Fall 1986

A Frequency Domain Digital Signal Processor for Laser Velocimeter Systems

Andreas Evangelos Savvakis
Old Dominion University

A FREQUENCY DOMAIN DIGITAL SIGNAL PROCESSOR

FOR LASER VELOCIMETER SYSTEMS

By

Andreas Evangelos Savvakis
B.S.E.E. May 1984, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF ENGINEERING

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY

November, 1986

Approved by:

Sharad V. Kanetkar, Director

John W. Stoughton

Stephen A. Zahorian

A FREQUENCY DOMAIN DIGITAL SIGNAL PROCESSOR

FOR LASER VELOCIMETER SYSTEMS

By

Andreas Evangelos Savvakis
B.S.E.E. May 1984, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF ENGINEERING

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY

November, 1986

ABSTRACT

A FREQUENCY DOMAIN DIGITAL SIGNAL PROCESSOR

FOR LASER VELOCIMETER SYSTEMS

Andreas Evangelos Savvakis
Old Dominion University, 1986
Director: Dr. Sharad V. Kanetkar

A new signal processor for laser velocimeter systems is proposed. The proposed processor incorporates automatic initialization, real time operation, and can efficiently process input signals comprised of as low as 150 photons, with mean oscillation frequency up to 100 MHZ, and input turbulence from 0 to 20%. A bank of digital bandpass filters is employed for the energy spectrum estimation of the input signal. A deterministic model is developed to describe the relationship between the filter output energies and the input signal parameters. The input frequency is estimated by linearly weighting the filter output energies. A spline function approximation approach is used to determine the coefficients that minimize the mean squared error. The same approach is used to develop

an error model that evaluates the processor performance. Simulation results demonstrated that the proposed processor measures mean input frequencies with less than 0.5% average error, and provides a minimum measure of turbulence between 0.2 and 0.5%. Compared to other currently existing systems the frequency domain processor is found to be superior in almost all cases.

To Dora, Vangelis, and Sophia.

ACKNOWLEDGEMENT

I wish to thank my advisor, Dr. Sharad V. Kanetkar, for his guidance and support throughout the course of this research. I would also like to thank Dr. John W. Stoughton, the principal investigator for this project and member of my thesis committee, for the time and effort that he invested in many stimulating discussions. I wish to express my appreciation to Dr. Stephen A. Zahorian, the third member of my thesis committee, for making available the digital filter design software and the computing facilities of the speech laboratory.

I would also like to thank Mr. James F. Meyers for the insight that he provided me on laser velocimetry, and for his help on parts of the simulation program. I acknowledge the financial support by the National Aeronautics and Space Administration under contract NAS1-17993-1.

Finally I express my gratitude to my fellow graduate students for their support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

SYMBOL	DEFINITION
f_D	Beat frequency
v	Particle velocity
λ	Laser light wavelength
θ	Angle between laser beams
β	Angle between particle motion and normal to laser beam bisector
F_1	Pedestal amplitude
α	Pedestal decay exponent
F_2	Visibility ratio
ϕ	Phase
f	Signal oscillation frequency
K	High frequency component amplitude
f_o	Variable oscillator frequency
f_i	Intermediate frequency
f_r	Spectrum analyzer range
x	Normalized frequency
f_s	Sampling rate
$P( )$	Energy transfer characteristic
$P_N( )$	Noise energy component

SYMBOL	DEFINITION
n_p	Number of photons per signal burst
S_h	Energy of high frequency component

CHAPTER 1

INTRODUCTION

A laser Doppler velocimeter (LDV) is an instrumentation system which measures velocities at points inside a flow field. The LDV schematic diagram is shown in Figure 1.1. An optical arrangement focuses laser light at the control volume, the region where the flow velocity is to be measured. Particles are injected in the flow via a particle seeding mechanism. When a particle passes through the control volume, it scatters the laser light that is focused there. The scattered light contains a Doppler shift in laser frequency due to the velocity of the particle. Some of the scattered light is directed via another lens arrangement to the cathode of a photomultiplier tube. The output of the photomultiplier is then processed by a signal processor which estimates the Doppler frequency of the signal. The actual particle velocity is a linear function of the Doppler frequency, and can be easily determined once the Doppler frequency is known.

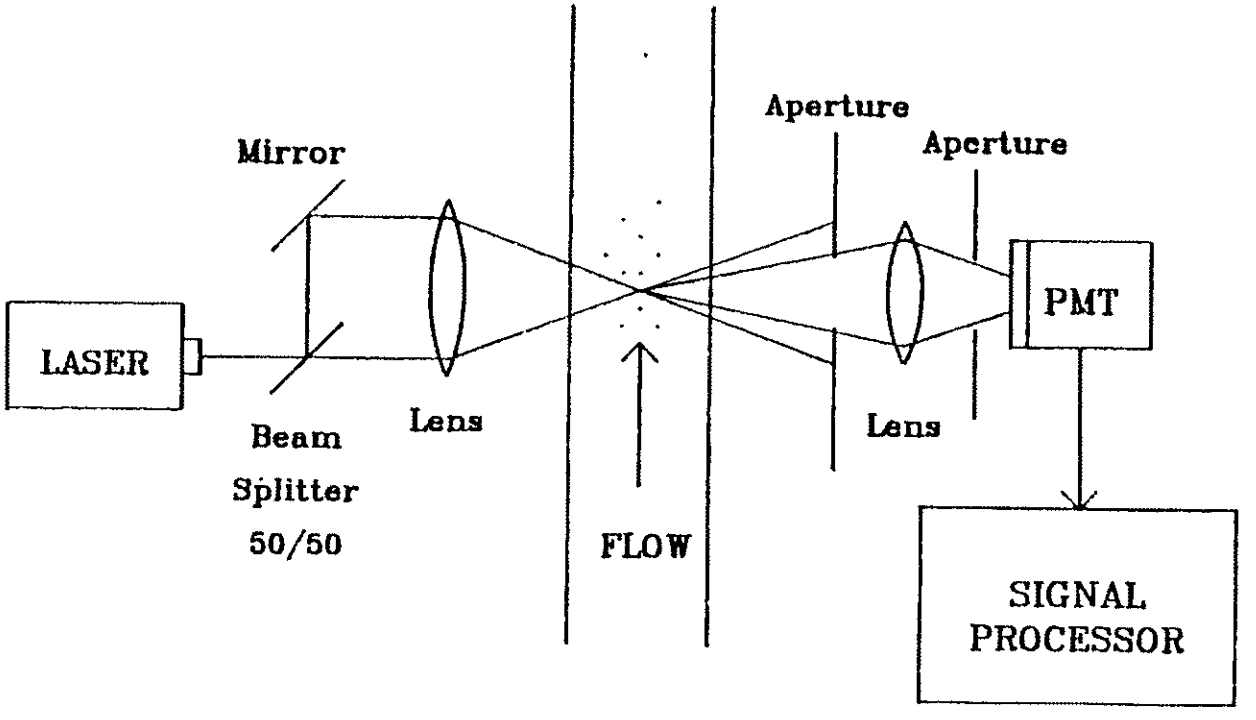


Figure 1.1. LDV System.

LDV systems provide several advantages over conventional mechanical probes, and they have been used for a variety of applications. The primary advantage is that the flow is not disturbed while measuring velocities, because only the laser light is transmitted to the control volume. In addition, laser light can be focused at a very small volume, and a very good spatial resolution can be achieved. Typical resolution is 20 to 100 micrometers, which cannot be obtained by any other method. Another advantage is that the LDV can operate in real time, and velocity variations due to turbulence can be followed. Finally a wide range of flow velocities can be measured, from millimeter per second to supersonic. However, LDV systems are not without disadvantages. The medium must be transparent to allow optical access to the control volume. Some mechanism of introducing particles in the flow is also required. Finally the cost of the signal processing systems increases for accurate processing of signals in high noise and high turbulence environments.

The most important LDV signal processing systems are the frequency tracker [1], [2], the high speed burst counter [3], [4], [5], and the photon correlator [6]. None of these systems takes full advantage of the digital signal processing techniques available, and as a result they

exhibit a number of limitations. The frequency tracker operation is dependent upon past measurements, and is limited to low speed liquid flows. The high speed burst counter is the best available instrument to date for gas flow measurements. Its operation is independent of past measurements, but it requires good signal-to-noise ratios. In addition, the user should set filter parameters during the system initialization. If the filter parameters are not set properly the input signal can be significantly attenuated, and large errors can result. The photon correlator is used in situations of low visibility signals. It cannot be used in real time, and it calculates only first order statistics.

1.1 Objective

The objective of this thesis is to design a new LDV signal processor which outperforms the existing processors. The new processor should operate in real time, independent of past measurements, and require no user intervention during the initialization phase. Its performance should not be significantly affected by variations in input frequency, turbulence intensity, and signal-to-noise ratio.

The proposed processor estimates the input frequency from the frequency characteristics of the input signal, thus it is named frequency domain processor (FDP). The most important feature of the new processor is the use of a bank of digital bandpass filters for the estimation of the signal energy spectrum. The input frequency estimation is based on a spline function approximation approach where the mean squared error is minimized with respect to the weighting coefficients. Nonlinear quantization is employed for enhancement of the Doppler signal characteristics. Automatic setup is accomplished by a variable sampling clock whose rate is adjusted by a controller network for efficient capture of the Doppler signals by the digital filterbank.

1.2 Thesis Overview

Chapter 1 is an introduction. In Chapter 2 the generation of the photomultiplier signal is described for the differential Doppler optical arrangement, and the characteristics of the signal are examined. The existing signal processing systems are reviewed, and the advantages and limitations of each system are outlined. Then a deterministic model for the proposed processor is presented. The relationship between the input signal, and

the filter output energies is examined, and a procedure for the normalization of the filter output energies is suggested. The input frequency is estimated by linearly weighting the filter output energies, and the mean squared error of the approximation is minimized with respect to the weighting coefficients.

The design of the proposed processor is presented in Chapter 3. Two processing stages are identified. In the first stage the input signal is amplified, sampled, quantized, and captured in a long shift register. In the second stage the captured signal is processed through the digital filterbank, the filter output energies are calculated, normalized, and linearly weighted for the input frequency estimation. An error model was developed and used for the evaluation of several filter sets.

The simulation testing of the frequency domain processor is presented in Chapter 4. The results of the mean input frequency estimation and the input turbulence estimation are presented. The simulation results are interpreted using the error model suggested in Chapter 3. The processor performance is compared to the performance of the high speed burst counter.

Chapter 5 includes concluding remarks, and some suggestions for further work.

CHAPTER 2

BACKGROUND

LDV systems measure flow velocities by measuring the Doppler shift of laser light scattered by particles embedded in the flow. The first LDV system was demonstrated in 1964 by Yeh and Cummins [7], who measured Doppler shifts of laminar flow in water pipes. However, the same principle has been utilized earlier by radar systems in a much lower part of the electromagnetic spectrum. The Doppler shifts which are observed in LDV systems are very small, because the velocities that are measured are much smaller than the velocity of light. As a result, optical spectrometers cannot be used because they do not provide the required resolution unless supersonic velocities are involved. The optical arrangements which are employed for measuring very small Doppler shifts utilize the principle of heterodyning or beating, which has been extensively used in radio communications. When two frequencies are simultaneously input to a nonlinear device, the output contains a component of their difference. The most widely used optical method where this principle is

utilized is the dual beam or differential Doppler technique.

2.1 The Differential Doppler Optical Arrangement

The differential Doppler optical arrangement is shown in Figure 1.1. The laser light is divided in two beams of equal intensity which are focused on the control volume. The interference of the two laser beams forms a fringe pattern inside the control volume as shown in Figure 2.1. When a particle moves through the light and dark regions of the fringe pattern, it scatters light whose intensity varies in an oscillatory manner. A portion of the scattered light is collected by the aperture lens, and it is directed to the photocathode surface of a photomultiplier. Light is scattered from both beams simultaneously, and the resulting beat frequency is equal to the difference of the Doppler shifts from two angles of scattering given in [8] by

$$f_D = \frac{v}{\lambda} \sin\left(\frac{\theta}{2}\right) \cos(\beta) \qquad (2.1)$$

where v is the velocity of the particle, λ is the wavelength of the laser light, θ is the angle between the laser beams, and β is the angle the direction of motion makes with the normal to the bisector of the beams. The

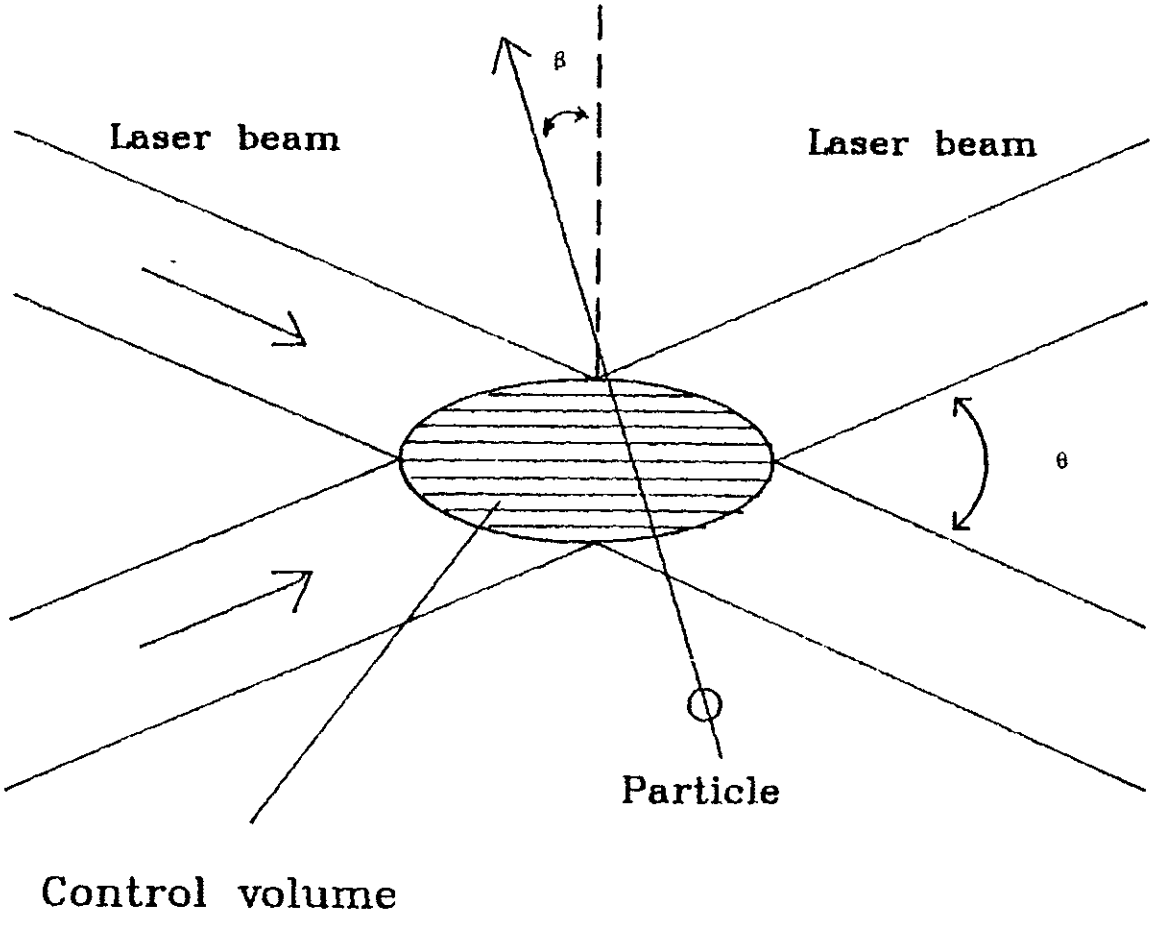


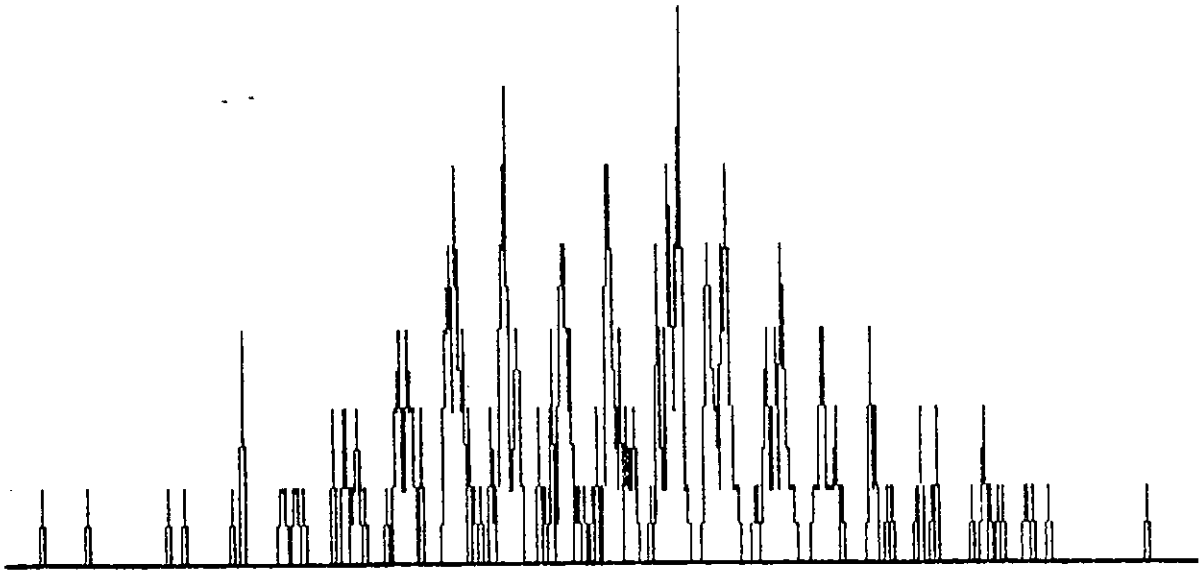
Figure 2.1. Control Volume.

beat frequency is independent of the direction of reception, therefore, a large aperture can be used for the signal detection. Thus, most of the scattered light contributes to the photomultiplier signal, and good signal to noise ratios can be obtained, especially in situations of low particle concentrations that occur in gas flows.

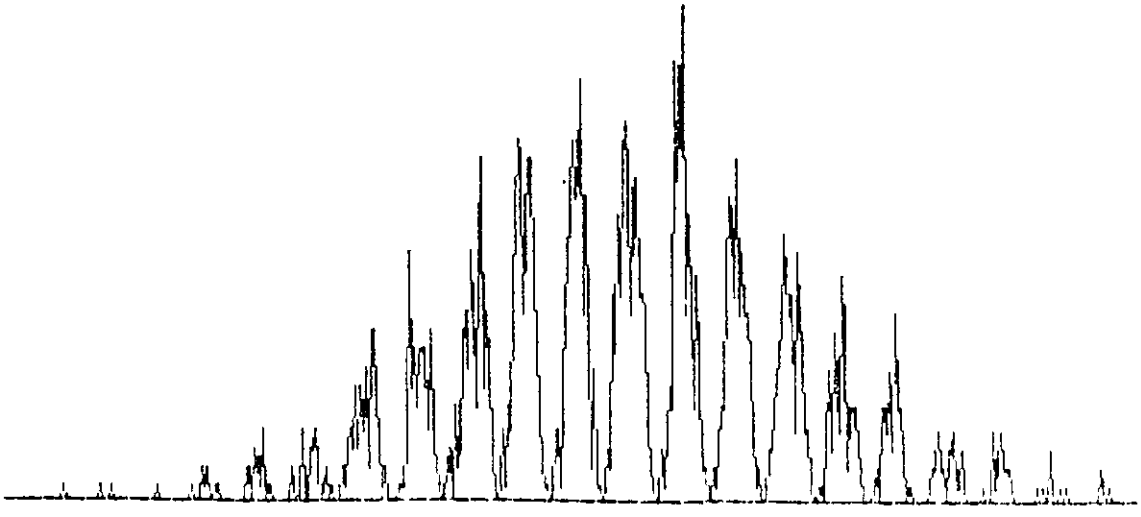
The photomultiplier converts the optical signal to an electronic signal which consists of a collection of Poisson distributed photo-electrons whose average occurrence rate is proportional to the instantaneous light intensity at the photocathode. For small intensities there is one photon per response time of the photomultiplier. As the light intensity at the photocathode increases, additional photon arrivals within the photomultiplier's response time add voltage to the output signal. The output signal quality depends on the number of photons that are present per response time of the photomultiplier. Two photomultiplier signals are shown in Figure 2.2. They are composed of 300 and 1500 photons respectively. As the number of photons increases, the output signal approaches the ideal waveform of Figure 2.3, and can be described by the expression :

$$s(t) = F_1 \exp(-\alpha^2 t^2)[1 + F_2 \cos(2\pi f t + \phi)] \qquad (2.2)$$

$$= F_1 \exp(-\alpha^2 t^2) + F_1 F_2 \exp(-\alpha^2 t^2) \cos(2\pi f t + \phi)$$



(a)



(b)

Figure 2.2. Signal bursts comprised of (a) 300 photons (b) 1500 photons.

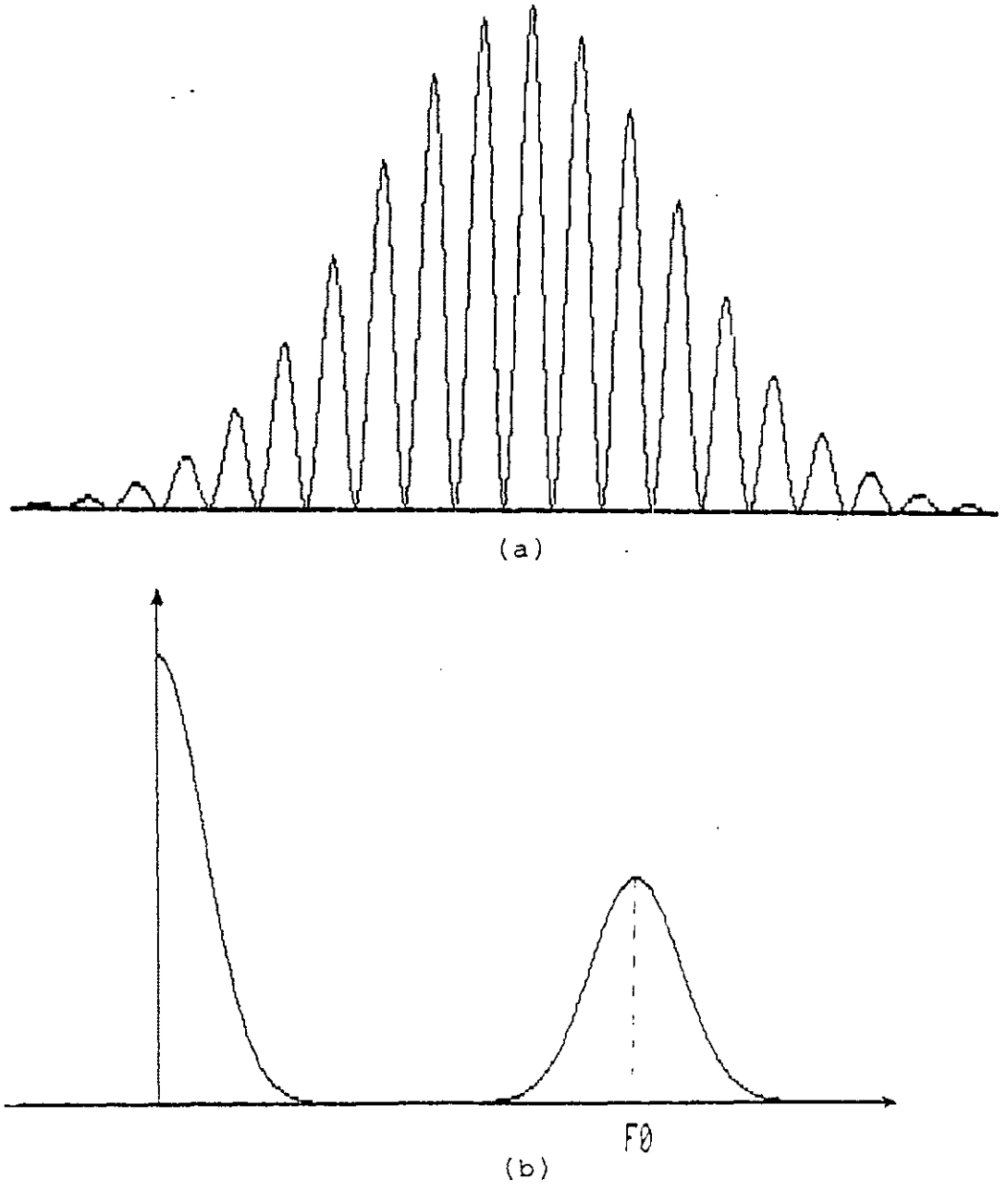


Figure 2.3. Ideal signal (a) Time characteristics
(b) Frequency characteristics.

The signal is composed of a Gaussian shaped low frequency component, and a high frequency component which contains the modulating Doppler frequency f. The low frequency component is called pedestal function, and it is a result of the laser beam intensity variation. Its amplitude F_1 depends on the laser power. The decay rate of the pedestal function is controlled by α, a constant that depends on the optical arrangement that is used. It can be assumed that approximately 15 to 20 cycles of the high frequency term are included in the time interval between the points of the pedestal function that have amplitude $F_1 \exp(-2)$.

The high frequency component contains the modulating frequency f with a uniformly distributed random phase ϕ, and has amplitude K where

$$K = F_1 F_2 \tag{2.3}$$

The ratio between the amplitudes of the high and low frequency components is F_2, and it is called visibility ratio. F_2 can take values between 0.5 and 1.0 depending on the laser power, the particle size and position, and the optics of the system. The modulating frequency f can be assumed constant throughout the duration of a signal burst, because the time that it takes a particle to cross the fringe pattern is very small for any appreciable changes in

velocity to take place. However, different signal bursts can have different velocities. Therefore, f must be measured for each input signal burst. If f is determined, the velocity of the particle can be found by multiplying f with the fringe spacing distance.

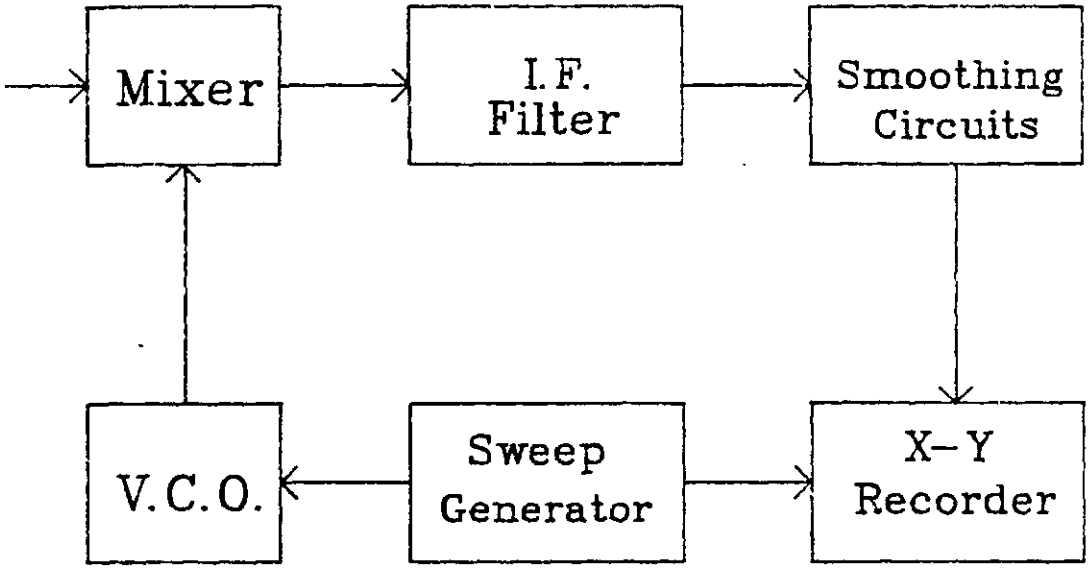
From the above discussion it becomes apparent that the objective of the LDV signal processors is to measure f. The most important signal processing systems that have been used for the measurement of f are presented in the next section.

2.2 Signal Processing Systems

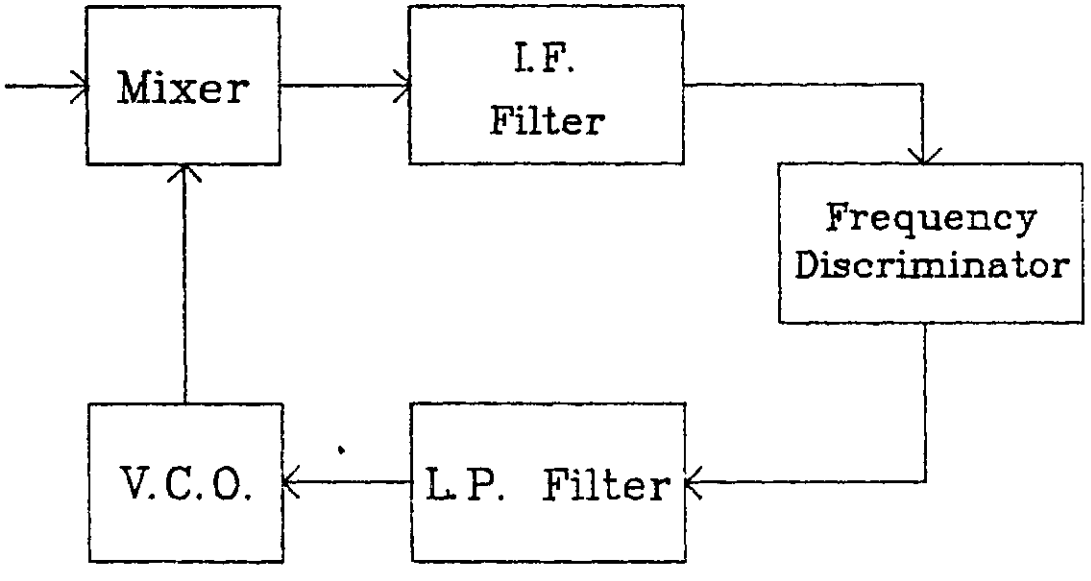
LDV signal processors should cope with the signal detection, signal corruption due to noise, a wide range of input frequencies, and frequency variations due to turbulence up to 20% from the mean. The most important processors are the spectrum analyzer, the frequency tracker, the high-speed burst counter, the analog filterbank processor, and the photon correlator. These systems vary in accuracy, processing speed, complexity of operation, cost, range, and performance in high noise and high turbulence environments. One of the major problems encountered involves signal conditioning for the pedestal removal. The pedestal function causes difficulty in the

Doppler frequency extraction, so the input signal is highpass filtered before it is processed. The cutoff frequency of the highpass filter is set by the operator, and if not chosen properly, filtering can significantly attenuate the Doppler signal in situations where the Doppler frequency is low, or the turbulence is high. The processing techniques of the existing processors are outlined in the following paragraphs. For detailed descriptions of these processing systems the reader is referred to [9].

The spectrum analyzer is an instrument that has been used for applications other than LDV, such as vibrations and electronic testing. The block diagram of a spectrum analyzer is shown in Figure 2.4(a). The input signal has frequency f, and it is mixed with the output of a variable oscillator of frequency f_o. The mixer output contains the frequency f_o+f, and is filtered through a bandpass filter tuned to an intermediate frequency f_i. The filter output is rectified, and recorded or displayed. The signal frequency is found by detecting the peak of the spectrum. If the variable oscillator frequency is swept from f_i to f_i+f_r, the system can detect frequencies from zero to f_r. The spectrum analyzer is easy to use, but it has several limitations. The input signal is used



(a)



(b)

Figure 2.4. Block diagram of (a) Spectrum analyzer (b) Frequency tracker.

inefficiently, because it is filtered through the tuned filter for a small part of the sweep time. In addition, the spectrum analyzer cannot be used in real time, because it must sweep many times through its range to obtain a spectrum that has a high confidence level. In situations of turbulent flow the instrument response is not fast enough to follow the frequency fluctuations. Therefore, the use of the spectrum analyzer is limited to laminar flow. Finally extraction of the mean frequency from the plotted data is time consuming, and often does not provide high accuracy.

The frequency tracker is a device that overcomes some of the limitations of the spectrum analyzer. Its block diagram is shown in Figure 2.4(b). The operation of the frequency tracker is based on the same principle used in radio reception of frequency modulated signals. The input signal is again mixed with the variable oscillator frequency. The mixer output is passed through an intermediate frequency filter, whose output is applied to a frequency discriminator. The discriminator output is used to control the frequency of the variable oscillator. This way the system remains in lock with the signal frequency. The major limitation of the frequency tracker is the requirement that a signal must be present at least one

percent of the time, otherwise the system drifts out of control. That restricts the use of the device to low speed liquid flows where sufficient particle concentration can be achieved. In addition, there is dependence upon past measurement history, which limits the changes in Doppler frequency that can be handled.

The signal processing system that is primarily used in gas flow measurements is the high-speed burst counter. The counter is triggered at some particular threshold voltage, and approximately ten cycles are counted. The time duration of these ten cycles is determined by counting the number of cycles of a fast clock during that time interval. Once the time duration of ten cycles is determined, the period of the Doppler frequency can be calculated. The counter measures the Doppler frequency of each signal burst independently, and allows large standard deviations from the mean frequency. Its major limitations are that it requires a good signal to noise ratio of at least 15 db, and it has a minimum measure of turbulence intensity around 0.5% due to time quantization. In addition, the input signal should be bandpassed through filters whose parameters must be set by the user. If the input filters are not set properly, large errors could result.

Another signal processing system is the analog filterbank processor [10]. It employs a bank of 50 analog bandpass filters which span the frequency range from 0.6 to 6.0 MHz. Successive filters overlap at the 3 db point. The output of each filter is monitored in real time, and an estimate of the Doppler frequency is obtained from the filter with the largest response. The filterbank is very efficient in processing signals that have low signal to noise ratio, but it does not provide good accuracy, and it has a limited range.

The photon correlator is used in situations where the scattered light intensity is very low. For such low light intensities there are not enough photoelectrons arriving at the cathode to produce continuous current, and the resulting waveform consists of individual pulses, each pulse corresponding to the arrival of a photoelectron at the cathode. There is greater probability of photon detection when the particle crosses the bright area of the fringes, than when it crosses the dark area of the fringes. Therefore, the signal will exhibit a sinusoidal rate of photon arrival at the cathode, due to the corresponding intensity variation at the scattering volume. The input frequency can consequently be obtained by measuring the autocorrelation function of the arriving

photons. The photon correlator requires an appreciable amount of computations for the autocorrelation function, and it cannot be used in real time. In addition, it can only calculate the mean of the input frequencies, it can measure only low frequencies, and it is sensitive to interference from light scattered from walls.

In this section the most important LDV signal processing systems have been discussed, and the advantages and limitations of each have been outlined. The existing systems utilize either analog processing techniques or time domain analysis, and do not take advantage of the frequency domain digital signal processing techniques available today. The frequency domain processor is proposed in this thesis with the intent to combine frequency domain analysis and modern digital signal processing technology. The processor modeling is developed in the next section, and the processor design is presented in Chapter 3.

2.3 Design Approach

During the course of this research two different algorithms were developed for the estimation of the input frequency. In both cases the estimation is based on the frequency characteristics of the input signal. The input frequency is determined from the peak of the output signal

spectrum shown in Figure 2.3(b). The signal spectrum can be computed directly using discrete Fourier transform methods. However, the Fast Fourier transform (FFT) computation is time consuming, and it provides the whole spectrum, while only a small part of the spectrum is needed. It is possible to compute just a few points of the Fourier transform using the Goertzel algorithm [11], but for large frequency fluctuations due to turbulence these points cannot be determined a priori. Another potential problem is that the FFT results are sensitive to input noise, and this could lead to large errors in the estimation of the peak.

Instead of computing the signal magnitude spectrum via the discrete Fourier transform, it is possible to estimate the signal energy spectrum using a bank of parallel bandpass filters. By Parseval's theorem the output energy of each filter can be computed in the time domain by squaring and adding the terms of the filter output sequence [12]. The filter output energies are used as an approximation to the signal energy spectrum at the filter center frequencies. The primary advantage of the digital filterbank approach stems from the fact that the design of digital filters is based on normalized frequency, and the same filter coefficients can be used to pass different

input frequencies by changing the sampling rate. Therefore, the digital filterbank processor can adapt to different input frequencies simply by adjusting the sampling rate. Another advantage of the digital filterbank approach is that digital filtering requires a small number of computations which can be pipelined by decomposing the filter in cascaded sections. Furthermore, the computations for each filter of the filterbank are performed in parallel to reduce the processing time.

2.3.1 Model

After it was decided that the input signal would be processed through the digital filterbank, a model was needed, which could describe the relationship between the output filter energies and the input signal characteristics. It is known that an input frequency f is mapped in the normalized frequency domain on to x, such that

$$x = f / f_s \qquad (2.4)$$

where f_s is the sampling frequency. If x is within the filterbank range, the filter output energies will assume certain nonzero values. The filter output energies as a function of x can be thought of as the system transfer characteristics, called energy transfer characteristics

(ETCs). The ETC of the ith filter is denoted by $P_i(x)$. The ETCs will be used for the estimation of the input frequency, therefore it is important to determine their functional dependence on the input signal parameters. As mentioned earlier, the P_i's primarily depend on x. Their value also depends on K, the amplitude of the high frequency component of the input signal given by (2.3). Thus the functional dependence of the ETCs can be expressed as $P_i(x,K)$. Finally the P_i's depend on the input noise. For simplicity in the analysis it was assumed that the input noise is additive and uncorrelated to the signal. Therefore, the presence of noise at the input can be modeled as an additive component to the output filter energies [13]. Since the level of the input noise depends on the number of photons n_p which comprise the input signal, the additive noise component P_N can be expressed as a function of n_p. Thus,

$$P_i = P_i(x,K) + P_N(n_p) \qquad (2.5)$$

The fact that the P_i's depend on parameters other than the input frequency suggests that some normalization procedure should be employed before using them for the input frequency estimation. Ideally the P_i's could be normalized by subtracting the noise component $P_N(n_p)$ and then dividing by S_h, the total energy of the high

frequency component of the input signal. In practice however, neither P_N or S_h are available. Therefore, some estimate of these two quantities will be used for the processor implementation. The way these estimates are obtained depends on the particular design used, and will be discussed in the next chapter.

2.3.2 Curvefitting Algorithm

The first approach in estimating the input frequency from the peak of the output spectrum was developed in cooperation with J. Meyers of N.A.S.A. Langley Research Center. Based on the assumption that the energy captured in each bandpass filter approximates the value of the energy spectrum curve at the center frequency of the filter, several points of the energy spectrum curve are obtained from the output energies of the filterbank filters. These points can be used to obtain an approximation of the energy spectrum by curvefitting the best polynomial function through them. The input frequency is estimated by finding the peak of the curvefitting polynomial. The major limitation of this model is that as the filter bandwidths become large with respect to the Doppler signal bandwidth, the signal energy is contained in only one or two filters, and the remaining filters capture

energy due to noise. This results in a crude and inaccurate curvefit. It is therefore essential to develop a second more effective model. Meyers developed an approach which combined the use of the curvefitting algorithm with a narrow filter set for low turbulence intensities, and the use of counting techniques at the filter output sequence for high turbulence intensities [14], [15]. In this thesis a different approach is taken. The foundations for this approach are presented next.

2.3.3 Linear Approximation Algorithm

In the second model the problem of estimating the input frequency from the output energies of the bandpass filters is viewed as a linear approximation problem. The input frequency which is not an observable is approximated by linearly weighting the filter energies which are observable. The weighting coefficients that minimize the mean squared error are determined as follows. Let the input frequency be x_1, and its approximation be x_1^*. Then

$$x_1^* = \sum_{i=1}^{k} a_i \, R_i(x_1) \tag{2.6}$$

where a_i is a real weighting coefficient, and R_i is the

ETC of the ith filter normalized as suggested in Section 2.3.1. The error in the estimation of the frequency x_1 is

$$e_1 = x_1 - x_1^* \tag{2.7}$$

If the approximation is performed over the interval $a<x<b$, the normalized filter ETCs can be viewed as spline functions which are linearly weighted to approximate x in (a,b) [16]. Then

$$x^* = \sum_{i=1}^{k} a_i \, R_i(x) \qquad a<x<b \tag{2.8}$$

The mean squared error in (a,b) is

$$E = \frac{1}{b-a} \int_a^b (x-x^*)^2 \, dx \tag{2.9}$$

Given the functions $R_i(x)$ it is possible to minimize the mean squared error E with respect to the coefficients a_i. The minimization procedure is included in Appendix A, where the matrix equation (A.1) is derived. This equation is rewritten here as

$$\left[\int_a^b R_i(x) \, R_j(x) \, dx\right]_{kxk} \left[a_i\right]_{kx1} = \left[\int_a^b x \, R_i(x) \, dx\right]_{kx1} \tag{2.10}$$

or

$$\underline{C} \, \underline{A} = \underline{B}$$

If the matrix $\underline{C}$ is invertible, it is possible to solve for the coefficient matrix $\underline{A}$ as follows:

$$\underline{A} = \underline{C}^{-1} \underline{B} \tag{2.11}$$

Once the coefficients a_i are determined, the input frequency is estimated from (2.8).

2.3.4 Section Summary

In this section two models that can be used for the estimation of the input Doppler frequency were presented. In the first model the input energy spectrum was approximated by a polynomial function which was curvefitted over the filter output energies. The Doppler frequency was determined from the peak of the approximated spectrum. In the second model the Doppler frequency was estimated by linearly weighting the filter output energies. The weighting coefficients were chosen such that the mean squared error of the approximation is minimized. The functional dependence between the ETCs and the input parameters was also examined, and an appropriate normalization procedure was suggested. The processor design and an error model for the design evaluation are presented in the next chapter.

CHAPTER 3

DESIGN

After taking into account the limitations of the existing LDV processors, the goal of this work was to design a new LDV processor that satisfies the following specifications:

1. Operation should be in real time with a throughput of 1000 particles per second.

2. The processor must operate automatically without user intervention.

3. The device must be accurate yielding average errors less than half percent.

4. Frequency variations due to turbulence should be allowed, for turbulence intensities up to 20%.

5. Signals with as low as 150 photons per burst should be efficiently processed.

6. Wide range of operation is required for mean input frequencies from 1 to 100 MHz.

7. Intermittent signals occurring in gas flows should be processed without difficulty.

The block diagram of the proposed processor is shown in Figure 3.1. The flowchart of the processor functions is shown in Figure 3.2. The processing of each input burst is carried out in two stages. In the first stage the waveform

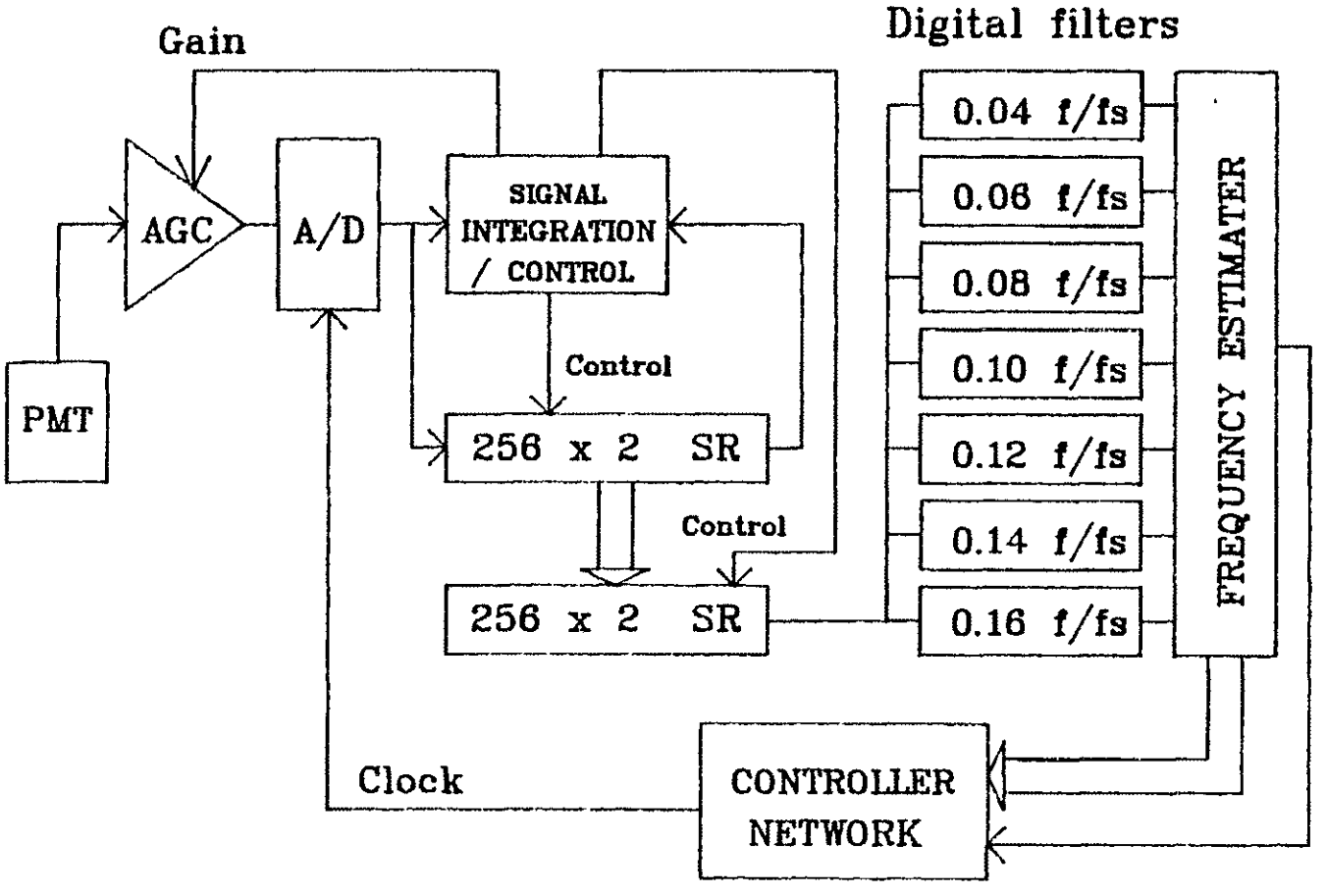


Figure 3.1. FDP block diagram.

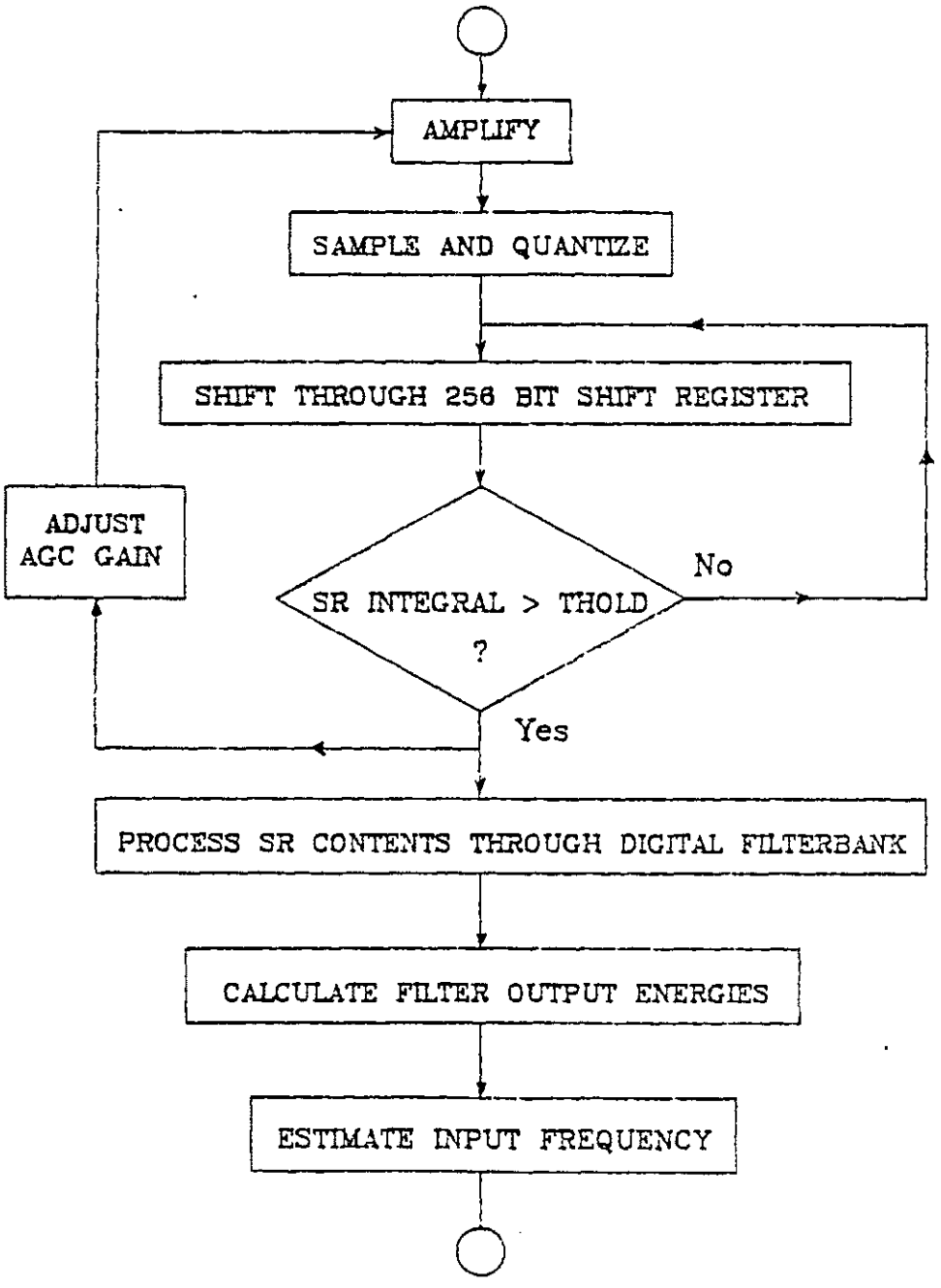


Figure 3.2. FDP processing functions.

is amplified to a constant level, it is sampled, and quantized. The quantized signal is shifted through a long shift register until the Doppler burst is detected and captured. The second stage of the processing begins after the Doppler signal has been captured. The shift register contents are processed through the bank of parallel bandpass filters. The output energies of the filters are calculated and used for the estimation of the input frequency. The sampling clock rate is set during the processor initialization at the value which maps the mean input frequency at the center of the filterbank. The clock setup procedure is automatic, and it is performed by the controller network. Each processing stage will be described in further detail in the following paragraphs.

3.1 First Processing Stage

The processing in the first stage is independent of the frequency estimation algorithm employed. At first the input waveform is amplified by an automatic gain controller network (AGC). The initial AGC gain is set manually by the operator to a value that is a power of two i.e. 1, 2, 4, 8, 16 etc. As the input signal bursts are processed, the gain is adjusted through a feedback network depending on the number of times that the captured signal exceeds the

highest quantization level. The gain is adjusted in small increments, so that its value does not change drastically due to few waveforms. If the required gain is above twice the manual gain or below half the manual gain, front panel lights are lit to indicate that the the manual gain should be adjusted. Completely automatic gain control is not employed to avoid noise amplification to levels that could be interpreted as signal.

After the input waveform has been amplified, it is sampled and quantized by a two bit analog to digital converter (ADC). Nonlinear rather than linear quantization is employed to compensate for the effect of the pedestal function. This way the highpass filtering of the input waveform is not necessary, and the problems that can be caused by it are eliminated. The ADC quantization levels should be chosen such that the quantized signal retains the frequency characteristics of the unquantized waveform. Thus it is desirable that each cycle of the waveform crosses at least one quantization level. The lowest quantization level should be placed as low as possible to include the maximum number of waveform cycles in the quantized signal. However, it cannot be placed at a very small value, because noise would cross it and would be interpreted as signal. The voltage level of individual

photons is 0.05 volts, so the lowest quantization level is chosen at 0.1 volts. The highest quantization level is placed at 0.4 volts at half the photomultiplier saturation voltage. The position of the middle quantization level is not crucial, and it is placed at 0.2 volts. ADC's with three and four bit quantization were also considered. The simulation results indicated that the increase in accuracy which resulted was not significant. Therefore, it was decided to use a two bit ADC resulting in less hardware.

The quantized waveform is not directly processed through the filterbank, because the input signal bursts are not present at all times. Each signal burst is first detected and captured in a 256 bit shift register. As the quantized signal is shifted through the shift register, a signal integration circuit computes the integral of the shift register contents. Since the signal amplitude takes on only positive values, the integral of the shift register contents can be easily computed, and it provides a measure of the size of the captured signal. The signal burst is detected when the value of the integral crosses some threshold. Then the contents of the shift register are shifted 64 times to ensure that the signal burst is centered, and they are parallel loaded in temporary registers in order to be processed through the filterbank.

This way only the portion of the input data that contains useful information is processed, which results in an increase of the processor efficiency. Shift register lengths of 128, 256, and 512 bits were tested in the simulation program. Larger shift registers require higher sampling rates to fill up the same portion. This yields higher resolution in the waveform representation, and provides more accuracy. However, the 256 bit shift register was preferred over the 512 bit shift register to relax the sampling rate requirements, and to reduce the processing time.

3.2 Second Processing Stage

The second processing stage begins by processing the shift register contents through the digital filterbank. The design of the filters is governed by the design objectives, the frequency estimation algorithm, and the projected processor cost. An error model based on the filter ETCs was developed, and it was used for the evaluation of the filter design. The filter coefficients were obtained using the Atlanta Signal Processors Digital Filter Design Package [16].

The design objectives require that mean input frequencies up to 100 MHz should be processed accurately.

For a maximum sampling frequency of 1 GHz the maximum frequency of 100 MHz is mapped at 0.1 f/f_s. Thus the center frequency of the middle filter is placed at 0.1 f/f_s. Another design requirement is that turbulence intensities up to 20% should be handled. Therefore, the filterbank should pass frequencies within almost three standard deviations above and below the mean frequency. The number of filters that is used to span this range depends on the required accuracy and the projected processor cost. As the number of filters increases, the accuracy improves, and the cost increases. A reasonable compromise between the two conflicting requirements is the use of seven filters, because five filters do not provide the required accuracy, and nine filters increase the processor cost beyond the desirable limit. The center filter is placed at 0.1 f/f_s, and the others are placed at 20, 40, and 60% above and below the center. Thus the center frequencies of the seven filters are at 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, and 0.16 f/f_s.

The number of filters that are linearly weighted for the frequency estimation depends on the filter separation. The larger that the filter separation becomes, the fewer filters contain information about the signal spectrum. In the case of the filter positioning suggested above, at most

three filters contain signal energy, and the remaining contain energy due to noise. Thus it was decided to use three filters for the frequency estimation, because more filters would not provide any more information about the input signal. The three filters that are chosen are the one with maximum energy, and the two filters that are next to it.

The type, order, and bandwidth of the filters are chosen to best suit the algorithm that is used for the input frequency estimation. In the estimation algorithm based on the linear approximation model it is required that the ETCs, which are viewed as spline functions, be smooth functions. Thus Butterworth filters are used, because they exhibit smooth, monotonic characteristics in the passband. Although the spline functions should be smooth, they should not be flat, so that different input frequencies can be detected from the filter output. The flatness of the filter characteristics in the passband was reduced by allowing a large deviation from unity in the passband. It should be noted that only in the case of Butterworth filters this is possible, because they are monotonic in the passband. It is also desirable to minimize the energy capture in the filters that are positioned away from the input frequency. Thus high filter orders are desirable, so

that the frequencies outside the passband are greatly attenuated. However, as the order of the filters increases, it becomes more difficult and more expensive to implement them. It was decided to use 8th order filters, because they provide sufficient attenuation outside the passband, and they can be implemented in real time by decomposing them in second order cascaded sections. Finally some overlap between the filter passbands is desirable, so that more than one of the filter energies contain information about the input signal. The amount of overlap was determined from the system error model which is presented next.

3.3 Error Model

The system error model was developed so that the system performance can be evaluated for any given set of filters. Once a particular set of filters is chosen, the system ETCs described by (2.5) should be obtained. After the ETCs have been obtained, it is possible to compute the coefficients which minimize the mean squared error by solving the matrix Equation (2.10). Each ETC was obtained by averaging the corresponding filter output energies over 30 consecutive inputs, each comprised of 750 photons. The input frequency was varied from 0.04 to 0.16 f/f_s to

obtain the filter ETCs for the filterbank range. The ETCs that are obtained should be normalized by subtracting the noise component P_N, and dividing by S_h, the total energy of the signal high frequency component, as suggested in Section 2.3.1. The noise component is not known, and has to be estimated. For simplicity P_N is set equal to the minimum filter energy by assuming that the minimum filter energy is exclusively due to noise. The same level of noise is assumed in all of the filters, therefore the smallest output energy is subtracted from all the filter energies. The energy of the signal high frequency component is also unknown. It is assumed that it is equal to the sum of three filter energies, the maximum filter energy and the energies of the two filters next to it. This approximation is crude due to the amount of overlap in the filters. It works however, because the amount of overlap between the filters is the same in all cases. Thus,

$$R_i = \frac{P_i - P_{min}}{(P_{max-1}-P_{min})+(P_{max}-P_{min})+(P_{max+1}-P_{min})} \qquad (3.1)$$

The normalized filter ETCs are the spline functions $R_i(x)$, which will be linearly weighted in order to estimate the input frequency. The weighting coefficients

which minimize the mean squared error of the approximation are obtained by solving the system of Equations (2.10). Three spline functions were weighted, the one with maximum value and the two from the filters next to it. The computer program MSE which is listed in Appendix B was used for the calculation of the weighting coefficients. The weighting coefficients are computed once for a particular set of filters, and they can be used for the input frequency estimation during the real time operation of the processor. The error between the input frequency and its estimation was plotted as a function of frequency, and it was used for the evaluation of each set of filters. The best error function was obtained for the filter set with passbands overlapping by 40%. The error functions that were obtained for filters with 20, 40, and 60% overlap are shown in Figure 3.3.

One further step was taken in improving the spline function characteristics, so that the error function is improved. The filters are designed close to the lower end of the normalized spectrum, and their frequency response characteristics are not symmetric in the interval of the approximation. The filter responses can be made more symmetric by slightly extending the passband at the lower end. The frequency responses of one filter with

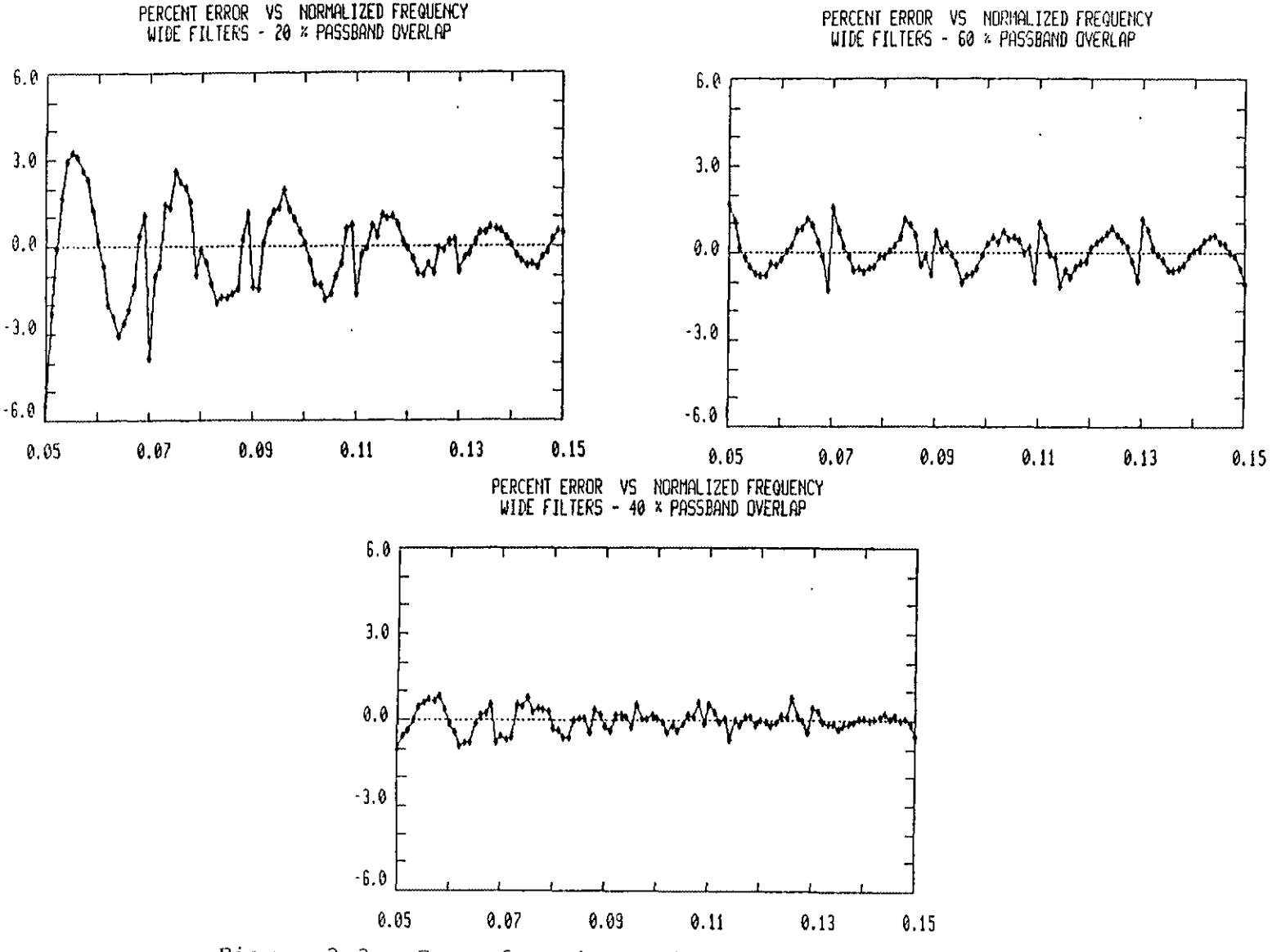


Figure 3.3. Error functions of the linear approximation.

nonsymmetric response and one filter with symmetric response are shown in Figure 3.4. The resulting error functions are shown in Figure 3.5. The filter set with symmetric frequency responses provides a slight improvement in the error function and is preferred. This error function has the desirable characteristic of being symmetric around zero, thus it shows no bias. It is therefore expected that if the input frequencies are mapped throughout the filterbank range, the average error will be close to zero. In addition the average error at any frequency should not exceed the magnitude of the error function, which is less than 1%.

In order to better illustrate the design procedure the frequency response of the three middle filters is shown in Figure 3.6. The ETCs for these three filters are also shown in Figure 3.6. Note that the ETCs are not as smooth as the frequency responses of the filters even after averaging the filter output energies of 30 consecutive input bursts. The ETCs do not assume zero values if the input frequency lies outside the filter passbands due to the noise energy captured. Finally the normalized ETCs which are the spline functions used in the linear approximation are shown in Figure 3.6. It is interesting to notice that the spline functions resemble triangular

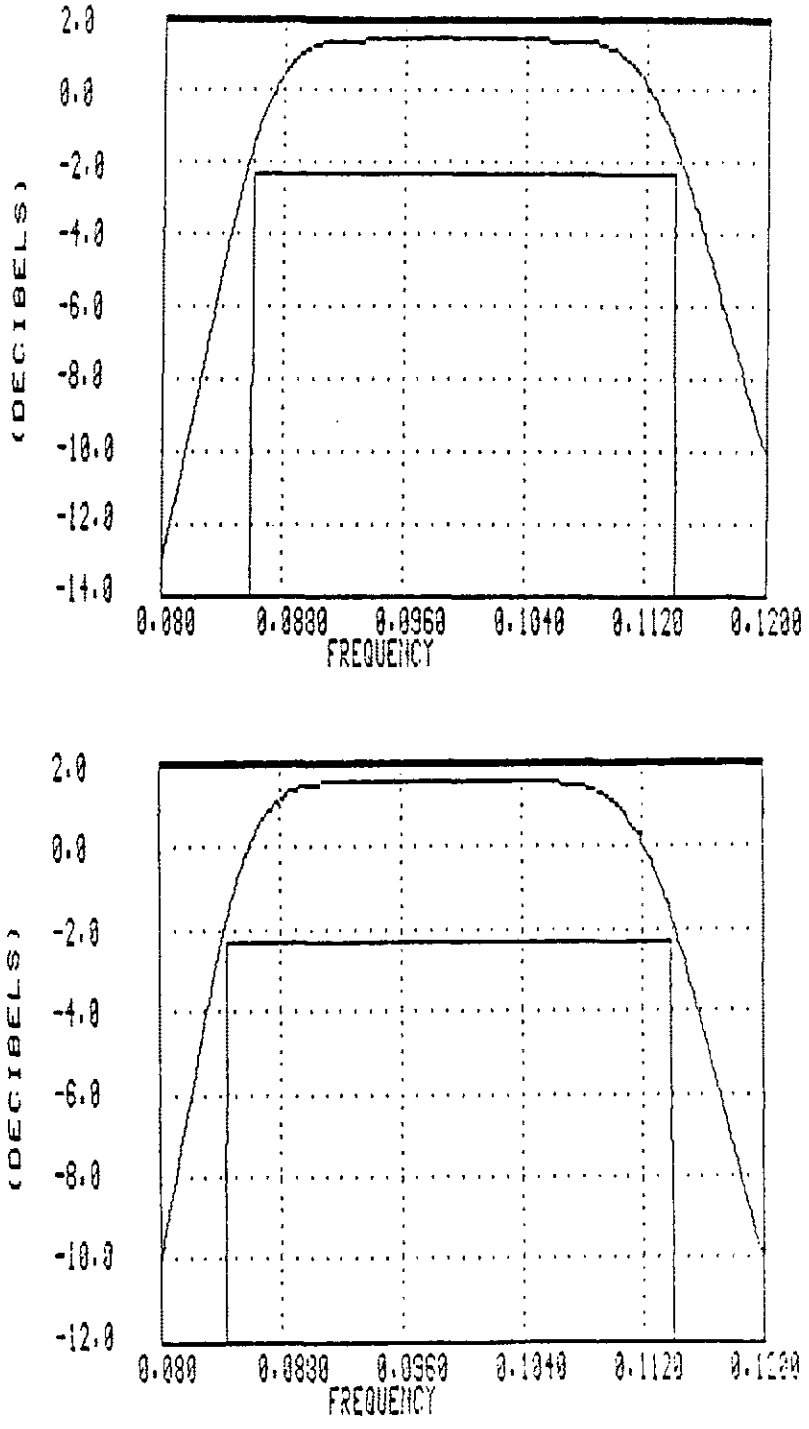


Figure 3.4. Filter frequency responses (a) Nonsymmetric (b) Symmetric.

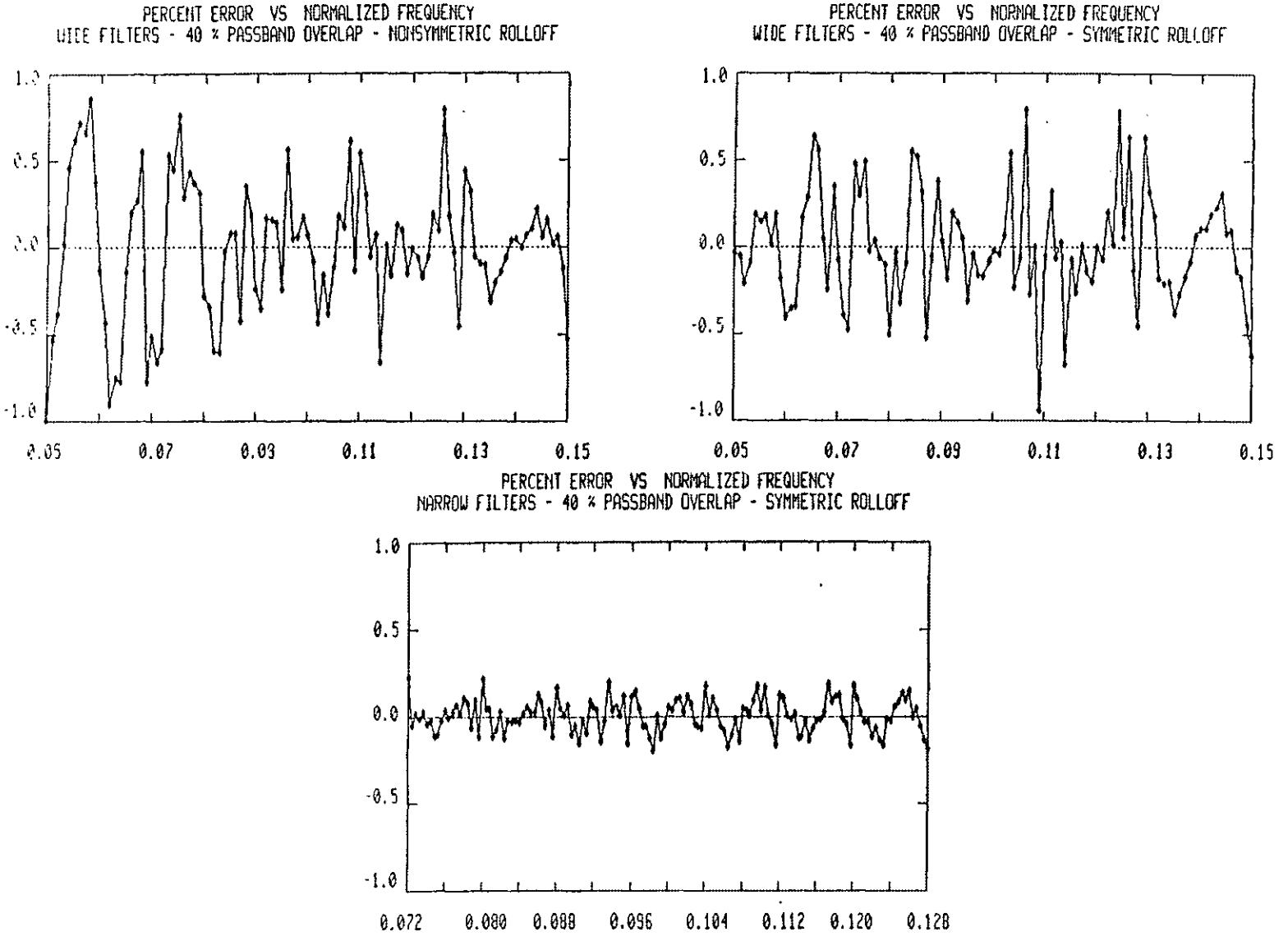


Figure 3.5. Error functions for wide and narrow filters.

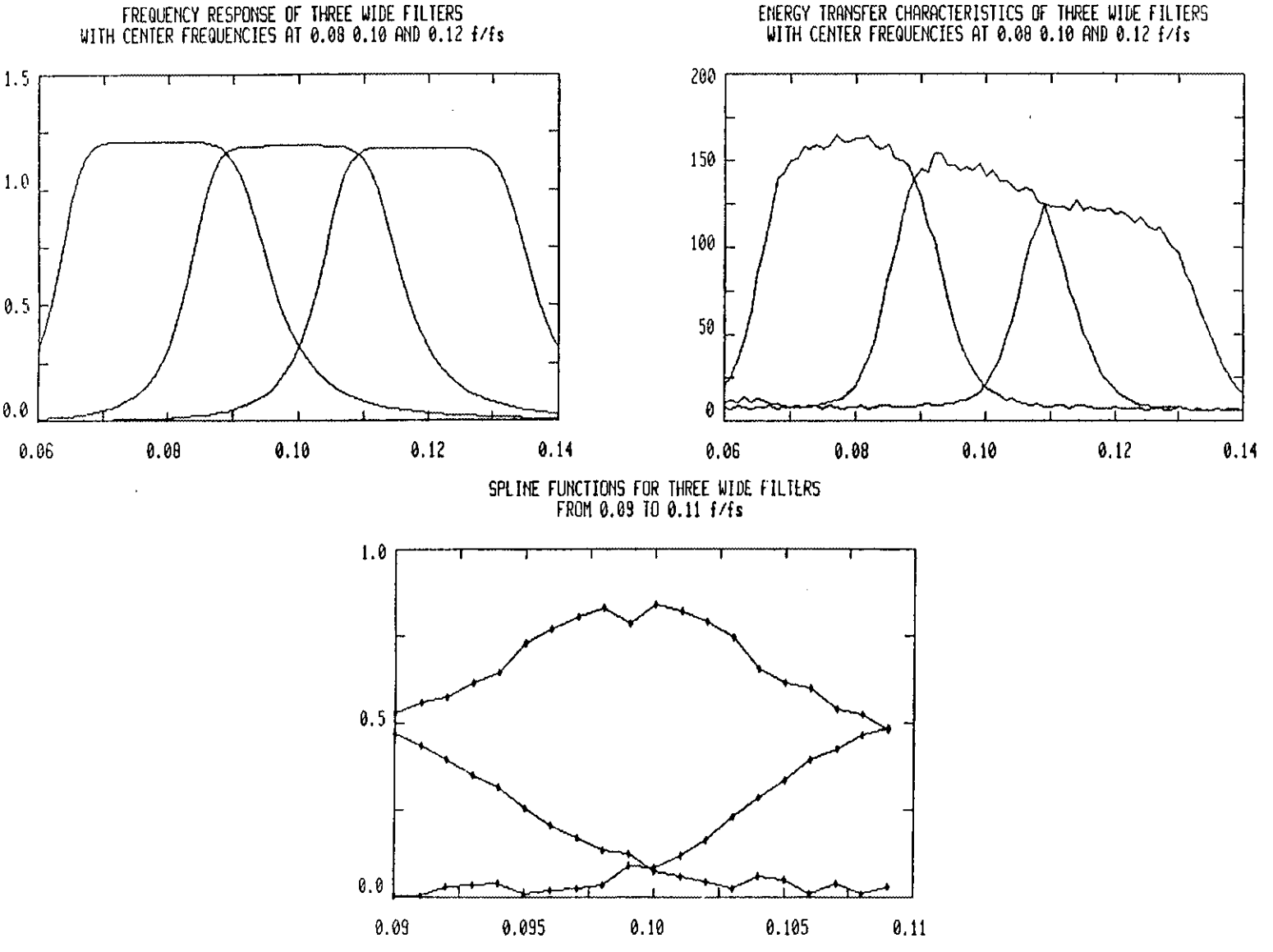


Figure 3.6. Frequency responses, ETCs, and spline functions for three wide filters.

functions, which is the ideal shape for spline functions used in approximations of this type [18]. Finally the procedure of the frequency estimation from the filter output energies is illustrated in the flowchart of Figure 3.7.

Although the error function of the wide filters is good, in cases where the input turbulence is not very high the range that the wide filters provide is not necessary. A better error function can be obtained if a narrow set of filters is used in situations where the turbulence is below 5%. The filters are again chosen as 8th order Butterworth with a large passband ripple and 40% overlap of their passbands. Nine filters were used at center frequencies of 0.068, 0.076, 0.084, 0.092, 0.100, 0.108, 0.116, 0.124, and 0.132 f/f_s, which form three filter sets with seven filters in each set. One filter set is chosen for a given run depending on the mean normalized input frequency. The error function that was obtained from this filter set is shown in Figure 3.5. This error function provides a significant improvement over the error function of the wide filter set.

The efficient application of the linear estimation algorithm heavily relies on the correct identification of the peak. In cases of high turbulence and low

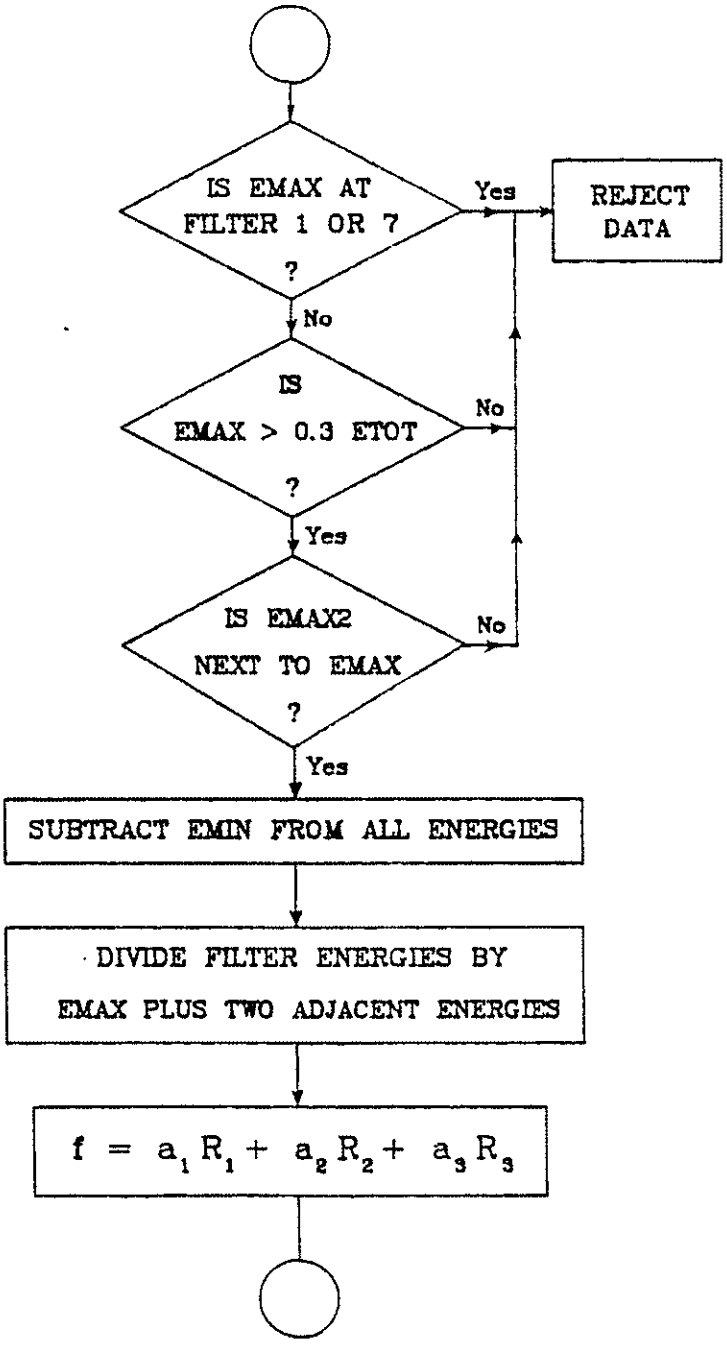


Figure 3.7. Frequency estimation procedure.

signal-to-noise ratio it is not always easy to correctly identify the peak. To avoid identification of the wrong peak that could lead to large errors, some degree of intelligence was built in the peak identification procedure, so that data points that could provide erroneous estimation are excluded. If the maximum filter energy was at the first or last filter of the filterbank, the input frequency was assumed to be outside the filterbank range, because the energies at both the filters around the maximum cannot be obtained. If the maximum filter energy is less than 30% of the total energy contained in the filters, the data is not processed, because the peak is not sufficiently large to indicate that a signal burst is captured in the shift register. Finally if the second largest energy does not occur at a filter next to the filter with maximum energy, the data is rejected as unreliable, because there are two peaks in the spectrum.

3.4 System Initialization

Before starting to actually measure input frequencies, the processor must be initialized. During the system initialization the sampling clock value is set, and it is decided whether the wide filter set or one of the narrow filter sets will be used.

The clock initialization procedure is shown in the flowchart of Figure 3.8, and can be performed automatically by the controller network. The sampling frequency f_s should be chosen such that the mean input frequency F is mapped at the center of the filterbank. This way the system would be able to process signals with the largest possible variation in frequency above or below the mean. The clock setup is accomplished by setting f_s at a very large value, and gradually decreasing it until F is mapped at the middle filter. Large sampling rates are desirable to provide a better representation of the signal burst, however there is a limit to the hardware speed of operation. Thus the maximum clock frequency is chosen at 1 GHz. At first for every particle that is processed the output energies are examined. If the maximum energy is less than 30% of the sum of all the filter energies, or if the maximum energy occurs at a filter below the middle one, the clock value is decreased to 0.8 of its value. When the maximum energy is at or above the middle filter, an estimate of the mean frequency is obtained from 10 particles. If the mean normalized frequency is mapped between 0.085 and 0.115 f/f_s, the clock value is fixed. If it is below 0.085, the clock value is decreased to 0.8 its value. If it is above 0.115, the clock value is

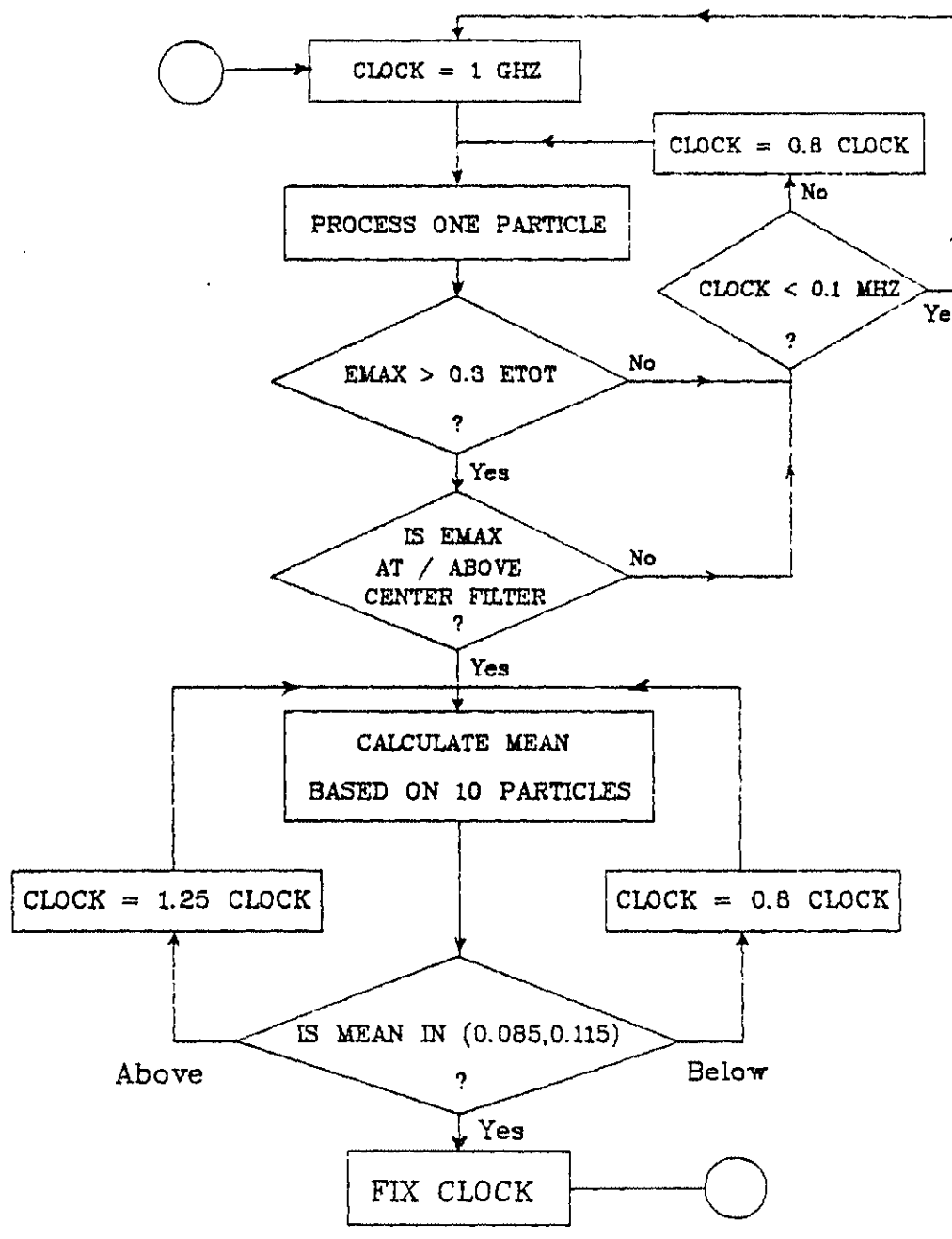


Figure 3.8. Clock initialization.

increased to 1.25 its value. These frequency differentials were chosen because they are easy to implement using three crystals and divide by two circuitry. If the clock frequency required is below 0.1 MHz, it is assumed that the correct sampling frequency was missed, and the clock is reinitialized to 1 GHz.

After the sampling clock has been fixed, the choice of a filter set is made after processing 30 particles through the wide filters. The mean normalized frequency and standard deviation for the 30 particles are computed. If the standard deviation is above 5%, the processing is performed through the wide filters. If the standard deviation is less than 5%, the narrow filter set is used. The mean normalized frequency is used for the choice of the narrow filter set. The procedure for the filter set choice is shown in the flowchart of Figure 3.9, and it is also performed automatically by the controller network.

3.4 Summary

In this chapter the frequency domain processor design was presented. Two stages were identified for the processing of each signal burst. In the first stage the input waveform is amplified through an AGC network. Then it is quantized through a two bit ADC with nonlinear

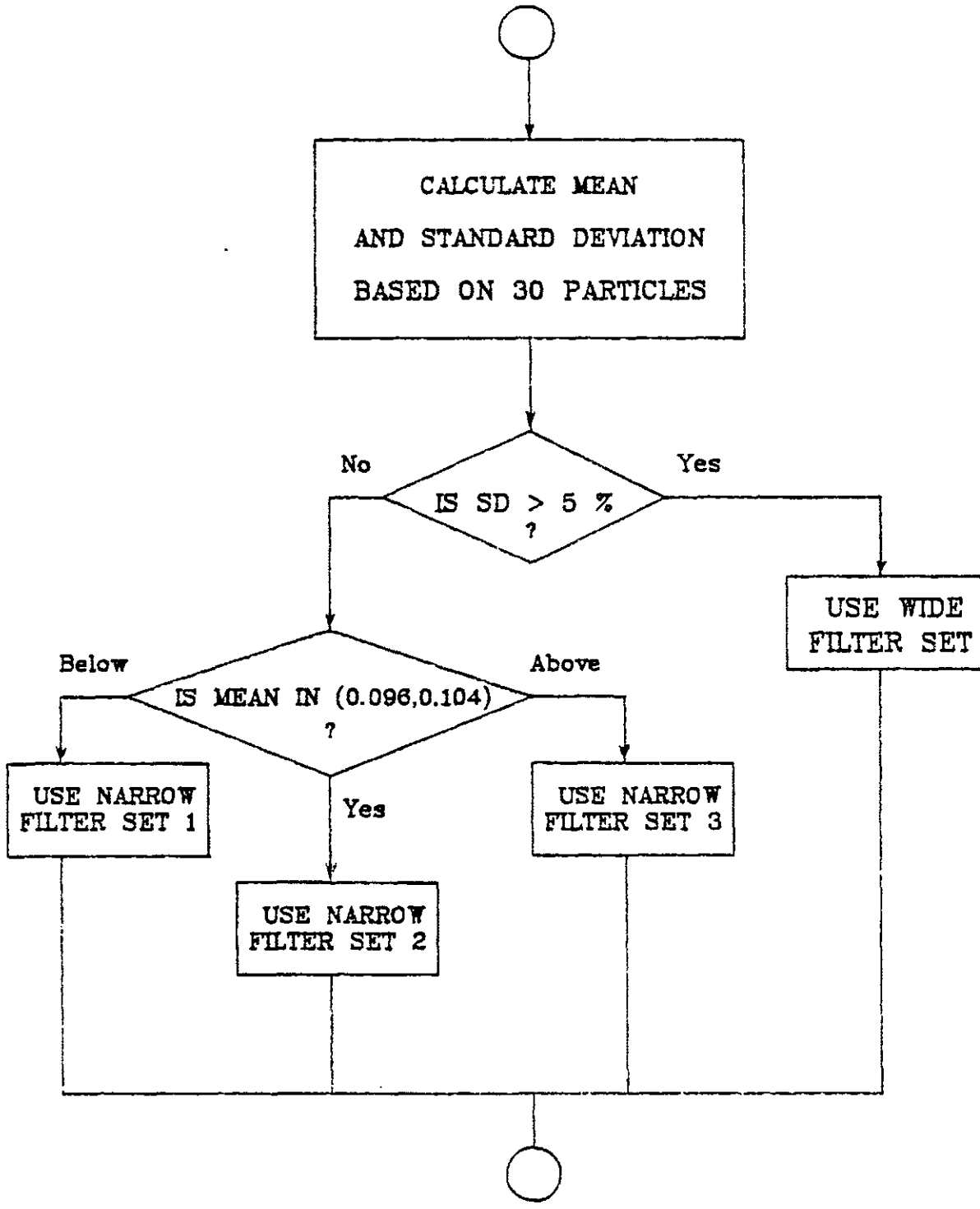


Figure 3.9. Choice of filter set.

spacing in the quantization levels for better representation of the frequency characteristics of the signal burst. The quantized signal is shifted through a 256 bit shift register until the signal burst is detected. The burst detection is accomplished when the integral of the shift register contents exceed some threshold. The second stage begins by processing the shift register contents through the digital filterbank. Two sets of filters are used, one wide set for turbulence more than 5%, and one narrow set for turbulence less than 5%. The filters were chosen as 8th order Butterworth with large passband ripple. The filter output energies are normalized by subtracting the minimum filter energy which is assumed to be the noise component, and dividing by the sum of the maximum filter energy and the energies of the filters next to it. After being normalized, these three energies are linearly weighted for the input frequency estimation. The weighting coefficients are chosen such that the mean squared error of the approximation is minimized. The system initialization is accomplished automatically by adjusting the sampling clock to the value that maps the mean input frequency at the center of the filterbank, and choosing the appropriate filter set. The processor performance was evaluated via simulation. The results are presented in the next chapter.

CHAPTER 4

SIMULATION AND RESULTS

In this chapter the results obtained from the simulation testing of the frequency domain processor are presented. The simulation testing was performed for various input frequencies, turbulence intensities, and numbers of photons per signal burst. The results demonstrate that the proposed processor meets the design objectives, and outperforms the high speed burst counter.

4.1 Simulation Program

The simulation program FDP, listed in Appendix C, was used for the processor evaluation throughout the course of the processor design. The simulated processor functions are shown in the flowchart of Figure 3.2. The photomultiplier signal burst is generated via Poisson shot noise models [19]. The program for the generation of the photomultiplier signal was made available by J. Meyers of N.A.S.A. Langley Research Center. Each signal burst is amplified through the AGC network, sampled, and quantized. Then it is processed through the digital filterbank, and

the output energies of the filters are calculated. The filter energies are normalized and linearly weighted for the input frequency estimation as shown in the flowchart of Figure 3.7. The sampling clock value is set at the beginning of the program as shown in the flowchart of Figure 3.8. The choice of the appropriate filter set is made after the clock has been fixed according to the flowchart of Figure 3.9. Once the clock has been fixed and the filter set has been chosen, the frequency estimation begins. The error between the input frequency and the estimated frequency is calculated for every signal burst. The error statistics are calculated based on 100 particles. The input and estimated turbulence intensities are also calculated and compared. The simulation results are presented in the following section.

4.2 Results

The parameters varied during the simulation testing are the input frequency, the input turbulence intensity, and the number of photons per signal burst. The input frequency range is from one to 100 MHz. The input turbulence varies from zero to 20% from the mean. The number of photons per burst can vary between 150 to 3000 photons. Since it is impossible to provide data for every

possible case, the following set for data is presented, which is believed to be sufficient for the evaluation of the processor performance. The simulation testing was performed for mean input frequencies of 5, 25 and 100 MHz. The input turbulence was varied from 0 to 20% for signals comprised of 1500 photons (good signal to noise ratio) and 300 photons (poor signal to noise ratio). The number of photons per signal burst was varied from 150 to 3000 for zero input turbulence.

4.2.1 Mean Frequency Estimation

During the estimation of the mean input frequency both the average percent error and the standard deviation of the error were calculated. In the graphs to be presented the results are represented by small circles. The solid lines connect the data points to show the trends of the processor performance, and do not convey any other meaning.

The average percent error in the estimation of the input frequency is shown in Figure 4.1 as a function of photons per burst at zero input turbulence. In all cases the average error does not exceed 0.3%, and it is less than 0.1% for signals comprised of 300 photons or more. The average percent error is very small and does not show any bias. These results are due to the fact that the error

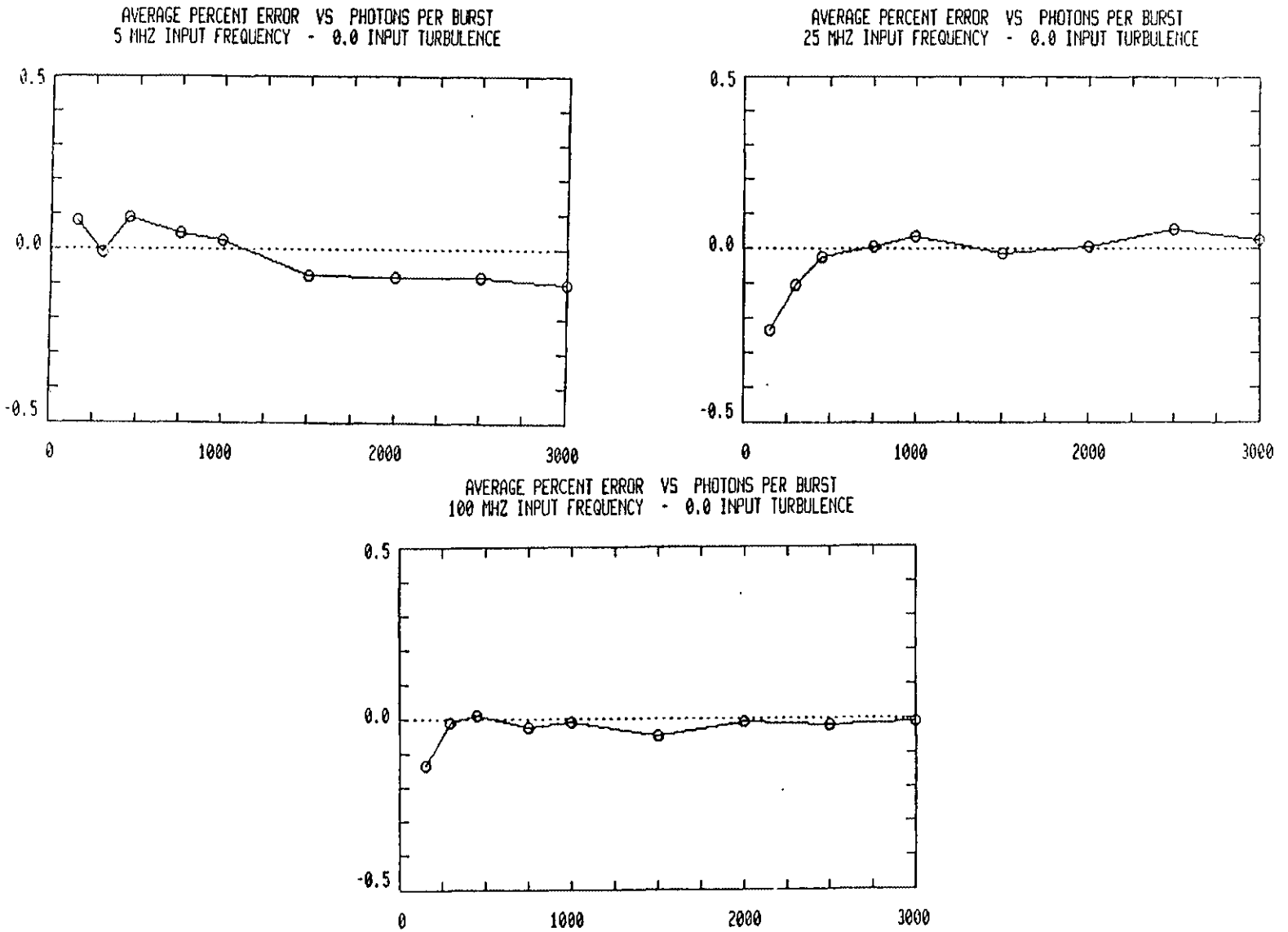
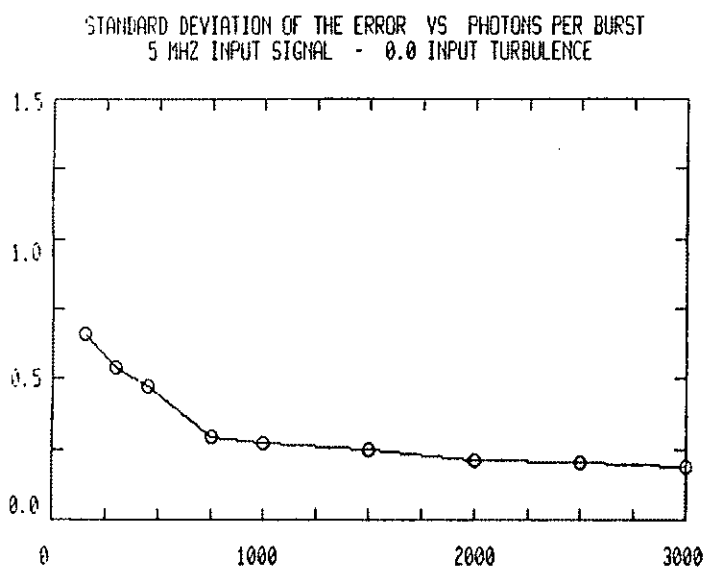


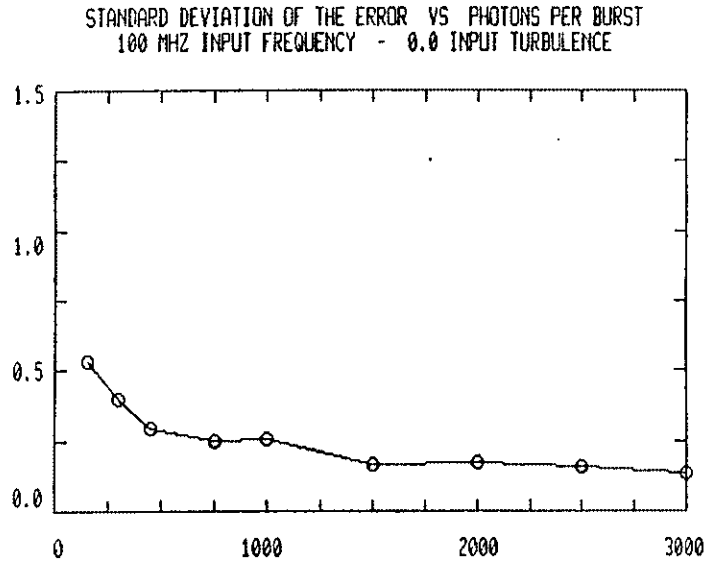
Figure 4.1. Average percent error vs number of photons per burst.

function of the narrow filter set shown in Figure 3.5 is unbiased and has very small amplitude.

The standard deviation of the error as a function of photons per burst is shown in Figure 4.2. The standard deviation of the error decreases as the number of photons per burst increases. This demonstrates that the system performance depends on the level of the input noise. The results show that the standard deviation of the error is below 0.3% for signals comprised of 750 photons or more, and it shows a slight improvement as the number of photons per burst increases.

The percent error in the estimation of the mean input frequency was also examined as a function of input turbulence for signal bursts comprised of 300 and 1500 photons. The average percent error as a function of turbulence is shown in Figure 4.3 for 1500 photons per burst. The error remains below 0.3% for input turbulence intensities up to 15%. When the input turbulence is above 15% the processor does not estimate the mean input frequency with the same consistency and the error can be as high as 0.5%. This happens because at high input turbulence the processor operates close to the limits of its range where its performance is not as good.





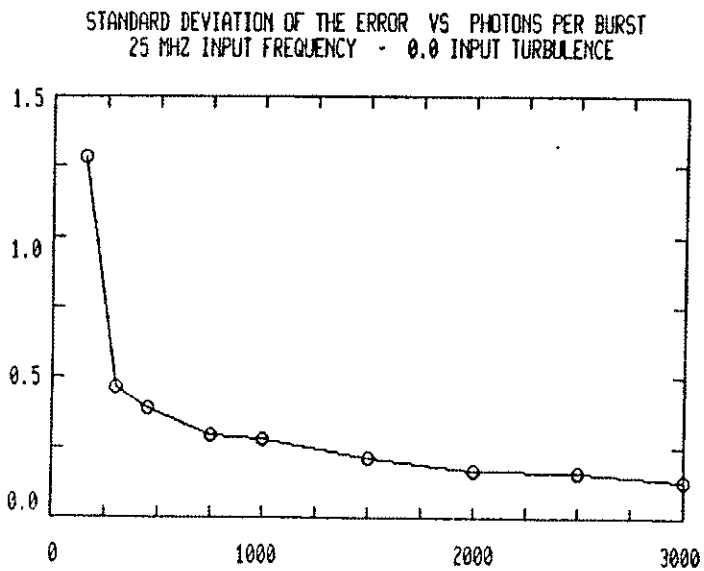


Figure 4.2. Standard deviation of the error vs number of photons per burst.

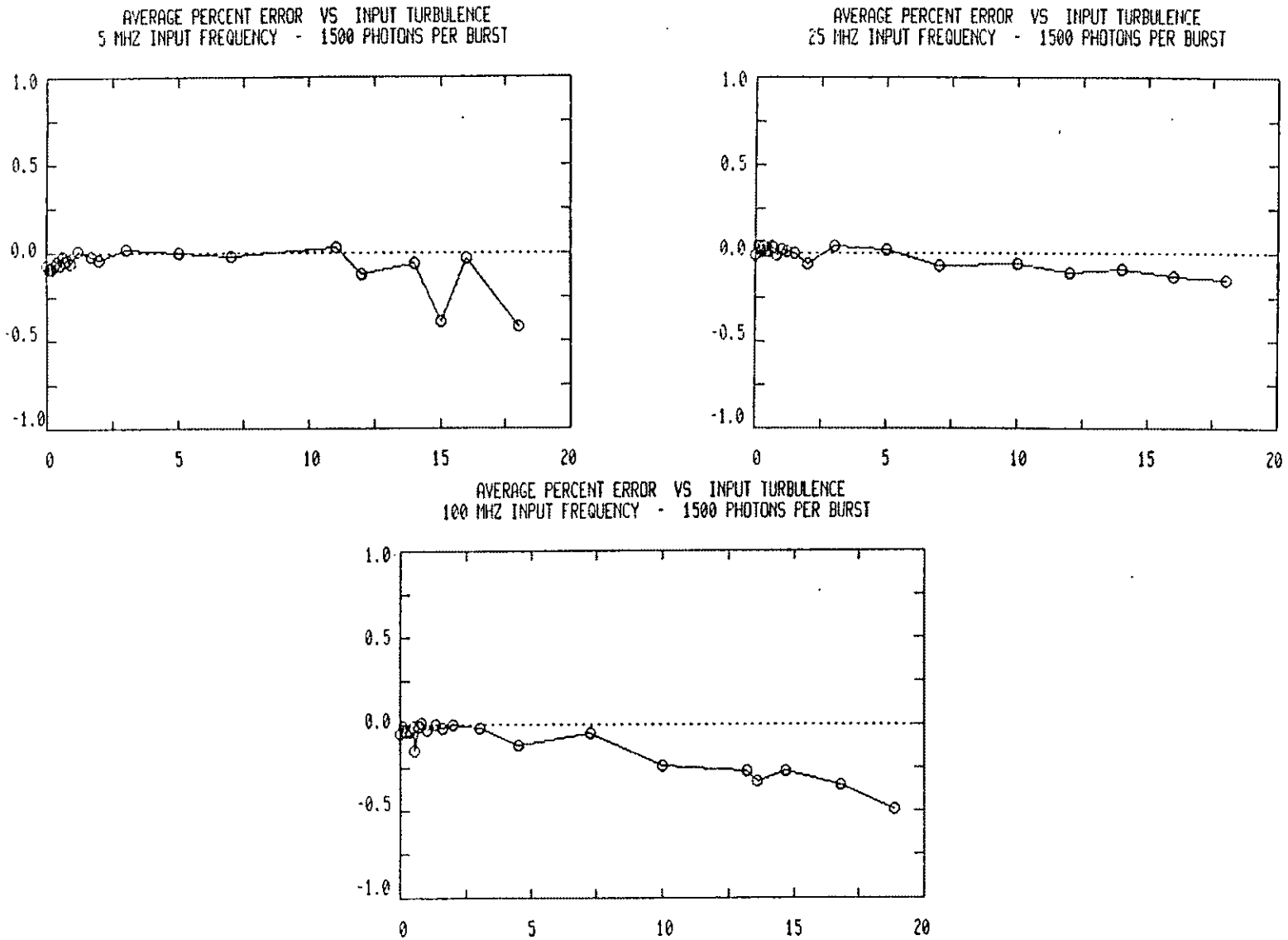


Figure 4.3. Average percent error vs input turbulence at 1500 photons per burst.

The standard deviation of the error as a function of input turbulence is shown in Figure 4.4 for signal bursts comprised of 1500 photons. As the turbulence intensity increases above 5% the wide filters are used for the processing. Note that for input turbulence higher than 5%, the standard deviation of the error increases, because the wide filters provide an error function of greater magnitude as shown in Figure 3.5. Note that for the case of average error the effect of transition from narrow to wide filters is not as apparent, because the error function of the wide filters is symmetric, and although the errors are larger in magnitude they average out.

The average percent error and the standard deviation of the error as a function of input turbulence for 300 photons per burst are shown in Figures 4.5 and 4.6 respectively. These results are not as good as in the case of 1500 photons per burst, but they follow the same trends that were discussed in the previous paragraphs. This is expected, because as shown in Figures 4.1 and 4.2 the average error and the standard deviation of the error increase as the number of photons per burst decreases.

At this point it is appropriate to compare the frequency domain processor performance in estimating the mean input frequency with that of the high speed burst

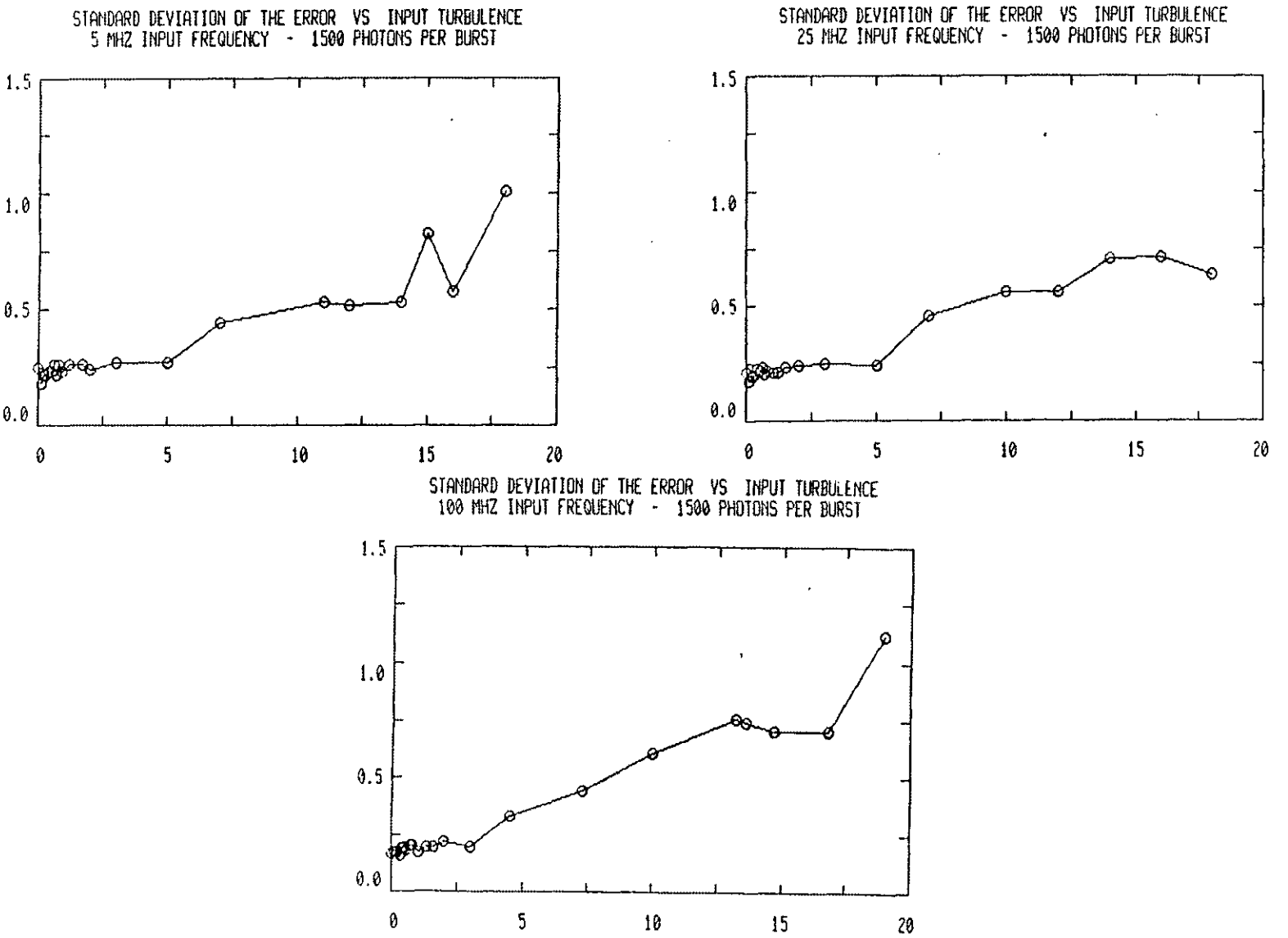


Figure 4.4. Error standard deviation vs input turbulence at 1500 photons per burst.

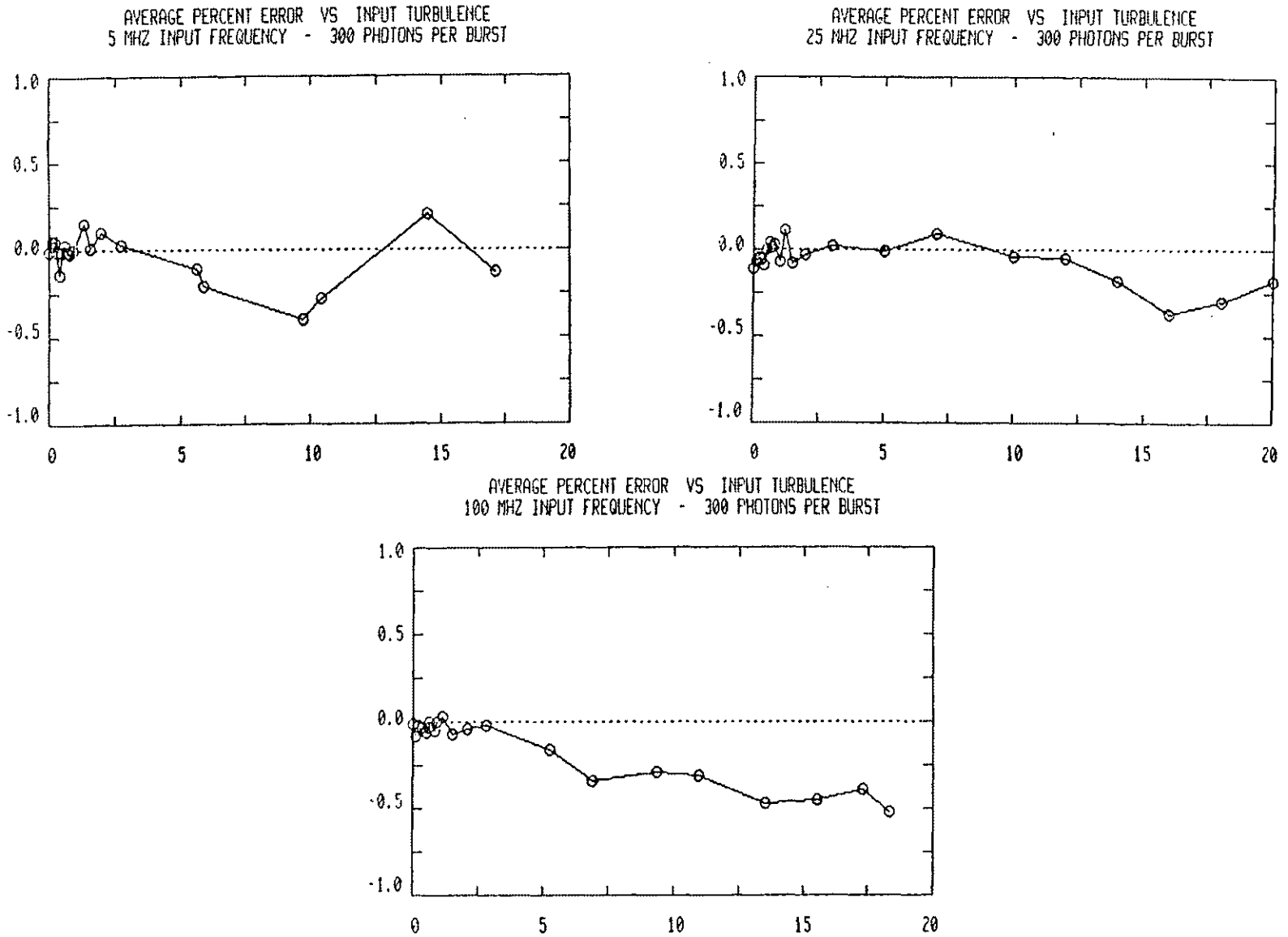


Figure 4.5. Average percent error vs input turbulence at 300 photons per burst.

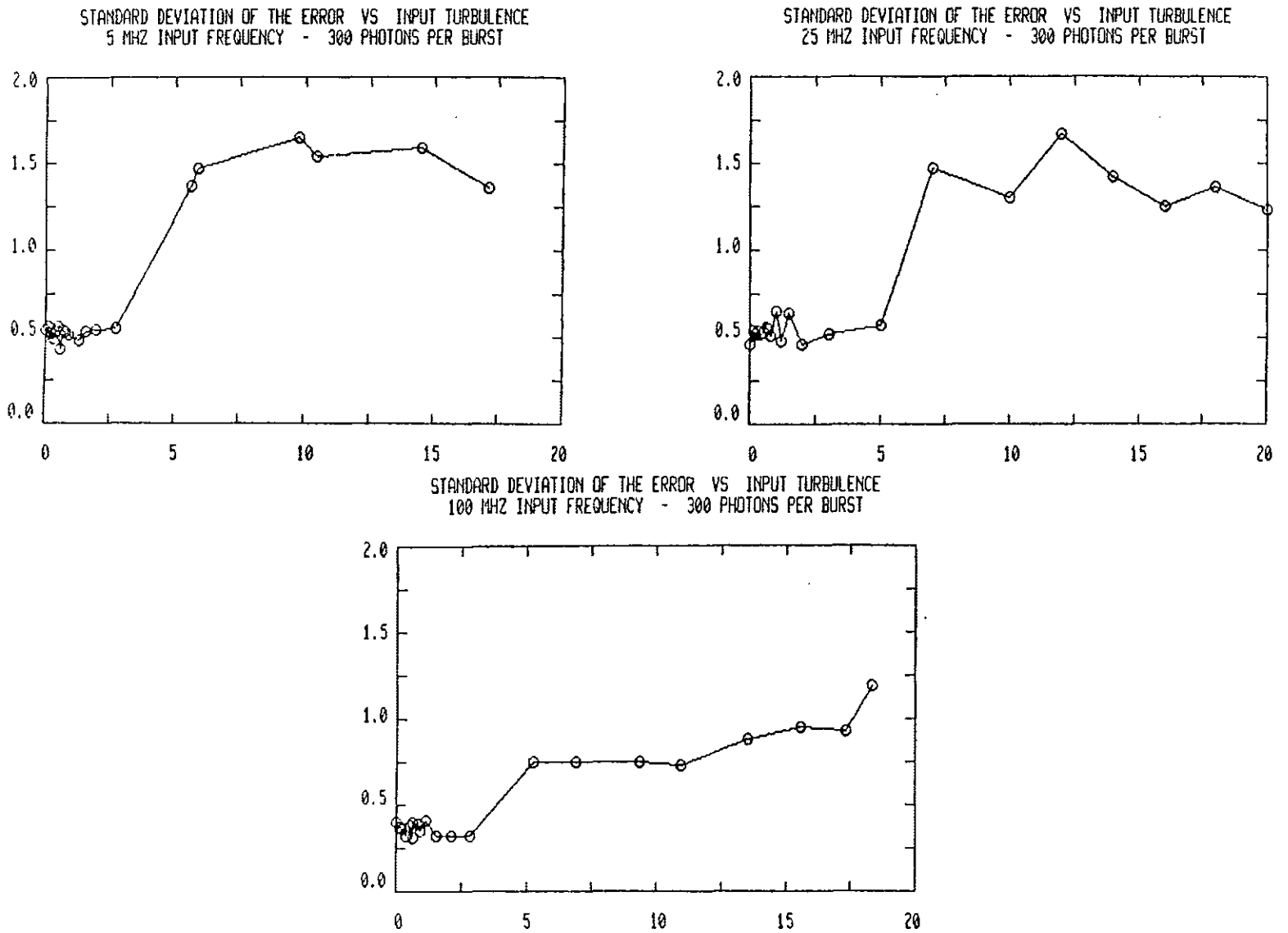


Figure 4.6. Error standard deviation vs input turbulence at 300 photons per burst.

counter. The performance characteristics of the high speed burst counter were obtained through personal conversation with J. Meyers. At zero input turbulence the high speed burst counter yields at best an average error around 0.3% with a standard deviation of 0.5%. At zero input turbulence the frequency domain processor yields an average error less than 0.2%, and a standard deviation of the error around 0.3%. When the input turbulence varies, the counter yields an average error between 0.5% and 0.6%. The frequency domain processor yields an average error less than 0.3% if the input turbulence is less than 15%, and 0.5% if the input turbulence is above 15%. As the input turbulence is varied the standard deviation of the error that the counter provides is between 0.5% and 0.6% for signals comprised of 1500 photons, but increases up to 5% at high turbulence for signals comprised of 300 photons. When the turbulence is below 5% the frequency domain processor yields standard deviation of the error around 0.3% for signals comprised of 1500 photons and 0.5% for signals comprised of 300 photons. When the turbulence is above 5% the standard deviation of the error increases to around 1% for 1500 photon signals and 1.5% for 300 photon signals.

The above discussion of the errors in the mean frequency estimation shows that the frequency domain processor is superior to the counter in almost all cases.

4.2.2 Turbulence Estimation

The next phase of the testing involved the measurement of turbulence for signal bursts comprised of 300 and 1500 photons. In the graphs to be presented the data points are represented by small circles. The solid line represents the ideal measurements where the measured turbulence is equal to the input turbulence, and will be compared to the data obtained from the processor simulation.

The measured turbulence is shown as a function of input turbulence in Figure 4.7 for signal bursts comprised of 1500 photons. The part of the graph for turbulence intensities up to 5% is shown in Figure 4.8. This graph illustrates that the frequency domain processor has a minimum measure of turbulence intensity around 0.2%.

The relationship between measured and input turbulence intensity is shown in Figure 4.9 and 4.10 for signals comprised of 300 photons. In this case the minimum measure of turbulence is around 0.5%, and for higher turbulence intensities there is not significant degradation of performance.

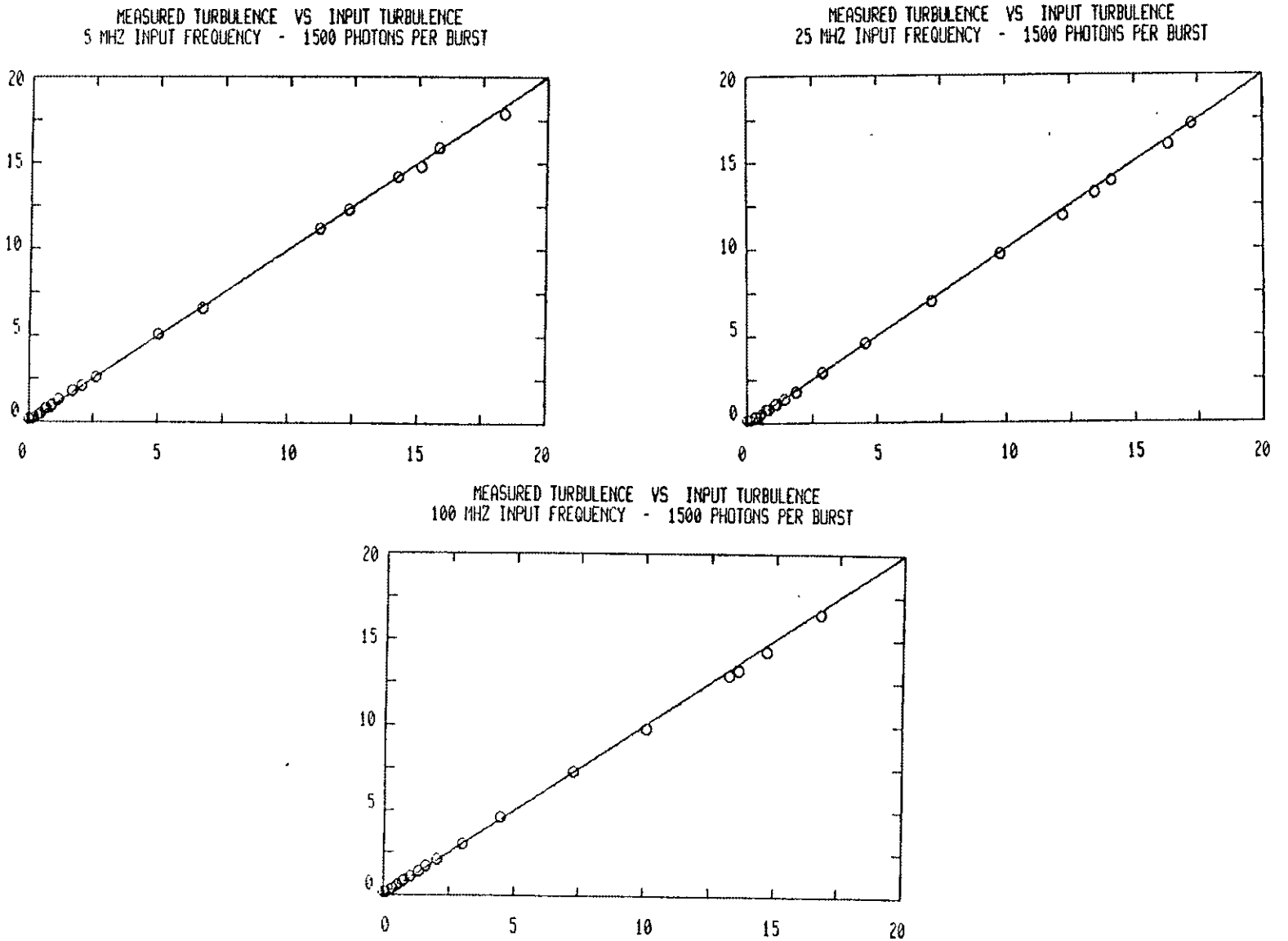
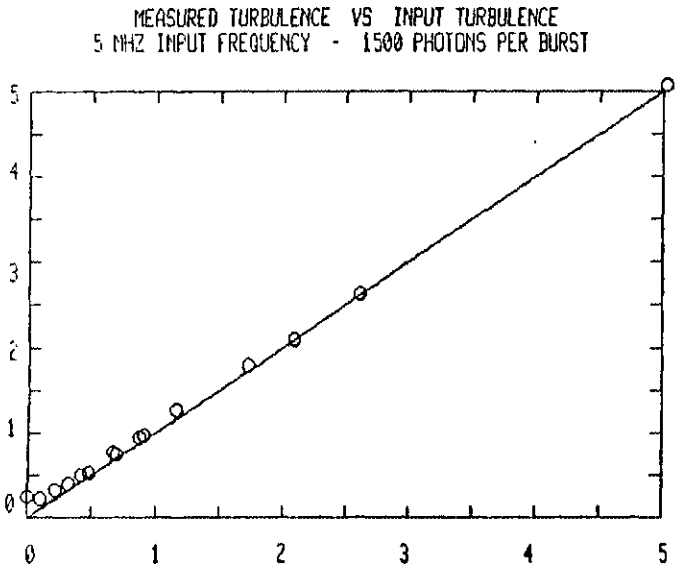
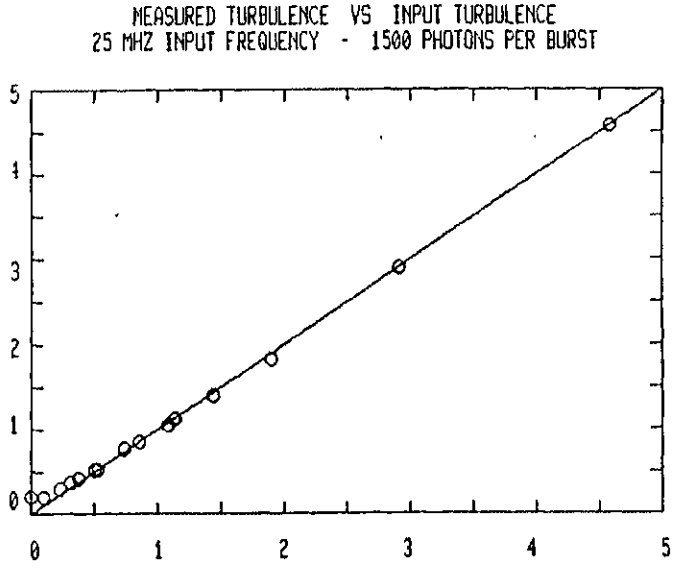


Figure 4.7. Measured turbulence vs input turbulence at 1500 photons per burst.





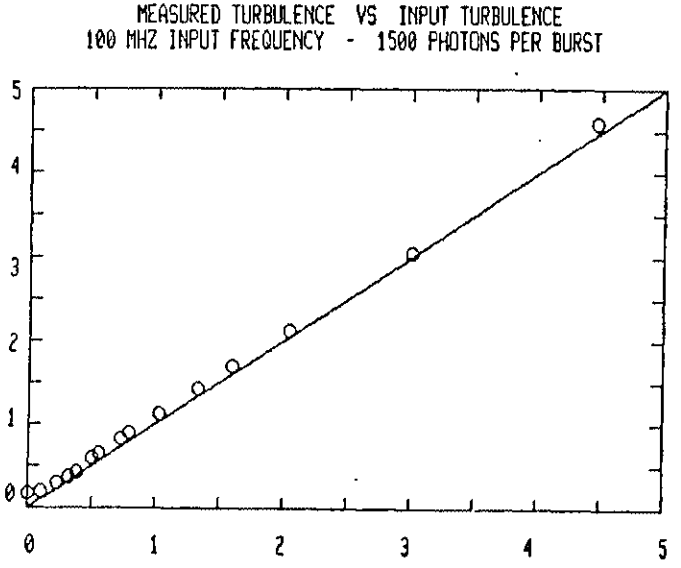


Figure 4.8. Measured turbulence vs input turbulence at 1500 photons per burst.

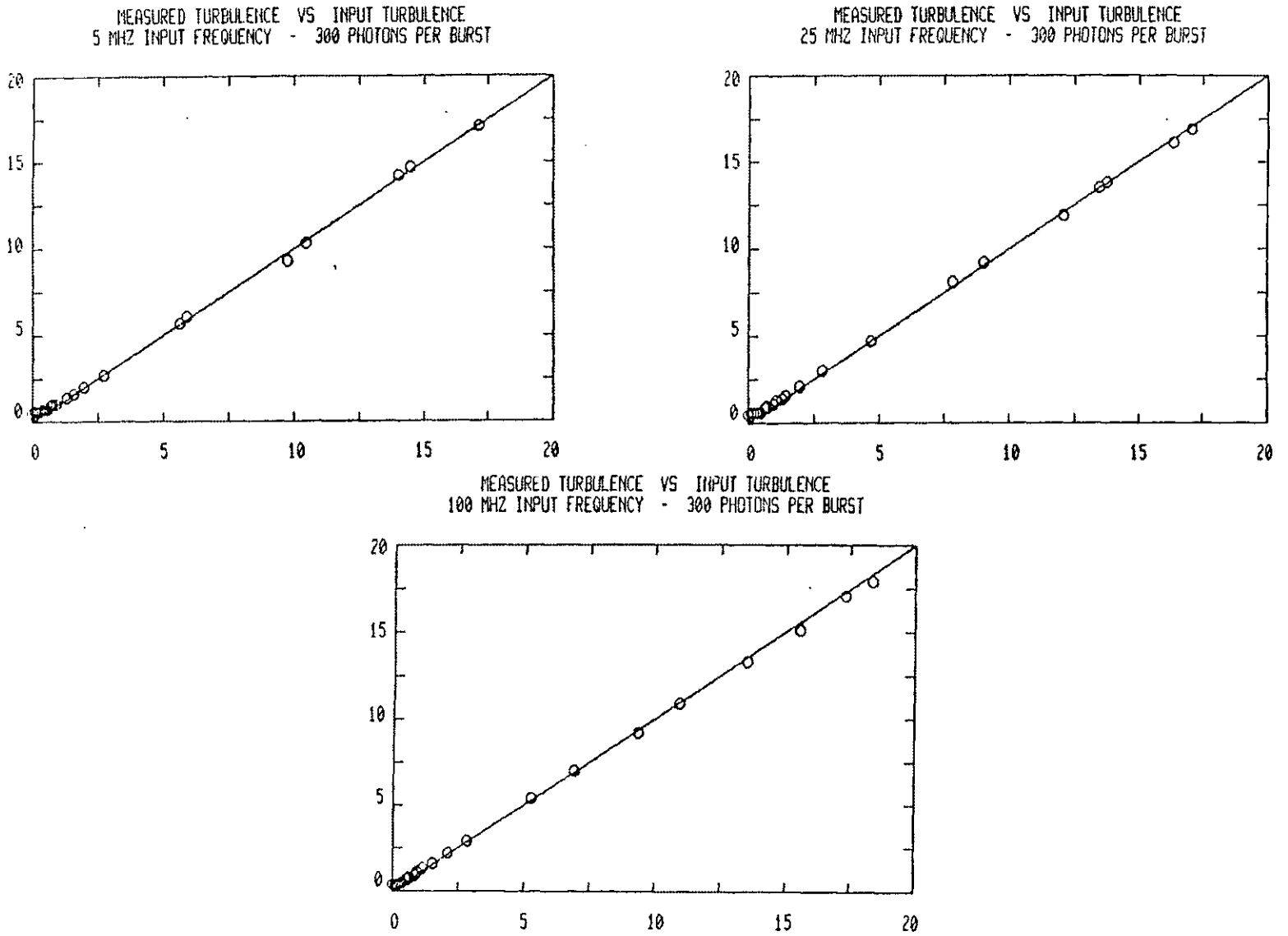


Figure 4.9. Measured turbulence vs input turbulence at 300 photons per burst.

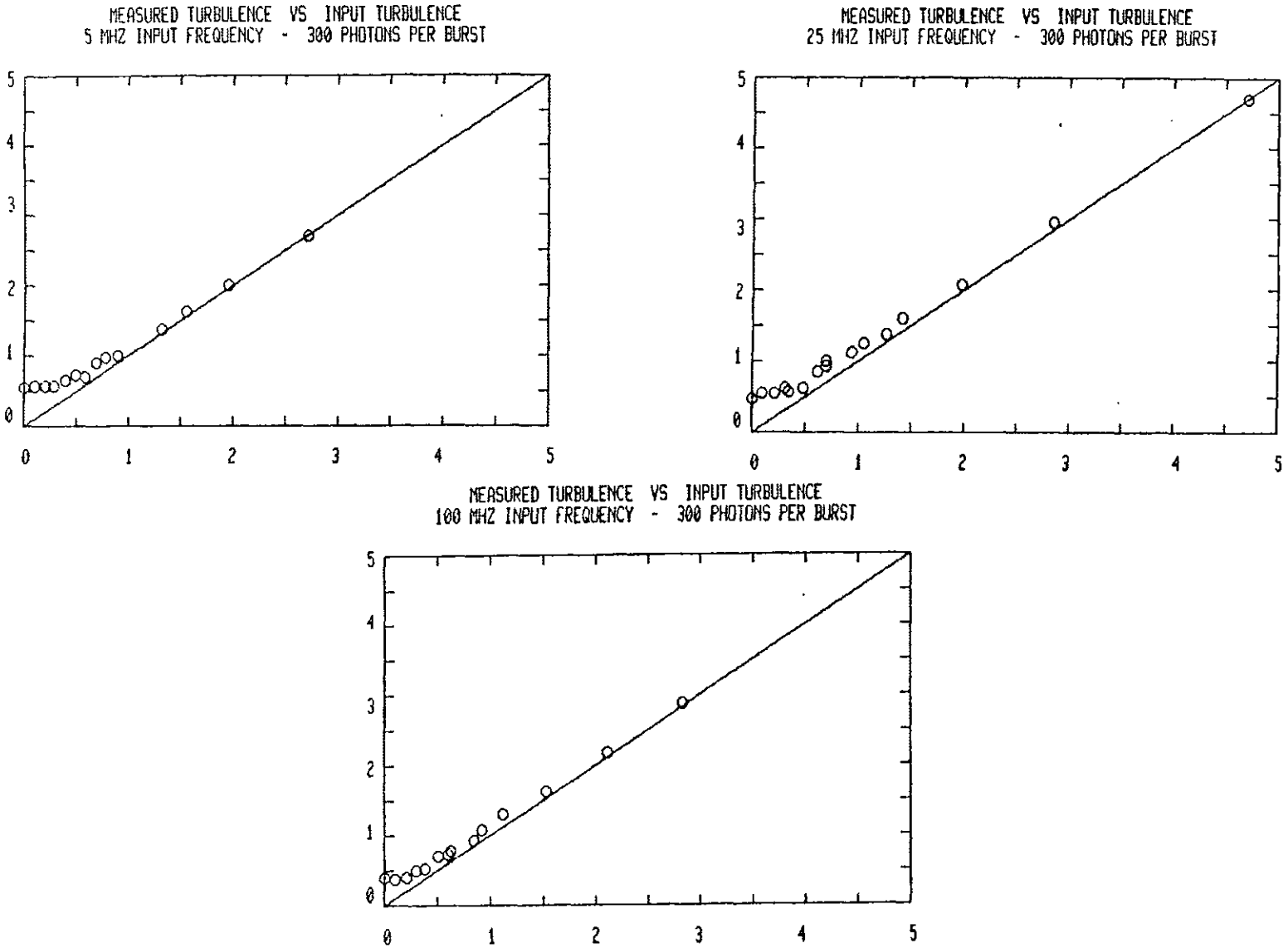


Figure 4.10. Measured turbulence vs input turbulence at 300 photons per burst.

The turbulence estimation results of the frequency domain processor are now compared with the high speed burst counter results. The counter has a minimum measure of turbulence around 0.5% when the signals are comprised of 1500 photons per burst. The counter performance depends on the value of the input frequency, and for higher input frequencies its performance becomes worse, because the time quantization of the counter clock becomes more significant with respect to the measured frequency. The frequency domain processor provides a minimum measure of turbulence around 0.2% independent of the input frequency. When the signals are comprised of 300 photons the counter shows a minimum measure of turbulence around 0.7% while the frequency domain processor minimum measure of turbulence is around 0.5%. In addition the frequency domain processor measures turbulence more accurately for low photon counts.

4.3 Summary

In this chapter the simulation results were presented and used for the evaluation of the frequency domain signal processor. The frequency domain processor yielded an average error around 0.1 to 0.2% at zero input turbulence when the number of photons per burst was varied from 150 to 3000. The standard deviation of the error was less than

0.3% for signals comprised of 750 photons or more. The average error was less than 0.5% when the input turbulence was varied from zero to 20%. The standard deviation of the error depended on the level of the input turbulence. When the input turbulence was below 5%, the standard deviation of the error was around 0.2 to 0.5% depending on the number of photons per signal burst. When the input turbulence was above five percent, the standard deviation of the error increased to values from 0.7 to 1.5% depending on the amount of turbulence and the number of photons per burst. The results also demonstrated that the frequency domain processor can measure the input turbulence very accurately. The processor minimum measure of turbulence is 0.2% for signals comprised of 1500 photons per burst, and 0.5% for signals comprised of 300 photons per burst. The above results are independent of the input frequency. The comparison between the frequency domain processor and the high speed burst counter shows that the frequency domain processor outperforms the counter in almost all cases, especially when the signal to noise ratio is low.

CHAPTER 5

CONCLUSIONS

5.1 Remarks

The goal of this research was to design a new signal processor for laser velocimeter systems. The design objectives required automatic setup procedures, real time operation, and good accuracy for input frequencies up to 100 MHz, turbulence intensities from 0 to 20%, for signals comprised of as low as 150 photons per burst.

The proposed processor employs a digital filterbank for the estimation of the energy spectrum of the input signal. The normalized output filter energies as a function of frequency are viewed as spline functions that are linearly weighted to estimate the input frequency. The frequency estimation is accomplished by linearly weighting the maximum filter energy and the energies of the two filters next to it. The mean squared error of the estimation is minimized with respect to the weighting coefficients. Two filter sets are employed, a wide set for turbulence intensities higher than 5%, and a narrow set for turbulence intensities less than 5%. Other features of the

processor include nonlinear quantization of the input waveform for efficient capture of the Doppler frequency, and capture of the signal burst in a 256 shift register for efficient processing of the input data. During the processor initialization the sampling clock is adjusted to the value that maps the mean input frequency at the center of the filterbank. Then the narrow or the wide filter set is chosen for the processing depending on the input turbulence level.

The processor performance was tested via the simulation program FDP. The simulation results demonstrate that the proposed processor meets the design objectives. The frequency domain processor performance is found superior to that of the high speed burst counter, the best available processing system to date. The results show that for low turbulence intensities the average error in the estimation of the mean input frequency is less than 0.2% even for signals with as low as 150 photons per burst. For high turbulence the average error is below 0.5%. The standard deviation of the error is between 0.3 and 0.5% for input turbulence less than 5%, and between 0.7 and 1.5% for input turbulence higher than 5%. The frequency domain processor is much better than the high speed burst counter in estimating the mean input frequency, especially in

situations of poor signal to noise ratios. The only case where the frequency domain processor is not as good as the counter is at the standard deviation of the error at high turbulence for signals that have good signal to noise ratio.

The frequency domain processor also measured the input turbulence very accurately. The processor has a minimum measure of turbulence of 0.2% for signals comprised of 1500 photons, and 0.5% for signals comprised of 300 photons. The results demonstrated that measurements of input turbulence are independent of the input frequency, and do not significantly degrade for low signal to noise ratios. These results are much better than the counter results when the signal to noise ratio is low.

Through the simulation testing of the frequency domain processor it was verified that the design objectives were accomplished.

5.2 Further Work

The first phase of further work for this project involves the implementation of the frequency domain processor. Already some implementation aspects have appeared in [13] and [14]. It is expected that within the next year a prototype for the frequency domain processor will be constructed at N.A.S.A. Langley Research Center.

Although the simulation results indicated that the frequency domain performance is within the required specifications, it is believed that further work could improve the processor performance. The deterministic model used in this thesis can be substituted by a more complete statistical model which would take into consideration the ETC statistics. This approach is expected to improve on the standard deviation of the error that the processor provides. More accuracy can be obtained if more intelligence is incorporated in the normalization procedure. For example the noise energy can be estimated from the average of the two smallest energies rather than from the smallest energy. A more conservative approach would require that both the second and third maximum energies are at the filters next to the maximum, so that the data is not processed unless it shows a well defined peak. However, the intelligence that is built in the processor results in an increase in cost and processing time.

In this thesis an error model was developed and used for the evaluation of a given filter set. The next step would involve the design of the optimal filter set. In such a design approach it is desirable to obtain the filter position, type, order, and bandwidth that minimize the

error of the approximation, given the frequency range of operation and the frequency estimation algorithm employed. The investigation of these issues could lead to a dissertation topic.

REFERENCES

[1] Huffaker, R. M.; Fuller, C. E.; and Lawrence, T. R.: "Application of Laser Doppler Velocity Instrumentation to the Measurement of Jet Turbulence," International Automotive Engineering Congress, SAE, Detroit, MI, January 13-17, 1969.

[2] Fridman, J. D.; Kinnard, K. F.; and Meister, K.: "Laser Doppler Instrumentation for the Measurement of Turbulence in Gas and Fluid Flows," Electro-Optical System Design Conference, New York, NY, September 16-18, 1969.

[3] Iten, P. D.; and Mastner, J.: "A Laser Doppler Velocimeter offering High Spatial and Temporal Resolution," Symposium o Flow - Its Measurement and Control in Science and Industry, Pittsburg, PA, May 10-14, 1971.

[4] Lennert, A. E.; Smith, F. H., Jr.; and Kalb, H. T.: "Application of Dual Scatter, Laser, Doppler Velocimeters for Wind Tunnel Measurenents," 4th International Congress on Instrumentation in Aerospace Simulation Facilities, von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium, June 21-23, 1971.

[5] Asher, J. A.: "LDV Systems Development and Testing," Workshop on the Use of the Laser Doppler Velocimeter for Flow Measurements, Purdue University, West Lafayette, IN, March 9-10, 1972.

[6] Pike, E. R.: "The Application of Photon-Correlation Spectroscopy to Laser Doppler Velocimeter," Workshop on the Use of the Laser Doppler Velocimeter for Flow Measurements, Purdue University, West Lafayette, IN, March 9-10, 1972.

[7] Yeh, Y.; and Cummins H. Z.: "Localized Flow Measurements with an He-Ne Spectrometer," Applied Physics Letters, 4, 176-78, 1964.

[8] Drain, L. E.: "The Laser Doppler Technique," John Wiley, 1980.

[9] Durst, F.: "Principles and practice of Laser Doppler Anemometry," Academic Press, 1981.

[10] Baker, R. J.: "A Filterbank Signal Processor for Laser Anemometry," AERE R7652, 1973.

[11] Burrus, C. S.; and Parks, T. W.: "DFT/FFT and Convolution Algorithms," John Wiley and Sons, 1985.

[12] Jong, M. T.: "Methods of Discrete signal and system analysis," McGraw Hill, 1982.

[13] Papoulis, A.: "Probability, Random variables, and Stochastic processes," McGraw Hill, 1984.

[14] Meyers, J. F.; and Stoughton, J. W.: "Frequency domain Laser Velocimeter Signal Processor," Third International Symposium on Applications of Laser Anemometry to Fluid Mechanics, Lisbon, Portugal, July 7-9, 1986.

[15] Meyers, J. F.; Clemmons, C. I.; Stoughton, J. W.; Kanetkar, S. V.; and Savvakis A. E.: "Frequency domain Laser Velocimeter Signal Processor, N.A.S.A. Langley Internal Report, June 1986.

[16] Rice, R. R.: "The Approximation of functions," Addison Wesley, 1964.

[17] Atlanta Signal Processors Incorporated: "Digital Filter Design Package," Version 1.02, March 1984.

[18] Stoughton, J. W.; and Parrish, E. A., Jr.: "Stochastic Pertrubation Model for Optimal Position Transfer Characteristics," IEEE Trans. on Instrumentation and Measurement," Vol. IM-26, No. 2, pp. 109-112, June 1977

[19] Mayo, W. T., Jr.: "Modeling Laser Velocimeter Signals as Triply Stochastic Poisson Processes," Proceedings of the Minnesota symposium on Laser Anemometry, Bloomington, MN, October 22-24, 1975.

APPENDICES

Appendix A: Derivation of the coefficients that minimize the mean squared error of the approximation.

Appendix B: Program MSE.FTN.

Appendix C: Program FDP.FTN.

APPENDIX A

In this appendix the mean squared error E of the linear approximation of Equation 2.8 is minimized with respect to the coefficients a_i. The mean squared error is given by the expression:

$$E = \frac{1}{b-a} \int_a^b (x-x^*)^2 dx$$

$$= \frac{1}{b-a} \int_a^b \left\{ x - \sum_{i=1}^{k} a_i R_i(x) \right\}^2 dx$$

To minimize E with respect to the j^{th} coefficient set

$$\frac{\partial}{\partial a_j} (E) = 0$$

or

$$\frac{\partial}{\partial a_j} \left[\frac{1}{b-a} \right] \int_a^b \left\{ x - \sum_{i=1}^{k} a_i R_i(x) \right\}^2 dx = 0$$

$$\frac{\partial}{\partial a_j} \int_a^b \left\{ x^2 + \left[\sum_{i=1}^{k} a_i R_i(x) \right]^2 - 2x \sum_{i=1}^{k} a_i R_i(x) \right\} dx = 0$$

The variables a_j and x are independent of one another, hence the order of integration and differentiation can be interchanged.

$$\int_a^b \left\{ \frac{\partial}{\partial a_j} (x^2) + \frac{\partial}{\partial a_j} \left[\sum_{i=1}^{k} a_i R_i(x) \right]^2 - 2x \frac{\partial}{\partial a_j} \left[\sum_{i=1}^{k} a_i R_i(x) \right] \right\} dx = 0$$

Since x does not depend on a_j,

$$\frac{\partial}{\partial a_j} (x^2) = 0$$

Similarly ,

$$\frac{\partial}{\partial a_j} \left[\sum_{i=1}^{k} a_i R_i(x) \right] = R_j(x)$$

because the a_j's are independent of a_j for $i \neq j$. Also,

$$\frac{\partial}{\partial a_j} \left[\sum_{i=1}^{k} a_i R_i(x) \right]^2 = 2 \left[\sum_{i=1}^{k} a_i R_i(x) \right] \frac{\partial}{\partial a_j} \left[\sum_{i=1}^{k} a_i R_i(x) \right]$$

$$= 2 \left[\sum_{i=1}^{k} a_i R_i(x) \right] R_j(x)$$

Therefore,

$$\int_a^b \left\{ 2\, R_j(x) \left[\sum_{i=1}^{k} a_i\, R_i(x) \right] - 2x\, R_j(x) \right\} dx = 0$$

or

$$\int_a^b \left\{ R_j(x) \left[\sum_{i=1}^{k} a_i\, R_i(x) \right] \right\} dx = \int_a^b x\, R_j(x)\, dx$$

or

$$\int_a^b \left\{ \sum_{i=1}^{k} a_i\, R_i(x)\, R_j(x) \right\} dx = \int_a^b x\, R_j(x)\, dx$$

or

$$\sum_{i=1}^{k} a_i \int_a^b R_i(x)\, R_j(x)\, dx = \int_a^b x\, R_j(x)\, dx$$

is the equation that is obtained if E is minimized with respect to a_j. If E is minimized with respect to all a_i, k equations will be obtained. For simplicity in notation let

$$I_{ij} = \int_a^b R_i(x)\, R_j(x)\, dx$$

and

$$I_{xj} = \int_a^b x\, R_j(x)\, dx$$

Then the k equations are:

$$\sum_{i=1}^{k} a_i\, I_{i1} = I_{x1}$$

$$\sum_{i=1}^{k} a_i\, I_{i2} = I_{x2}$$

$$\vdots$$

$$\sum_{i=1}^{k} a_i\, I_{ik} = I_{xk}$$

These equations can be expressed in matrix form as:

$$\begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1k} \\ I_{21} & & & I_{2k} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ I_{2k} & I_{3k} & \cdots & I_{kk} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_k \end{bmatrix} = \begin{bmatrix} I_{x1} \\ I_{x2} \\ \cdot \\ \cdot \\ \cdot \\ I_{xk} \end{bmatrix} \qquad (A.1)$$

APPENDIX B

```
C     PROGRAM MSE CALCULATES THE WEIGHTING COEFFICIENTS
C THAT MINIMIZE THE MEAN SQUARED ERROR OF THE APPROXIMATION
      DIMENSION R(3,300),SPAD(300),XFREQ(300)
      DIMENSION COEF(3,3),BCOL(3),CINV(3,3),W(3),BET(7)
      DIMENSION A1(3),A2(3),A3(3)
      INTEGER*4 SEED
      REAL X,BET
      REAL R,PROD,SINT,TEMP,XFREQ
      REAL   COEF,CINV,BCOL
      COMMON/BLK1/A1,A2,B0
C
C
      WRITE(5,*) ' ENTER STARTING FREQUENCY OF THE APPROXIMATION'
      READ(5,*)XINIT
      WRITE(5,*) ' ENTER # OF POINTS OF THE APPROXIMATION'
      READ(5,*)NPOINT
      WRITE(5,*) ' ENTER FREQUENCY INCREMENT'
      READ(5,*)DELX
      WRITE(5,*) ' ENTER # OF MIDDLE FILTER IN THE APPROXIMATION'
C
      OPEN(UNIT=2,FILE='SPLINE.DAT',STATUS='OLD')
C
      DO 770 N=1,NPOINT
      DELN=FLOAT(N)-1.0
C
C READ FILTER ETCs
699   READ(2,*)X,(BET(J), J=1,7)
      IF(X.LT.(XINIT-0.5*DELX))GOTO 699
C
C NORMALIZE ETCs
      BMIN=BET(1)
      DO 345 JI=1,7
      IF(BET(JI).GT.BMIN)GOTO 345
      BMIN=BET(JI)
      JMIN=JI
345   CONTINUE
C
      DO 346 JI=1,7
346   BET(JI)=BET(JI)-BET(JMIN)
C
      BSUM=0.
      DO 289 JI=(LFIL-1),(LFIL+1)
289   BSUM=BSUM+BET(JI)
C
C GENERATE SPLINE FUNCTIONS
      R(1,N)=BET(LFIL-1)/BSUM
      R(2,N)=BET(LFIL)/BSUM
      R(3,N)=BET(LFIL+1)/BSUM
C
      WRITE(5,657)X,(R(J,N), J=1,3)
657   FORMAT(8(2X,F9.4))
C
      XFREQ(N)=X
      SPAD(N)=0.0
770   CONTINUE
C
      CLOSE(UNIT=2)
C
```

```
C CALCULATE THE COEFFICIENTS OF THE C MATRIX
C
      DO 810 I=1,3
      DO 810 J=1,3
C
         DO 803 N=1,NPOINT
803   SPAD(N) = R(I,N) * R(J,N)
C
      SINT=0.0
      DO 801 N=1,NPOINT-1
      TEMP=(SPAD(N)+SPAD(N+1))/2.0
801   SINT=SINT+(TEMP*DELX)
C
      COEF(I,J)=SINT
810   CONTINUE
C
C
C CALCULATE THE COEFFICIENTS OF THE B MATRIX
C
      DO 850 I=1,3
C
      DO 855 N=1,NPOINT
855   SPAD(N)=R(I,N)*XFREQ(N)
C
      SINT=0.0
      DO 857 N=1,NPOINT-1
      TEMP=(SPAD(N)+SPAD(N+1))/2.
875   SINT=SINT+(TEMP*DELX)
C
      BCOL(I)=SINT
850   CONTINUE
C
C
C SOLVE FOR THE WEIGHTING COEFFICIENTS
C
      CALL MATINV(COEF,CINV,3,NMAX)
C
      DO 933 I=1,3
C
      W(I)=0.0
      DO 930 J=1,3
930   W(I)=W(I)+BCOL(J)*CINV(I,J)
C
      WRITE(5,*)
      WRITE(5,*)I,W(I)
933   CONTINUE
C
C CALCULATE THE ERROR FUNCTION OF THE APPROXIMATION
C
      DO 983 JI=1,NPOINT
      XEST=0.0
      DO 984 L=1,3
984   XEST=XEST+W(L)*R(L,JI)
C
      XERR=XEST-XFREQ(JI)
      ERR=100.*XERR/XFREQ(JI)
      WRITE(5,'(4(X,F9.5))')XFREQ(JI),XEST,XERR,ERR
983   CONTINUE
C
      STOP
      END
```

APPENDIX C

```
C********* FREQUENCY DOMAIN PROCESSOR SIMULATION PROGRAM ***************
        DIMENSION FIN(100),FEST(100),FERR(100)
        DIMENSION XSCAT(514),FMEAS(100),SMEAS(100)
        DIMENSION ENERGY(100),SCAT(256),HT(100)
        DIMENSION WE1(4,7),WE2(4,7),WE3(4,7)
        DIMENSION XIN(7),YIN(7),X(7),Y(7),XN1(7),XN2(7),XN3(7)
        DIMENSION YHOLD(7),YINT(7),TURBU(23),AMPLIT(99),AGC(99)
        DIMENSION B2(7),B1(7),B0(7),C1(7),C0(7)
        REAL LEVEL(15),NOISE,TURBU
        REAL*8 XN
        INTEGER*4 SEED
        CHARACTER IFILE*12
        DATA TURBU/0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,
     *  1.2,1.5,2.0,3.0,5.0,7.0,10.0,12.0,14.0,16.0,18.0,20.0/
        COMMON/BLK3/B2,B1,B0,C1,C0
        SEED=1234567890
C
C
C********* INITIALIZATION ***********************************************
C
        WRITE(5,*) ' ENTER PARTICLE VELOCITY'
        READ(5,*)PVELX
        FBAR=PVELX/8.424
        WRITE(5,*) ' ENTER AMPLITUDE'
        READ(5,*)FONE
        WRITE(5,*) ' ENTER MANUAL GAIN'
        READ(5,*)GAINM
        WRITE(5,*) ' ENTER INITIAL AND FINAL TURBULENCE INDEX '
        READ(5,*)JTURS
        READ(5,*)JTURE
C
C READ WEIGHTING COEFFICIENTS FOR THE WIDE FILTER SET
        WE1(1,2)=0.0376136
        WE2(1,2)=0.0602266
        WE3(1,2)=0.0760162
        WE1(1,3)=0.0578469
        WE2(1,3)=0.0802682
        WE3(1,3)=0.0972581
        WE1(1,4)=0.0787457
        WE2(1,4)=0.0999237
        WE3(1,4)=0.1179420
        WE1(1,5)=0.0990443
        WE2(1,5)=0.1200941
        WE3(1,5)=0.1386377
        WE1(1,6)=0.1196454
        WE2(1,6)=0.1395580
        WE3(1,6)=0.1607123
C
C READ WEIGHTING COEFFICIENTS FOR THE FIRST NARROW FILTER SET
        WE1(2,2)=0.0680715
        WE2(2,2)=0.0769496
        WE3(2,2)=0.0837466
        WE1(2,3)=0.0761374
        WE2(2,3)=0.0840291
        WE3(2,3)=0.0917368
        WE1(2,4)=0.0839507
        WE2(2,4)=0.0919789
        WE3(2,4)=0.1000021
        WE1(2,5)=0.0917053
        WE2(2,5)=0.1000199
        WE3(2,5)=0.1080692
        WE1(2,6)=0.0998476
        WE2(2,6)=0.1079804
        WE3(2,6)=0.1161274
```

```
C READ WEIGHTING COEFFICIENTS FOR THE SECOND NARROW FILTER SET
        WE1(3,2)=0.0761374
        WE2(3,2)=0.0840291
        WE3(3,2)=0.0917360
        WE1(3,3)=0.0839807
        WE2(3,3)=0.0919769
        WE3(3,3)=0.1000831
        WE1(3,4)=0.0917053
        WE2(3,4)=0.1000190
        WE3(3,4)=0.1080092
        WE1(3,5)=0.0998476
        WE2(3,5)=0.1079804
        WE3(3,5)=0.1161276
        WE1(3,6)=0.1077082
        WE2(3,6)=0.1160439
        WE3(3,6)=0.1239043
C
C READ WEIGHTING COEFFICIENTS FOR THE THIRD NARROW FILTER SET
        WE1(4,2)=0.0839807
        WE2(4,2)=0.0919769
        WE3(4,2)=0.1000831
        WE1(4,3)=0.0917053
        WE2(4,3)=0.1000190
        WE3(4,3)=0.1080092
        WE1(4,4)=0.0998476
        WE2(4,4)=0.1079804
        WE3(4,4)=0.1161276
        WE1(4,5)=0.1077082
        WE2(4,5)=0.1160439
        WE3(4,5)=0.1239043
        WE1(4,6)=0.1157973
        WE2(4,6)=0.1239229
        WE3(4,6)=0.1320736
C
        F1=FONE*1.0E-08
        F2=1.0
        BRAGF=0.0
        BRAGF=BRAGF*1.0E+6
C
        GAINL=0.5*GAINM
        GAINH=2.0*GAINM
C
        OPEN(UNIT=2,FILE='DATA.DAT')
C
C START TURBULENCE LOOP
        DO 90 JTUR=JTURS,JTURE
        TIIN=TURBU(JTUR)
C
        WRITE(2,*)
        WRITE(2,*)' MEAN FREQ =',FBAR,'  TURBULENCE =',TURBU(JTUR)
C
C READ WIDE FILTER SET COEFFICIENTS
        CALL IFILW
C
C SET QUANTIZATION LEVELS
        LEVEL(1)=0.1
        LEVEL(2)=0.2
        LEVEL(3)=0.4
        DO 92 I=4,15
92      LEVEL(I)=10000.
C
C INITIALIZE CLOCK, GAIN
        CLOCK=1.E+09
        NBIT=256
        GAIN=GAINM
        M=7
        ITEST=1
        FRSPC=8.424E-06
        DA=200.E-06
        H=6.626E-34
        C=3.0E+08
        G=2.E+06
        E=1.6E-19
        XLIG=0.5145E-06*0.5*0.21/H/C
```

```
      LFILT=1
101   AVGFIL=0.
      SIGFIL=0.
      ACOUNT=0.
100   SN=0.
      F2IN=0.
      F1IN=0.
      FNIN=0.
      E2IN=0.
      E1IN=0.
      ERIN=0.
      PHOTON=0.
      NOISE=0.
      WIDE=0.
      PEAK=0.
C
      TI=TIIN*PVELX/100.
C
      DO 10 I0=1,100
C
C********** IDEAL WAVEFORM GENERATION ********************************
C
      RNU=SQRT(-2.*ALOG(RAN(SEED)))
      RNU=RNU*COS(2.*3.14159*RAN(SEED))
C
      PVEL=PVELX+TI*RNU
      CONS=2.*3.14159*(PVEL/FRSPC+BRAGF)
      AVPSI=2.*3.14159*RAN(SEED)
      FREQIN=1.0E-06*(PVEL/FRSPC+BRAGF)
C
      TRNST=DA/PVEL
      DRELT=TRNST/512.
      TRELS=-TRNST/2.
      DRELX=DA/512.
      XRELS=-DA/2.
C
      DO 12 I=1,512
      TEMP=-8.0*XRELS*XRELS/DA/DA
      PED=0.0
      IF((ABS(TEMP)).LT.80.)PED=EXP(TEMP)
      XSCAT(I)=PED*F1*(1.0+F2*COS(CONS*TRELS-AVPSI))
      XSCAT(I)=XL1G*XSCAT(I)*DRELT
      TRELS=TRELS+DRELT
12    XRELS=XRELS+DRELX
C
C********** REAL WAVEFORM GENERATION *********************************
C
      DO 13 I=2,511
13    XSCAT(I)=0.5*(XSCAT(I+1)+XSCAT(I))+XSCAT(I-1)
      XSCAT(512)=XSCAT(511)
      XSCAT(513)=XSCAT(512)
C
      XN=0.0
      RN=RAN(SEED)
      RNN=ALOG(1.0/RN)
C
      DO 14 I=1,512
      SUM=0.0
      XL=0.5*(XSCAT(I+1)+XSCAT(I))
C
15    IF(RNN.GT.XL)GOTO 14
      SUM=SUM+1.0
      XN=XN+1.0
      RN=RAN(SEED)
      RNN=RNN+ALOG(1.0/RN)
      GOTO 15
C
14    XSCAT(I)=SUM
C
      LO=2.0E-09/DRELT
      IF(LO.LT.1)GOTO 16
C
```

```
        KQ=2*LQ+1
        DO 17 K=1,KQ
        J=K-1-LQ
        XJ=J
        XQ=LQ
        XKQ=KQ
17      HT(K)=2.0*(1.0-ABS(XJ/XQ))/(XKQ-1.0)
C
        IQQ=LQ+1
        JQ=511-LQ
        K=1
        DO 18 K1=IQQ,JQ
        XSCAT(K)=0.0
        DO 19 K2=1,KQ
        J=K2-1-LQ
19      XSCAT(K)=HT(K2)*XSCAT(K1-J)+XSCAT(K)
18      K=K+1
C
        K=K-1
        IQQ=IQQ+1
        DO 20 I=JQ,IQQ,-1
        XSCAT(I)=XSCAT(K)
20      K=K-1
        DO 21 I=1,IQQ
21      XSCAT(I)=XSCAT(IQQ+1)
        DO 22 I=JQ,512
22      XSCAT(I)=XSCAT(JQ-1)
C
C
16      R=50.0
        FAC=E*G*R/DRELT
        XSCMAX=0.0
        DO 25 I=1,512
25      XSCAT(I)=FAC*XSCAT(I)
C
C PMT SATURATION
C
        DO 23 K=1,512
23      IF(XSCAT(K).GT.0.8) XSCAT(K)=0.8
C
C********** AGC AMPLIFICATION ***********************************************
C
        DO 29 K=1,512
29      XSCAT(K)=GAIN*XSCAT(K)
C
C********** WAVEFORM SAMPLING ************************************************
C
        DO 26 K=1,512
26      SCAT(I)=0.0
C
        STIME=1.0/CLOCK
C
        K=1
        DO 60 I=1,NBIT
        T1=STIME*FLOAT(I-1)
        DO 61 J=K,512
        T2=DRELT*FLOAT(J-1)
        IF(T2.GE.T1)THEN
        SCAT(I)=XSCAT(J)
        K=J
        GOTO 60
        ENDIF
61      CONTINUE
60      CONTINUE
C
C********** WAVEFORM QUANTIZATION ********************************************
C
        CALL VMAX(SCMAX,SCAT,1,NBIT)
        PVOLT=SCMAX
        PAMP=0.
C
```

```
        DO 62 I=1,NBIT
        AD=0.
        DO 63 J=1,15
        IF(SCAT(I).GT.LEVEL(J))THEN
        AD=AD+1.0
        ELSE
        GOTO 64
        ENDIF
63      CONTINUE
64      SCAT(I)=AD
        IF(AD.EQ.3.0) PAMP=PAMP+1.0
62      CONTINUE
C
C********** AGC GAIN ADJUSTMENT ***********************************************
C
        IF(PAMP.LT.32.0)GAIN=GAIN+0.1875
        IF(PAMP.GT.95.0)GAIN=GAIN-0.1875
        IF(GAIN.LT.GAINL)THEN
        GAIN=GAINL
        WRITE(5,*) ' REQUIRED GAIN IS TOO LOW'
        ENDIF
        IF(GAIN.GT.GAINH)THEN
        GAIN=GAINH
        WRITE(5,*) ' REQUIRED GAIN IS TOO HIGH'
        ENDIF
C
C********* PROCESSING THROUGH THE FILTERBANK ***************************
C
        CALL FILT(SCAT,ENERGY)
C
C********** INPUT FREQUENCY ESTIMATION ***********************************
C CALCULATE THE RATIO BETWEEN THE LARGEST FILTER ENERGY AND THE SUM
C OF ALL THE FILTER ENERGIES
        CALL VSUM(ESUM,ENERGY,1,7)
        IF(ESUM.LE.0.0)GOTO 10
        ESUM=1/ESUM
        CALL VSMY(ESUM,ENERGY,1,YHOLD,1,7)
        CALL VMAX(JMAX,YHOLD,1,7)
        EMAX=YHOLD(JMAX)
C
        CALL VMAX(JMAX,ENERGY,1,7)
C
        IF(ITEST.EQ.1)GOTO 711
C
C FIND THE SECOND LARGEST FILTER ENERGY
        CALL VMOV(ENERGY,1,YHOLD,1,7)
        YHOLD(JMAX)=0.0
        CALL VMAX(JMAX2,YHOLD,1,7)
C
C EXCLUDE 'BAD' DATA POINTS
        IF(EMAX.LT.0.3)THEN
        NOISE=NOISE+1.0
        GOTO 10
        ENDIF
        IF(JMAX.EQ.1.OR.JMAX.EQ.7)THEN
        WIDE=WIDE+1.0
        GOTO 10
        ENDIF
        IF(JMAX2.LT.(JMAX-1).OR.JMAX2.GT.(JMAX+1))THEN
        PEAK=PEAK+1.0
        GOTO 10
        ENDIF
C
C NORMALIZE THE FILTER ENERGIES
        CALL VMOV(ENERGY,1,YHOLD,1,7)
        EMIN=ENERGY(1)
        JMIN=1
        DO 705 JI=2,7
        IF(ENERGY(JI).GT.EMIN)GOTO 705
        EMIN=ENERGY(JI)
        JMIN=JI
705     CONTINUE
C
```

```
      DO 707 JI=1,7
      ENERGY(JI)=ENERGY(JI)-EMIN
707   CONTINUE
C
      ESUM=0.
      DO 708 JI=JMAX-1,JMAX+1
709   ESUM=ESUM+ENERGY(JI)
C
      DO 710 JI=1,7
710   ENERGY(JI)=ENERGY(JI)/ESUM
C
C ESTIMATE THE INPUT FREQUENCY
      STATFR=WE1(LFILT,JMAX)*ENERGY(JMAX-1)
      STATFR=STATFR+WE2(LFILT,JMAX)*ENERGY(JMAX)
      STATFR=STATFR+WE3(LFILT,JMAX)*ENERGY(JMAX+1)
C
      STATFR=STATFR*CLOCK*1.0E-06
C
C********* ADJUST SAMPLING CLOCK ***********************************
C THIS PART OF THE PROGRAM IS EXECUTED ONLY DURING THE SYSTEM SET UP
711   IF(ITEST.EQ.1)THEN
      CLOFR=1.0E-06*CLOCK
      WRITE(2,'(I4,8F9.3)')IQ,CLOFR,(ENERGY(JI),JI=1,7)
C
            IF(EMAX.LT.0.3.OR.JMAX.LT.4)THEN
            CLOCK=0.8*CLOCK
            IF(CLOCK.LT.1.0E+6)CLOCK=1.0E+9
            GOTO 10
            ELSE
            ITEST=0
            WRITE(2,*) 'RUN 10 POINTS TO ESTIMATE MEAN'
            GOTO 100
            ENDIF
      ENDIF
C
C
       IF(ITEST.EQ.0)THEN
       AVGFIL=AVGFIL+STATFR
       ACOUNT=ACOUNT+1.
            IF(ACOUNT.EQ.10.)THEN
            AVGFIL=0.1*AVGFIL/CLOFR
            WRITE(2,*) ' AVERAGE FS/FC = ',AVGFIL
                  IF(AVGFIL.LT.0.085)CLOCK=CLOCK*0.8
                  IF(AVGFIL.GT.0.115)CLOCK=CLOCK/0.8
                  CLOFR=CLOCK*1.0E-06
                  IF(AVGFIL.GE.0.085.AND.AVGFIL.LE.0.115)THEN
                  ITEST=-1
            CLOFR=CLOCK*1.0E-06
            WRITE(2,*) ' FIX CLOCK FREQ = ',CLOFR
                  GOTO 101
                  ENDIF
            GOTO 101
            ENDIF
      ENDIF
C
C********* CHOOSE FILTER SET ****************************************
C THIS PART OF THE PROGRAM IS EXECUTED ONLY DURING THE SYSTEM SET UP
      IF(ITEST.EQ.-1)THEN
            AVGFIL=AVGFIL+STATFR
            SIGFIL=SIGFIL+(STATFR)**2
            ACOUNT=ACOUNT+1.
                  IF(ACOUNT.EQ.30.0)THEN
                  AVGFR=AVGFIL/ACOUNT
                  SIGFIL=(ACOUNT*SIGFIL-AVGFIL*AVGFIL)/ACOUNT/ACOUNT
            IF(SIGFIL.GT.0.0)SIGFIL=SQRT(SIGFIL)
                  SIGFIL=100.*SIGFIL/AVGFR
                  ITEST=-2
            LFILT=1
            WRITE(2,*) ' TURB CHECK YIELDS TEST SIGMA = ',SIGFIL
      IF(SIGFIL.GT.5.0)GOTO 100
            FSFC=AVGFR/CLOFR
      IF(SIGFIL.LE.5.0)THEN
            IF(FSFC.LT.0.096)THEN
```

```
C LOAD COEFFICIENTS OF THE FIRST NARROW FILTER SET
                LFILT=2
                CALL IFILN1
                ENDIF
                IF(FSFC.GE.0.096.AND.FSFC.LE.0.104)THEN
C LOAD COEFFICIENTS OF THE SECOND NARROW FILTER SET
                LFILT=3
                CALL IFILN2
                ENDIF
                IF(FSFC.GT.0.104)THEN
C LOAD COEFFICIENTS OF THE THIRD NARROW FILTER SET
                LFILT=4
                CALL IFILN3
                ENDIF
        ENDIF
        WRITE(2,*) ' FILTER SET TOO WIDE, DROP TO NEXT LEVEL'
        WRITE(2,*) ' NARROW FILTER = ',LFILT,' FSFC=',FSFC
                GOTO 101
                ENDIF
        ENDIF
C
C********* CALCULATE STATISTICS AND ERRORS ***************************
C
72      CALVS=(STATFR*1.0E+06-BRAGF)*FRSPC
        PHOTON=PHOTON+XN
        SN=SN+1.0
        ISN=SN
        SMEAS(ISN)=CALVS
        FIN(ISN)=FREQIN
        FEST(ISN)=STATFR
        FERR(ISN)=100.*(STATFR-FREQIN)/FREQIN
        ERROR=100.*(STATFR-FREQIN)/FREQIN
        WRITE(2,'(7F9.3)')(YHOLD(JI),JI=1,7)
        WRITE(2,'(7F9.3)')(ENERGY(JI),JI=1,7)
        WRITE(2,'(X,I3,3F9.3,I3,F9.5)')IQ,XN,STATFR,FREQIN,JMAX,ERROR
        FNIN=FNIN+1.0
        FREQIN=FREQIN-FBAR
        F1IN=F1IN+FREQIN
        F2IN=F2IN+FREQIN*FREQIN
        ERIN=ERIN+1.0
        E1IN=E1IN+ERROR
        E2IN=E2IN+ERROR*ERROR
C
10      CONTINUE
C
        PHOTON=PHOTON/SN
        FRAVG=F1IN/FNIN+FBAR
        FRSIG=(FNIN*F2IN-F1IN*F1IN)/FNIN/FNIN
C
        IF(FRSIG.LE.0.0)THEN
        FRSIG=0.0
        ELSE
        FRSIG=SQRT(FRSIG)
        ENDIF
        FREQTI=100.*FRSIG/FRAVG
C
        ERRAVG=E1IN/ERIN
        FRRSIG=(ERIN*E2IN-E1IN*E1IN)/ERIN/ERIN
C
        IF(ERRSIG.LE.0.0)THEN
        ERRSIG=0.0
        ELSE
        ERRSIG=SQRT(ERRSIG)
        ENDIF
C
        ISN=SN
        CALL VSUM(S1,SMEAS,1,ISN)
        AVGV=S1/SN
        DO 95 JI=1,ISN
95      SMEAS(JI)=SMEAS(JI)-AVGV
        CALL VDOT(S2,SMEAS,1,SMEAS,1,ISN)
```

```
      SIGV=SQRT(S2/SN)
      AVGF=AVGV/FRSPC*1.0E-06
      SIGF=SIGV/FRSPC*1.0E-06
      TI=100.*SIGF/AVGF
      FVEL=8.424*FRAVG
      Z1=FONE
      Z2=GAIN
      Z3=PHOTON
      Z4=SN
      Z5=AVGF
      Z6=SIGF
      Z7=TI
      Z8=FNIN
      Z9=FRAVG
      Z10=FRSIG
      Z11=FREQTI
      Z12=ERRAVG
      Z13=ERRSIG
      WRITE(2,333)Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11,Z12,Z13
333   FORMAT(2X,F6.4,X,F5.1,X,F7.1,X,2(F5.1,X,3F8.3,X),2F10.4)
      WRITE(2,*)' NOISE=',NOISE,' WIDE=',WIDE,' PEAK=',PEAK
C
90    CONTINUE
C
      WRITE(2,*)
C
      CLOSE(UNIT=2)
      STOP
      END
C
C*************************************************************************
C
      SUBROUTINE FILT(SR,ENERGY)
      DIMENSION SR(256),ENERGY(7)
      DIMENSION B0(7,4),B1(7,4),B2(7,4),C1(7,4),C0(7,4)
      DIMENSION WN(7,4),WN1(7,4),WN2(7,4)
      COMMON/BLK3/B2,B1,B0,C1,C0
C
      DO 710 J=1,7
      ENERGY(J)=0.
      DO 710 K=1,4
      WN1(J,K)=0.
710   WN2(J,K)=0.
      DO 711 I=1,256
      DO 712 J=1,7
      Y=SR(I)
C
      DO 713 K=1,4
      WN(J,K)=Y-C1(J,K)*WN1(J,K)-C0(J,K)*WN2(J,K)
      Y=B2(J,K)*WN(J,K)+B1(J,K)*WN1(J,K)+B0(J,K)*WN2(J,K)
713   CONTINUE
C
      ENERGY(J)=ENERGY(J)+Y**2
712   CONTINUE
C
      DO 714 J=1,7
      DO 714 K=1,4
      WN2(J,K)=WN1(J,K)
      WN1(J,K)=WN(J,K)
714   CONTINUE
711   CONTINUE
C
      ENERGY(1)=ENERGY(1)/1.53
      ENERGY(2)=ENERGY(2)/1.52
      ENERGY(3)=ENERGY(3)/1.48
      ENERGY(4)=ENERGY(4)/1.46
      ENERGY(5)=ENERGY(5)/1.44
      ENERGY(6)=ENERGY(6)/1.43
      ENERGY(7)=ENERGY(7)/1.42
C
      RETURN
      END
C
```

```
C*************************************************************************
C
      SUBROUTINE IFILW
      DIMENSION B0(7,4),B1(7,4),B2(7,4),C1(7,4),C0(7,4)
      COMMON/BLK3/B2,B1,B0,C1,C0
      OPEN(UNIT=3,FILE='BFILW.DAT',STATUS='OLD')
C
      DO 810 I=1,7
      DO 810 J=1,4
      READ(3,*)C1(I,J)
      READ(3,*)C0(I,J)
      READ(3,*)B2(I,J)
      B1(I,J)=0.
810   B0(I,J)=-B2(I,J)
C
      CLOSE(UNIT=3)
      RETURN
      END
C
C
      SUBROUTINE IFILN1
      DIMENSION B0(7,4),B1(7,4),B2(7,4),C1(7,4),C0(7,4)
      COMMON/BLK3/B2,B1,B0,C1,C0
      OPEN(UNIT=3,FILE='BFILN1.DAT',STATUS='OLD')
C
      DO 810 I=1,7
      DO 810 J=1,4
      READ(3,*)C1(I,J)
      READ(3,*)C0(I,J)
      READ(3,*)B2(I,J)
      B1(I,J)=0.
810   B0(I,J)=-B2(I,J)
C
      CLOSE(UNIT=3)
      RETURN
      END
C
C
      SUBROUTINE IFILN2
      DIMENSION B0(7,4),B1(7,4),B2(7,4),C1(7,4),C0(7,4)
      COMMON/BLK3/B2,B1,B0,C1,C0
      OPEN(UNIT=3,FILE='BFILN2.DAT',STATUS='OLD')
C
      DO 810 I=1,7
      DO 810 J=1,4
      READ(3,*)C1(I,J)
      READ(3,*)C0(I,J)
      READ(3,*)B2(I,J)
      B1(I,J)=0.
810   B0(I,J)=-B2(I,J)
C
      CLOSE(UNIT=3)
      RETURN
      END
C
C
      SUBROUTINE IFILN3
      DIMENSION B0(7,4),B1(7,4),B2(7,4),C1(7,4),C0(7,4)
      COMMON/BLK3/B2,B1,B0,C1,C0
      OPEN(UNIT=3,FILE='BFILN3.DAT',STATUS='OLD')
C
      DO 810 I=1,7
      DO 810 J=1,4
      READ(3,*)C1(I,J)
      READ(3,*)C0(I,J)
      READ(3,*)B2(I,J)
      B1(I,J)=0.
810   B0(I,J)=-B2(I,J)
C
      CLOSE(UNIT=3)
      RETURN
      END
C
```

```
C*************************************************************************
C
      SUBROUTINE VMOV(V1,ISTART,V2,JSTART,JSTOP)
      REAL V1(999),V2(999)
      L=ISTART
      DO 10 K=JSTART,JSTOP
      V2(K)=V1(L)
10    L=L+1
      RETURN
      END
C
C
      SUBROUTINE VMAX(MAX,V,JSTART,JSTOP)
      REAL V(999)
      VMAXI=0.0
      DO 10 K=JSTART,JSTOP
      IF(V(K).LT.VMAXI)GOTO 10
      VMAXI=V(K)
      MAX=K
10    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE VSMY(SCALAR,V1,ISTART,V2,JSTART,JSTOP)
      REAL V1(999),V2(999)
      L=ISTART
      DO 10 K=JSTART,JSTOP
      V2(K)=SCALAR*V1(L)
10    L=L+1
      RETURN
      END
C
C
      SUBROUTINE VDOT(DOT,V1,ISTART,V2,JSTART,JSTOP)
      REAL V1(999),V2(999)
      DOT=0.0
      L=ISTART
      DO 10 K=JSTART,JSTOP
      DOT=DOT+V1(L)*V2(K)
10    L=L+1
      RETURN
      END
C
C
      SUBROUTINE VSUM(SUM,V,JSTART,JSTOP)
      REAL V(999)
      SUM=0.0
      DO 10 K=JSTART,JSTOP
10    SUM=SUM+V(K)
      RETURN
      END
```

```
 ********* DATA FILE BFILW.DAT ***************************************
-1.768764
0.834471
0.083900
-1.836957
0.873936
0.075150
-1.818329
0.917878
0.067300
-1.930263
0.956776
0.124609
-1.691412
0.833412
0.084522
-1.762469
```

0.860019
0.082366
-1.727640
0.920091
0.064792
-1.869308
0.947797
0.140762
-1.587907
0.834853
0.083633
-1.666449
0.854026
0.085991
-1.609060
0.922714
0.062040
-1.781852
0.943091
0.148792
-1.458513
0.837417
0.082100
-1.545648
0.851663
0.087979
-1.463917
0.925299
0.059314
-1.666216
0.940573
0.152277
-1.305223
0.838455
0.081510
-1.401798
0.849471
0.089426
-1.294836
0.926732
0.058013
-1.526064
0.938607
0.155149
-1.130789
0.839718
0.080820
-1.236017
0.848225
0.090127
-1.104774
0.928070
0.056961
-1.361588
0.937268
0.156719
-0.938237
0.840512
0.080417
-1.051328
0.847035
0.090756
-0.895368
0.929044
0.056272
-1.176380
0.936111
0.158068

```
********** DATA FILE SFILN1.DAT **********************************
-1.750262
0.933819
0.033443
-1.775757
0.937599
0.039622
-1.766356
0.970928
0.023342
-1.824130
0.974840
0.057474
-1.706570
0.933826
0.035699
-1.734097
0.937180
0.034142
-1.720174
0.971065
0.023667
-1.783595
0.974526
0.062260
-1.658561
0.933835
0.033088
-1.688136
0.936820
0.040707
-1.669626
0.971177
0.025499
-1.738610
0.974264
0.052743
-1.606370
0.933834
0.035164
-1.637962
0.936520
0.035535
-1.614837
0.971268
0.022717
-1.689286
0.974042
0.064295
-1.550121
0.933838
0.033625
-1.583699
0.936251
0.037982
-1.555956
0.971349
0.028969
-1.635731
0.973848
0.049655
-1.489942
0.933837
0.033741
-1.525487
0.936018
0.039136
-1.493126
0.971417
0.022904
```

```
-1.578081
0.973679
0.061094
-1.426006
0.933833
0.034995
-1.463449
0.935812
0.035669
-1.426517
0.971477
0.024226
-1.516468
0.973528
0.061423

********** DATA FILE BFILN2.DAT ****************************************
-1.706570
0.933826
0.035699
-1.734097
0.937180
0.034142
-1.720174
0.971065
0.023667
-1.783595
0.974526
0.062260
-1.658561
0.933835
0.033088
-1.688136
0.936820
0.040707
-1.669626
0.971177
0.025499
-1.738610
0.974264
0.062743
-1.606370
0.933834
0.035164
-1.637962
0.936520
0.035535
-1.614837
0.971268
0.022717
-1.689286
0.974042
0.064295
-1.550121
0.933838
0.033625
-1.583699
0.936251
0.037982
-1.555956
0.971349
0.029969
-1.635731
0.973948
0.049655
-1.489942
0.933837
0.033741
-1.525487
0.936018
0.039138
```

```
-1.493126
0.971417
0.022904
-1.578081
0.973679
0.061094
-1.426006
0.933833
0.034995
-1.463449
0.935812
0.035669
-1.426517
0.971477
0.024226
-1.516468
0.973528
0.061423
-1.358468
0.933827
0.033115
-1.397738
0.935624
0.042678
-1.356294
0.971533
0.023519
-1.451050
0.973390
0.056149

********** DATA FILE BFILN3.DAT **********************************************
-1.658561
0.933835
0.033088
-1.688136
0.936820
0.040707
-1.669626
0.971177
0.025499
-1.738610
0.974264
0.052743
-1.606370
0.933834
0.035164
-1.637962
0.936520
0.035535
-1.614837
0.971268
0.022717
-1.689286
0.974042
0.064295
-1.550121
0.933838
0.033625
-1.583699
0.936251
0.037932
-1.555955
0.971349
0.023959
-1.635731
0.973848
0.049655
-1.489942
0.933837
0.033741
-1.525487
```

0.?36013
0.039138
-1.493124
0.971417
0.022904
-1.578081
0.973679
0.061094
-1.426006
0.933833
0.034995
-1.463449
0.935812
0.035669
-1.426517
0.971477
0.024226
-1.516468
0.973528
0.061423
-1.358468
0.933827
0.033115
-1.397738
0.935624
0.042678
-1.356294
0.971533
0.023519
-1.451050
0.973390
0.056149
-1.287489
0.933820
0.035915
-1.328534
0.935456
0.035025
-1.282635
0.971582
0.022555
-1.381986
0.973265
0.066064