure 32. One pulse method results for gate fully open. Mechanically estimated sped is 1.38 m/s. (a) distribution of measured velocities. (b) Baye's method. (c) Maximum energy method. (d) Inverse system method. (e) maximum energy of inverse system method. (f) Correlation of probabilities method.

Table 3. Results for one pulse velocity speed determination for moderately dense and dense flows.

Method of Speed Estimate	Gate Height of 1 inch	Gate Fully Open
Mechanical Speed	1.135 m/s	1.377 m/s
Mean of Measured Speed	1.0969 m/s	1.3662 m/s
Mean of Baye's Distribution	1.0753 m/s	1.2053 m/s
Maximum Energy Point	0.8189 m/s	0.9449 m/s
Maximum Point of Inverse System	1.4136 m/s	1.0366 m/s
Maximum Point of Energy of Inverse System	0.8167 m/s	0.911 m/s
Mean of Distribution from Correlation of Probabilities	0.8913 m/s	1.112 m/s

reduction of higher velocities produced by distortion. The suppression of higher speeds effectively reduces the average. The natural flow speed distribution is unknown. However, in fully developed flows, such as the two measured, a shape symmetric about the mean is expected because of the high number of collisions. None of these distributions exhibit this behavior near the reference values.

Results for the two pulse method are shown in figure 33 for the dense flow profile. Only one measurement out of the 75 taken is near the mechanically estimated velocity of 1.377m/s. Clearly, the two pulse method is not functioning properly. This is due to the separation distance between detectors. The error correction scheme has trouble

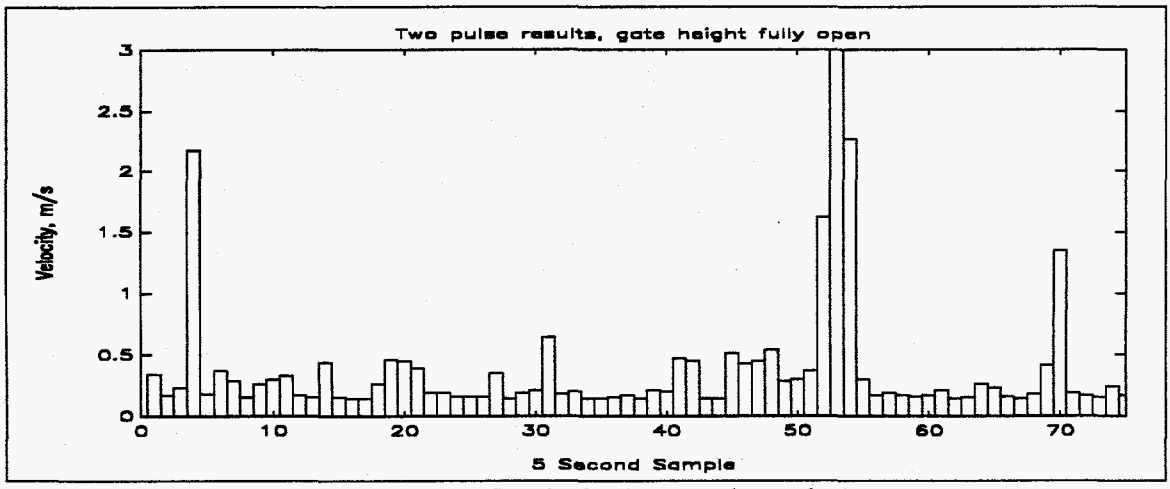

Figure 33. Test Results for the two pulse method.

synchronizing in dense flows because at system start-up: detection's occur immediately at both sensors and one or possibly two particles are located between the detector focal points. The combination of these two events increases the difficulty of buffer pointer alignment. Several methods are attempted to achieve synchronization. The first method off-sets the trail pointer from the lead pointer by 2 and 3 addresses, allowing for the occurrence for particles between the sensors. The search routine will cause the trail pointer to catch up and synchronize to the lead pointer. The second is a time out for the beginning of calculations. This allows the lead pointer to store a number of values before the trail pointer is enabled. Again the search routine will cause the trail pointer to catch up and synchronize to the lead pointer. These methods were attempted separately and combined. In neither case did the pointers synchronize. The same results are produced for the 1 inch gate height thus they are not shown.

The mass flow rate per unit area depends upon the number of particles detected within the five second sampling period. Figure 34 shows the number of particles detected versus time for both flow profiles. The mass flow rate is calculated by averaging the number of particles per second over 130 seconds and then plugging this value into equation 7. The results are listed in table 4. The measured flow rate results are not as close as expected to the mechanically measured reference. There are several possible

Table 4. Mass flow rate results.

	Mech. Mass Flow Rate	Measured Mass Flow Rate
Gate Fully Open	1078.3 kg / m / s	109.1 kg / m / s
Gate Height is 1 inch	249.5 kg / m / s	49.9 kg / m / s

reasons for this. The most probable cause is the placement of the detector. If an undulating flow occurs and the detector is too far away then only the peaks are detected. Also, the detector is positioned near the side wall where the flow is less strong than in the

middle of the chute. Further experimentation is required to understand the difference in mass flow rate values.

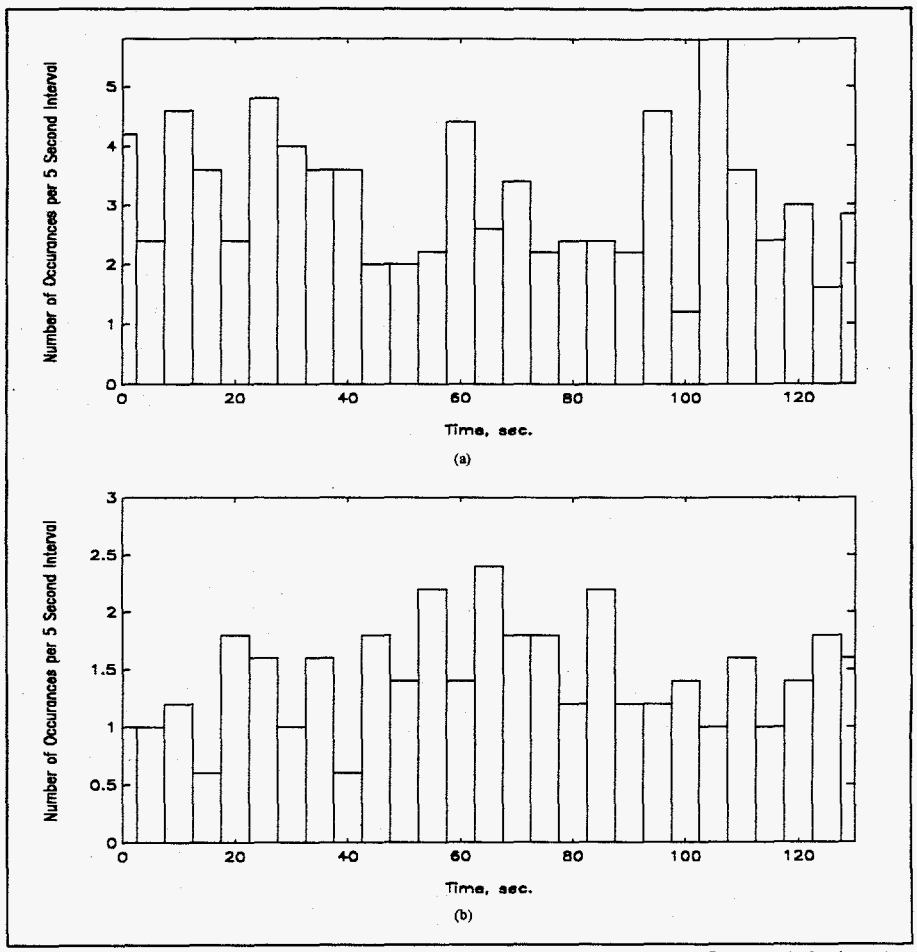

Figure 34. Experimental flow rate results. a) Gate fully open. b) Gate height is 1 inch.

CHAPTER 5
DISCUSSION OF RESULTS, CONCLUSIONS

The two pulse speed method works very well in a controlled environment but functions poorly in the chute. This is due to the distance between the focal points being 3 times greater than the particle diameter. It is desirable to overcome this problem because the two pulse method can measure individual particles with a high degree of accuracy, has a wide range, is not dependent upon particle geometry and is simpler to implement. There are two options in solving this problem. The first is post processing the data using a exhaustive search and match algorithm. The other is reduction of the focal point distance using fiber optic sensors. The first option is similar to performing a cross-correlation on the signals produced by each detector and is not in the scheme of a real time processing device. At the time of this writing, a new fiber optic reflective sensor(E32-DC200B) is available from the OMRON company. The diameter of the detector head is 1.5 mm two detector heads may be positioned such that the focal point distance will be less than 3 mm. The focal length of each sensor is 5 mm which is slightly greater than the HBCS-1100. The OMRON E3XA-CC4A photoelectric amplifier is used in conjunction with the fiber optic head to produce analog and digital outputs. This solution is recommended because little modification is required for assimilation into the processor unit and the sensor system is cost effective when considering the amount of development time required for redesign.

The one pulse method provides good speed estimations. Conclusive results cannot be determined about the natural distribution of speeds as no evidence is available at this time for comparison. Of all the distortion correction schemes, the Baye's algorithm proves to be the best in regards to average velocity determination. The maximum energy, inverse system energy and correlation of probability techniques provide results similar to

each other, although at lower values from the reference. Further experimentation which will include a working two pulse system will determine the utility of these techniques as analysis tools. The inverse system method did not provide good results for one pulse analysis due to its sensitivity to large spikes in the upper end of the velocity range. It is probable that increasing the number of data points will reduce the sensitivity to high velocities. Further experimentation is required to obtain conclusive results.

The mass flow rate measurement results did not coincide with the mechanically produced reference. Further experimentation is required with the detector head mounted in a different position, such as the side wall. This placement should minimize any effects due to undulating flows, however, it is expected that the mass flow rate measurements will still be less than the mechanically measured average due to side wall effects. Another possible system configuration is the use of the OMRON fiber head as a dedicated probe for flow rate analysis. The probe can be mounted in different positions inside the chute and would have an application specific program to determine flow rates. Another feature can be added to this program which will simultaneously determine mean time between detection's. The HP sensors are not recommended for this approach because the size of the detector head and housing have shown tendencies to interfere with saltating particles. A smaller diameter head reduces the probability of this occurrence.

The processing system is adequate for this application however, too much processor time is spent on detection and counter maintenance. Ideally, a separate circuit for pulse width measurement and event counting would be used to free the processor for other functions such as real time transmission of output. A peripheral such as the Intel 8251 provides three 16 bit timers which function in event count and pulse width modes, however, in order to maintain the present system velocity range, at least a 24 bit timer is required. Thus one i8251 is necessary for each detector which increases the chip count and power consumption for a multi-detector device. Another option is the Motorola 68332 micro-controller. This device includes a separate Timer Processor Unit(TPU)

which has 16 channels for timing events. Each channel supports the event count and pulse width modes. The unique feature of this device is that the TPU releases the CPU from all timing tasks. The CPU is interrupted only for a change in status. Timer values and program data are passed back and forth between the CPU core and the TPU through shared RAM. Another feature of the 68332 is the inclusion of peripheral addressing circuitry which eliminates the need for both PAL22V10's. Also, a UART is included which is capable of communication at 19.2 Kbit/s. This feature eliminates the 68901 and reduces the detection sampling interval. The 68332 like the 68000 operates at 16 Mhz, however, the 68332 has 3 machine cycles per instruction rather than 4 which results in a 25% increase in execution time. Motorola sells a relatively inexpensive evaluation board having the same functional components as the processing element designed for this system. This system is the processing element of choice.

The detection system presented in this thesis is a prototype design. Presently, it is good platform for evaluating sensor heads and testing analysis techniques. However, enhancements are necessary to accomplish the goal of accurately measuring the speed of individual particles. The two most important improvements are: reducing the detector separation distance to improve the functionality of the two pulse method and improving the processor unit to increase the velocity range thus sharpening the resolution of the system output. These improvements can be accomplished using fiber optic probes and a separate timer processor unit such as the one in the 68332 micro-controller.

GLOSSARY

68HC000	- Hybrid CMOS 16 MHz 16-bit micro-processor
74LS244	- Octal tri-state buffer
AS	- 68000 Address Strobe
ASCII	- American Standard Code for Information Interchange
BPS	- bits per second
CMOS	- Complimentary Metal Oxide Semiconductor
CPU	- Central Processor Unit
DS	- 68000 Data Strobe
DTACK	- 68000 Data Acknowledge
EEPROM	- Electrically Erasable Programmable Read Only Memory
FC	- 68000 Function Codes
GND	- Signal Ground
I/O	- Input/Output
ISR	- Interrupt Subroutine
MAX232	- RS-232 line driver
MC68901	- CMOS Multi-function peripheral
MFP	- Multifunction Peripheral
PAL22V10	- Programmable Array Logic; 22 inputs and 10 outputs
RAM	- Random Access Memory
ROM	- Read Only Memory
RW	- 68000 Read Write
TTL	- Transistor Transistor Logic
UART	- Universal Asynchronous Receiver Transmitter
UCR	- 68901 USART Control Register