To the Graduate Council:

I am submitting herewith a thesis written by John Eric Breeding entitled "The Design and Development of Real-Time Beamformers Using Multirate Digital Signal Processing Techniques." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Dr. D. B. Koch, Major Professor

We have read this thesis and
recommend its acceptance:

Dr. B. W. Bomar

Dr. H. P. Neff

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

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

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

Signature John Eric Breeding

Date March 26, 1994

THE DESIGN AND DEVELOPMENT OF REAL-TIME BEAMFORMERS USING MULTIRATE DIGITAL SIGNAL PROCESSING TECHNIQUES

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

John Eric Breeding

Oak Ridge National Laboratory*

May 1994

*Managed by Martin Marietta Energy Systems, Inc.
for the
U. S. Department of Energy
Under Contract No. DE-AC05-840R21400

ACKNOWLEDGMENTS

I am grateful to all the people who have contributed to this thesis. First, I thank my major professor, Dr. Daniel B. Koch, and my committee members, Dr. Herbert P. Neff and Dr. Bruce W. Bomar. I also thank the following team members for their important contributions to this work: Jerry Bentz, Tom Karnowski, Wayne Manges, and Robert Tate. I am also grateful to Dwight Clayton and Bill Zuehsow for significant technical guidance and fruitful discussions throughout the course of this work. A special word of gratitude goes to my management, Dan McDonald and Jim Jansen, who provided the opportunity and support which enabled this work.

Finally, I thank my wife, Jean, for her invaluable support, patience, and encouragement. I also thank my parents and grandmother for the many opportunities they have provided throughout my life.

ABSTRACT

This thesis presents the design, development, and implementation of real-time beamformers for passive sonar applications requiring the production of a single beam. An efficient interpolation beamforming algorithm based on the polyphase filter structure is developed, and the adaptation of this algorithm for a near-field application is presented. The architecture of a real-time beamforming system based on the polyphase filter structure is discussed. Real-time processing concepts such as multiple data buffers, pipeline processing, and overlap-save processing are emphasized.

In addition, the thesis discusses circular convolution effects that occur when the delay property of the fast Fourier transform (FFT) is used to achieve fractional sample delays in a frequency domain beamformer. The concept of integer and fractional sample delays developed for the polyphase filter algorithm is applied to the frequency domain to eliminate the effects of circular convolution. The implementation of an FFT beamformer based on this concept is presented. Special emphasis is placed on the use of windowed delay filters for fractional sample generation and a set of prestored basis vectors for integer delay generation.

PREFACE

The focus of this document is the design and development of real-time interpolation beamformers for passive sonar applications. The beamformers are important components of the underwater acoustic measurement systems developed by the Real-Time Computer Systems group of Oak Ridge National Laboratory's Instrumentation and Controls Division. Specifically, this document details the author's work in the areas of interpolation beamforming algorithm development, hardware specification and configuration, software design and development, and system integration. Certain aspects of this work such as overall system architecture and human interface are the result of a collective effort on the part of the author and other members of the development team. The beamformers described herein play a vital role in the functional and performance objectives of these underwater acoustic measurement systems.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 BEAMFORMING THEORY 5

Passive Sonar Systems 5

Spatial Filters 10

Ambient Noise 10

Line Hydrophone and the Aperture Function 12

Discrete Sensor Arrays 16

Beam Pattern Characteristics 19

Beam Steering 23

Array Shading 25

Summary 29

CHAPTER 3 INTERPOLATION BEAMFORMING ALGORITHM DESIGN 31

Time-domain Beamforming 31

Discrete Delay-and-Sum Beamforming 34

Interpolation Beamforming 39

Interpolation Beamforming Algorithm Development 42

Sampling Rate Conversion Using Polyphase Filters 43

Index Determination for a Focused Beamformer 46

Design Considerations: Overlap Processing 48

Summary 54

CHAPTER 4 SYSTEM ARCHITECTURE OF A REAL-TIME BEAMFORMER 55

Hardware Architecture 58

Signal Selector 58

Anti-aliasing Filters 60
Analog-to-Digital Converter 64
Array Processors 66
Software Architecture 70
Hardware Configuration 71
Time Shift Computation 73
Beamforming Operation 74
Cache Mapping for 7-Tap FIR Filter 79
Summary 81
CHAPTER 5 A FREQUENCY-DOMAIN APPROACH TO INTERPOLATION BEAMFORMING 83
FFT Beamforming 84
Fractional Samples 90
Fractional Samples and the DFT Delay Property 92
Fractional Samples Using FIR Filters 96
Real-time Implementation of an FFT Beamformer 99
Summary 105
CHAPTER 6 RESULTS 106
CHAPTER 7 CONCLUSIONS 119
LIST OF REFERENCES 121
VITA 124

LIST OF FIGURES

Figure 2.1. Classical data acquisition system. 7

Figure 2.2. Example passive sonar system. 8

Figure 2.3. One dimensional spatial filter 14

Figure 2.4. Beam pattern for a line hydrophone. 17

Figure 2.5. Beam power pattern from a 7-element line array, element spacing = $\frac{\lambda}{2}$ 20

Figure 2.6. Beam power pattern for 21-element line array, element spacing = $\frac{\lambda}{2}$. 21

Figure 2.7. Beam power pattern for 7-element line array, element spacing = $\frac{\lambda}{4}$. 22

Figure 2.8. Spatial aliasing due to sampling the line hydrophone. 26

Figure 2.9. Beam power pattern for 7-element line array, element spacing = λ. 27

Figure 2.10. Beam steered to –36°, element spacing = λ. 28

Figure 2.11. 51-element line array with Taylor shading, sidelobe suppression is 26 dB with 5 close-in sidelobes. 30

Figure 3.1. Plane wave having an angle of incidence, ψ, on a discrete line array with element spacing, d. 33

Figure 3.2. Synchronous beams for d = 1.0 m, f_S = 3 kHz. 36

Figure 3.3. Synchronous beams for d = 1.0 m, f_S = 30 kHz. 37

Figure 3.4. Scalloping loss for a 51-element discrete line array, d = 1.0 meter and f_S = 30 kHz. 38

Figure 3.5. Block diagram of classical interpolation beamformer. 40

Figure 3.6. Zero padding for upsampling $x(n)$ by L = 4. 41

Figure 3.7. Counterclockwise commutator model for 1-to-L interpolator. 44

Figure 3.8. Operation of a 1-to-4 interpolator using polyphase filters. 45

Figure 3.9. Direct form structure for a 7-tap finite impulse response (FIR) filter. 50
Figure 3.10. Overlap-save method for convolving blocks of 64 points with a 7-tap interpolation filter. 51
Figure 3.11. 4×2 Planar Array. 53
Figure 4.1. Interface boundaries for the beamformer system. 56
Figure 4.2. Multi-aperture array representing three 7×3 Arrays and a single 15×3 array. 57
Figure 4.3. Acoustic signal processor block diagram. 59
Figure 4.4. Physically realizable lowpass filter compared with the ideal filter assumed by the Nyquist sampling theorem. 62
Figure 4.5. Illustration of relationship between sampling frequency and anti-aliasing filter's shape factor. 63
Figure 4.6. Block diagram of the analog-to-digital converter subsystem. 65
Figure 4.7. Commutator configuration of scan list memories. 67
Figure 4.8. Buffer resulting from commutated scan list memories. 68
Figure 4.9. ST-50 hardware configuration. 69
Figure 4.10. Diagram of beamformer's software architecture. 72
Figure 4.11. Finite impulse response of 20:1 interpolation filter. 75
Figure 4.12. Pipeline architecture of real-time beamformer. 77
Figure 4.13. Circular buffer used for input data. 78
Figure 4.14. Cache mapping and arithmetic control processor execution for real-time implementation of Eq. (3.10). 80
Figure 5.1. Block diagram of Basic FFT Beamformer. 85
Figure 5.2. Illustration of the circular shift due to delay property of the discrete Fourier transform. 88
Figure 5.3. 3-element array example illustrating errors due to direct application of

discrete Fourier transform's delay property..... 89

Figure 5.4. Overlap-save processing corrects for errors due to circular shift..... 91

Figure 5.5. Magnitude response of two-tone signal with (a) no time shift; (b) time shift of 0.7 samples using delay property of the discrete Fourier transform..... 94

Figure 5.6. Error signal resulting from applying delay property of the discrete Fourier transform with Fractional τ..... 95

Figure 5.7. (a) Magnitude of two-tone signal shifted by 0.7 samples using finite impulse response (FIR) filter technique; (b) error signal produced by FIR filter technique..... 98

Figure 5.8. Tasks required for fast Fourier transform (FFT) Beamformer..... 100

Figure 5.9. Coefficient generation algorithm for a fast Fourier transform (FFT) beamformer..... 103

Figure 6.1. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°..... 107

Figure 6.2. Far-field pattern: source frequency = $\frac{f}{2}$; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°..... 108

Figure 6.3. Far-field pattern: source frequency = $2f$; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°..... 109

Figure 6.4. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 15.0°..... 110

Figure 6.5. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = –30.0°..... 111

Figure 6.6. Near-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform

shading; focal point = 1.52 m at 0.0°..... 112

Figure 6.7. Near-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; focal point = 1.52 m at 15.0°..... 113

Figure 6.8. Near-field pattern: source frequency = $\frac{f}{2}$; element spacing = $\frac{c}{2f}$; uniform shading; focal point = 1.52 m at 15.0°..... 114

Figure 6.9. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°. 115

Figure 6.10. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = –15.0°. 116

Figure 6.11. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; Taylor shading; steering angle = 0.0°. 117

CHAPTER 1

INTRODUCTION

The transportation of energy through space by means of wave motion is a fundamental principle of physics. Humans are equipped with eyes and ears which sense and process electromagnetic and acoustic wave motion to obtain information vital to existence. Disciplines such as radar and sonar have emerged to extend and improve the human ability to sense and process energy propagating through a medium.

In both radar and sonar systems, multiple sensors have been employed to increase the capability to discern information about sources of propagating waves. Examples of such systems include phased array radar and towed array sonar. An array of sensors is superior to a single sensor because of its inherent capability to reduce the degrading effects of noise. While the reduction may improve a system's capability to detect a source, it is, however, typically insufficient to glean more advanced information such as movement or identity. Thus, additional processing techniques are needed for identification and tracking. One array processing technique that is often employed is beamforming.

Beamforming is a multidimensional filtering technique used in the processing of signals carried by propagating waves. Beamforming uses the spatial and temporal aspects of both the sensor array and the propagating medium to attain directivity. The property of directivity allows signals originating from the direction of interest to pass unaffected while attenuating signals propagating from all other directions. The direction of interest coupled with the temporal frequency band of the beamformer may be thought of as the passband of the spatial filter [1]. By possessing directional properties, beamforming

serves two major functions in a measurement system. First, beamforming enables a measurement to determine the location of a source. In addition, noise cancellation occurs due to the attenuation of the spatial filter. Hence, an improvement in the signal-to-noise ratio (SNR) of the measurement system is achieved.

Conceptually, a beamformer achieves directivity by delaying and advancing the output from each sensor in the array so that all signals are in phase alignment. The required delays and advances are a function the direction of interest, or steering direction. Once phase alignment has been obtained, the signals are summed together to produce a beam. Traditionally, phase alignment has been accomplished by physically moving the sensor array toward the steering direction; however, with the emergence of high speed digital computers and integrated circuit technology, many measurement systems accomplish beam steering electronically. Electronic beam steering is superior to the traditional method in that multiple beams may be formed simultaneously. Furthermore, if sensor array data are stored, electronic steering enables an analyst or operator to beamform the data off-line.

In most modern acoustic measurement systems, signals from the sensor array are digitized prior to the beamforming operation. A major disadvantage of digitizing the signals is that the number of steering directions is both finite and proportional to the sampling frequency. Depending on hardware limitations that drive the sampling frequency, the number of achievable beams may render the measurement system useless. In an effort to increase the number of achievable beams, digital signal processing techniques have been used, and as a result, beamforming is considered a classical application for digital signal processing (DSP). Several different beamforming methods exist that use a wide range of DSP techniques. These methods include interpolation beamforming, fast Fourier transform (FFT) beamforming, and quadrature beamforming

[2]. Each method possesses its own set of advantages and limitations, and the choice of a method depends upon a variety of factors including the number of sensors, the temporal bandwidth of interest, and the computational capacity available.

This thesis details the design and development of real-time beamformers for two acoustic measurement systems. Each system may be characterized as a passive sonar system. Both beamformers produce a time series that serves as an input to a spectral analysis system, and each accommodates arbitrary sensor geometries. Although similar in terms of the hardware platform on which they were implemented, the two beamformers are designed to satisfy distinctly different sets of requirements.

The first beamformer considered is an interpolation beamformer that was developed as part of a data acquisition and analysis system for a large water tunnel. The primary purpose of this beamformer is to improve the SNR of the measurement system. In addition, the beamformer may also be used to isolate sources of noise by steering along a track that is known *a priori*. This approach is credible since steady state conditions are maintained by controlling the tunnel parameters. The dimensions of the tunnel's test section are such that wavefront curvature cannot be neglected, and thus, the beamformer is a focused beamformer [2]. As a result, the beam steering directions are specified as focal points.

The second beamformer considered is a frequency-domain beamformer that was developed for the analysis of data from sensor arrays located in the ocean. Unlike the first, this beamformer receives steering information from a real-time tracker. The operating environment permits the assumption that the wavefronts are planar (far-field). Thus, the steering information is a bearing angle rather than a focal point.

This thesis presents the theory, design, and implementation of the discrete time beamformers for the real-time acoustic measurement systems previously described. The

document begins by discussing beamforming theory with special emphasis on the aperture function, array shading, and figures of merit. Following these topics, a detailed treatment is given on both interpolation beamforming and frequency-domain beamforming. These presentations emphasize the use of multirate DSP to increase beam steering resolution and the application of the theory to the acoustic environment.

Although beamforming is a well-understood topic, certain aspects of the subject are virtually absent from the literature. Through the presentation of this material, the intent of this author is to describe in sufficient detail some of these areas.

CHAPTER 2

BEAMFORMING THEORY

Beamforming is a spatial filtering technique generally associated with passive sonar systems. The primary objectives of a beamformer may be categorized as follows [1]:

1. improve the SNR of a measurement system,
2. permit detection and resolution on a directional basis, and
3. estimate the direction of arrival of a signal.

The beamforming operation itself is an outgrowth of line antenna theory, and while most modern sonar systems employ discrete sensor arrays, line antenna theory provides an effective means of shaping the beam pattern. The purpose of this chapter is to present the motivation for the use of spatial filters and some basic theory on beamforming discrete sensor arrays.

Passive Sonar Systems

Sonar is typically divided into two categories: active and passive [2]. An active sonar system initiates the information gathering process by generating a known signal and listening for reflections from acoustic sources of potential interest. Although an active sonar system is an effective means of detecting and determining the position of an object,

the user of such a system inherently compromises his own position. If an important objective is maintaining silence, as is often the case in a tactical environment, active sonar is not the preferred method of gathering information.

As part of an attempt to improve quietness, test facilities have been developed for measuring the acoustic noise of vessels and individual components. Performance evaluation requires an accurate measure of acoustic emissions. Many of the tests performed in these facilities are not appropriate for active sonar systems. The signal generated from an active source could mask a problem in a design by dominating the dynamic range of a transducer. In addition, if the transducers are located in a tunnel or close to the water's surface, reflections of the generated signal may increase the noise in the measurement.

Acoustic design evaluation and noise measurement require the nonintrusive capability provided by passive sonar. A passive sonar system is simply a listening device taking the basic form of the classical data acquisition system shown in Figure 2.1. In classical data acquisition and analysis, the transducer converts the acoustic energy to electrical energy. The electrical signal is then conditioned, processed, and analyzed. An example of a passive sonar system is shown in Figure 2.2. For this system the transducer is the hydrophone, and conditioning includes filtering for anti-aliasing and digitization. The processing portion of the system includes both real-time and post processing. For this example, real-time processing generally consists of some digital signal processing technique such as spectral estimation using the FFT. Post processing functions may include such operations as hydrophone sensitivity correction, calibration to sound pressure levels, and gain correction.

The operation of many hydrophones is based on the piezoelectric phenomenon which is characteristic of certain crystalline and ceramic materials [3]. When such material

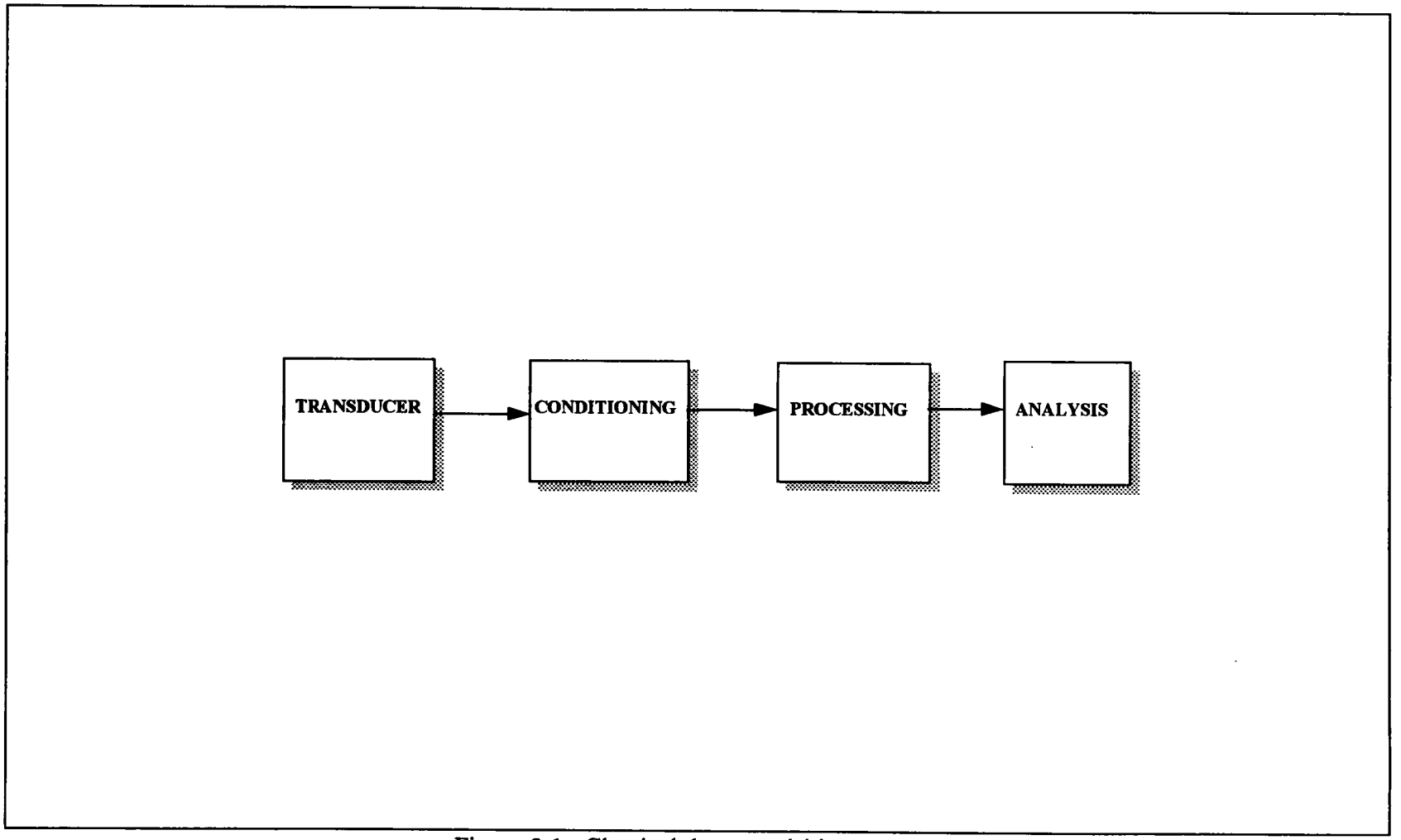

Figure 2.1. Classical data acquisition system.

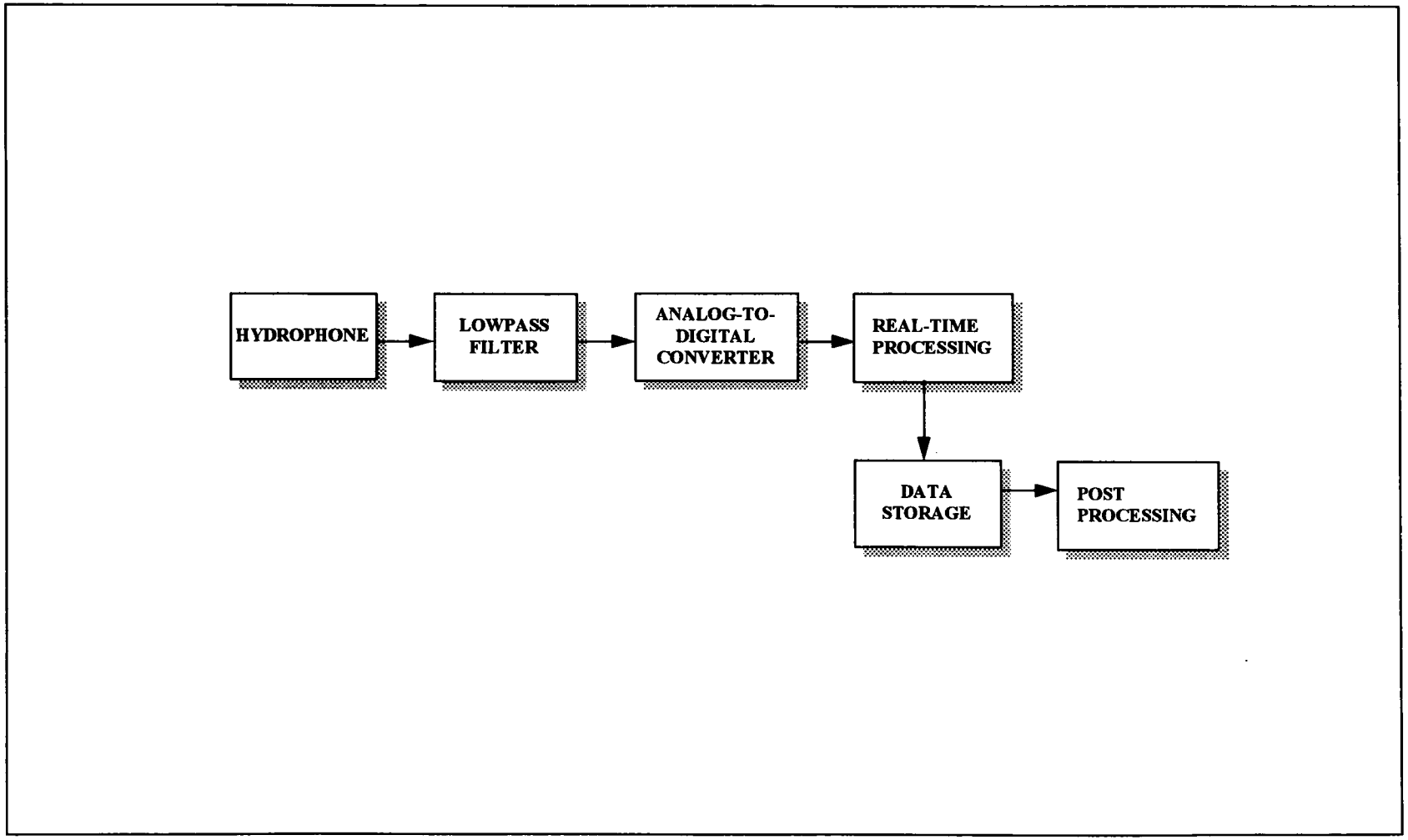

Figure 2.2. Example passive sonar system.

is subject to a mechanical stress, an electrical voltage occurs on its surface. For instance, a ceramic transducer disk typically has a thin conducting electrode covering each face. Sound waves cause the transducer to expand and contract, thus producing positive and negative voltages, respectively, across the electrodes. The piezoelectric phenomenon has the property of reciprocity which allows a transducer to act as an acoustic projector. By placing a voltage on the electrodes of the disk, the material expands and contracts, thus producing sound waves. Although the piezoelectric phenomenon is a common means of transduction, it is not the only one used for hydrophones. Other phenomena used for transduction include magnetostrictive, electrodynamic, and thermoacoustic [3].

As with most measurement systems, performance of a passive sonar system is significantly affected by the quality of the transducer. Hydrophone performance is measured in terms of sensitivity. Sensitivity for a hydrophone is defined as the ratio of the output voltage to the input pressure and is typically expressed in terms of volts per microPascals. Sensitivity depends on several parameters including material coefficients, geometry, and type of polarization used for the system [1]. The hydrophone block shown in Figure 2.2 actually consists of both a hydrophone and the amplifier to which it is connected. The output of this block is the linear combination of the hydrophone output and the electronic noise of the amplifier referred to the input [1]. From a system design perspective, the output acoustic SNR should be several decibels (dB). The output SNR is defined in terms of the lower bound on the expected input acoustic noise level, N_S, as [1]

$$SNR = N_s + H - N_a \,, \tag{2.1}$$

where H is the hydrophone's sensitivity and N_a is the electronic noise of the amplifier. The lower bound, N_S, is typically taken to be the minimum value of ambient acoustic

noise. Ambient noise establishes a lower limit on the intensity of useful acoustic signals in an underwater acoustic measurement system [1].

Spatial Filters

Passive sonar systems that process signals from a single hydrophone achieve performance that is dependent on the ambient noise. Such a system's ability to detect or resolve the direction of low-level signals is significantly reduced or even eliminated. To measure these signals, sensor arrays have been employed to effectively reduce the ambient noise. A sensor array is a set of hydrophones arranged in a specified spatial geometry. The geometry is determined by both the desired temporal bandwidth and spatial filter response. A sensor array coupled with the appropriate processing results in a spatial filter that enables the measurement system to extract low-level signals, and, in some instances, to determine the direction of these signals.

Ambient Noise

Acoustic emissions of interest are often masked by ambient noise. Ambient noise is a term that describes background acoustic noise. Ambient noise in the ocean is created by a combination of natural and man-made phenomena. Examples of natural contributions include wind and biological organisms, whereas man-made contributions are made up primarily of shipping noise. For a controlled environment such as a water tunnel, ambient noise is made up largely of flow noise.

Ambient noise may be generally described as a broadband signal having an intensity level that is inversely proportional to frequency; however, the specific

characteristics of ambient noise are dependent on many factors. Consequently, the first measurements made by passive sonar operators and analysts are of ambient conditions. In many cases, ocean-based test facilities for vessel designs are selected on the basis of a favorable noise characteristic. Several noise field models have been developed for describing ambient noise. Some noise field models include a surface-generated noise field, an impulsive field, a semi-isotropic field, and an isotropic field [1]. Among these models, the isotropic noise field is the most straightforward. An isotropic noise field is made up of many random waves propagating in all possible directions with equal probability. The noise intensity for a frequency, f, is given as [1]

$$\int_{4\pi} |N|^2 d\Omega = \int_{4\pi} K d\Omega = 4\pi K . \tag{2.2}$$

The power spectral density function for the isotropic noise field is given by [4].

$$S(\mathbf{k},\omega) = F(\omega)\delta\left(|\mathbf{k}| - \frac{\omega}{c}\right), \tag{2.3}$$

where c is the speed of sound in water, ω is the temporal frequency in radians per second, $F(\omega)$ is the distribution of power as a function of temporal frequency, and $\mathbf{k}$ is the wavenumber vector having a magnitude of

$$|\mathbf{k}| = \frac{\omega}{c} . \tag{2.4}$$

By taking the inverse Fourier transform of $S(\mathbf{k},\omega)$, the autocorrelation function has the following form [4]:

$$R(x,\tau) = A\cos(\omega\tau)\cdot\frac{\sin\left(\frac{\omega x}{c}\right)}{\frac{\omega x}{c}}. \qquad (2.5)$$

As seen in this equation, sampling the ambient noise at a spacing of

$$x = \frac{\lambda}{2} = \frac{\pi c}{\omega} \qquad (2.6)$$

results in uncorrelated samples [4]. This result is important in determining the optimal sensor spacing for the passband of a spatial filter.

Line Hydrophone and the Aperture Function

To measure low-level emissions, a passive sonar system must improve the signal-to-ambient noise ratio of the transducer and conditioning circuitry without compromising signal integrity. The SNR may be maximized by passing the signal from the transducer through a matched filter. For a given signal, $s(t)$, the impulse response of the matched filter is given by $s(T-t)$ [5]. Although the matched filter technique provides the best possible SNR, it is not practical for passive sonar systems due to the inherent assumption that the signal is known *a priori*. This filtering technique is more suitable for active sonar systems where the transmitted signal is known and the expected reflection from an object is similar to the transmitted signal. The matched filter also produces distortion in the input signal, which is often undesirable. This technique does, however, demonstrate the principle that a filter may be used to gain an SNR improvement.

Traditionally, designers of passive sonar systems have sought to achieve an improved SNR through spatial filtering. Spatial filtering is similar to temporal filtering in

that a band of interest is passed while the remaining bands are attenuated. The spatial filter band is typically specified in terms of some spatial metric such as incidence angle. For the spatial filter shown in Figure 2.3, the SNR for a plane wave having a temporal frequency, f_0, and propagating in an isotropic noise field at an angle of incidence, φ_0, is given by [1]

$$SNR = \frac{a^2}{K\psi_b}, \tag{2.7}$$

where a is the signal amplitude, K is the noise power, and φ_b is the equivalent noise bandwidth (ENBW) for a spatial filter having an angular response, $G(\psi)$, defined as [1,5]

$$\psi_b = \int_{-\pi}^{\pi} \frac{|G(\psi)|^2}{|G(\psi_0)|^2} d\psi . \tag{2.8}$$

As indicated by Eq. (2.7), the SNR is improved by decreasing the width of the passband, ψ_b.

The spatial filter response, $G(\psi)$, is the Fourier transform of a one-dimensional aperture function, $g(x)$. The term aperture originates from antenna theory and refers to the area from which a receiving antenna extracts power from an incoming signal [6]. The aperture function denotes a response of a sensor that is continuous in one dimension. For the purpose of measuring acoustic emissions, this sensor is the line hydrophone. The output of the line hydrophone whose input is a plane wave signal, $p_i(t)$, having an angle of incidence, ψ, is given by [1]

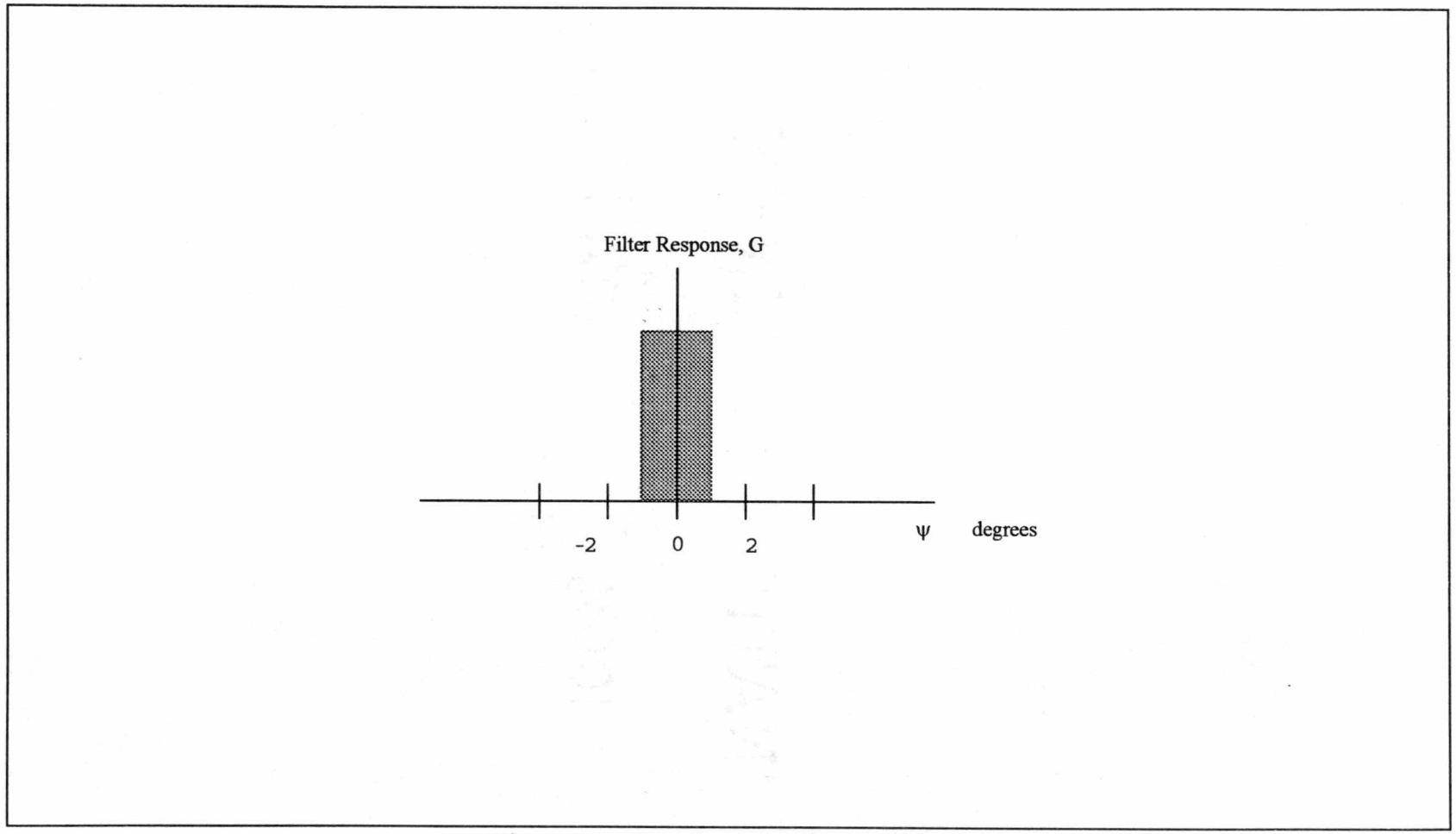

Figure 2.3. One dimensional spatial filter.

$$p_o(t,\psi) = \int_{-\infty}^{+\infty} g(x)\, p_i\left(t + \frac{x\sin(\psi)}{c}\right) dx \quad . \tag{2.9}$$

Alternatively, the response for the line hydrophone is expressed in terms of temporal frequency response as [1]

$$p_o(t,\psi) = \int_{-\infty}^{+\infty} P_i(f)\, G(f,\psi) \exp(j\,2\pi f\, t) df \ , \tag{2.10}$$

where $P_i(f)$ is the Fourier transform of $p_i(t)$, and the single-frequency beam pattern, $G(f,\psi)$, is defined as

$$\mathrm{G}(f,\psi) = \int_{-\infty}^{+\infty} \mathrm{g(x)} \exp\left[\mathrm{j}2\pi x \frac{f\sin(\psi)}{c}\right] dx \ . \tag{2.11}$$

$G(f,\psi)$ is a function of the temporal frequency of the incoming plane wave and the aperture function. The normalized aperture function and the corresponding spatial filter response, or beam pattern, are given in Eqs. (2.12) and (2.13), respectively [1]:

$$\mathrm{g}(x) = \frac{1}{L}\,\mathrm{rect}\left(\frac{x}{L}\right) \tag{2.12}$$

$$\mathrm{G}(f,\psi) = \mathrm{sinc}\left[\frac{f\sin(\psi)}{Lc}\right]. \tag{2.13}$$

A plot of this beam pattern as a function of temporal frequency is shown in Figure 2.4. It is evident from Eq. (2.13) that the width of the main lobe increases as the temporal frequency decreases. As a result, the directivity is decreased, and the SNR improvement is reduced.

Discrete Sensor Arrays

The line hydrophone is an example of a continuous transducer. In practice continuous transducers are approximated by discrete hydrophones. Although each hydrophone has its own performance characteristics, a common assumption is that the array consists of point transducers that are equally matched in both phase and directivity. If the directivity is uniform over all directions, the ideal point transducers are often called omnidirectional hydrophones, or omnis.

Examination of Eq. (2.9) reveals that the line hydrophone achieves a spatial filtering operation by integration. For the discrete sensor array, this operation corresponds to the summation of the outputs of each hydrophone. Thus, a spatial filter that passes signals from a certain direction while attenuating signals from others may be expected by summing the outputs of each hydrophone [2]. Consider a plane wave signal, $p_i(t)$, given by

$$p_i(t) = \exp\left[j(2\pi ft - \mathbf{r}_i \cdot \mathbf{k})\right], \tag{2.14}$$

where $\mathbf{r}_i$ denotes the position of the i-th sensor. The output of the spatial filter produced by an M-element array is

$$p_o(t) = \exp(j2\pi ft)\sum_{i=1}^{M}\exp(-j\mathbf{r}_i \cdot \mathbf{k}), \tag{2.15}$$

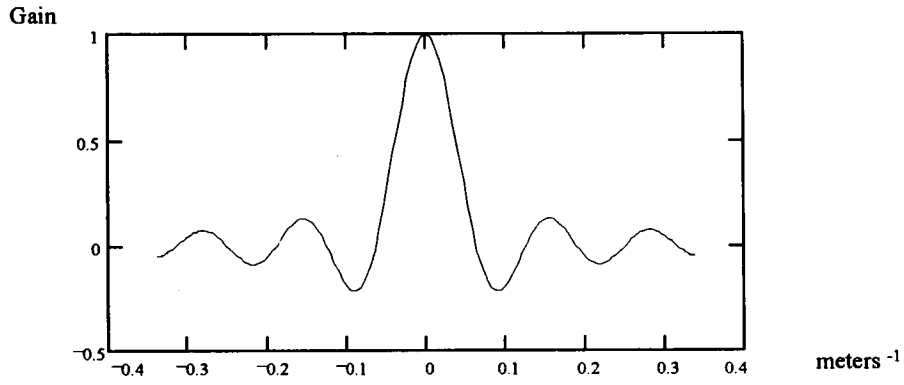

Figure 2.4. Beam pattern for a line hydrophone.

where the array pattern [7], or complex beam pattern [2], is given by

$$W(\mathbf{k}) = \sum_{i=1}^{M} \exp(-j\mathbf{r}_i \cdot \mathbf{k}) \,. \tag{2.16}$$

For the discrete line array having an element spacing, d, the array pattern becomes [2]

$$W(k) = \sum_{i=1}^{M} \exp\left[jk(i-1)d\sin(\psi)\right] \,. \tag{2.17}$$

Using the geometric progression formula

$$\sum_{m=1}^{M} a^m = \frac{a\left(1-a^M\right)}{1-a} \,, \tag{2.18}$$

a closed-form expression may be obtained for this array pattern:

$$W(k) = \frac{\sin\left[\dfrac{kMd\sin(\psi)}{2}\right]\exp\left[\dfrac{jkd(M-1)\sin(\psi)}{2}\right]}{\sin\left[\dfrac{kd\sin(\psi)}{2}\right]} \,. \tag{2.19}$$

The magnitude of the array pattern, or beam pattern, is given by [2]

$$|W(k)| = \left|\frac{\sin\left[\dfrac{kMd\sin(\psi)}{2}\right]}{\sin\left[\dfrac{kd\sin(\psi)}{2}\right]}\right| = \left|\frac{\sin\left[\dfrac{\pi Mfd\sin(\psi)}{c}\right]}{\sin\left[\dfrac{\pi fd\sin(\psi)}{c}\right]}\right| \,. \tag{2.20}$$

A plot of the beam power pattern , $|W(k)|^2$, is shown in Figure 2.5.

Beam Pattern Characteristics

The beam power pattern shown in Figure 2.5 is characterized by a single main lobe and several sidelobes. The main lobe corresponds to the passband of the spatial filter. The width of the passband is a function of the sensor spacing, the number of sensors, and the temporal frequency. Since a small passband improves the SNR, a narrow main lobe is generally desirable. One method for achieving a narrow main lobe is by increasing the number of sensors in the array. The beam power pattern in Figure 2.5 was produced by a 7-element array. A significant improvement is realized if the number of sensors is increased to 21 as shown in Figure 2.6.

The beam power pattern is also a function of temporal frequency, f. The temporal frequency and distance traveled are related through the wavenumber, **k**. Hence, for a fixed element spacing, the beam power pattern varies with temporal frequency. The design frequency, f_d, of an array with fixed spacing, d, is typically defined as the frequency at which the wavelength is twice the distance between sensors. This relationship is given by

$$d = \frac{\lambda_d}{2} = \frac{c}{2 f_d} . \tag{2.21}$$

According to Eq. (2.6), this array spacing corresponds to uncorrelated noise samples. The design frequency is generally considered the upper useful operating frequency of the array [1]. Source frequencies below the design frequency result in a broader main lobe as shown in Figure 2.7, Thus, the noise included in the measurement is increased.

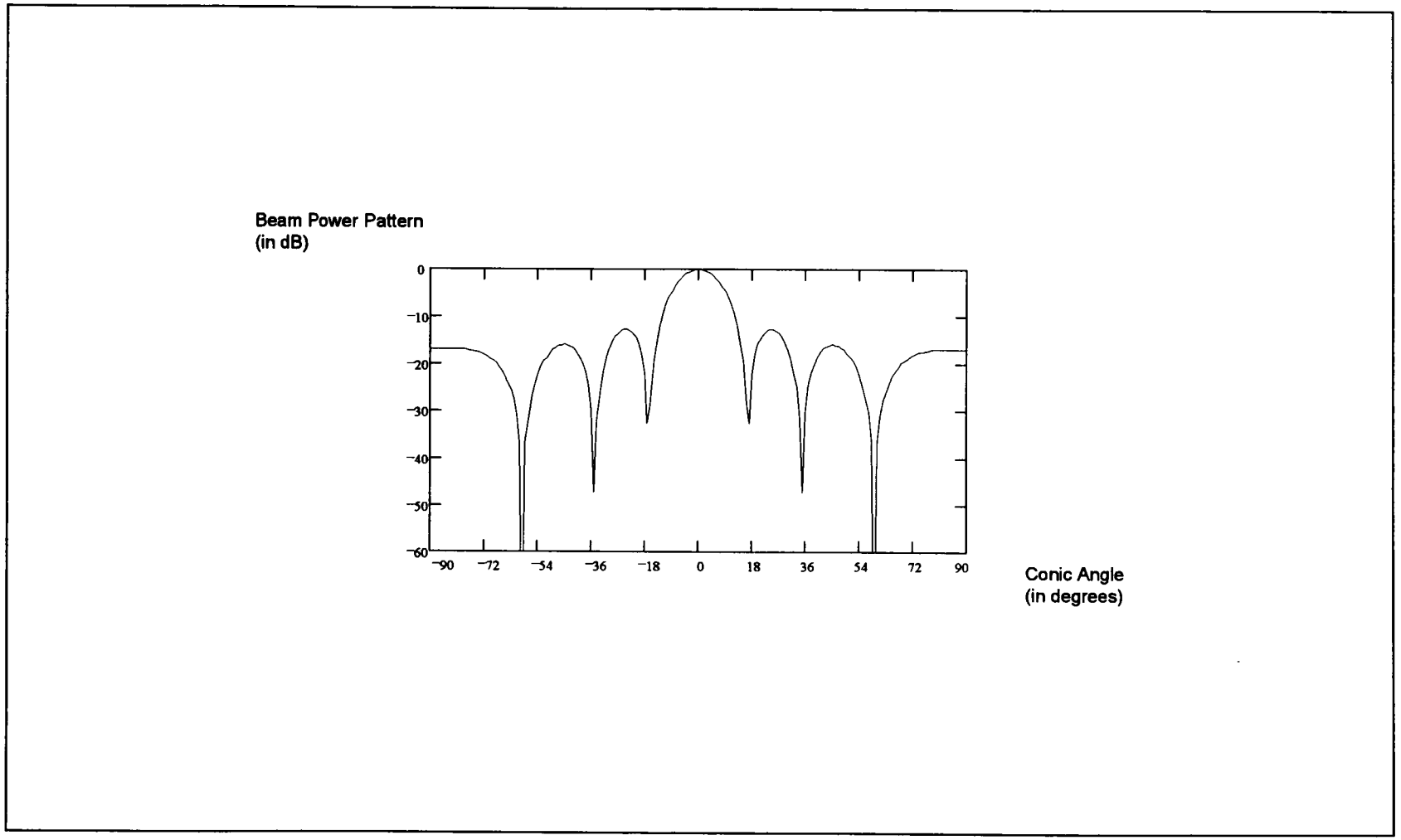

Figure 2.5. Beam power pattern from a 7-element line array, element spacing = $\frac{\lambda}{2}$.

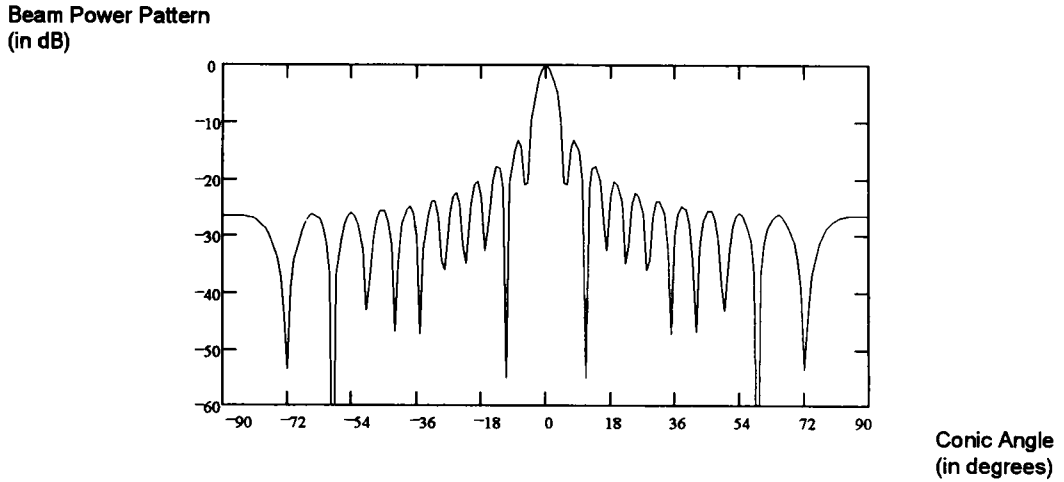

Figure 2.6. Beam power pattern for 21-element line array, element spacing = $\frac{\lambda}{2}$.

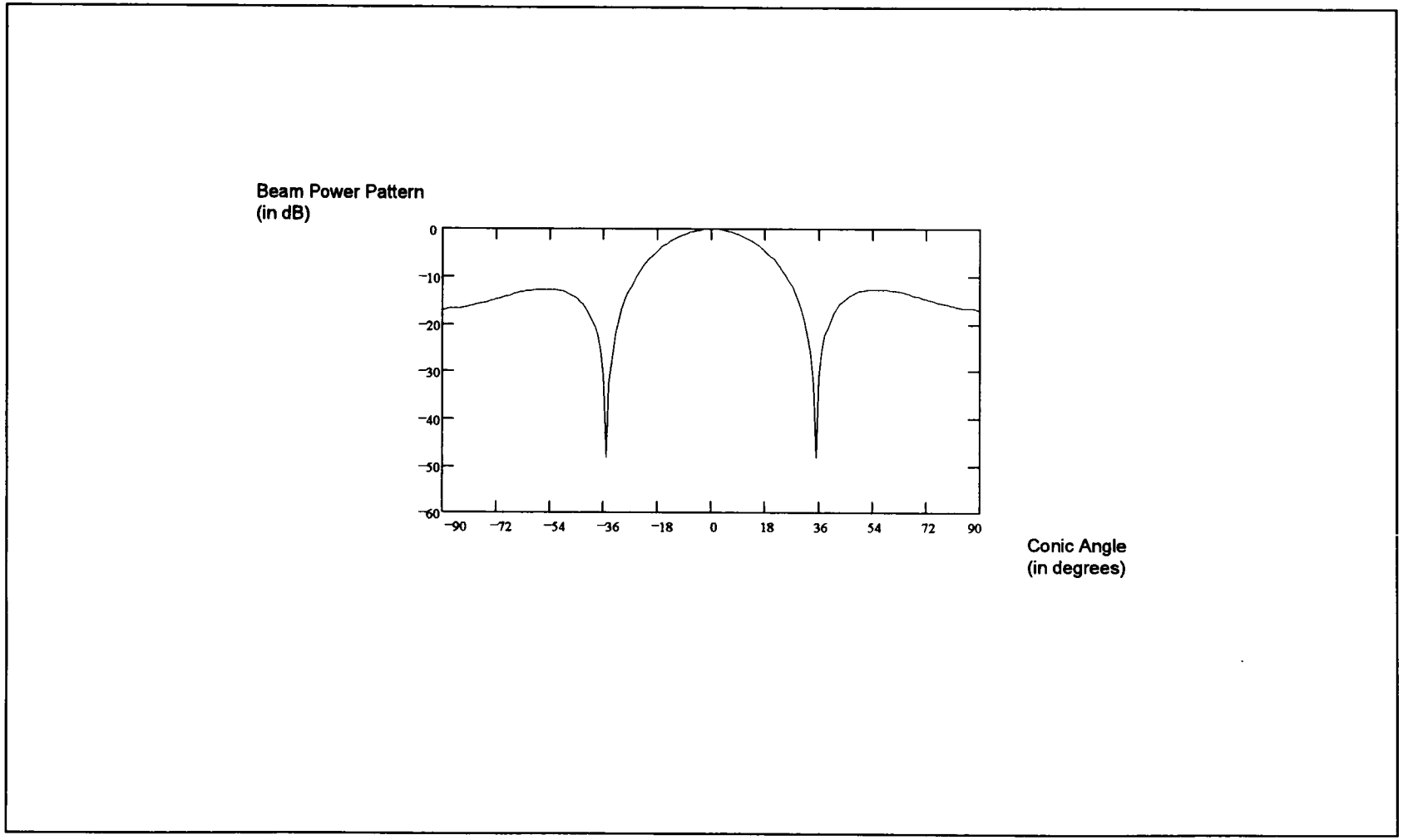

Figure 2.7. Beam power pattern for 7-element line array, element spacing = $\frac{\lambda}{4}$.

Array gain is a measure of the SNR improvement of an array relative to a single hydrophone. The array gain is defined as the ratio of the noise power out of an omnidirectional hydrophone to the noise power out of the beam power pattern [2]. For an isotropic ambient noise model, the array gain is also called the directivity index [1]. The goal of a system designer is to achieve the highest possible array gain. For the M-element line array, the directivity index is given as [1,2]

$$DI = \frac{M}{1+\frac{2}{M}\sum_{i=1}^{M-1}(M-i)\cos\left[\frac{2\pi fdi\sin(\psi_0)}{c}\right]\frac{\sin\left(\frac{2\pi fdi}{c}\right)}{\left(\frac{2\pi fdi}{c}\right)}}. \tag{2.22}$$

According to Eq. (2.22), the maximum value of the array gain is equal to the number of sensors in the array when the temporal frequency of the noise is at the design frequency of the array. As the frequency decreases below f_d, the summation in Eq. (2.22) becomes nonzero, and thus, the performance of the array is degraded.

Beam Steering

The existence of the beam pattern is a result of the geometry of the transducer. By employing multiple transducers, the passive sonar system achieves spatial filtering; however, the spatial filter is inflexible in that the passband, or maximum response axis (MRA), is always at broadside with respect to the sensor array. This characteristic of the array response makes the sensor array limited in its usefulness if it cannot be steered toward new directions. The process of steering the array response is known as beam

steering, or beamforming. Steering may be accomplished either mechanically or electronically. Mechanical beamforming requires that the array be physically moved toward the direction of interest, ψ_0. Electronic steering may be accomplished in either the conditioning or processing blocks of the passive sonar system. Conceptually, electronic steering is achieved by using time delays and advances on the signal from each hydrophone. The array pattern that results from steering the beam toward a direction of propagation, $\mathbf{k}_0$, is given by [7]

$$W(\mathbf{k}-\mathbf{k}_0)=\sum_{i=1}^{M}\exp\left[-j\mathbf{r}_i\cdot(\mathbf{k}-\mathbf{k}_0)\right]. \tag{2.23}$$

For the M-element line array, the beam pattern resulting from steering toward ψ_0 is given as [2]

$$|b(f,\psi)|=\left|\frac{\sin\left\{\pi fMd\left[\sin(\psi)-\sin(\psi_0)\right]\right\}}{\sin\left\{\pi fd\left[\sin(\psi)-\sin(\psi_0)\right]\right\}}\right| \tag{2.24}$$

The beam pattern given by Eq. (2.24) is obtained by spatially sampling the line hydrophone. The aperture function of the discrete line array, $g'(x)$, is

$$\mathrm{g}'(x)=\sum_{m=\frac{-L}{2}}^{\frac{L}{2}}\delta\,(x-md)\,\mathrm{g}(x) \tag{2.25}$$

From sampling theory, it is evident that $\mathrm{G}(f,\psi)$ becomes periodic and some spatial aliasing occurs. The degree of aliasing is determined not only by the spacing of the samples as in the case of temporal sampling theory, but also the temporal frequency of the

source and the angle, ψ, at which the beam is steered. For the discrete line array, the beam power pattern for $\psi_0 = 0°$ is periodic with a period of 180° as shown in Figure 2.8. The aliased main lobes are called grating lobes. If the spatial sampling is too low for a given source frequency, the grating lobes appear in the field of interest as shown in Figure 2.9. Grating lobes become more prevalent as the beam is steered off broadside as shown in Figure 2.10. The aliasing effect causes the main lobe to narrow considerably compared to the beam power pattern produced at the design frequency. If the grating lobes are not in the field of interest, the aliasing actually improves the performance of the beamformer by producing a narrow main lobe.

Array Shading

Array shading is the assignment of numerical weights to each element in a discrete array for the purpose of achieving a predetermined array pattern. Array shading is analogous to using window functions in applications of the discrete Fourier transform (DFT). The beam power patterns displayed in Figures 2.5 - 2.10 are for a discrete line array whose elements are equally weighted. Such an array is characterized as uniformly shaded. The effect of uniform shading on the spatial filter response is equivalent to that produced by a rectangular window function on the frequency response of the DFT. For instance, the highest sidelobe level of the beam power pattern in Figure 2.5 is approximately 13 dB below the peak of the main lobe, and the sidelobes fall off at 6 dB per octave [8].

For most beamforming applications, the ideal shape for the beam power pattern is a narrow main lobe accompanied by low sidelobe levels. As with temporal windows, a

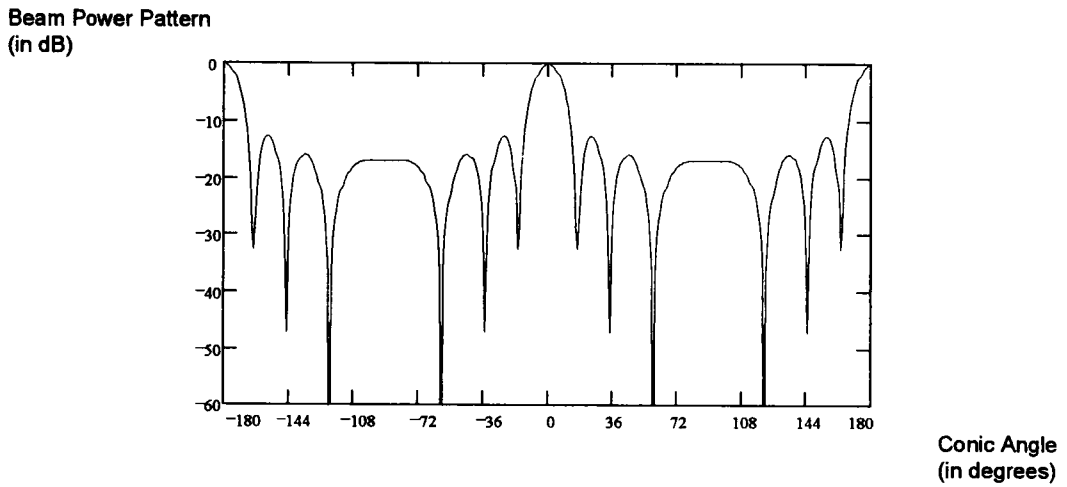

Figure 2.8. Spatial aliasing due to sampling the line hydrophone.

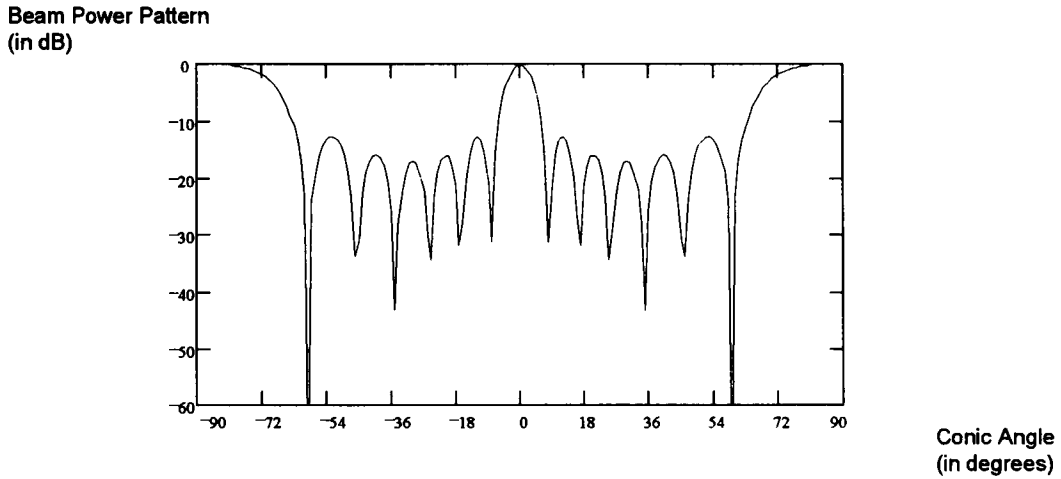

Figure 2.9. Beam power pattern for 7-element line array, element spacing = λ.

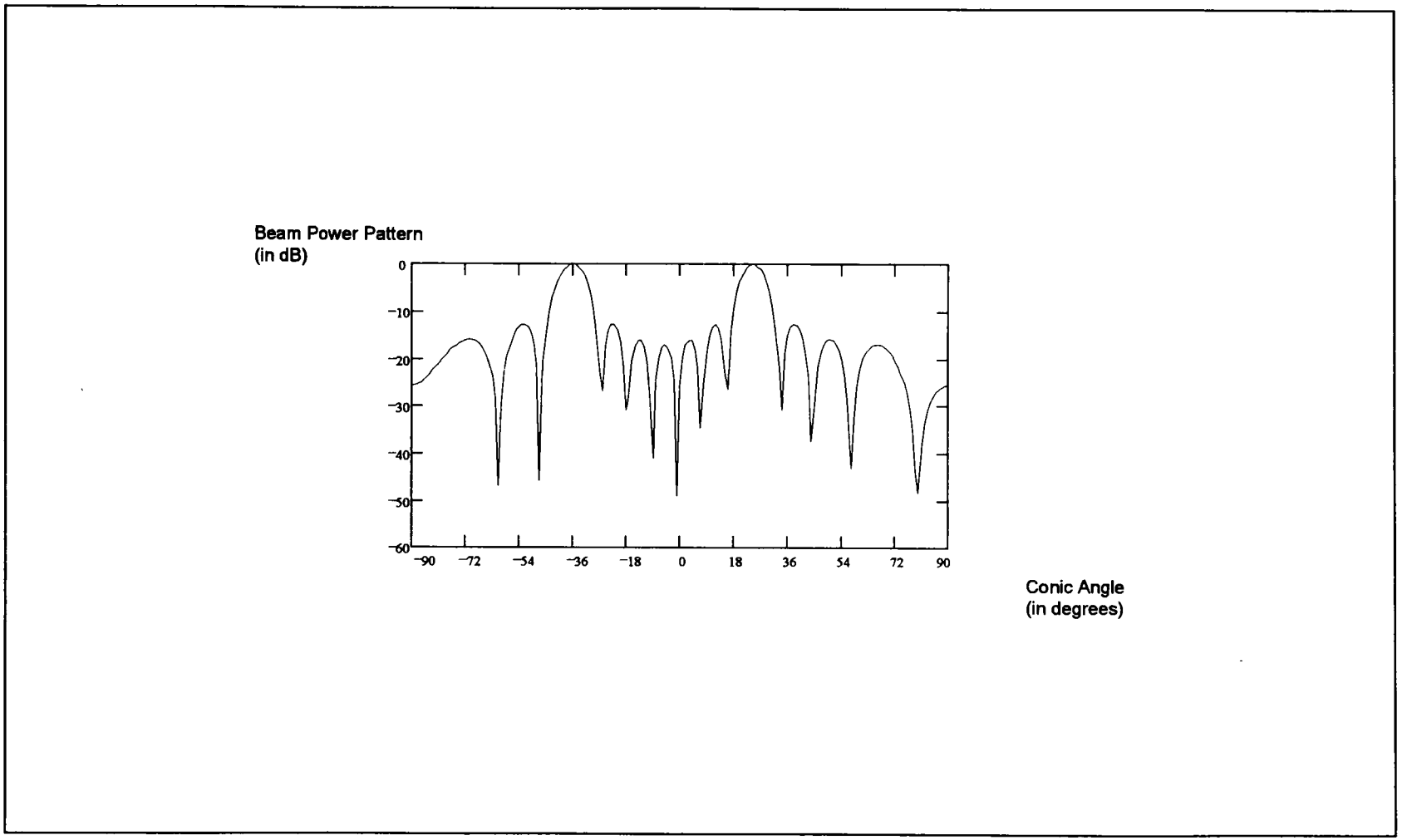

Figure 2.10. Beam steered to −36°, element spacing = λ.

tradeoff exists between these two characteristics; however, certain shading functions yield very low sidelobe levels without significantly increasing the width of the main lobe. For discrete linear arrays, the Taylor shading function is widely used. The Taylor shading technique is based on the Dolph-Tchebyshev window function [9]. The Dolph-Tchebyshev function provides the minimum main lobe width for a given sidelobe level. Taylor shading is an approximation of the Dolph-Tchebyshev function developed for continuous antenna applications [8]. Taylor shading requires two input parameters: the number of close-in sidelobes and the sidelobe levels. A beam power pattern produced by a 51-element line array with Taylor shading is shown in Figure 2.11.

Summary

Spatial filters are used in passive sonar systems to effectively reduce the ambient noise level. Discrete sensor arrays produce a spatial filter known as an array pattern when their outputs are summed together. Beamforming is the process of steering the passband (MRA) of the array pattern. In addition to steering, the shape of the array pattern is alterable by using a technique known as shading.

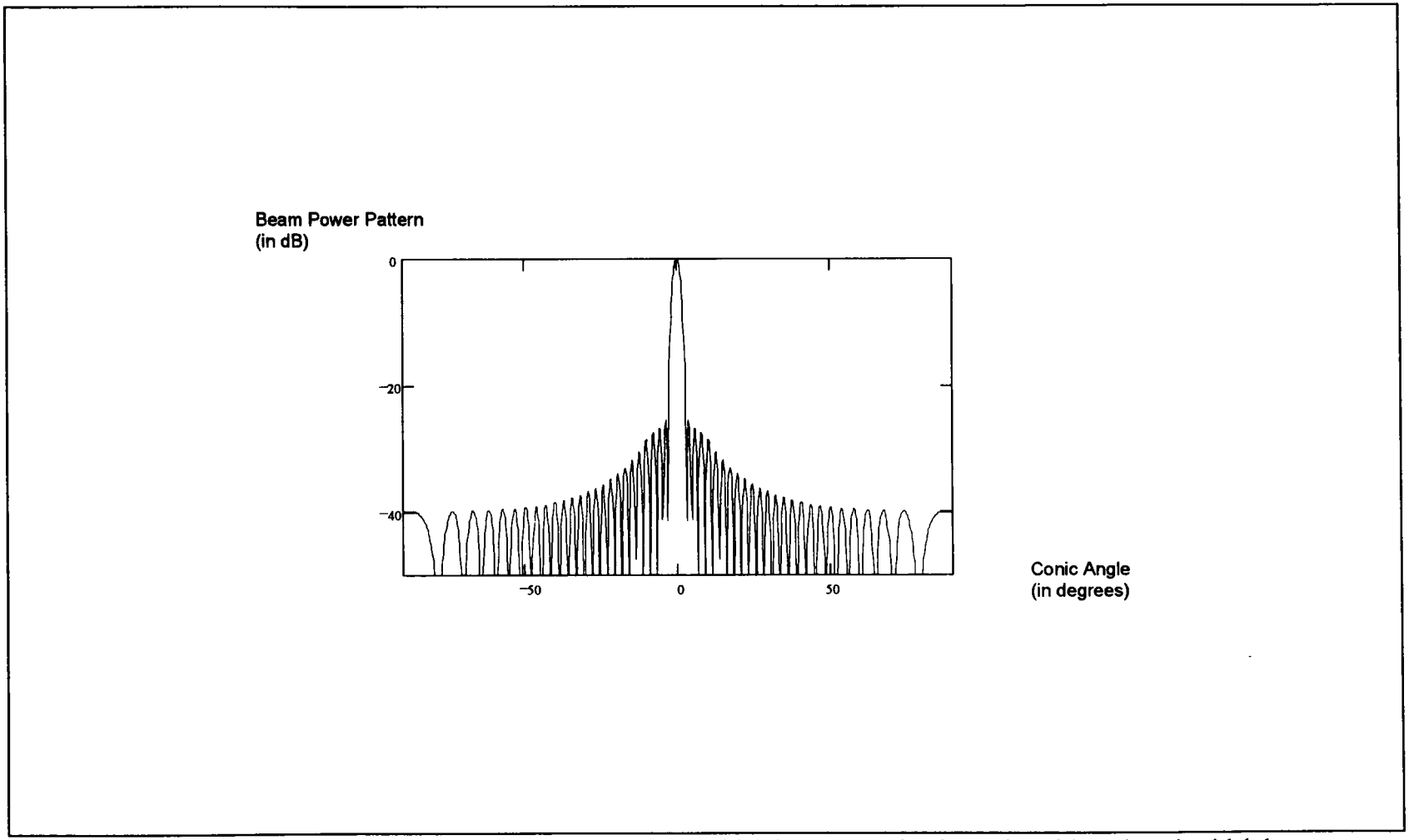

Figure 2.11. 51-element line array with Taylor shading, sidelobe suppression is 26 dB with 5 close-in sidelobes.

CHAPTER 3

INTERPOLATION BEAMFORMING ALGORITHM DESIGN

This chapter describes the design of a real-time interpolation beamforming algorithm for a test facility. For this facility, an acoustic measurement system was developed which digitizes elements of an array and produces a single beam having an MRA that updates at a fixed rate. Because of the dimensions of the test section, far-field conditions cannot be assumed. Thus, the MRA is specified as a focal point, and the beamformer is referred to as focused [2].

Time-domain Beamforming

As shown in the previous chapter, an array of hydrophones performs a spatial filtering operation. This filter response, also known as the beam power pattern, is produced by summing the signals from all sensors together. The response is characterized by a single main lobe accompanied by several sidelobes. The MRA of the resulting beam occurs broadside to the array. As stated in the previous chapter, the beam may be steered such that the MRA occurs at a new direction. Steering is accomplished either mechanically or electronically. Mechanical beam steering is accomplished by physically altering the orientation of the hydrophone array. Often this maneuver is impractical. For instance, in a test facility the position of the array(s) is often fixed. An alternative to mechanical steering is beamforming.

Beamforming may be defined as the process of changing the MRA of the array response without altering the position or physical orientation of the sensor array. A beamformer is a device that performs the operation in Eq. (3.1) [7]

$$b(t) = \sum_{i=1}^{M} p_i(t - \tau_i) , \tag{3.1}$$

where τ_i is a time shift for the i-th sensor. Each time shift is selected on the basis of the desired MRA. For example, given the discrete line array shown in Figure 3.1, the time shifts required to steer the beam in the direction ψ are given by [1]

$$\tau_i = \frac{(i - i_{ref})d \sin(\psi)}{c} , \tag{3.2}$$

where i_{ref} is the reference sensor, d is the sensor spacing, and c is the speed of sound in water ($\approx$ 1500 meters/second). The beamforming operation of Eq. (3.1) assumes that the speed of propagation of the pressure field is constant over the frequency band of interest [7]. Theoretically, τ_i is a continuous variable, thus allowing the MRA to be steered to any arbitrary location; however, in practice, the value of τ_i is subject to limitations in the hardware that implements the necessary time shifts. If analog electronics is used to steer the beam, the range of MRAs is relatively smooth; however, if the signals from the array are digitized prior to beamforming, the range of possible MRAs is finite and discontinuous. The remainder of this document focuses on the issues and techniques surrounding the design of a digital beamformer.

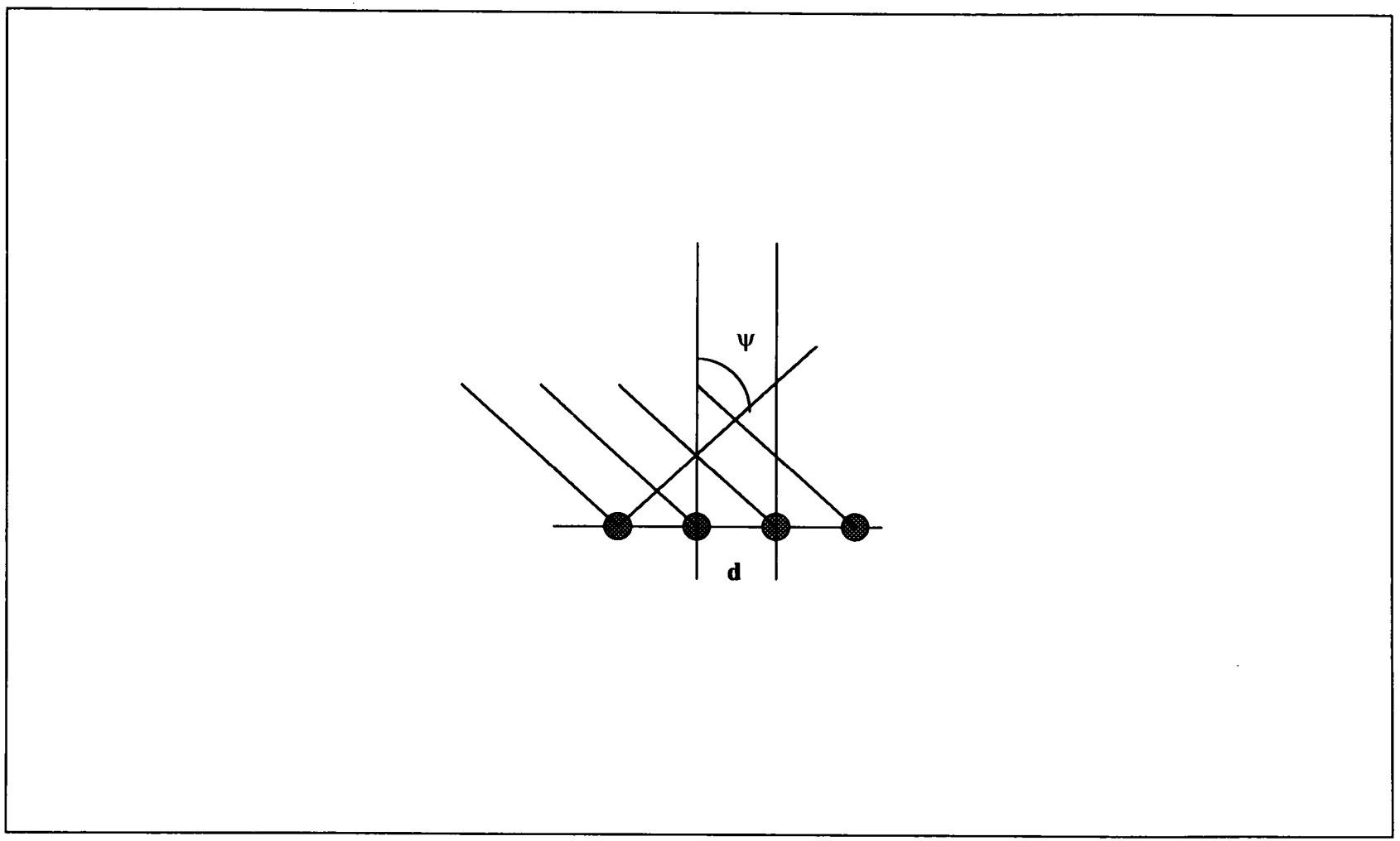

Figure 3.1. Plane wave having an angle of incidence, ψ, on a discrete line array with element spacing, d.

Discrete Delay-and-Sum Beamforming

If the signal from each array element is digitized, the beamforming operation of Eq. (3.1) becomes

$$b(nT) = \sum_{i=1}^{M} p_i\left(nT - n_i T\right), \tag{3.3}$$

where T is the temporal sampling interval. This operation is known as a discrete delay-and-sum beamformer. If array shading is employed, the operation is known as a weighted delay-and-sum beamformer and is given by [7]

$$b(nT) = \sum_{i=1}^{M} w_i\, p_i\left(nT - n_i T\right), \tag{3.4}$$

where w_i denotes the shading coefficient for the i-th sensor. By digitizing the signals, the achievable time delays are limited to a set of quantized delays, and thus, the steering directions obtainable from the available set of quantized delays are significantly reduced. The steering directions obtainable from the set of quantized delays are referred to as natural, or synchronous, beams [10].

The number of synchronous beams increases as the sampling frequency, f_S, increases. For instance, the synchronous beams that can be realized for the discrete line array in Figure 3.1 are given by [2]

$$\psi_0 = \sin^{-1}\left(\frac{hc}{df_s}\right) \qquad h = 0, \pm 1, \ldots \pm\left\lfloor \frac{df_s}{c} \right\rfloor. \tag{3.5}$$

Figures 3.2 and 3.3 demonstrate the effect of sampling frequency on the number of synchronous beams for a line array with element spacing of 1.0 meter and a sampling frequency of 3 kHz and 30 kHz, respectively. Although Figures 3.2 and 3.3 show only positive beam indices, it is understood that beam index, h, defined in Eq. (3.5) takes on positive and negative values. For the 3 kHz sampling frequency, the number of synchronous beams is 5, whereas the 30 kHz sampling frequency yields 41 synchronous beams.

From a system design perspective, the required number of synchronous beams is determined by the definition of the beamwidth coupled with the spatial coverage requirements. Figure 3.4 demonstrates the phenomenon called scalloping loss which is characteristic of synchronous beams. Scalloping loss is the maximum amount of attenuation that occurs for signals propagating from directions that lie between synchronous beams [2]. The number of required synchronous beams decreases as the maximum allowed attenuation increases or if array shading is employed since a shading function produces a broader main lobe. Figure 3.4 shows the scalloping loss as a function of bearing for a discrete line array consisting of 51 elements. Since the acceptable maximum attenuation is generally 3 dB [2], even a sampling frequency of 30 kHz is not sufficient to satisfy system goals.

By decreasing the time between samples, the number of synchronous beams is increased. Thus, a goal of passive sonar systems that include beamforming is to achieve the highest possible sampling rate in order to decrease the time between samples. Because of the number of sensors in an array, and hence the amount of digital data to process, is often large so as to obtain a narrow main lobe, the sampling frequency is limited by hardware processing capability. In addition, the sampling frequency needed may not be obtainable with analog-to-digital converter (ADC) technology. A relatively low temporal

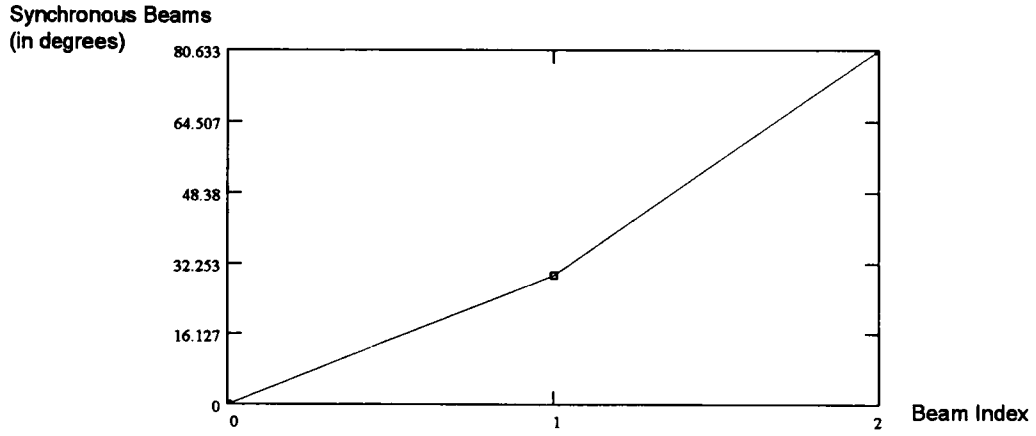

Figure 3.2. Synchronous beams for $d = 1.0$ m, $f_S = 3$ kHz.
($d \equiv$ element spacing, $f_S \equiv$sampling frequency)

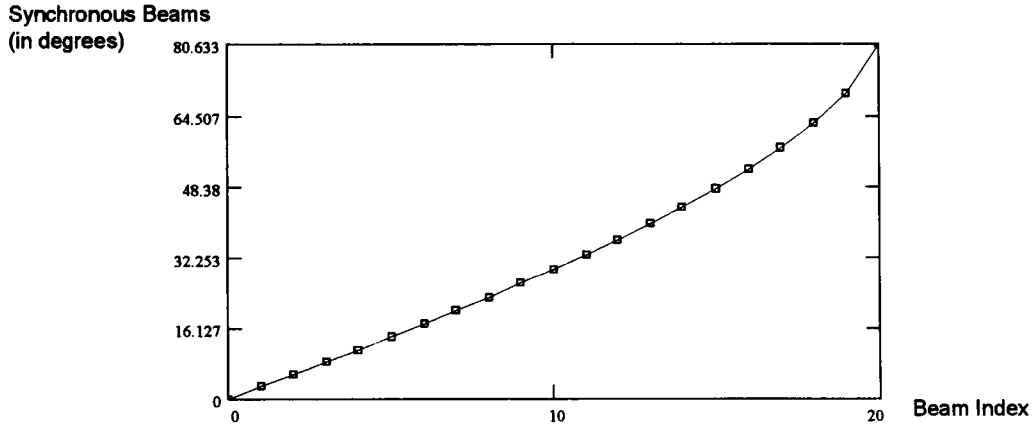

Figure 3.3. Synchronous beams for $d = 1.0$ m, $f_S = 30$ kHz.
($d \equiv$ element spacing, $f_S \equiv$sampling frequency)

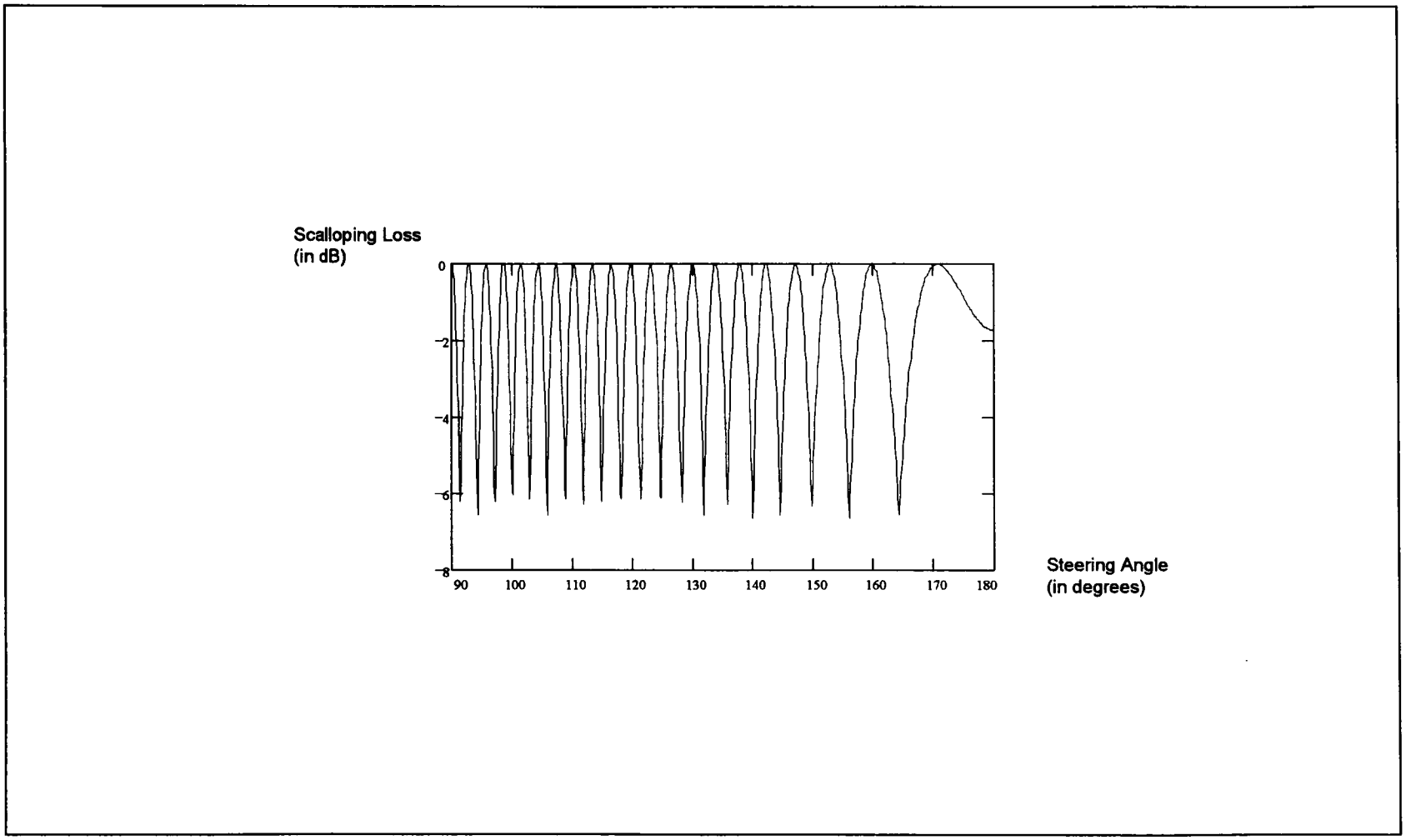

Figure 3.4. Scalloping loss for a 51-element discrete line array, d = 1.0 meter and f_S = 30 kHz.
($d \equiv$ element spacing, $f_S \equiv$sampling frequency)

bandwidth coupled with off-line beamforming may allow an increase in the sampling frequency sufficient for beam power pattern specifications; however, if real-time processing is required or if the temporal bandwidth is relatively high, delay-and-sum beamforming becomes computationally prohibitive. Thus, another approach to decreasing the quantile must be used. One such approach is interpolation beamforming.

Interpolation Beamforming

Interpolation beamforming is a technique that uses multirate digital signal processing to obtain the high sampling rates necessary to minimize scalloping loss [10]. A block diagram of an interpolation beamformer is shown in Figure 3.5. For this design, each element in the array is sampled at a rate on the order of the Nyquist rate. The sampling rate is then increased by an integer factor, L, using an interpolation algorithm, thus increasing the set of possible time shifts that are used by the beamformer. Finally, the beam is formed by implementing the discrete delay and sum beamformer of Eq. (3.4) using the expanded set of quantized delays. The interpolation beamforming operation is defined by

$$b(nT) = \sum_{i=1}^{M} w_i \, p_i \left[nT - \left(n_i + \frac{m_i}{L} \right) T \right], \tag{3.6}$$

where n_i and m_i represent integer and fractional time shift indices, respectively.

The classical method of interpolating by an integer factor is illustrated in Figure 3.6. In order to interpolate, or upsample, a digitized sequence, $x(n)$, by a factor of L, $L-1$ zeros are inserted between adjacent samples to obtain the sequence, $w(m)$. Known as

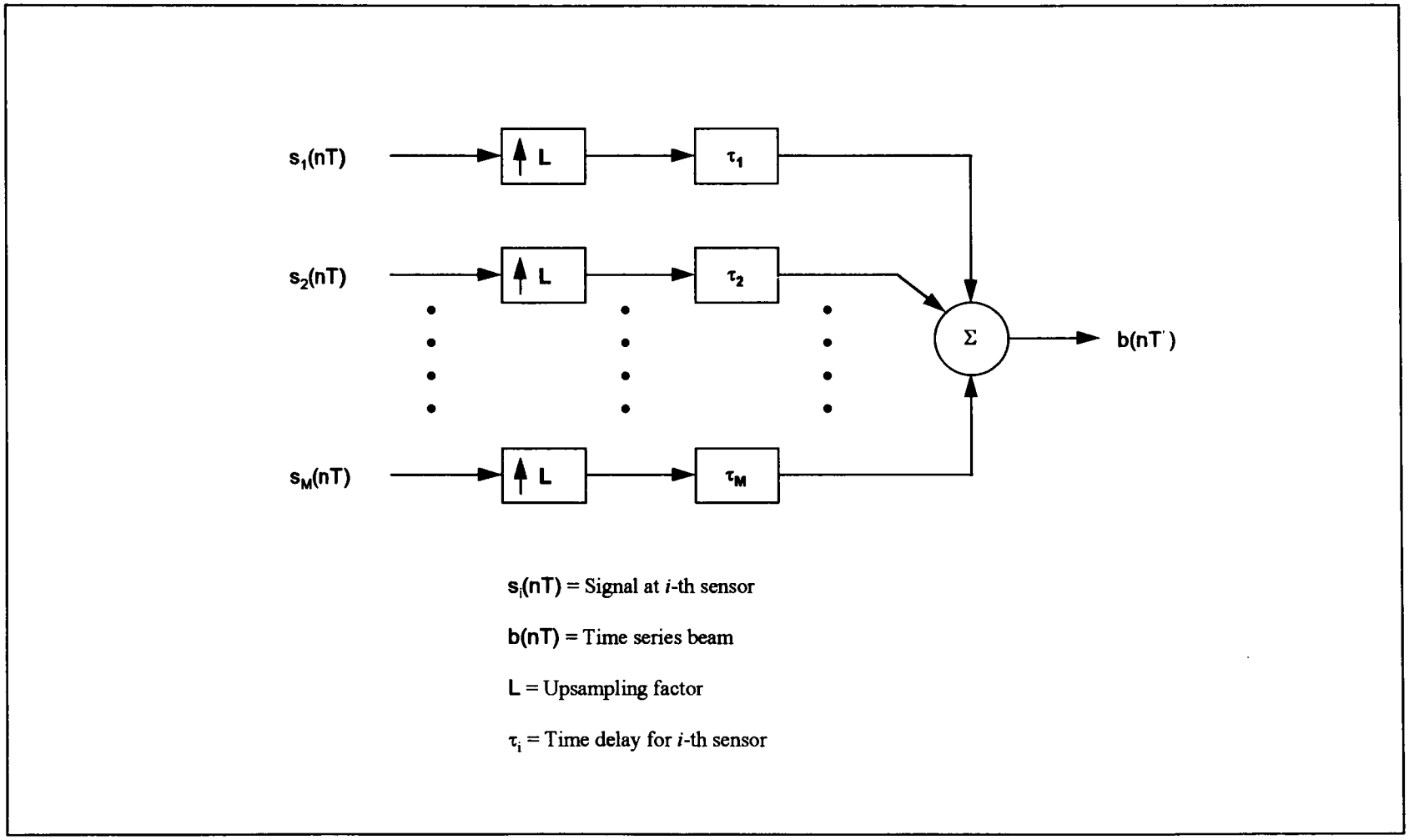

Figure 3.5. Block diagram of classical interpolation beamformer.

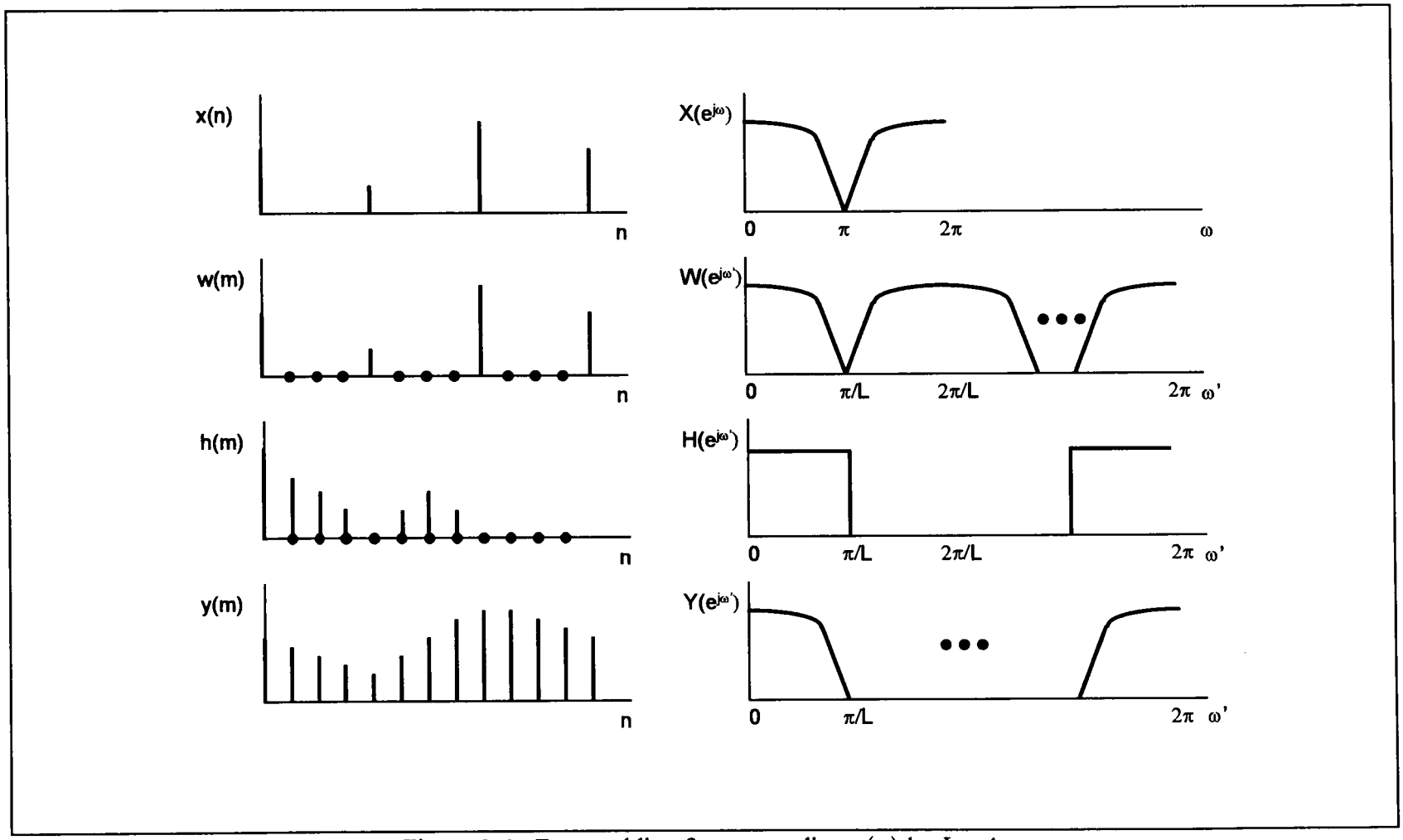

Figure 3.6. Zero padding for upsampling $x(n)$ by $L = 4$.

zero padding, this technique produces images in the frequency spectrum of the digitized signal. The zero padded sequence is then processed with a lowpass finite impulse response (FIR) filter, $h(m)$, to obtain the upsampled sequence, $y(m)$ [11]. The FIR filter acts as an anti-imaging filter having the response that approximates the ideal frequency response given by [11]

$$H(e^{j\omega'}) = \begin{cases} L & |\omega'| \le \dfrac{\pi}{L} \\ 0 & \text{otherwise} \end{cases}, \tag{3.7}$$

where

$$\omega' = 2\pi fT' = \frac{2\pi fT}{L}. \tag{3.8}$$

Interpolation Beamforming Algorithm Development

The operation defined by Eq. (3.6) consists of two major functional blocks: sampling rate conversion and index determination. Sampling rate conversion consists of producing upsampled points required for time shifting. Index determination is the process of selecting which upsampled points to use. The following sections describe the algorithm used for each functional block and the design considerations associated with these algorithms.

Sampling Rate Conversion Using Polyphase Filters

Because only a single beam is required for the test facility's measurement system, the classical method of interpolation illustrated in Figure 3.6 is not an efficient approach primarily because each upsampled point is produced. The production of L samples for every input sample translates into a significant increase in both computation and storage requirements. A single beam, $b(nT)$, only requires one interpolated sample per sensor. Thus, the beam samples have the same digitization rate as a single sensor.

An interpolation filter technique that implements the single beam case very well is the polyphase structure shown in Figure 3.7. This particular structure is referred to as a counterclockwise commutator model for a 1-to-L interpolator [11]. Each filter, $p_\rho(n)$, is called a polyphase subfilter, or simply polyphase filter. The polyphase filters are related to the anti-imaging filter, $h(m)$, through the relationship [11]

$$\begin{aligned} p_\rho(n) &= h(nL+\rho) \\ \rho &= 0, 1, 2, \ldots, L-1 \end{aligned} \tag{3.9}$$

In the polyphase structure, the order of each subfilter, i.e., the number of taps, is N/L, where N is the order of prototype filter, $h(m)$. The operation of the polyphase structure is demonstrated in Figure 3.8. The ρ-th subfilter produces the ρ-th interpolated sample between adjacent physical samples, and the output of each subfilter occurs at the input sampling interval, T. The commutator operates at a sampling interval of T/L, thus producing L output samples for each input sample. For the single-beam case, only one interpolated sample is required for each input sensor. Thus, for an interpolation beamformer, the selection of a particular time shift to steer the MRA of the beam involves

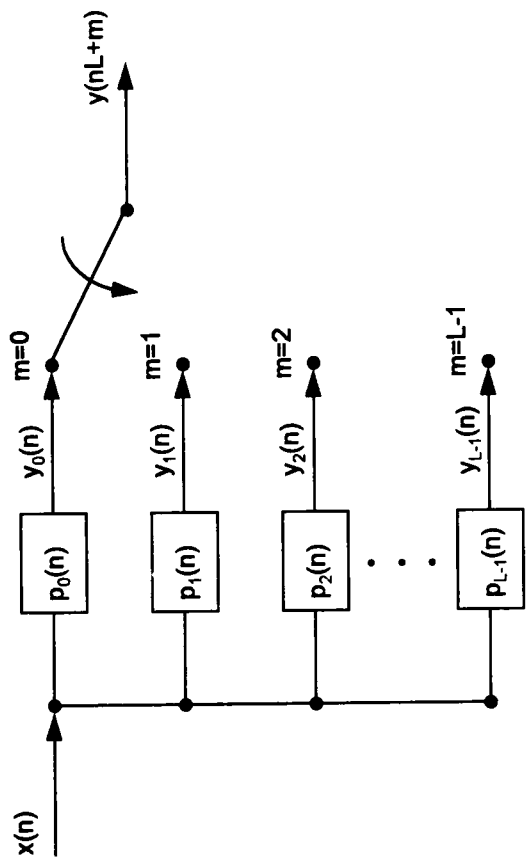

Figure 3.7. Counterclockwise commutator model for 1-to-L interpolator.

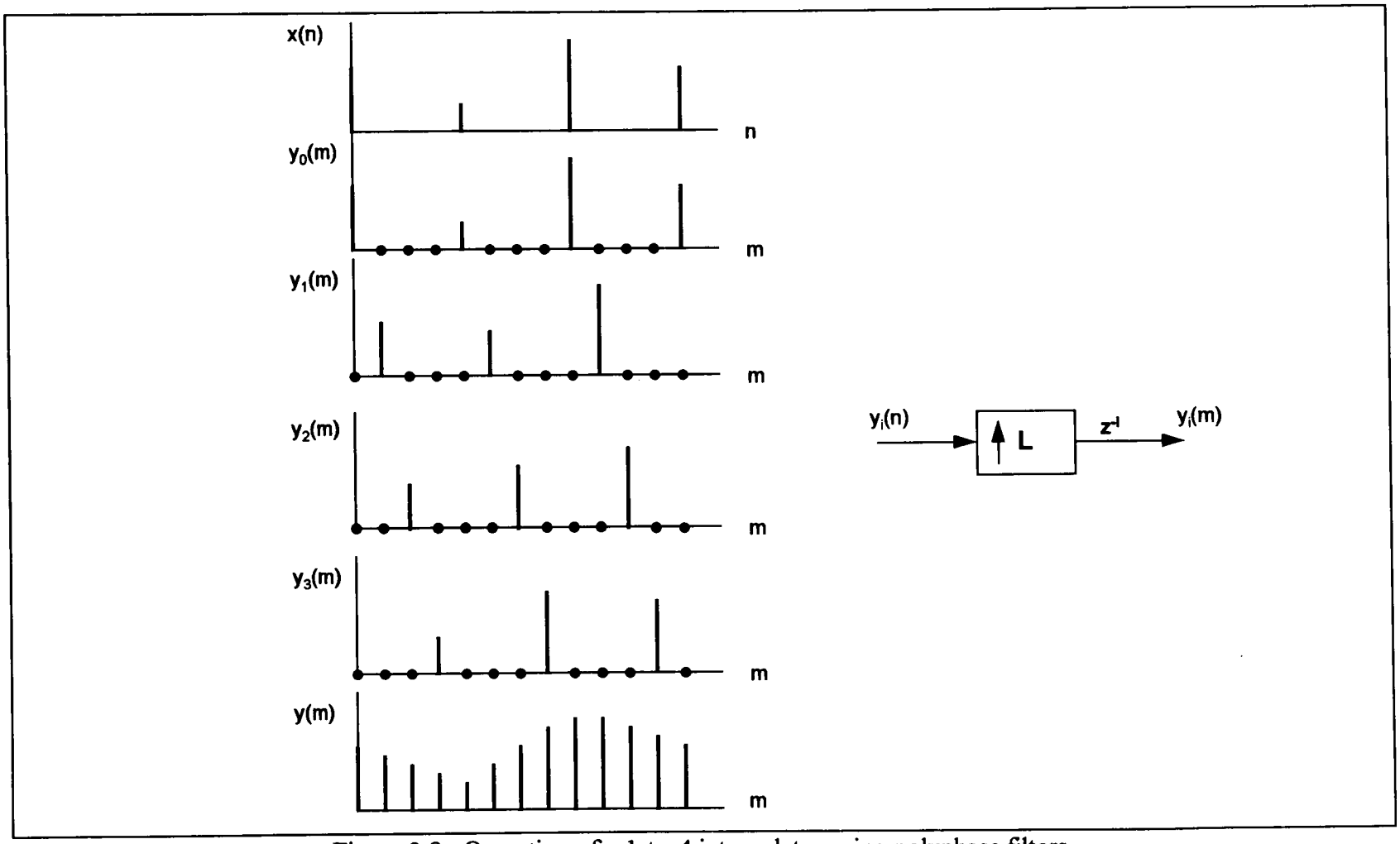

Figure 3.8. Operation of a 1-to-4 interpolator using polyphase filters.

the selection of the polyphase subfilter. For an M-element array, the interpolation beamformer using a N-tap prototype filter is represented by

$$b(nT) = \sum_{i=1}^{M} \left[\sum_{k=0}^{\frac{N}{L}-1} \mathrm{p}_{\rho i}(k) \mathrm{S}_i\{[(n+n_i-k)T]\} \right] , \tag{3.10}$$

where $S_i(nT)$ is the signal from the i-th sensor. The inner summation is the discrete convolution of $S_i(nT)$ with the polyphase subfilter, $p_\rho(n)$. The approximation for the time shift element, τ_i, in Eq. (3.1) is a function of the indices, n_i and ρ_i. The indices themselves represent time shifts in units of physical and interpolated samples, respectively. These indices are determined by the model used to compute the appropriate time shifts for each sensor.

Index Determination for a Focused Beamformer

The indices n_i and ρ_i are the integer and fractional time shifts, respectively, and their values are computed on the basis of a model of propagation. The most common assumption for beamforming with passive sonar is a far-field model. For instance, the expression for τ_i in Eq. (3.2) assumes planar wavefronts. In the case of the test facility, the test section dimensions prohibit the far-field assumption; instead, a near-field model is employed. In the near-field model, noise sources are interpreted as a set of points radiating equally in all directions. For instance, a continuous wave consisting of a single tonal frequency, f, and propagating without loss from a point in three-dimensional space designated (Xs, Ys, Zs) produces an output at the i-th sensor given by

$$S_i(t) = A\sin\left[2\pi f\left(t - \frac{D_i}{c}\right)\right] , \tag{3.11}$$

where D_i is the distance from the source to the i-th sensor given by [2]

$$D_i = \sqrt{(Xs - X_i)^2 + (Ys - Y_i)^2 + (Zs - Z_i)^2} . \tag{3.12}$$

The absolute time shifts required for steering the beam toward (*Xs, Ys, Zs*) are given by

$$Tabs_i = \frac{D_i}{c} . \tag{3.13}$$

In practice, the absolute time shifts are not used; instead a reference coordinate is selected so that a bulk delay [2] common to all sensors is obtained. Generally, the reference is either an actual sensor in the array or a virtual sensor having coordinates that are close to those of the physical elements. For instance, if the reference is selected to be the center of a line array having an odd number of elements, the reference would correspond to a physical sensor; however, if the number of elements in the array is even, the center of the array does not correspond to an actual element, and thus, the reference is characterized as virtual. This bulk delay is subtracted from the absolute delay to obtain relative time shifts as shown in Eq. (3.14):

$$\tau_i = \frac{D_i - D_{ref}}{c} \tag{3.14}$$

The fractional shift may be expressed in terms of the relative time shift as

$$\rho_i = \lfloor \tau_i f_s L + 0.5 \rfloor \oplus L , \tag{3.15}$$

where

$$(y) \oplus L = \begin{cases} R & y \geq 0 \\ L - |R| & y < 0 \end{cases}$$
$$R = \text{integer remainder of } \frac{y}{L} \tag{3.16}$$

and $\lfloor x \rfloor$ is defined as the greatest integer less than or equal to x. The addition of 0.5 is required so as to round ρ_i to the nearest integer. The integer time shift is expressed in terms of τ_i as

$$n_i = \lfloor \tau_i f_s \rfloor \tag{3.17}$$

Note that for the reference sensor, the values of n_i and ρ_i are zero. According to Eqs. (3.15) and (3.17), the time shift required for the i-th sensor is determined by first computing the total number of interpolated samples away from the reference sensor that most closely approximates the relative delay, τ_i. n_i represents the number of physical samples the signal from the i-th sensor must be shifted, and ρ_i determines the number of interpolated samples beyond n_i the signal must be shifted from the reference.

Design Considerations: Overlap Processing

As seen in Eq. (3.10), interpolation beamforming involves linear convolution. The linear convolution of two sequences, $x_1(n)$ and $x_2(n)$ is defined by the operation

$$x_3(n) = \sum_{k=-\infty}^{+\infty} x_1(k) x_2(n-k) \; . \tag{3.18}$$

If $x_1(n)$ and $x_2(n)$ have lengths of A and B, respectively, the length of $x_3(n)$ is given by [12]

$$C = A + B - 1 \; . \tag{3.19}$$

To correctly produce A samples of $x_3(n)$, C samples of $x_1(n)$ must be used. The impact of this fact depends mainly on the application and the platform used to compute the convolution.

For interpolation beamforming , one sequence, $x(n)$, corresponds to the input signal while the second sequence, $h(n)$, represents a polyphase subfilter. Typically, the length of $h(n)$ is very small compared with that of $x(n)$ since the input sequence is generally present for a long period of time. If the platform is a dedicated hardware device such as a digital filter integrated circuit having a structure shown in Figure 3.9, the impact is minimal because the device does not pay a penalty for transferring single words; rather, a single word is transferred into and out of the filter on every clock cycle. If the platform used to implement the filter is based on a central processing unit (CPU), a high penalty may be incurred for transferring only one word at a time. For data processing and transfers on most CPU-based platforms, high efficiency is realized by operating on blocks, or buffers, of data. In this case, the linear convolution operation requires that overlap processing be used. The algorithm selected for implementing block convolution is the overlap-save method [12]. Figure 3.10 illustrates the overlap-save technique for filtering two 64-point buffers of an input signal with a 7-tap interpolation filter. As seen in this example, the first valid output is $y(6)$. In order to produce 64 valid output samples, input data from the second buffer are required. Specifically, computation of $y(69)$ requires input samples $x(64)$, $x(65)$, $x(66)$, $x(67)$, $x(68)$, and $x(69)$ from the second buffer.

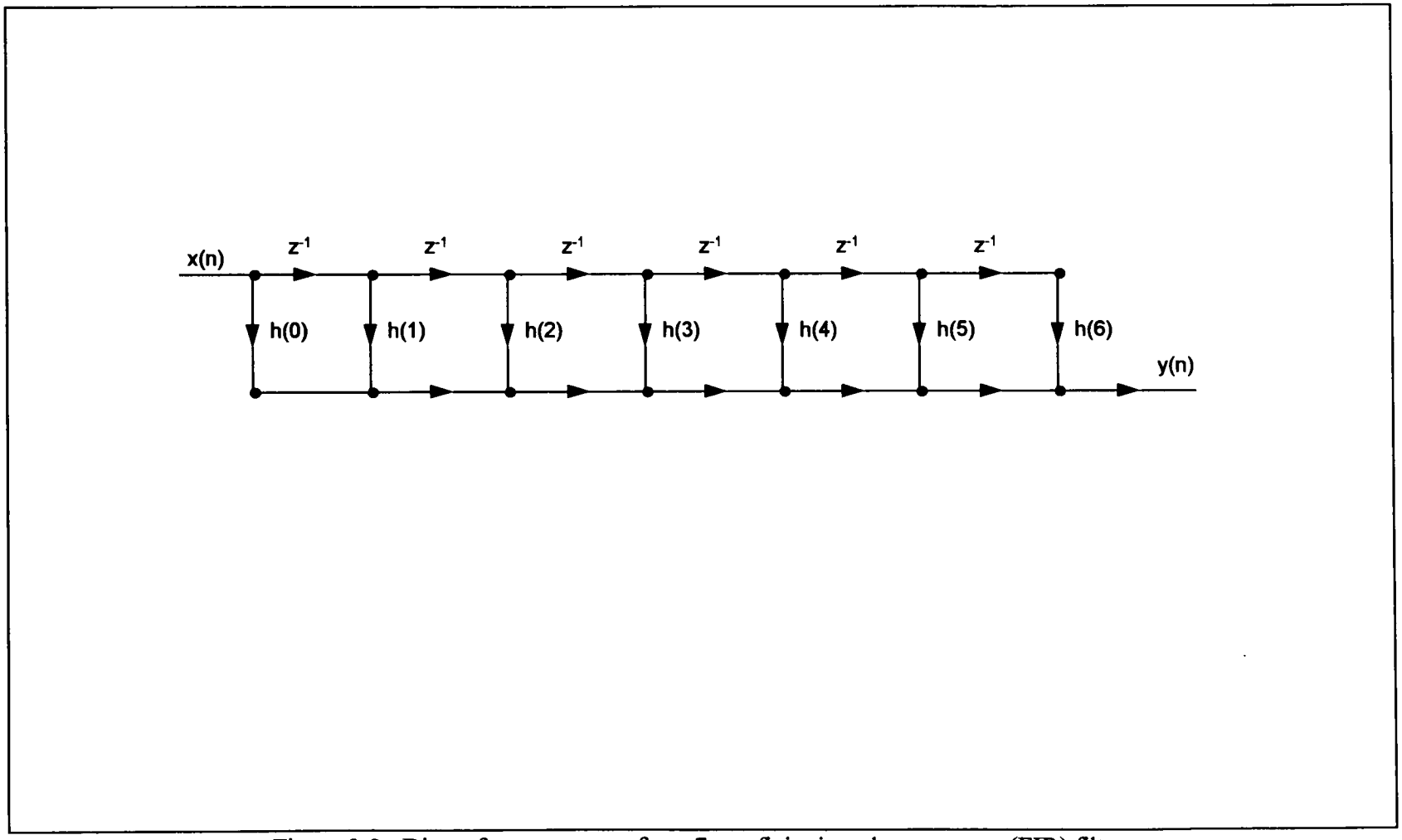

Figure 3.9. Direct form structure for a 7-tap finite impulse response (FIR) filter.

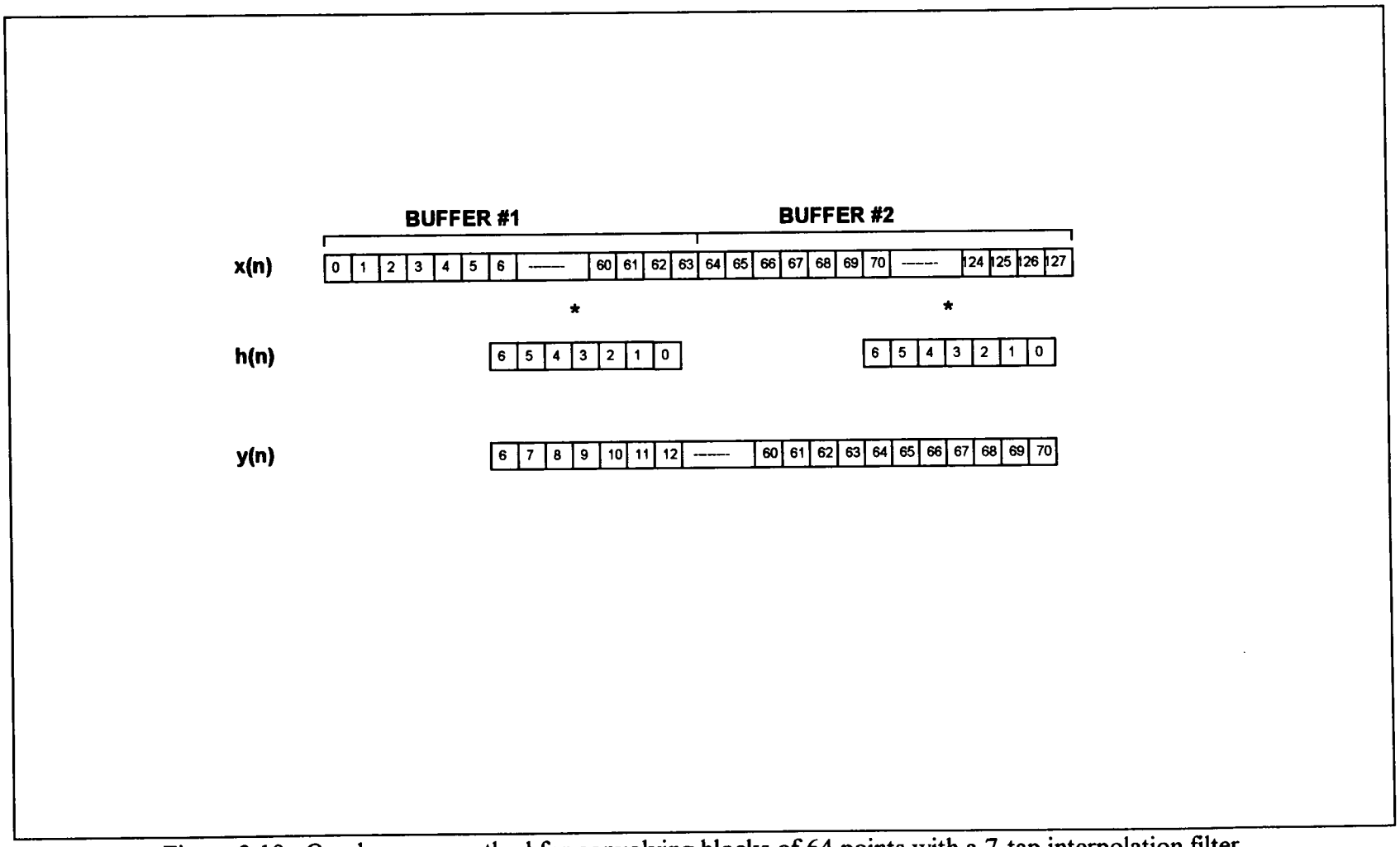

Figure 3.10. Overlap-save method for convolving blocks of 64 points with a 7-tap interpolation filter.

The overlap required due to the length of the polyphase subfilter does not account for the integer time shift, n_i, shown in Eq. (3.10). The coarse shift is due entirely to the beamforming operation and is a function of the focal point of the beam. The amount of overlap between adjacent buffers must be sufficient to accommodate the maximum relative time delay, τ_{max}. The maximum relative time shift is a function of the array geometry, the speed of propagation, and the field of view of the beam. The field of view is defined as the spatial region over which the beam is steered. For the 4 × 2 planar array shown in Figure 3.11, the maximum relative time shift, τ_{max}, is determined by computing the time required for an endfire beam to propagate across the array:

$$\tau_{\max} = \frac{\sqrt{(3d)^2 + d^2}}{c} . \tag{3.20}$$

The corresponding value for n_i is obtained by substituting τ_{max} into Eq. (3.17):

$$n_{\max} = \lfloor \tau_{\max} f_s \rfloor . \tag{3.21}$$

The relationship between n_i and the amount of overlap required depends on the choice for the array reference. If the array reference is chosen to be the center of the array, the samples of overlap needed for an endfire-to-endfire field of view is

$$OVERLAP = \pm \frac{n_{\max}}{2} \tag{3.22}$$

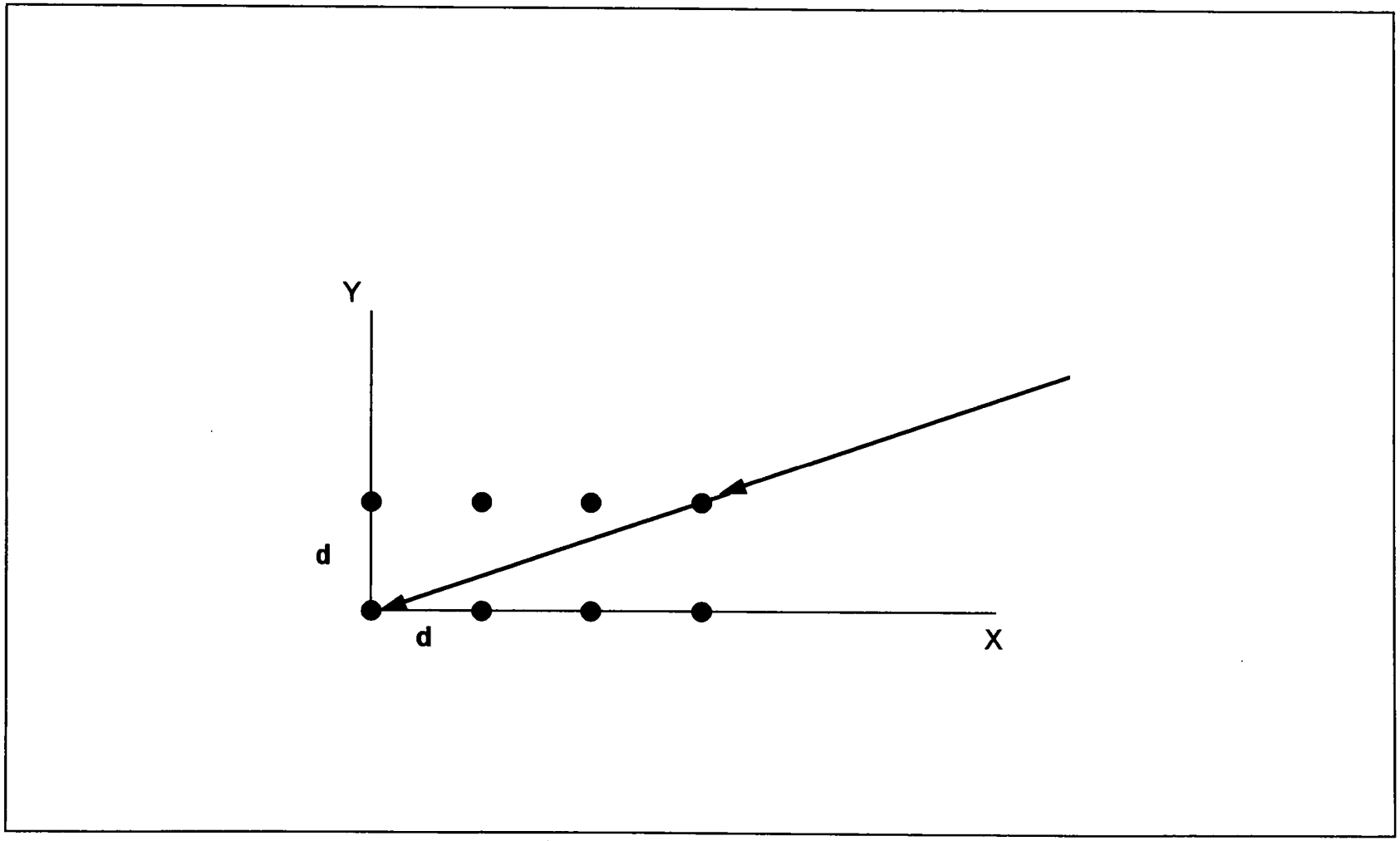

Figure 3.11. 4×2 Planar Array.
($d \equiv$ element spacing)

This expression states that the buffer design must accommodate both a time delay and a time advance equal to half the maximum relative time shift. If, however, the array reference is moved to either corner of the array, the buffer design must accommodate both a time delay and a time advance of n_{max} samples. Thus, a reference sensor at the end of an array increases the amount of overlap required by a factor of two.

Summary

By producing a single interpolated sample per physical sample, polyphase subfilters are a very efficient means of performing interpolation beamforming in the time domain; however, the real-time performance is determined by the platform on which the algorithm executes. The next chapter describes the system architecture of a beamformer based on the polyphase subfilter algorithm.

CHAPTER 4

SYSTEM ARCHITECTURE OF A REAL-TIME BEAMFORMER

The polyphase subfilter-based interpolation beamformer is an efficient algorithm that enables real-time processing of hydrophone arrays. The term real-time denotes the processing of signals while they occur as opposed to simply acquiring and storing data for off-line processing. In the case of a time-domain beamformer, the term "real-time" means that one sample of a time series beam is produced for each sample received from a single element in the array. Although necessary for a real-time beamformer, a time efficient algorithm is not sufficient to meet other system requirements. Several other factors are equally important to the goal of implementing a system that achieves both functional and performance requirements. These factors include analog signal integrity, user access to the algorithm, and a proper level of flexibility.

A block diagram illustrating the interface boundaries of the beamformer is shown in Figure 4.1. The beamformer must accommodate the selection of multi-aperture arrays. These arrays actually share some common elements as shown in Figure 4.2. The beamformer must also update the steering direction, i.e., the focal point, on the basis of a track specified by the user prior to starting data acquisition. Finally, the beamformer must produce a discrete-time output from which periodograms are produced.

This chapter describes the architecture of the acoustic signal processing system that performs interpolation beamforming based on the algorithm described in Chapter 3. In addition to system architecture, results from the beamformer are also presented in this chapter.

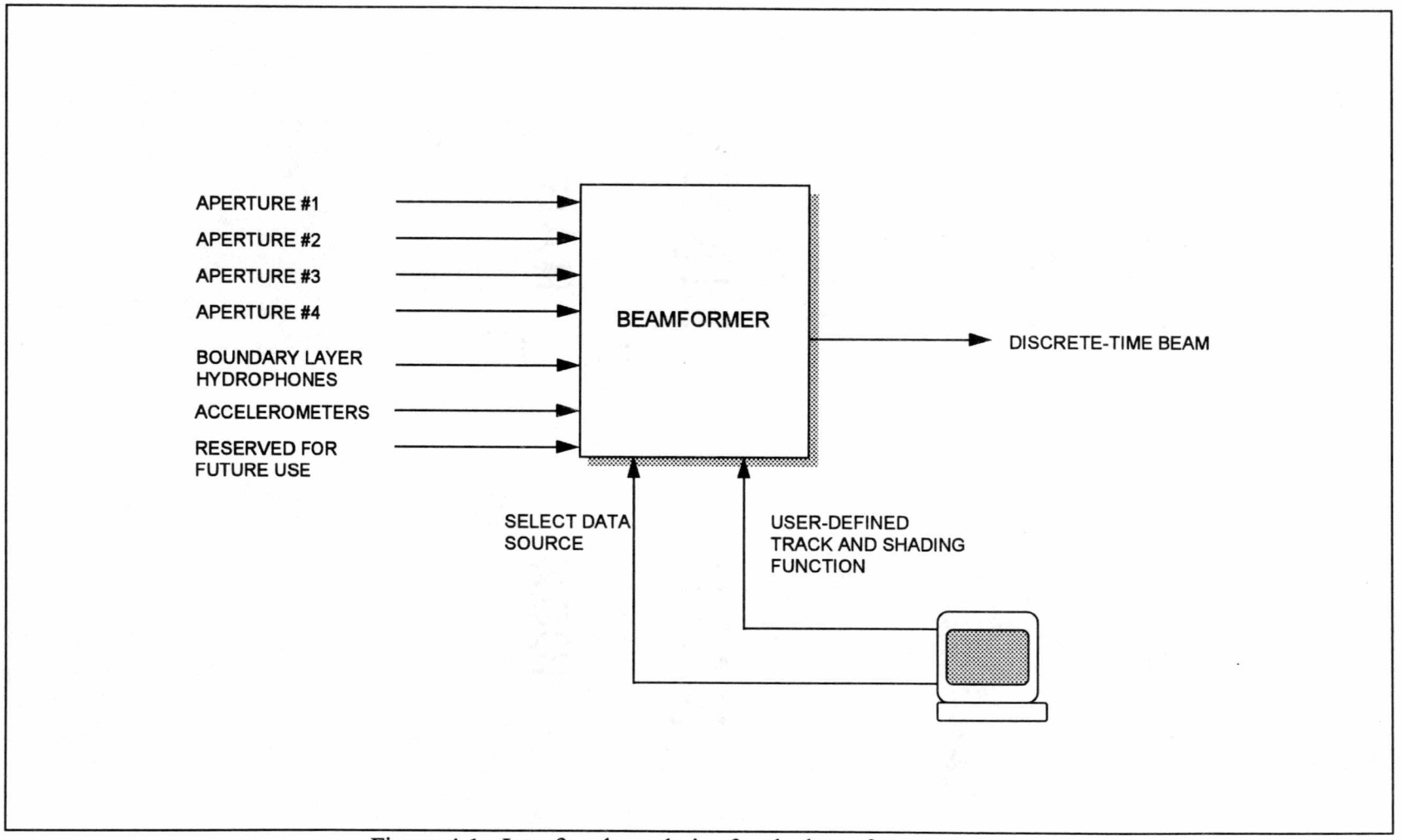

Figure 4.1. Interface boundaries for the beamformer system.

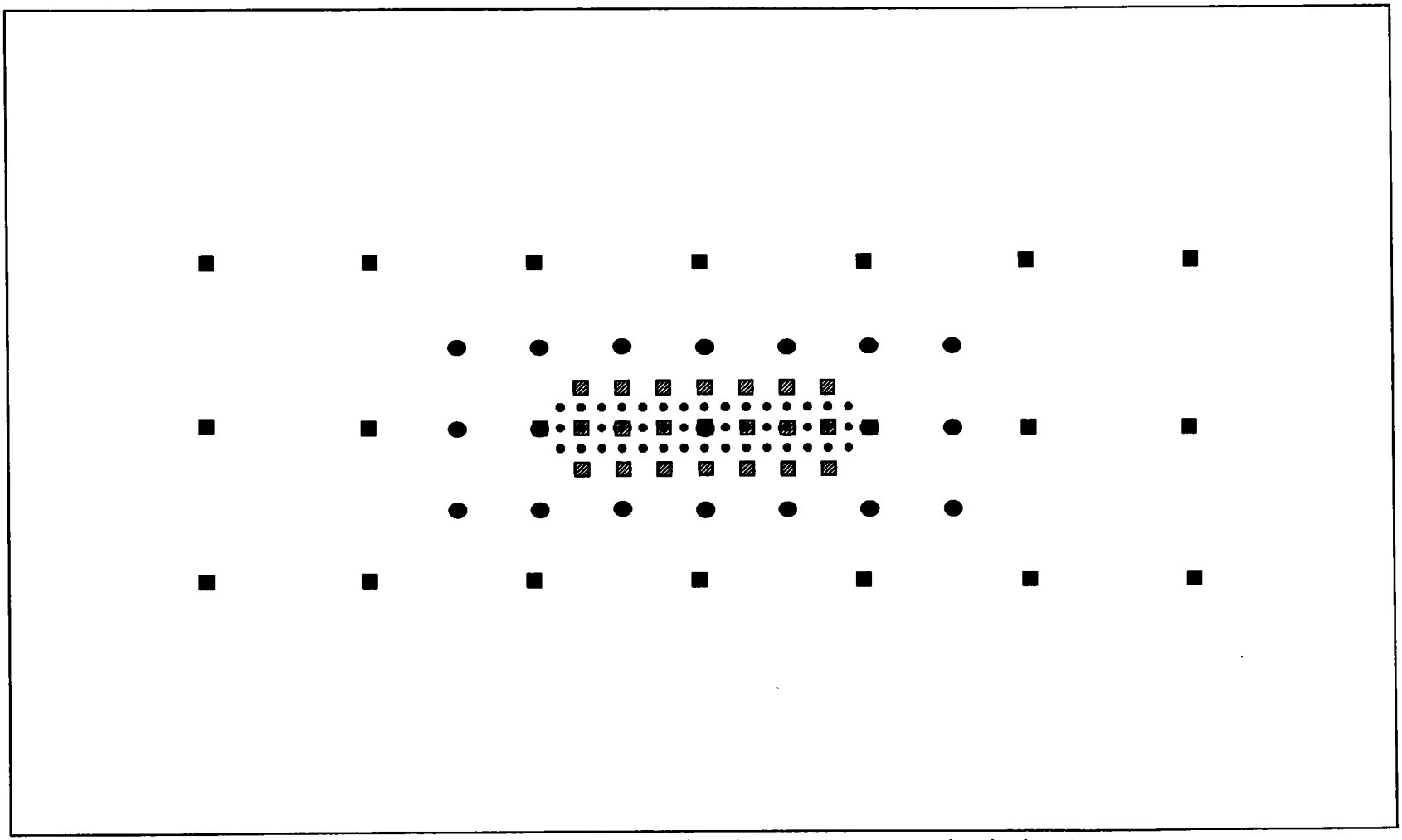

Figure 4.2. Multi-aperture array representing three 7×3 Arrays and a single 15×3 array.

Hardware Architecture

A block diagram of the acoustic signal processor is shown in Figure 4.3. The system consists of a signal selector, low-pass anti-aliasing filters, an ADC subsystem, and four array processors. Each hardware block shown in Figure 4.3 is configured by a MicroVAX 3900 host computer. In addition to configuring the hardware, the host computer starts and stops the data acquisition by controlling the ADC. Since all hardware represented in Figure 4.3 is commercially available, the host based architecture allows for future enhancements that may be more difficult with application-specific hardware. The following sections describe these hardware subsystems and their functions in the beamforming system.

Signal Selector

For a large facility such as that for which the acoustic signal processor was developed, a large number of sensors are deployed throughout the test section. Typically, all sensors are not used at the same time either because some sensors are not applicable for a particular experiment or some limitation in the measurement system prohibits the simultaneous processing of every sensor. An example of the latter is an ADC subsystem consisting of an analog multiplexer. The analog mutliplexer enables a single ADC to digitize multiple inputs rather than having one dedicated for each input. As the number of inputs increases, the required conversion rate of the ADC eventually exceeds the maximum operating frequency of the device. If the number of sensors does not exceed the specifications of the ADC, the data rate may become too high for the processing system.

The system shown in Figure 4.3 must beamform four multi-aperture arrays representing four separate octaves. Although desirable to produce the four time series

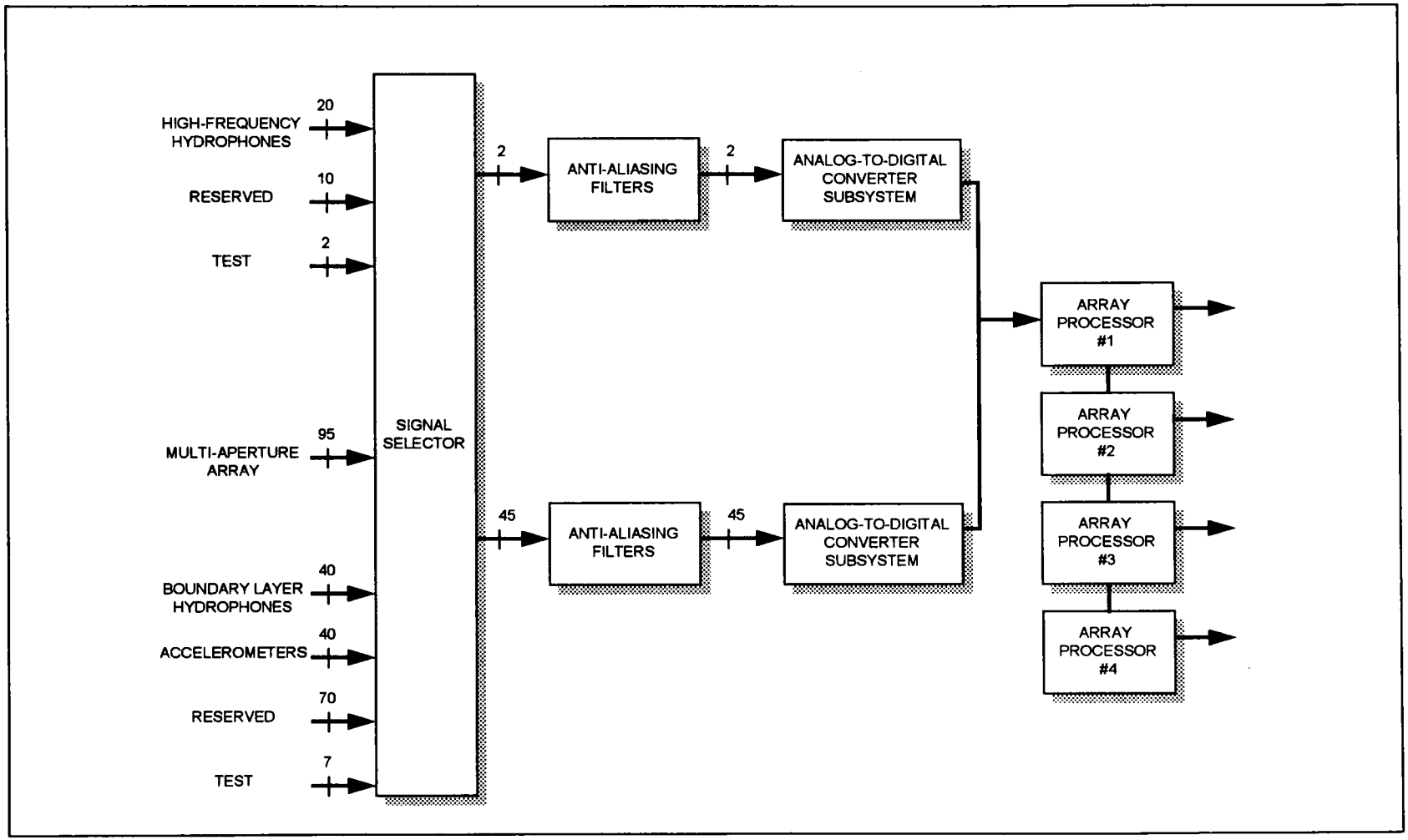

Figure 4.3. Acoustic signal processor block diagram.

beams simultaneously, the acoustic signal processor requires the user to process the four octaves consecutively due because of ADC and data rate considerations. (Actually, the acoustic signal processor allows the user to process the octaves in any combination. In fact, the user can choose to eliminate a particular octave.) In order to quickly switch the arrays that connect to the ADC subsystem, a signal selector is used. This signal selector is actually two System 4096 units manufactured by Precision Filters. Inc. Each System 4096 consists of four separate 36×12 matrix switch modules. As seen in Figure 4.3, one module is dedicated to channels that are used by the high-frequency processing system, while the remaining seven make up the signal selector for the beamformer.

Each module is capable of connecting any input to any output. In addition, a single input may be connected to up to 12 outputs. Because the System 4096 contains four modules, seven 36×12 matrix switches are not equivalent to a single 252 × 84 matrix switch. As a result, connecting the elements of the arrays to the signal selector cannot be done arbitrarily. The assignment of the input channels is further complicated by the fact that some elements are shared between arrays.

Anti-aliasing Filters

Digitizing a signal produces copies of its spectrum centered at integer multiples of the sampling frequency. Lowpass filtering is necessary to reduce the presence of aliased frequencies in the octave band of interest. The Nyquist sampling theorem states that a bandlimited signal, $x(t)$, given by

$$X(f) = 0 \text{ for } |f| > f_c \tag{4.1}$$

is uniquely determined by its samples if the signal is digitized at the a sampling frequency

$$f_s = 2f_c \ , \tag{4.2}$$

where f_C and f_S are defined as the Nyquist frequency and Nyquist rate, respectively. The bandlimiting requirement imposed by the Nyquist sampling theorem is not completely obtainable by physical components such as active and passive filters. Realizable filters consist of a passband, a stopband, and a transition region as shown in Figure 4.4. The Nyquist sampling theorem assumes a perfectly bandlimited signal, i.e., the transition region is zero, and the stopband attenuation is infinite.

Although the Nyquist sampling theorem cannot be satisfied, the effects of aliasing can be significantly reduced through proper use of analog filters. An analog filter is often characterized by its corner frequency, shape factor, and stopband attenuation. For a lowpass filter, the corner frequency defines the edge of the passband. The stopband edge is typically specified in terms of the shape factor, *SF*. The shape factor is the number by which the corner frequency is multiplied to obtain the lower frequency of the stopband. The shape factor sets a lower bound on the sampling frequency. As illustrated in Figure 4.5, the minimum sampling frequency that will attenuate the aliased spectra by the stopband level is given by

$$f_s = (SF + 1) f_c \ . \tag{4.3}$$

For the system shown in Figure 4.3, a Precision Filters, Inc., System 6000 is used for anti-aliasing. The System 6000 consists of linear phase filter modules having a shape factor of 3.34 and a stopband attenuation of 80 dB. Each filter module contains

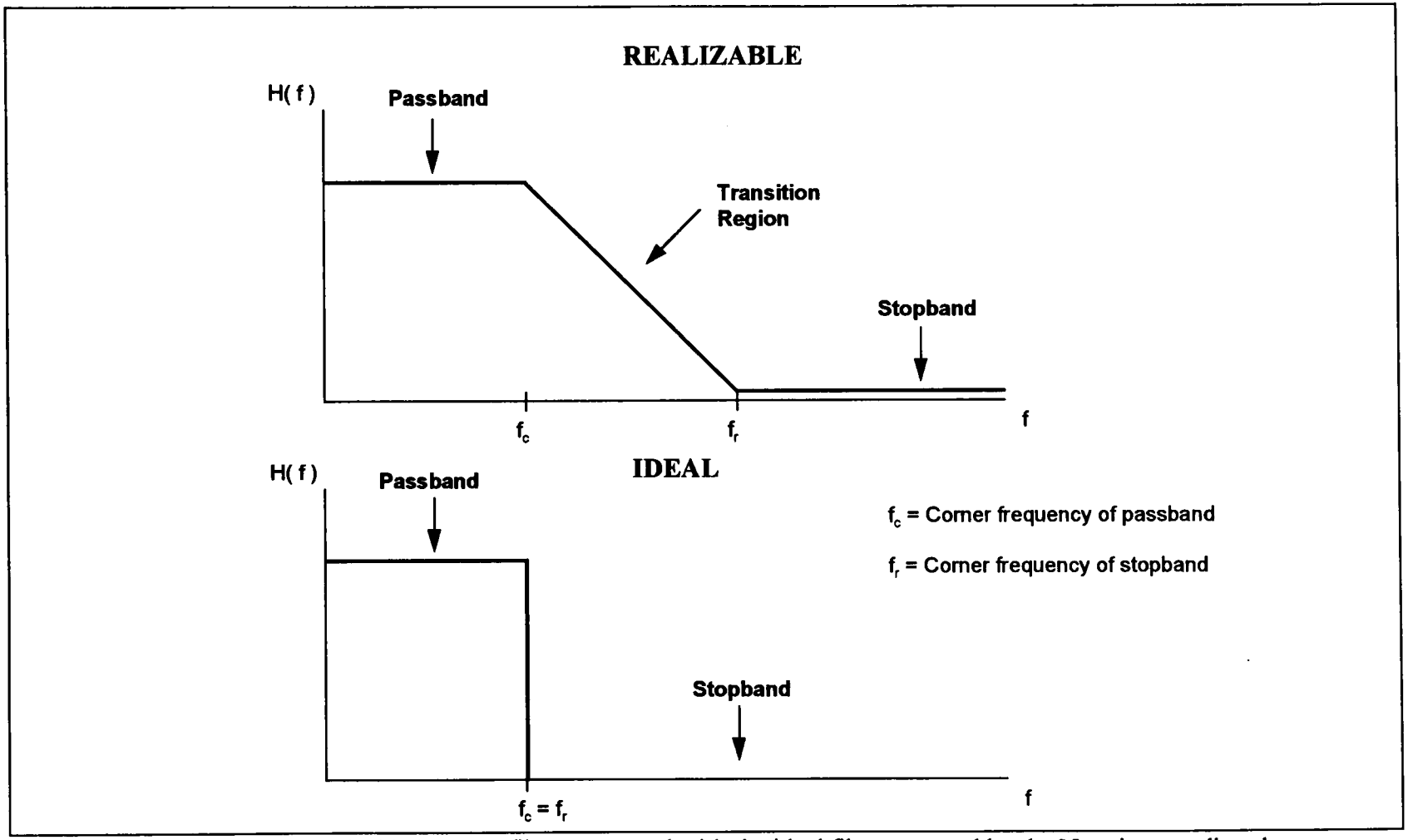

Figure 4.4. Physically realizable lowpass filter compared with the ideal filter assumed by the Nyquist sampling theorem.

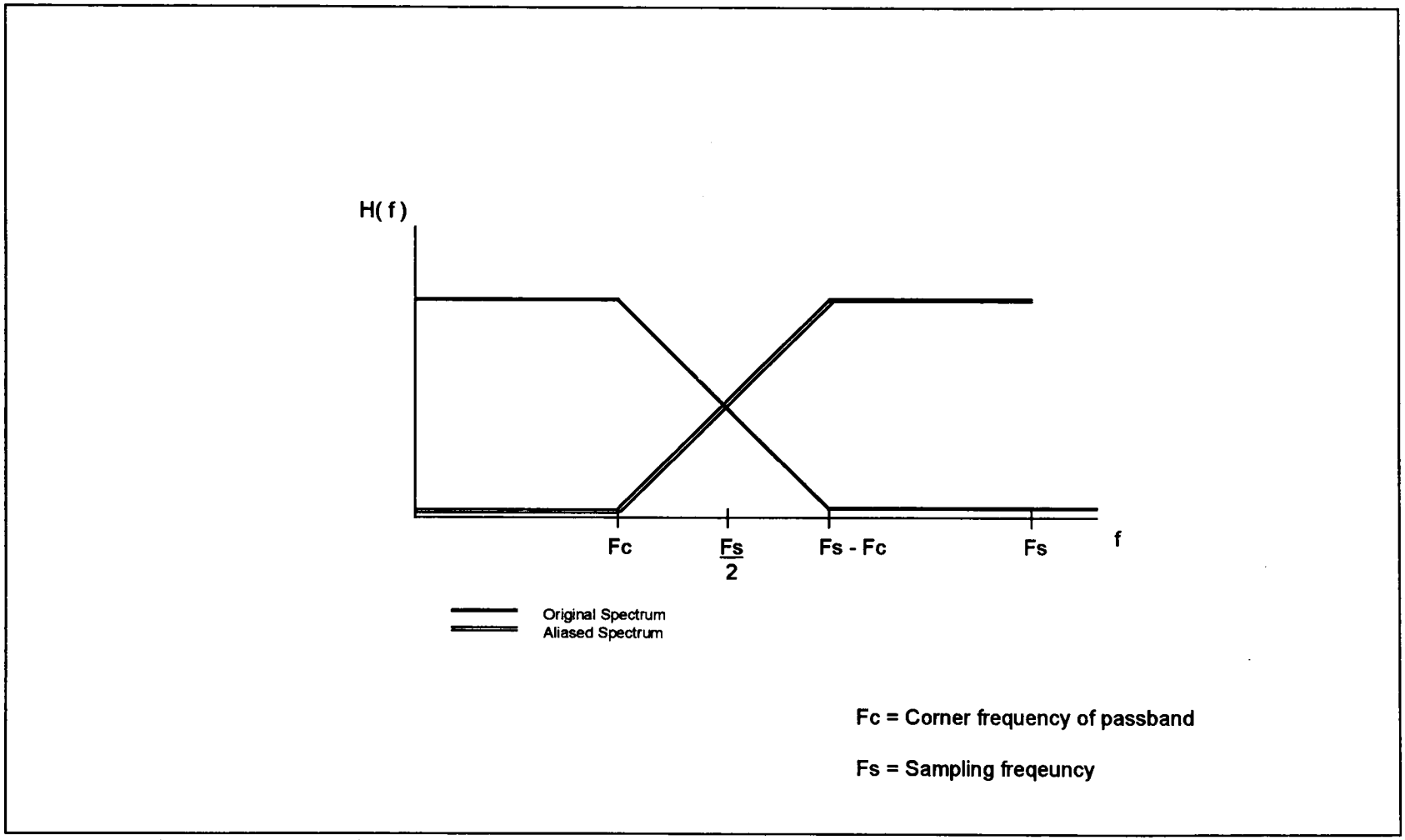

Figure 4.5. Illustration of relationship between sampling frequency and anti-aliasing filter's shape factor.

electronics for adding gain in both the prefilter and postfilter stages. In addition, the corner frequency of each filter is programmable in 1 Hz to 1023 Hz in 1 Hz steps and from 1000 Hz to 102.3 kHz in 100 Hz steps. The programmable corner frequency allows the use of one anti-aliasing filter subsystem for all apertures. All filter channels in the System 6000 are phase matched to within ± 1°.

Analog-to-Digital Converter

The ADC subsystem samples signals from the array and converts each sample to a 15-bit, twos-complement binary representation. A block diagram of the ADC subsystem manufactured by Tustin Electronics, Inc., is shown in Figure 4.6. The subsystem consists of six remote ADCs and a single central control and interface subsystem (CCIS). Each remote subsystem consists of a 1 MHz 15-bit analog-to-digital converter, an analog multiplexer, a scan list memory, an external start input, an interface to the CCIS, and simultaneous sample-and-hold (S/H) amplifiers. Five remote subsystems contain 8 analog channels, while the sixth subsystem contains only 5 channels. The simultaneous S/H amplifiers are a necessary part of any digital beamformer system since the various algorithms assume that the elements are digitized in an identical manner.

Digitization begins when the S/H amplifiers simultaneously sample the analog signal on their respective channels. Each channel is then digitized in the order specified by the scan list memory. The scan list memory instructs the analog multiplexer as to the order in which channels are to be converted. The actual conversion is performed by the ADC. The digitized data are then placed in a first-in-first-out (FIFO) buffer where they remain until collected by the CCIS. For the architecture of this subsystem, a distinction exists between sampling frequency and conversion rate. The sampling frequency relates to the rate at which the S/H amplifiers acquire a new analog sample. The conversion rate

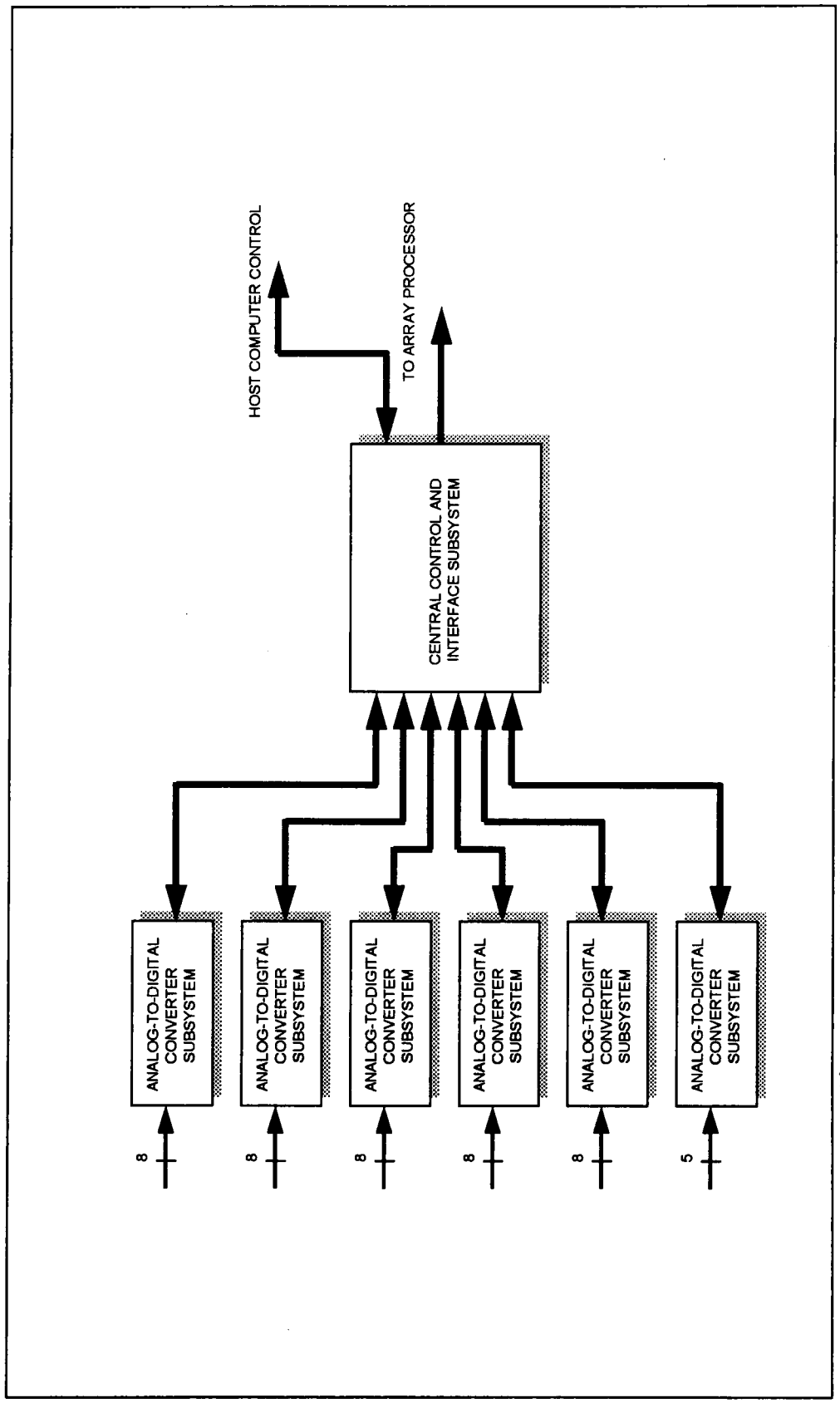

Figure 4.6. Block diagram of the analog-to-digital converter subsystem.

corresponds to the rate at which the samples are quantized. The samples from the S/H are converted in a burst fashion so as to minimize drooping.

The CCIS architecture is driven by the high throughput requirements of the beamformer. Specifically, the octave bands represented by the multi-aperture array of Figure 4.2 demand an aggregate sampling rate on the order of 5 million samples per second. To achieve this rate, the scan list memories of the remote subsystems and the CCIS must be configured properly. Specifically, the scan list memory must implement a commutator such as that shown in Figure 4.7. For properly digitizing the 45-element aperture, the scan list for each remote subsystem must scan through each channel. The scan list of the CCIS is then configured to scan through each subsystem. The output resulting from this commutator is illustrated in Figure 4.8. Data are transferred to the array processor in this commutated format.

Array Processors

The computational heart of the acoustic signal processor is the set of four STAR Technologies, Inc., ST-50 array processors. The ST-50 is a 50-MFlop processor consisting of 4 MWords of main memory, a 48 KWord data cache, and several internal processors. A block diagram of the ST-50 hardware configuration is shown in Figure 4.9. The internal processors include four input/output processors (IOPs), an arithmetic control processor (ACP), and a storage move processor (SMP). One IOP is dedicated to the array processor; the remaining IOPs are used to transfer data between main memory and external devices. For the system shown in Figure 4.3, Array Processors #2 and #3 connect to two other ST-50s. Array Processor #1 connects to one ST-50 and the CCIS, while Array Processor #4 connects to a single ST-50. Note that all array processors connect to the MicroVAX 3900 host computer. The ACP controls the arithmetic

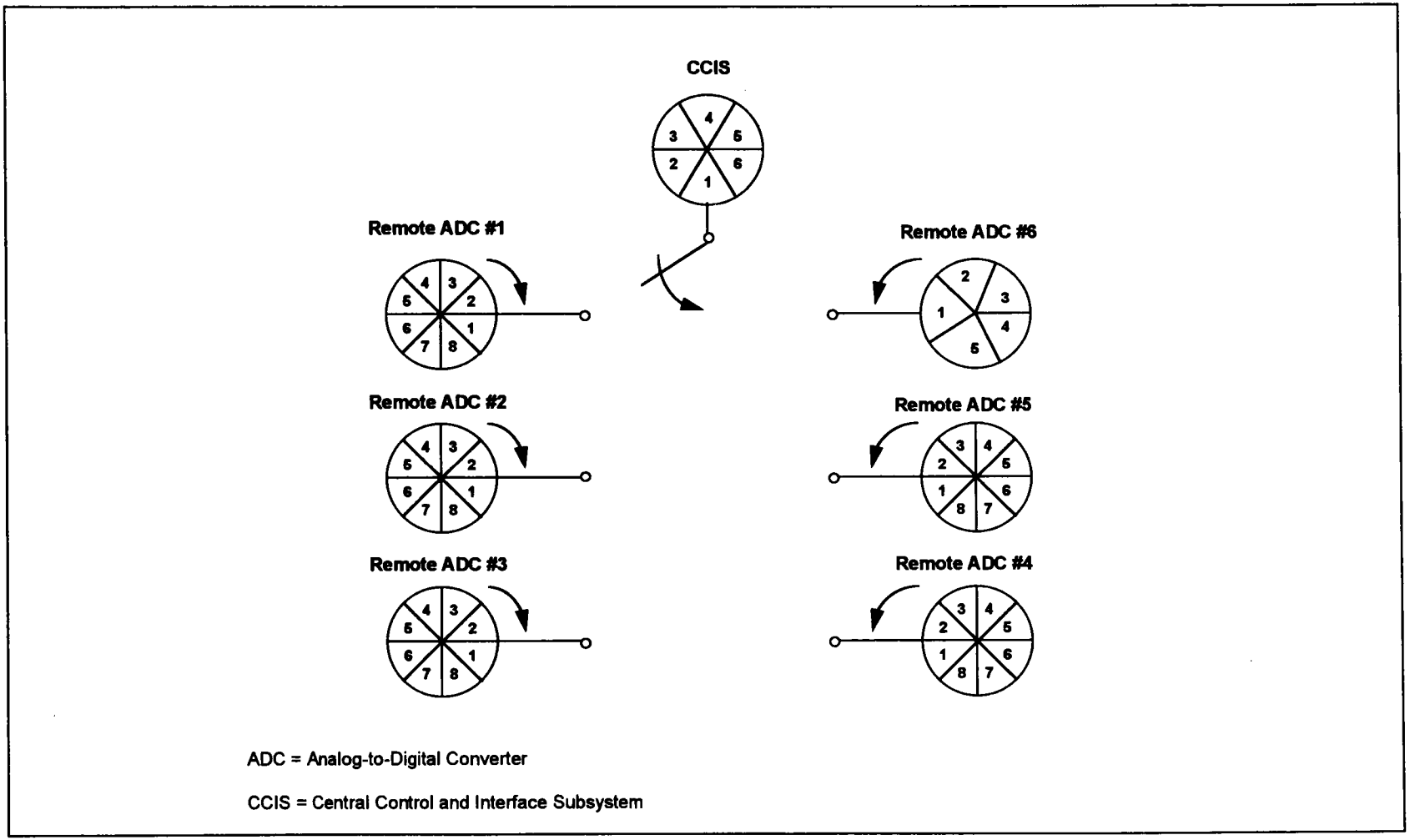

Figure 4.7. Commutator configuration of scan list memories.

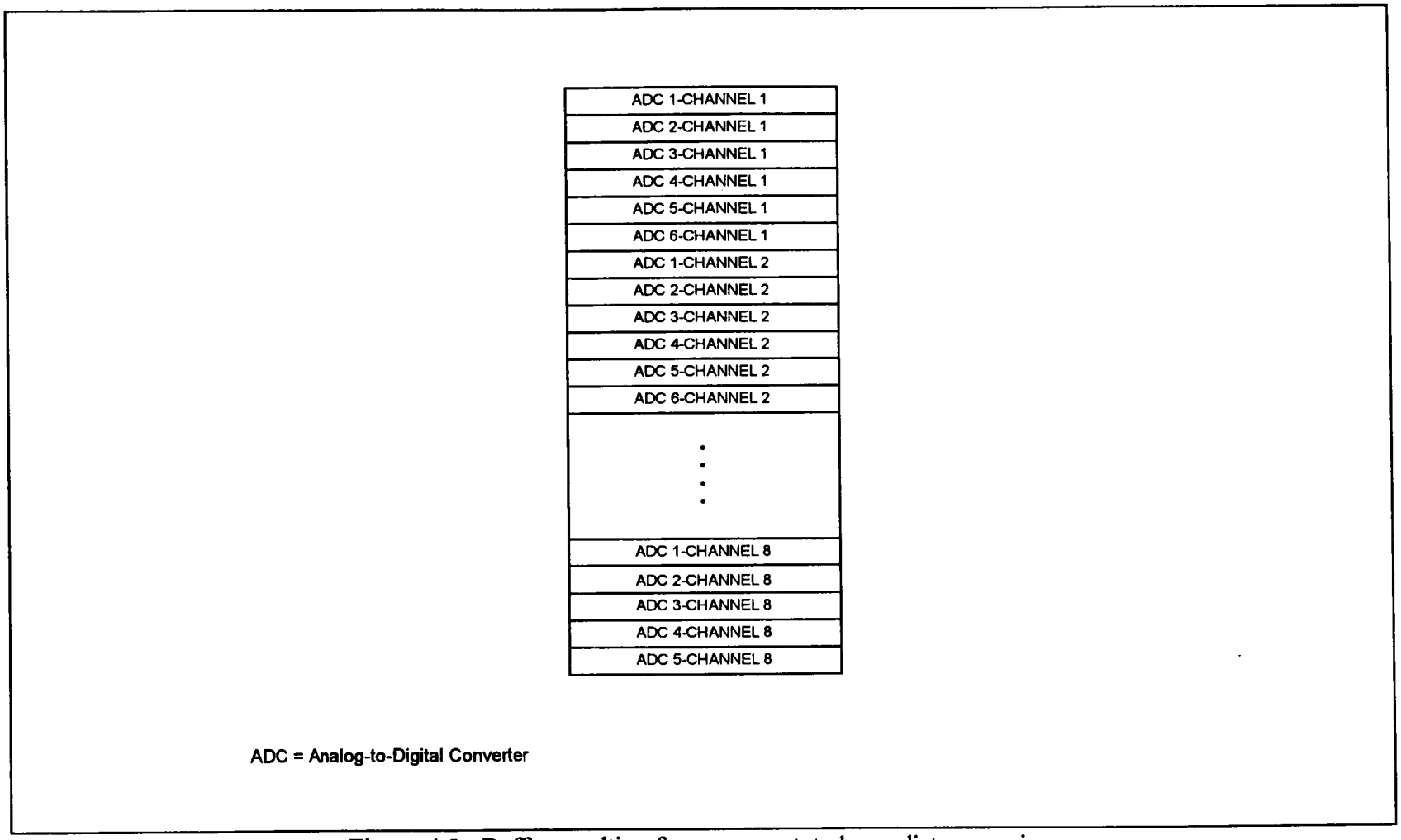

Figure 4.8. Buffer resulting from commutated scan list memories.

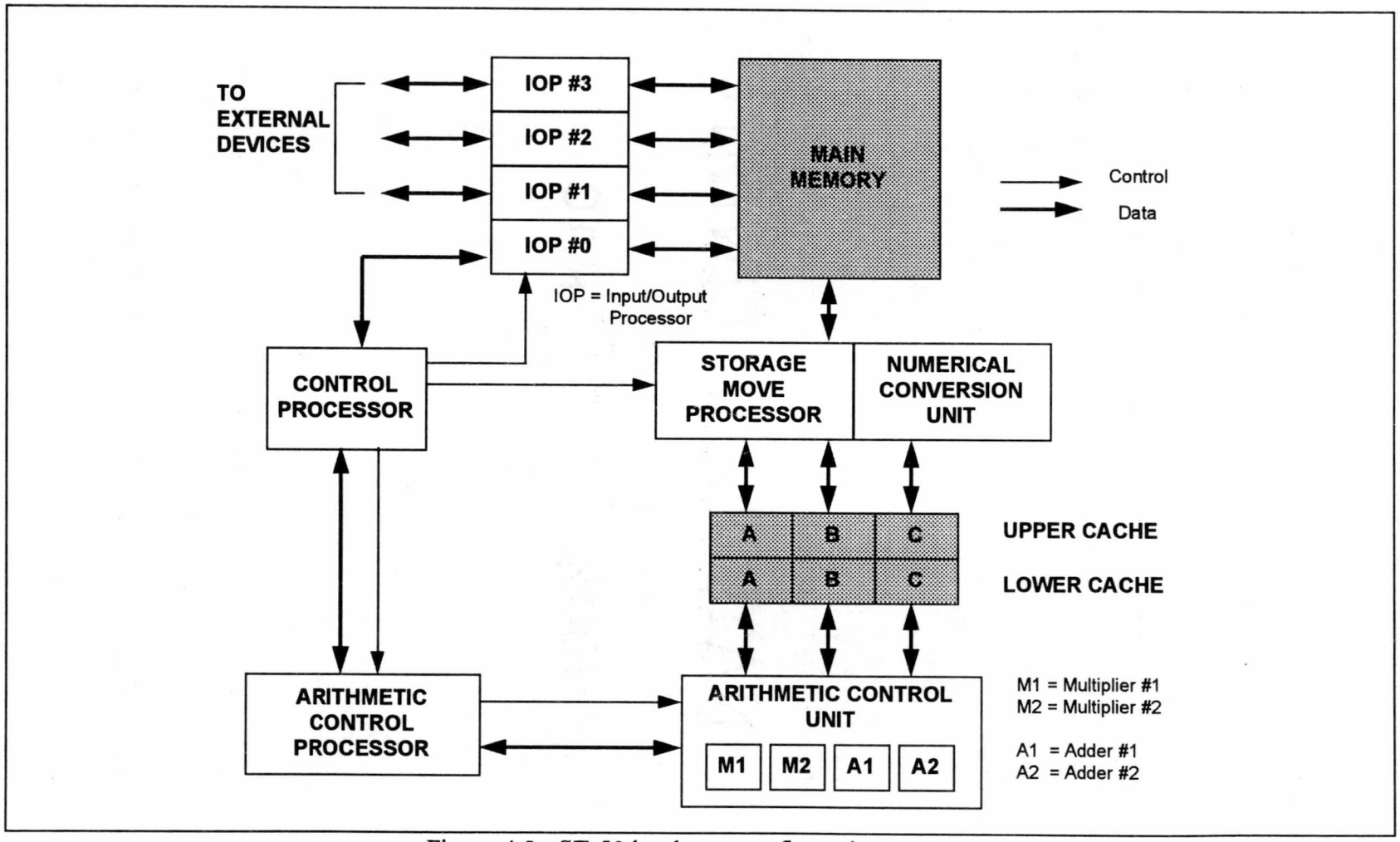

Figure 4.9. ST-50 hardware configuration.

computation unit (ACU) which consists of two multipliers and two adders. In addition, the ACP controls data movement between the cache and the ACU. The SMP controls data movement and corresponding format conversion between the main memory and cache.memory.

The ST-50 uses cache memory to achieve high throughput. The cache memory is partitioned into three 16-KWord banks. Typically, two banks, A and B, contain two vectors which are operands of a particular ST-50 instruction. The result obtained by executing the instruction is placed in the third cache, C. Each bank may be further partitioned into a two 8-KWord sections. In order to increase throughput, the architecture of the ST-50 allows the banks to be used by either the SMP or ACP, but not both simultaneously. For instance, while the ACP is executing an instruction on a set of vectors in the top cache, the SMP loads the next set of vectors from main memory into the bottom cache. Then, the ACP and SMP switch sections of cache so that the ACP executes an instruction using the operands in the bottom cache while the SMP moves the previously computed result to main memory and loads a new set of operands into the top cache.

Software Architecture

The hardware architecture previously described consists of commercially available, off-the-shelf subsystems. Such a platform is capable of a variety of signal processing applications. For instance, the STAR Technologies ST-series of array processors is used for processing applications in the aerospace and medical fields. Since the hardware architecture is not designed for the purpose of beamforming in real-time, the polyphase

subfilter algorithm must be partitioned into tasks that are assigned to various components within the acoustic signal processor. In addition to task assignment, a time line that defines the order of the tasks must be developed. For this beamformer, the tasks are categorized as either setup or run-time. Task assignment and time line development are governed by the following principle: the array processors are allocated for performing interpolation and sum defined in Eq. (3.10). Remaining tasks such as polyphase filter selection and shift indices are performed prior to acquiring data from the array.

A diagram illustrating the software architecture for the beamformer is shown in Figure 4.10. The bubbles represent objects that perform specific tasks. These objects interface with both data files known as stores and ultimately terminators. Terminators for the beamformer include the analog front-end, the periodogram generator, and the user. The following sections describe each bubble, its slot in the time line, and the method of implementation.

Hardware Configuration

The task of setting up the hardware is performed by SETUP HARDWARE which executes on the MicroVAX 3900 host computer prior to data acquisition. Configuring the hardware for data acquisition requires input from the user and knowledge of the hardware configuration. The user specifies the array to process by selecting the elements in the array by name. The use of signal name rather than array designation provides the user with the capability to use geometries other than the four planar arrays shown in Figure 4.2.

In addition to the set of signal names to process, the user defines the signals connected to the matrix switch input. The store designated HARDWARE CONFIGURATION defines the analog front-end's interconnection from the output of the matrix switch subsystem to the input of the ADC subsystem. On the basis of this

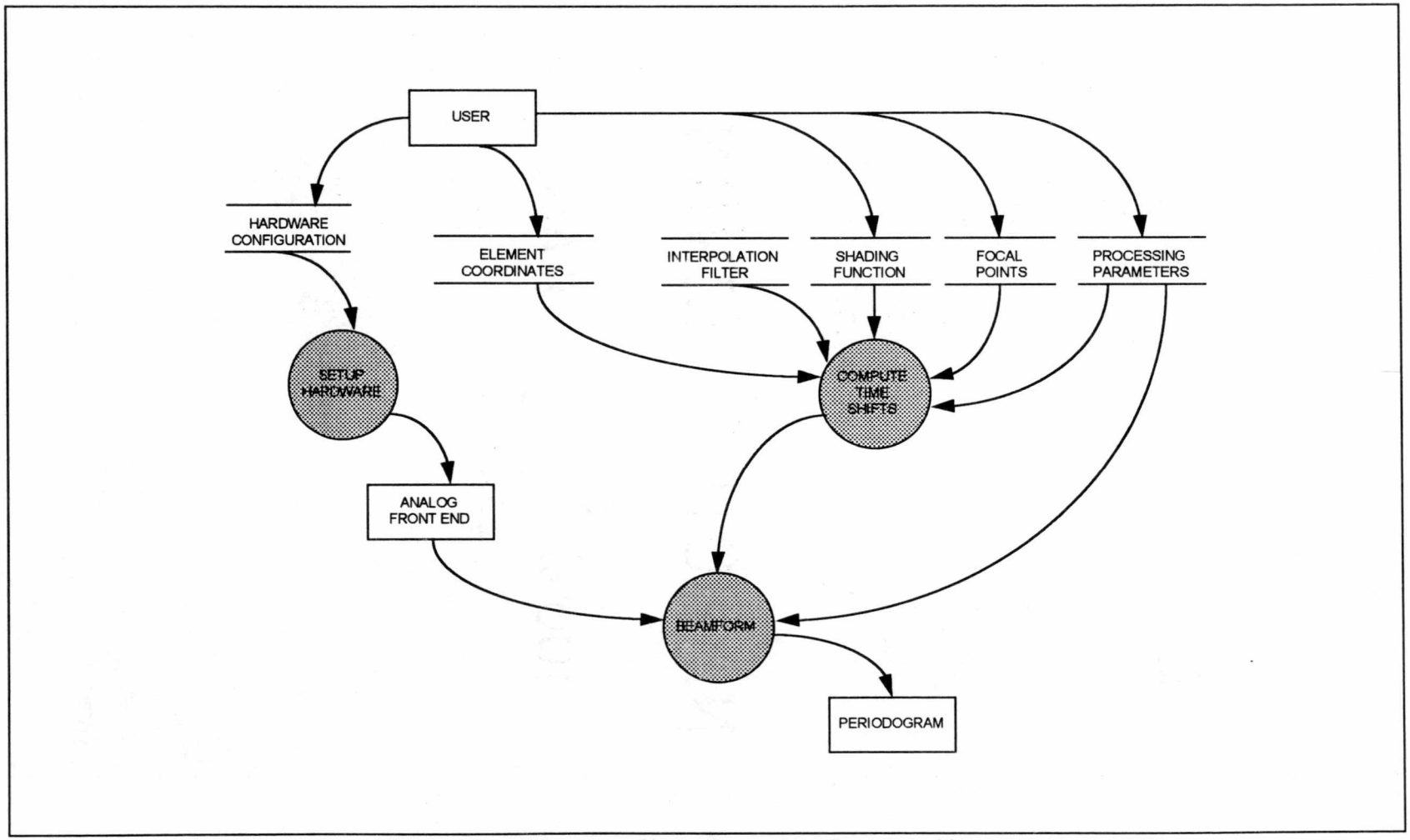

Figure 4.10. Diagram of beamformer's software architecture.

hardware configuration, SETUP_HARDWARE computes the required switch paths through the matrix switch subsystem and programs these settings through an IEEE-488 interface.

Besides configuring the matrix switch subsystem, SETUP_HARDWARE also programs via IEEE-488 the gain and corner frequency of each anti-aliasing filter channel. This information is contained in the store called PROCESSING PARAMETERS. Other processing parameters relate to the periodogram. These parameters include the octave band, integration period, window function, and the redundancy (percentage of overlap). SETUP_HARDWARE determines the sampling frequency of the ADC subsystem on the basis of the octave band and beamformer's oversampling factor.

In addition to computing the sampling frequency, SETUP_HARDWARE computes the scan list memories for the CCIS and the remote analog-to-digital converter subsystems. SETUP_HARDWARE contains an algorithm that determines the scan lists which will achieve maximum throughput. The algorithm implements the commutator model shown in Figure 4.7. Computing the scan list relieves the user of the responsibility of generating a complex scan list. Having computed the scan list, SETUP_HARDWARE downloads it along with the clock divisor to the ADC subsystem's CCIS via the DRV11-WA interface. SETUP_HARDWARE also communicates the new scan list to the task that computes the time shifts for beamforming.

Time Shift Computation

The beamforming operation shown in Eq. (3.10) requires a two-dimensional matrix of coarse and fine time shifts. One dimension of this matrix is indexed by scanned element, and the other is indexed by focal point. This matrix is computed by COMPUTE TIME SHIFTS which executes on the MicroVAX 3900 prior to data acquisition. The

information used for the computation of this matrix is contained in two separate data files. The first, ELEMENT COORDINATES, is a store whose contents are virtually static. This file contains the rectangular coordinates of each element in the multi-aperture array. While the element coordinates may easily have been made inaccessible to the user, the existence of this file serves two important functions. First, the file may be modified to allow for new arrays in the future. Second, this file may be changed to reflect the exact dimensions of the array rather than nominal values which would have been used since the completion of the beamformer preceded the completion of the array. The second, FOCAL POINTS, is a store whose contents may change depending on the test. FOCAL POINTS consists of up to 50 different steering directions for the beam specified by the user. Whereas each element in the coarse time shift matrix contains a single value, an element of the fine time shift matrix represents a polyphase subfilter. The fine time shifts are achieved by a 140-tap, 20:1 upsampling filter. The impulse response of this filter is shown in Figure 4.11. Thus, the fine time shift matrix is a set of 7-tap FIR filters. COMPUTE TIME SHIFTS selects the subfilter for each sensor and focal point and then multiplies the impulse response by the shading coefficient for the appropriate sensor. Thus, the additional multiplication required for the shading coefficient is accounted for in the gain of the polyphase subfilter. Having computed the shift matrices, COMPUTE TIME SHIFTS downloads them to the array processors where Eq. (3.10) is implemented. The completion of this task signifies the end of setup tasks.

Beamforming Operation

The real-time realization of Eq. (3.10) is performed by three ST-50s connected in the series configuration shown in Figure 4.3. This series configuration is conducive for pipeline processing. A diagram illustrating a three-stage pipeline for beamforming is

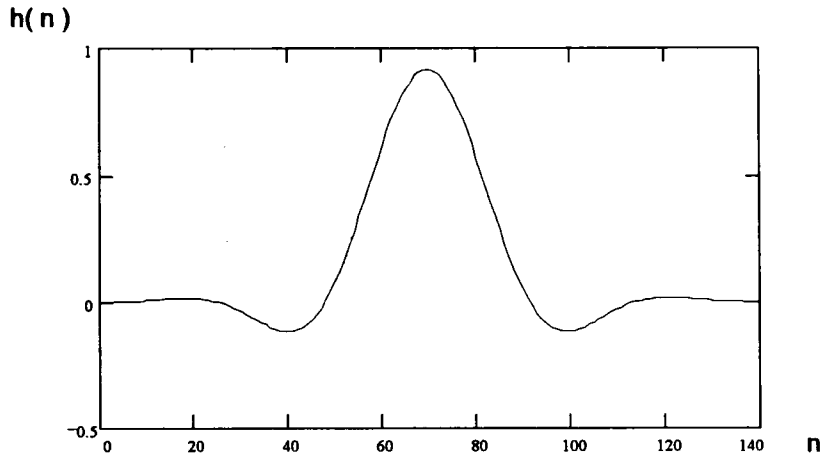

Figure 4.11. Finite impulse response of 20:1 interpolation filter.

shown in Figure 4.12. The pipeline is implemented by having each array processor perform the same function. Specifically, each ST-50 is assigned the task of beamforming one-third of the sensors in the array up to a maximum of 15 channels. Having beamformed its respective set of channels, the ST-50 must add the partial beam to the result from the previous stage. As seen in Figure 4.12, data from the ADC are received by the first stage of the pipeline, Array Processor #1. Array Processor #1 beamforms channels 1–15 according to Eq. (3.10) and transfers the resulting beam and the time series to the second stage, Array Processor #2. Array Processor #2 performs the identical set of operations on channels 16 – 30. The result from this operation is added to the partial beam received from Array Processor #2. The updated partial beam and time series is passed to the third stage, Array Processor #3. Array Processor #3 repeats the operations of Array Processor #2 on channels 31–45. Since Array Processor #3 is the last stage of the pipeline, the partial beam corresponds to the full beam. Thus, only the beam is transferred to the periodogram terminator. The pipeline architecture is very modular in that the need to beamform more channels may be accommodated by adding more array processors with minimal modifications required to the setup software.

In order to process long streams of data efficiently in a machine such as the ST-50, multiple buffers must be employed to hold incoming data while processing is being performed on a single buffer. For this beamformer, the five-section circular buffer shown in Figure 4.13 is used. Four buffers are active at any one time; the remaining one provides additional buffering at the input in case of an occasional perturbation in the data flow. For instance, while the beamforming operation is performed on Buffer #3, data from the ADC subsystem are placed in Buffer #5. Buffer #2, which contains the time series and a partial beam, is sent to the next stage in the pipeline. Buffer #4 is required to beamform Buffer #3 because of the overlap considerations outlined in Chapter 3.

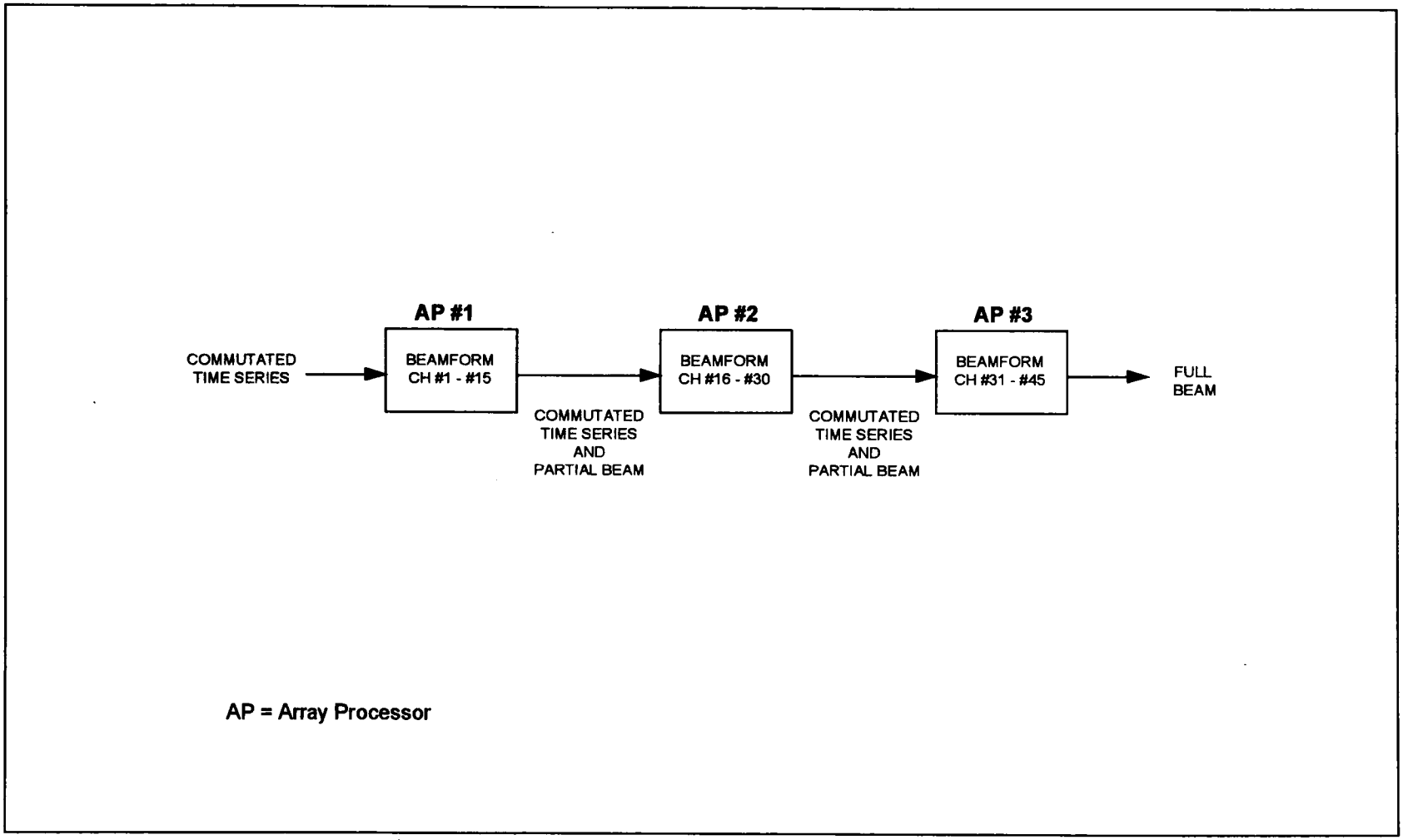

Figure 4.12. Pipeline architecture of real-time beamformer.

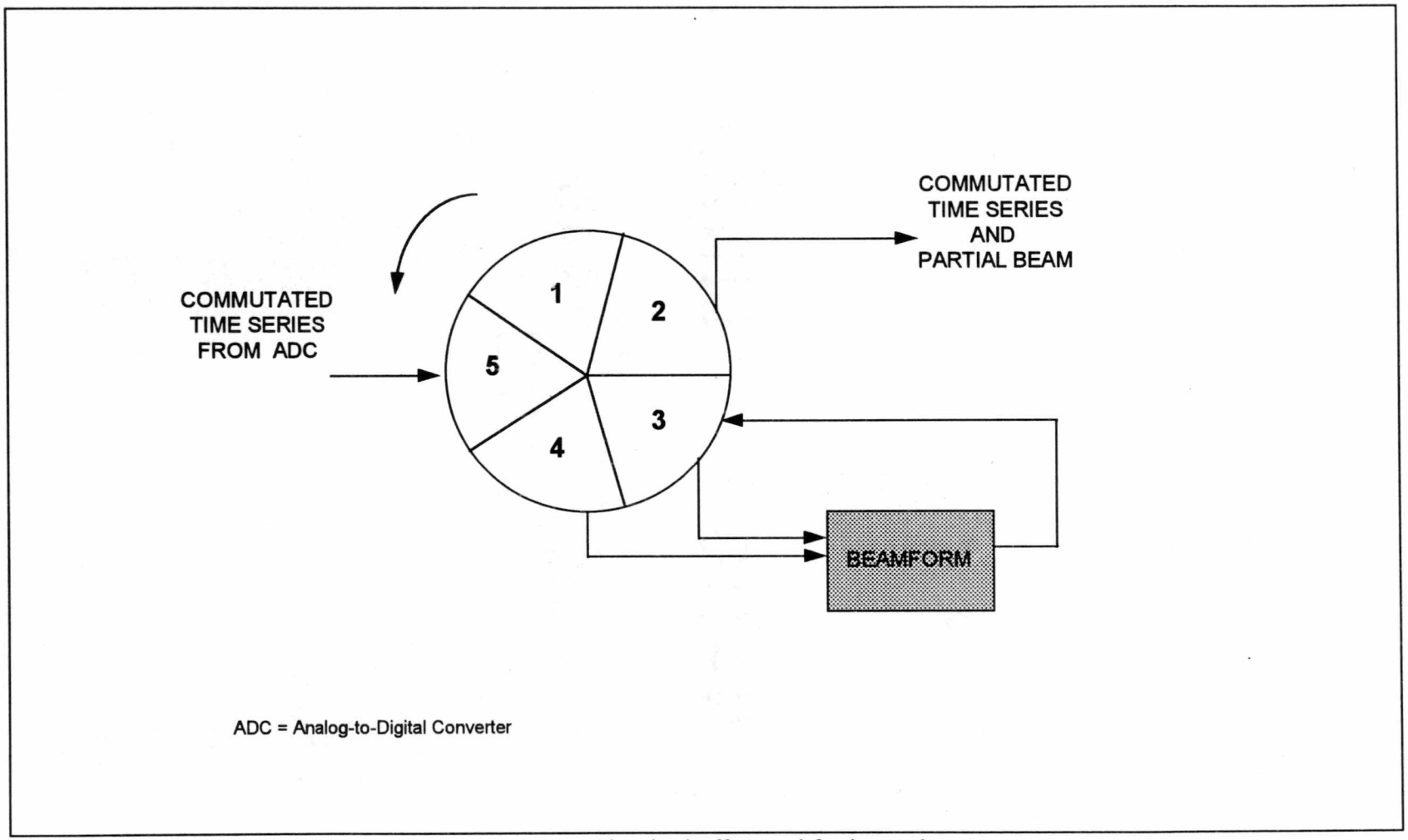

Figure 4.13. Circular buffer used for input data.

Cache Mapping for 7-Tap FIR Filter

The beamforming block shown in Figure 4.13 is performed by the ACP. Because the sustained input data rate is on the order of 10 MBytes per second, an efficient method of implementing Eq. (3.10) is required. The cache mapping and ACP execution scheme used for the polyphase subfilter based beamforming algorithm is based on an FIR filtering technique developed for the ST-50 array processor by W. D. Zuehsow, an engineer in the Instrumentation and Controls Division at Oak Ridge National Laboratory. The scheme shown in Figure 4.14 makes use of the AVADD (Vector Add) and PENTA macros available in the ST-50 ACP macro library. The PENTA macro is the most efficient of the ACP macros because it computes two multiplications and three additions per sample in a single machine cycle, which for the ST-50 corresponds to 80 nanoseconds.

For the scheme illustrated in Figure 4.14, one beamformed sample is produced every seven cycles. This cache mapping is identical for both the top and bottom cache banks. While the ACP is executing the processing outlined in this scheme in one cache bank, array data are loaded into the other cache bank. Each block in Figure 4.14 represents a vector having a length consistent with the data buffer size. For instance, blocks containing the time series data have a length of 2055 accounting for both the 2048 output samples and the 7-tap impulse response of the polyphase subfilter. T1, T2, T3, T4, S1, S2, and BEAM are temporary result arrays having a length of 2048. (Note that the cache arrays, S1, S2, and T2 are actually equivalenced to other arrays; however, separate labels have been used for clarity.)

The linear convolution required for the interpolation filter may be written in the piecewise manner defined by

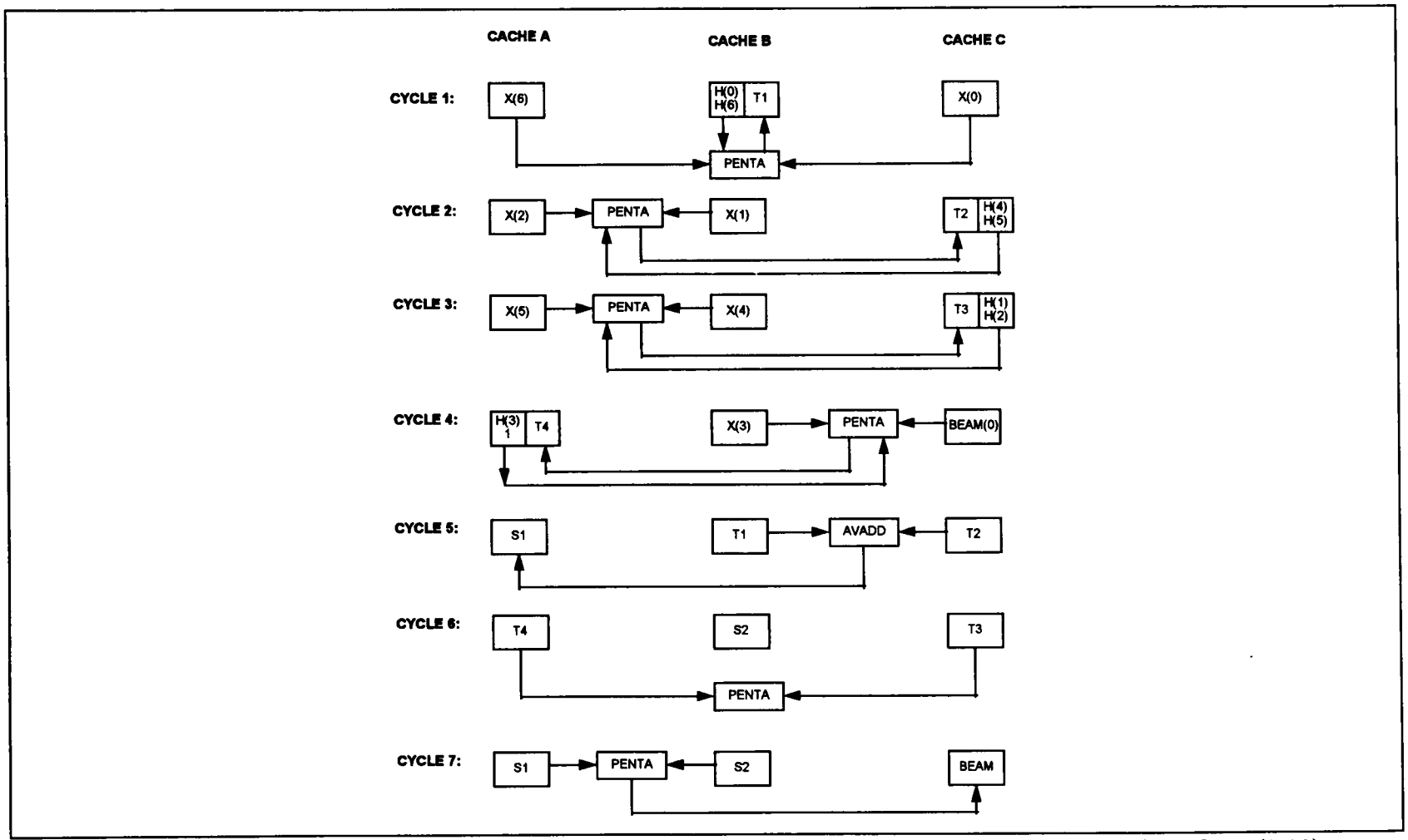

Figure 4.14. Cache mapping and arithmetic control processor execution for real-time implementation of Eq. (3.10).

$$y(n)=\sum_{k=0}^{6}h(k)x(n-k)=\left[h(0)x(n)+h(6)x(n-6)\right]+\left[h(4)x(n-4)+h(5)x(n-5)\right]+\left[h(1)x(n-1)+h(2)x(n-2)\right]+h(3)x(n-3) \quad (4.4)$$

Referring to Figure 4.14, Cycles 1–4 compute the bracketed terms in Eq. (4.4). Cycle 4 is special in that the partial beam, BEAM, is added to the output from the center tap, $h(3)x(n-3)$. Cycles 5, 6, and 7 compute the final three summations in Eq. (4.4). By using this scheme, real-time beamforming of a 15 element array in a single ST-50 is feasible for channel digitization rates which adhere to

$$f_s \le \left(\frac{1}{(7\text{ cycles})(80\times10^{-9}\text{ sec per cycle})(15\text{ channels})}\right) \le 119.048\ \frac{\text{kHz}}{\text{channel}} \quad (4.5)$$

Summary

Both the usefulness and performance of any beamforming algorithm depends on the architecture of the processing system. Although the ideal point transducer assumption made by most algorithms is not attainable, the hardware used to condition, filter, and digitize must be selected so as to approach this goal. In particular, the phase match of the hydrophone and analog electronics is a key characteristic in determining suitable hardware components. In addition to maintaining signal integrity, the hardware architecture of the processing system must accommodate both the sampling frequencies and accompanying

sustained data transfer rates characteristic of array processing. In addition, the software architecture must be designed so as to allow real-time processing of these data rates. To attain real-time performance, processors should be dedicated for a specific task — in this case, the interpolation beamforming algorithm — rather than performing tasks that are better handled elsewhere. As seen in this chapter, real-time interpolation beamforming in the time domain is attainable on an ST-50 based platform for a 45-element array whose elements are sampled at a nominal rate of 100,000 samples per second.

CHAPTER 5

A FREQUENCY-DOMAIN APPROACH TO INTERPOLATION BEAMFORMING

In the literature, the beamforming operation is almost always introduced from a time-domain perspective. Although the most intuitive approach, time-domain beamforming is not always the preferred method. A technique that is becoming more feasible for real-time applications with the progress in high-speed computing is frequency-domain beamforming. Frequency-domain beamforming offers some computational advantages over the time-domain approach, particularly when an application calls for simultaneous production of multiple beams. The purpose of this chapter is not to develop the tradeoffs between time-domain beamforming and frequency-domain beamforming since this is thoroughly discussed in many references [13]. Rather, this chapter details the development of a beamformer based FFT. Like the interpolation beamformer described in Chapters 3 and 4, this FFT beamformer executes on an ST-50 array processor with a MicroVAX 3900 host computer. Although the bandwidth of the arrays for which this beamformer was developed is low relative to the multi-aperture array described in Chapter 4, the frequency-domain beamformer is required to generate phase-shift coefficients in real-time based on the output of a target tracker.

FFT Beamforming

Frequency-domain beamforming is a term that describes algorithms that involve the use of the Fourier transform as part of the beamforming process [13]. Perhaps the most widely used algorithm for discrete time applications is the FFT beamformer. A block diagram of a general FFT beamformer is shown in Figure 5.1. As illustrated in Figure 5.1, the time series from each sensor is transformed to the frequency domain by an FFT. The DFT pair is defined as

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\exp\left(\frac{-j2\pi kn}{N}\right) \quad k = 0,1,\ldots N-1$$
$$x(n) = \sum_{k=0}^{N-1} X(k)\exp\left(\frac{j2\pi kn}{N}\right) \qquad n = 0,1,\ldots N-1 \tag{5.1}$$

The radix-2 FFT is simply a computationally efficient version of the DFT subject to the restriction on N given by

$$N = 2^p$$
$$p = \text{ a positive integer} \tag{5.2}$$

Following the production of each sensor's discrete frequency spectrum, the weighted delay-and-sum beamforming operation defined by Eq. (3.4) is accomplished by a shift and sum operation defined by

$$\hat{Y}(k) = \sum_{i=1}^{M} V_i(k) X_i(k) \qquad k = 0,1,\ldots N\text{-}1 \; . \tag{5.3}$$

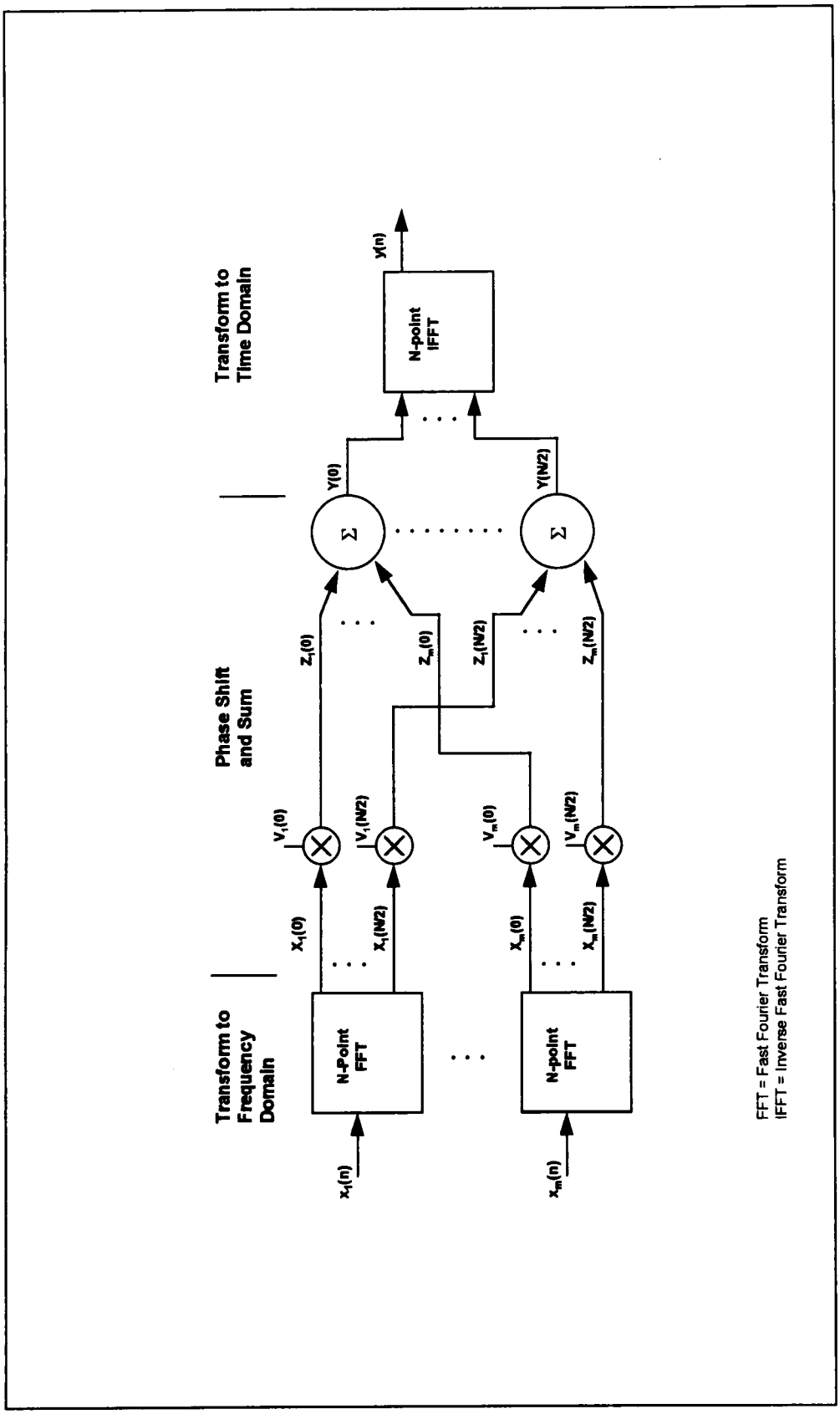

Figure 5.1. Block diagram of Basic FFT Beamformer.

The complex vector, $\mathbf{V_i}$, is expressed in terms of the shading coefficient, w_i, and the time shift, τ_i, for the i-th element as

$$V_i(k) = w_i \exp\left(-j2\pi k \frac{\tau_i}{N}\right). \tag{5.4}$$

The time series from each element is shifted by multiplying the spectrum by a complex vector, $\mathbf{V_i}$. The complex vector's relationship to the desired time shift, τ_i, is defined by the delay property of the DFT given by [12]

$$x_i(n-\tau_i) \leftrightarrow \exp\left(\frac{-j2\pi k\tau_i}{N}\right) X_i(k). \tag{5.5}$$

Having phase shifted each spectrum by the desired number of time samples, the result from the complex vector multiplies are summed on a per-bin basis to obtain an approximation to the beam's discrete frequency spectrum, $Y(k)$. $\hat{Y}(k)$ is then transformed back to the time domain to produce $\hat{y}(n)$.

According to Mucci [13], $\hat{y}(n)$ obtained from Figure 5.1 is an approximation to the discrete-time beam. Although not explicitly stated, Mucci appears to refer to circular convolution effects due to the direct application of Eq. (5.5). The desired result of a true time shift is not produced by applying the delay property of the DFT because [12]

$$x(n-\tau) \neq x(n-\tau) \bmod(N). \tag{5.6}$$

By taking the DFT of the sequence, $x(n)$, an inherent periodicity has been introduced; specifically, the spectrum, $X(k)$, is the Fourier series of a periodic sequence, $\tilde{x}(n)$, which

is related to the original sequence by

$$\tilde{x}(n) = x(n) \bmod(N) . \tag{5.7}$$

Thus, the convolution performed by Eq. (5.5) is circular rather than linear. For instance, circularly shifting a time signal, $x(n)$, by a single sample is accomplished by

$$x(n-1) \leftrightarrow \exp\left(\frac{-j2\pi k}{N}\right) X(k) . \tag{5.8}$$

The result of this operation is illustrated in Figure 5.2. If the circular convolution effects of the DFT are ignored by the beamforming operation defined by Eq. (5.3), significant errors are produced in the discrete-time beam. As illustrated in the 3-element example of Figure 5.3, these errors occur at the ends of the DFT window.

Although the delay property of the DFT corresponds to circular convolution, techniques are available that enable it to perform linear convolution. A well-known technique for performing linear convolution by multiplying two DFTs is the overlap-save method [12]. This technique is applied when the length of one sequence is essentially infinite with respect to the other sequence. In the overlap-save method, the DFT of a filter, $h(n)$, is multiplied by the DFT of a portion of the infinite length input sequence, $x(n)$. The windows over which the DFTs are taken correspond to overlapping sections of the infinite sequence. The amount of overlap required is determined by the length of the filter, $h(n)$. In Eq. (5.5) the multiplication of $X(k)$ by the linear phase term corresponds to convolution of $x(n)$ with a finite length filter having an impulse response defined by

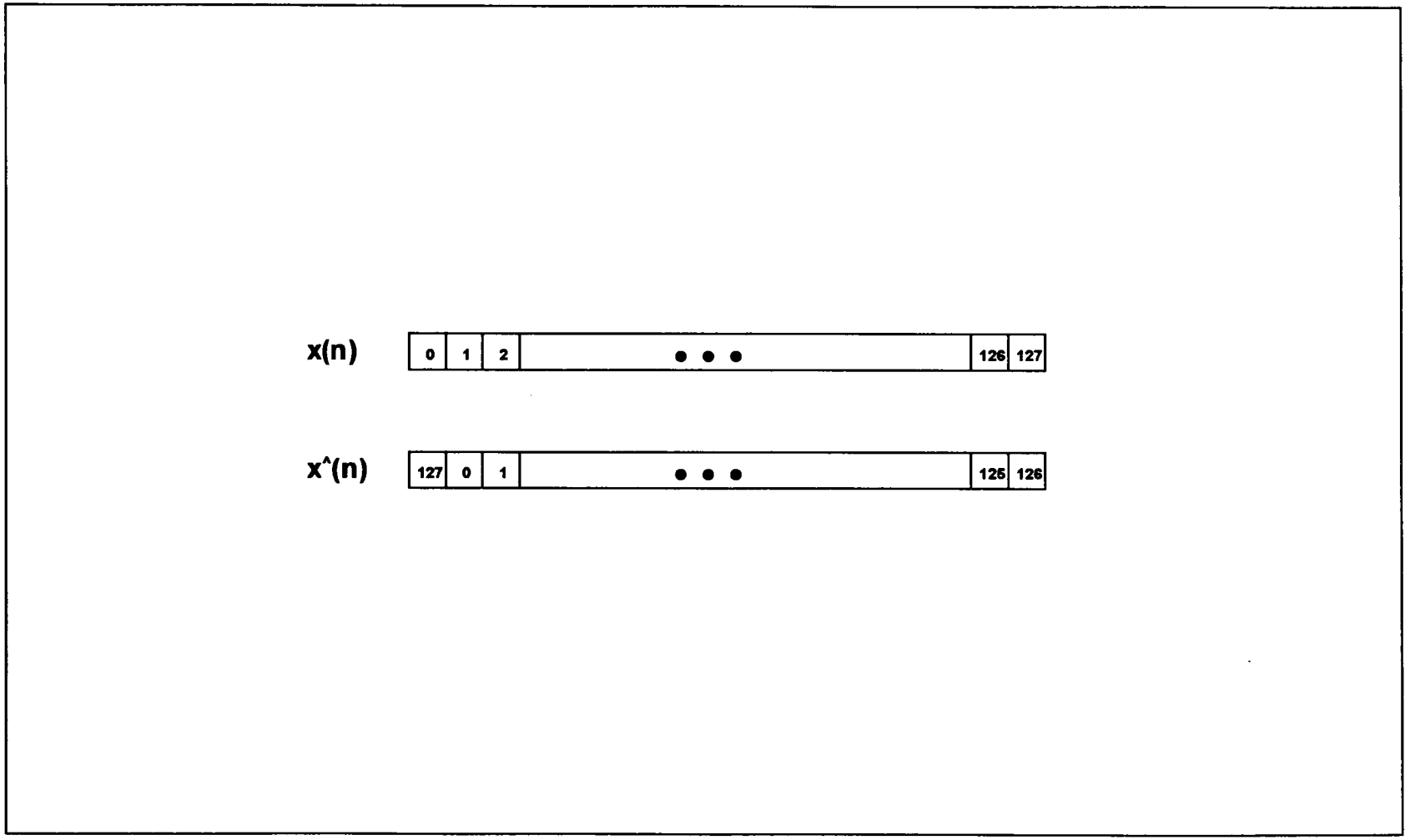

Figure 5.2. Illustration of the circular shift due to delay property of the discrete Fourier transform.

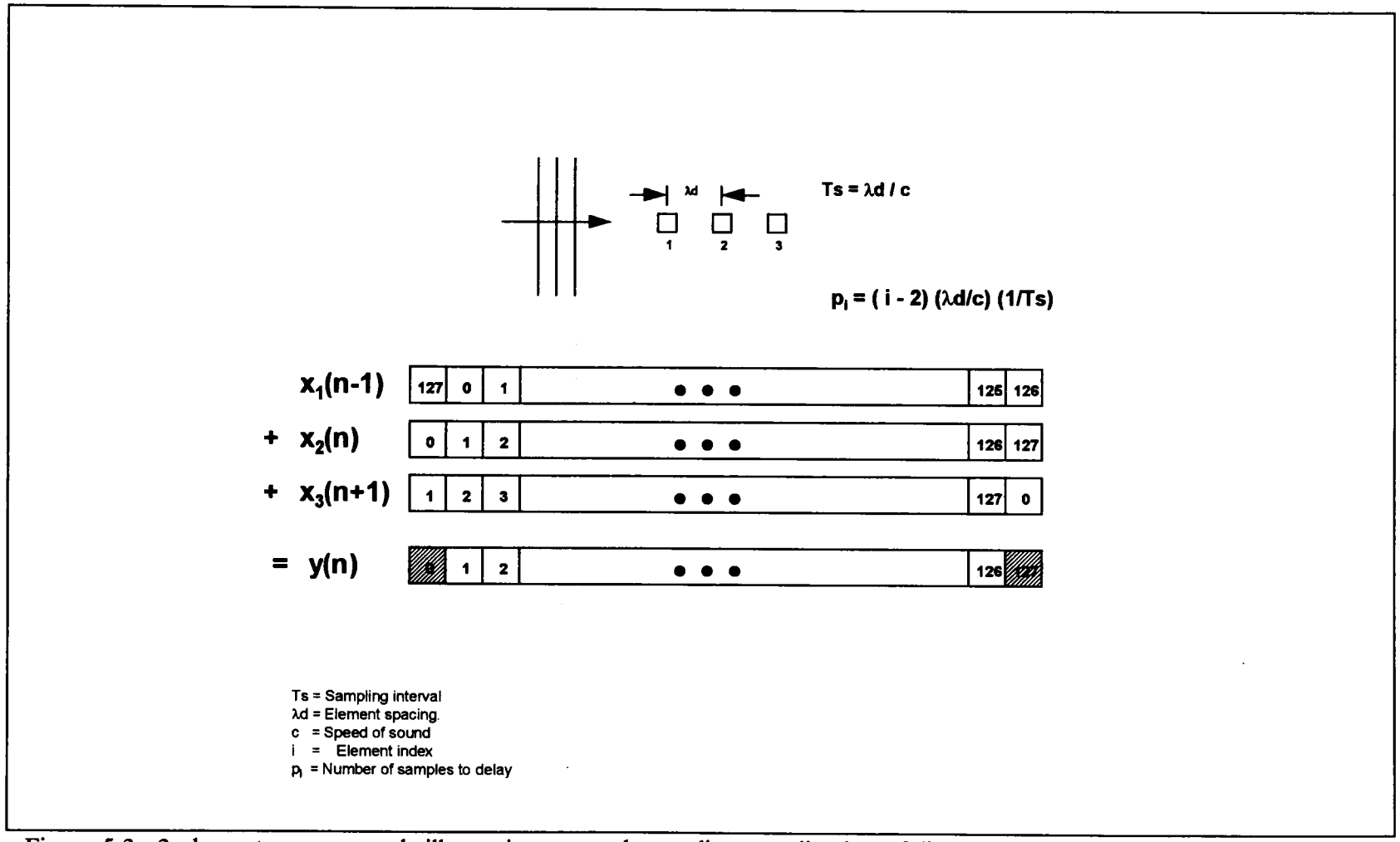

Figure 5.3. 3-element array example illustrating errors due to direct application of discrete Fourier transform's delay property.

$$h(n) = DFT^{-1}\left\{\exp\left(\frac{-j2\pi k\tau_i}{N}\right)\right\} = \delta\left(n-\tau_i\right)_{\bmod(N)} \quad (5.9)$$

Thus, the amount of overlap required to eliminate circular convolution is equal to τ_i. If overlap-save processing is used in conjunction with the FFT beamforming algorithm described by Eq. (5.3), the true beam, $y(n)$, is obtained [2]. The application of overlap-save processing to the 3-element example is shown in Figure 5.4. Because the reference is the middle sensor in the array, the minimum overlap required is two samples. The resulting beam, $y(n)$, is obtained by discarding one sample at each end of the buffer.

Using the delay property of the DFT coupled with overlap-save processing is equivalent to generating the integer time shifts described in Chapter 3. Since delay-and-sum beamforming typically does not provide sufficient beam resolution, a technique must be developed to achieve interpolation beamforming in the frequency domain. This technique requires the capability to effectively shift a signal by a fractional sample.

Fractional Samples

One misunderstanding regarding FFT beamforming is that fractional samples may be produced in the same manner as the integer shifts in the delay-and-sum beamformer. For instance, this assertion erroneously implies that a time series may be shifted by one-half of a sample by simply applying Eq. (5.5) with $\tau_i = 0.5$. This misunderstanding is fueled by lack of information on this subject in much of the literature on frequency-domain beamforming. Some references emphatically state that fractional values of τ_i can be used

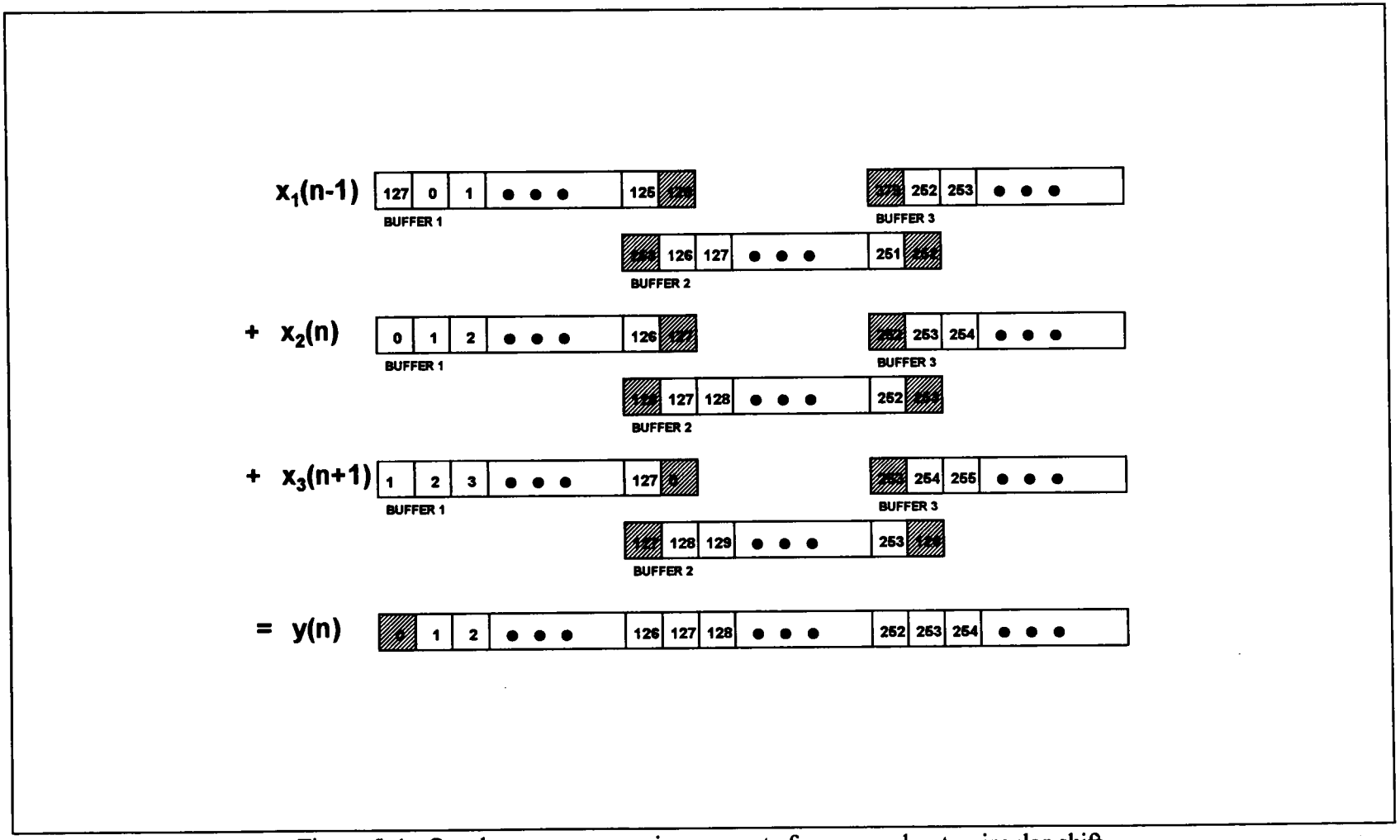

Figure 5.4. Overlap-save processing corrects for errors due to circular shift.

in Eq. (5.3) [4,p. 174], but do not follow up with specific details. In fact, this approach has problems because of the nature of the impulse response of the filter corresponding to a fractional value of τ_i [14]. This section outlines the problems and presents techniques for properly producing fractional time samples.

Fractional Samples and the DFT Delay Property

In conventional beamforming, real-valued signals from an array of sensors are shifted and summed to obtain a real-valued beam. If the delay property of the DFT is used directly to achieve a fractional sample, the shifted signal is complex, and thus, the beam is complex. For a signal to be real, its DFT must be conjugate symmetric as defined by [12]

$$X(k) = X^*(N-k) \qquad 0 \le k \le \frac{N}{2}. \tag{5.10}$$

Thus, a necessary condition for $x(n)$ to be real is

$$\begin{aligned} X\left(\frac{N}{2}\right) &= X^*\left(\frac{N}{2}\right) \\ \mathrm{Im}\left[X\left(\frac{N}{2}\right)\right] &= 0 \end{aligned} \tag{5.11}$$

If an integer value of τ_i is used, the shifted signal is given by

$$X\left(\frac{N}{2}\right)\exp\left[\frac{-j2\pi\left(\frac{N}{2}\right)\tau_i}{N}\right] = X\left(\frac{N}{2}\right)\cos(\pi\tau_i) \tag{5.12}$$

which is real. If a fractional value of τ_i is used, a complex time series will result because conjugate symmetry is violated as seen in Eq. (5.13):

$$X\left(\frac{N}{2}\right)\exp\left[\frac{-j2\pi\left(\frac{N}{2}\right)\tau_i}{N}\right] = X\left(\frac{N}{2}\right)\left[\cos(\pi\tau_i) - j\sin(\pi\ \tau_i)\right] \quad . \tag{5.13}$$

In addition to producing a complex output, a fractional value for τ_i corresponds to an impulse response given by [14]

$$h(n,\tau_i) = \exp\left[\frac{-j\pi(n-\tau_i)}{N}\right]\frac{\sin\left[\pi(n-\tau_i)\right]}{N\sin\left[\frac{\pi(n-\tau_i)}{N}\right]} \tag{5.14}$$

which is also nonzero for all values of n. Thus, circular convolution effects cannot be completely eliminated regardless of the amount of overlap [14]. The effect of this impulse response on the original sequence is illustrated in Figure 5.5. Figure 5.5(a) is the magnitude of the 2048-point DFT of a two-tone signal, $x(n)$, which is quantized by an ideal 15-bit ADC. Figure 5.5(b) is the spectrum that results when the delay property of the DFT is applied with $\tau_i = 0.7$ and an overlap of 20 samples. Note the significant distortion in the magnitude response due to improper application of Eq. (5.5). An error signal is defined as the difference between the fractionally shifted sequence and a theoretical sequence obtained by physically sampling the $x(n)$ such that a shift of $\tau_i = 0.7$ is achieved with physical samples. A plot of this error signal shown in Figure 5.6 illustrates that using the DFT delay property for obtaining fractional samples produces a prominent periodic component which is due to circular convolution effects.

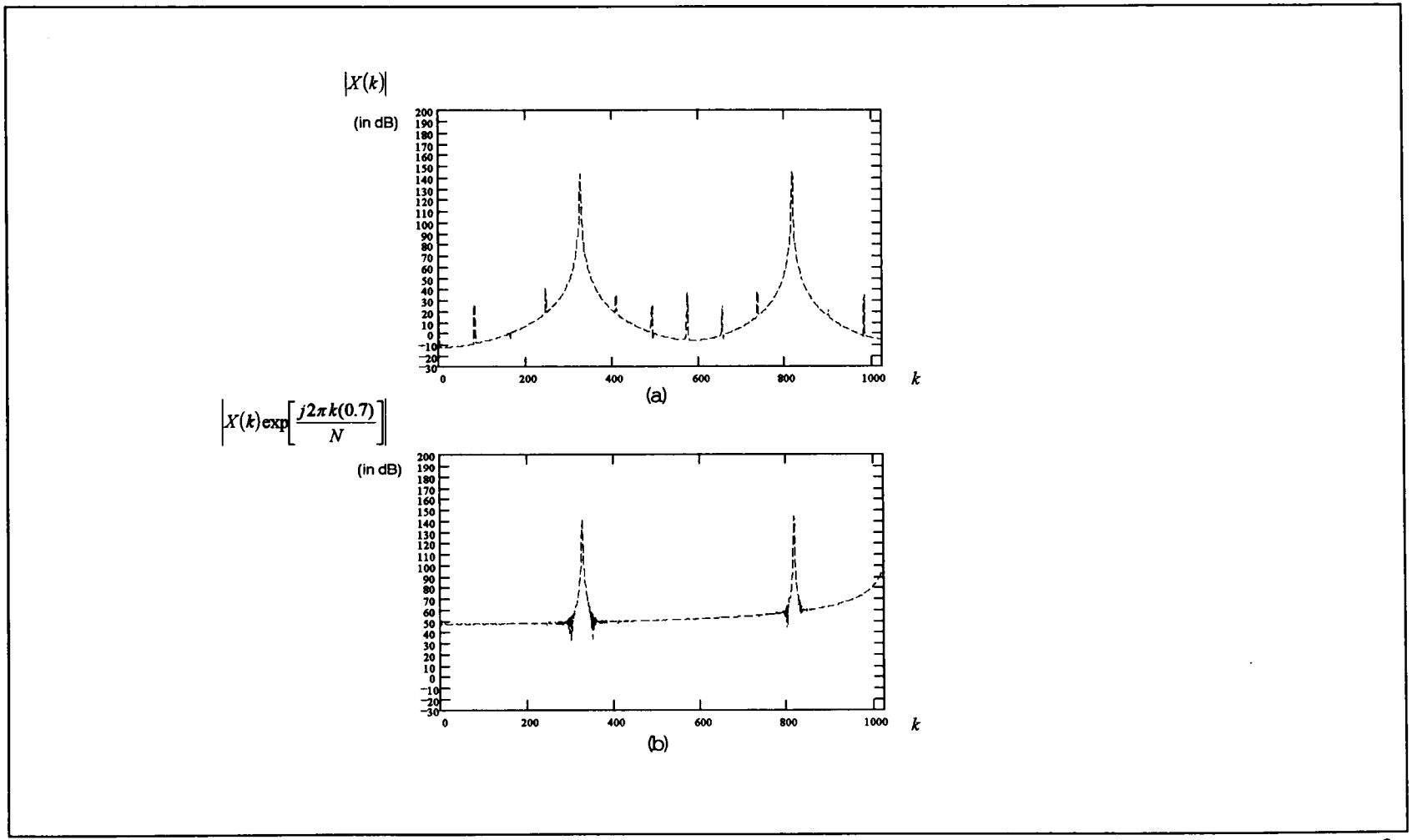

Figure 5.5. Magnitude response of two-tone signal with (a) no time shift; (b) time shift of 0.7 samples using delay property of the discrete Fourier transform.

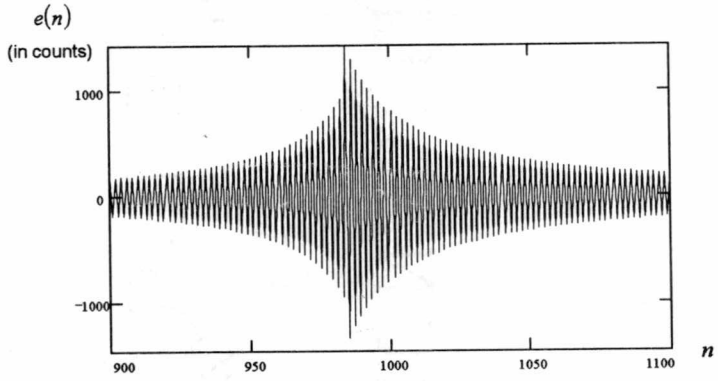

Figure 5.6. Error signal resulting from applying delay property of the discrete Fourier transform with Fractional τ.

Fractional Samples Using FIR Filters

Interpolation to produce fractional samples can be properly performed by using FIR filters. Because they are finite length by definition, FIR filters eliminate the effects of circular convolution when used with overlap-save processing. A well-known technique for designing FIR filters is the windowing method [15]. The object of the windowing method of FIR filter design is to approximate the ideal delay filter with a finite number of points. This approximation is accomplished by multiplying the ideal response by a window function as shown in Eq. (5.15):

$$h_w(n,\tau_i) = w(n,\tau_i)h(n,\tau_i) \quad . \tag{5.15}$$

The Hamming window is often used for this technique because it exhibits a good combination of transition bandwidth and maximum stopband ripple [15] while being relatively simple to compute. The Hamming window is given by [14]

$$w(n,\tau_i) = \begin{cases} 0.54 + 0.46\cos\left(\dfrac{\pi(n-\tau_i)}{L}\right) & -L+\tau_i \le n \le L+\tau_i \\ 0 & \text{otherwise} \end{cases} \quad . \tag{5.16}$$

The impulse response for the ideal filter which generates fractional samples is given by the inverse Fourier transform

$$\begin{aligned} h(n,\tau_i) &= \int_{-\pi}^{\pi} \exp(-j\omega\tau_i)\exp(-j\omega n)d\omega \\ &= \frac{\sin[\pi(n-\tau_i)]}{\pi(n-\tau_i)} \end{aligned} \quad . \tag{5.17}$$

For this application, a 21-tap FIR filter designed with this technique provides a good approximation to the ideal allpass delay filter [14] over the passband of interest. Using this FIR filter to shift the two-tone signal by $\tau_i = 0.7$ yields a result that more closely matches the original signal as shown in Figure 5.7(a). The error signal that results is shown in Figure 5.7(b). Although the error signal does not become zero, the periodic component due to circular convolution is no longer present.

Besides the windowing technique, a polyphase FIR filter structure can be used to obtain fractional samples by performing the identical interpolation operation in the frequency domain. The procedure is similar to that outlined in Chapter 3. A prototype filter is designed based on an upsampling factor, L. This filter is then implemented as a counterclockwise commutator. Having generated the commutator structure, the DFT of each polyphase filter is generated. To maintain consistency in the overlap-save buffer, each DFT is multiplied by a linear-phase term to shift the impulse response so that the first and last M points of a buffer are discarded as shown in Eq. (5.18):

$$V_i(k) = a_i \, P_i(k) \exp\left(\frac{j\pi kM}{LN}\right), \tag{5.18}$$

where M is the order of the prototype filter, and $P_i(k)$ is the DFT of the polyphase filter. Unlike the windowing technique which can be used for any value of τ_i [14], the polyphase approach is limited to quantized values of τ_i as described in Chapter 3.

In terms of Eq. (5.3), using FIR filters to generate the fractional time shifts instead of improperly applying the DFT delay property requires no additional computation because only the components of the time shift vector, **V**, are different between the techniques. For a given throughput, however, slightly more computational capacity is

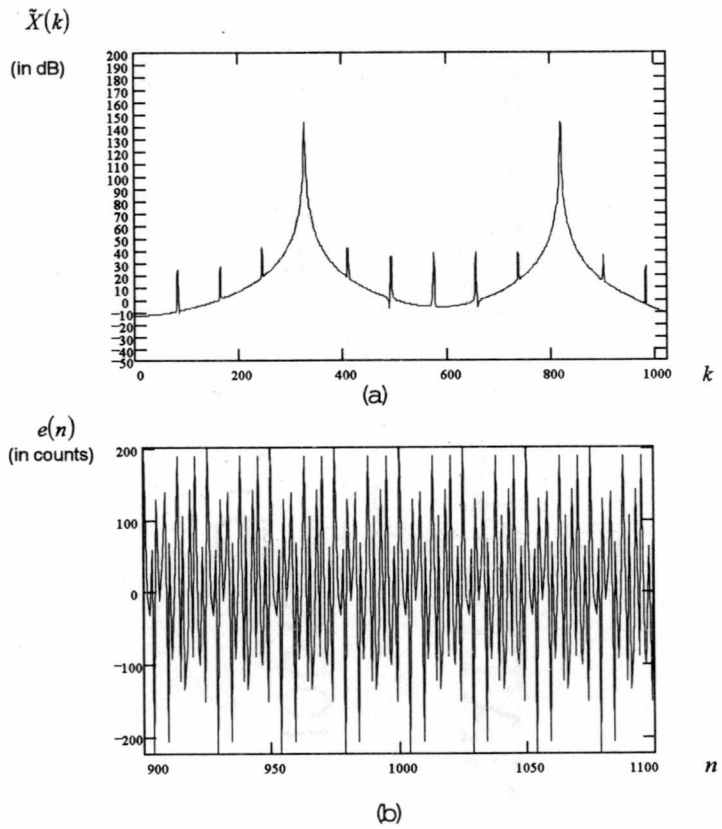

Figure 5.7. (a) Magnitude of two-tone signal shifted by 0.7 samples using finite impulse response (FIR) filter technique; (b) error signal produced by FIR filter technique.

needed for the FIR filter technique because of the additional overlap required to account for the length of the filter.

Real-time Implementation of an FFT Beamformer

While describing the FFT beamforming operation and the technique for obtaining fractional samples, the discussion thus far does not address the issues involved in the actual generation of **V**. The method of generating **V** is determined by system requirements and computational capacity. For instance, while the polyphase filter approach is valid and in some ways superior to the windowing approach [14], the computation of the prototype filter in real-time to achieve arbitrary τ_i is impractical for real-time processing applications; however, if quantized values of τ_i meet system goals, then a single prototype filter can be used as in the case of the interpolation beamformer described in Chapter 4. Time shift generation for the application in Chapter 4 is further simplified by the fact that beamforming is done along a track that is known *a priori*. Thus, the time shifts can be computed and stored prior to initiating real-time data processing.

The application outlined at the beginning of this chapter requires that **V** be updated in real-time. The beamformer for this application executes on a single ST-50 array processor controlled by a MicroVAX 3900 host computer. The tasks that must be performed are shown in Figure 5.8. The time-frequency and frequency-time transformations are based on an FFT routine developed by STAR Technologies, Inc. With the proper use of the cache memory, the time required for these operations may be predicted by multiplying the time required for a single FFT by the number of sensors to beamform. When choosing the FFT size, several factors must be considered. For

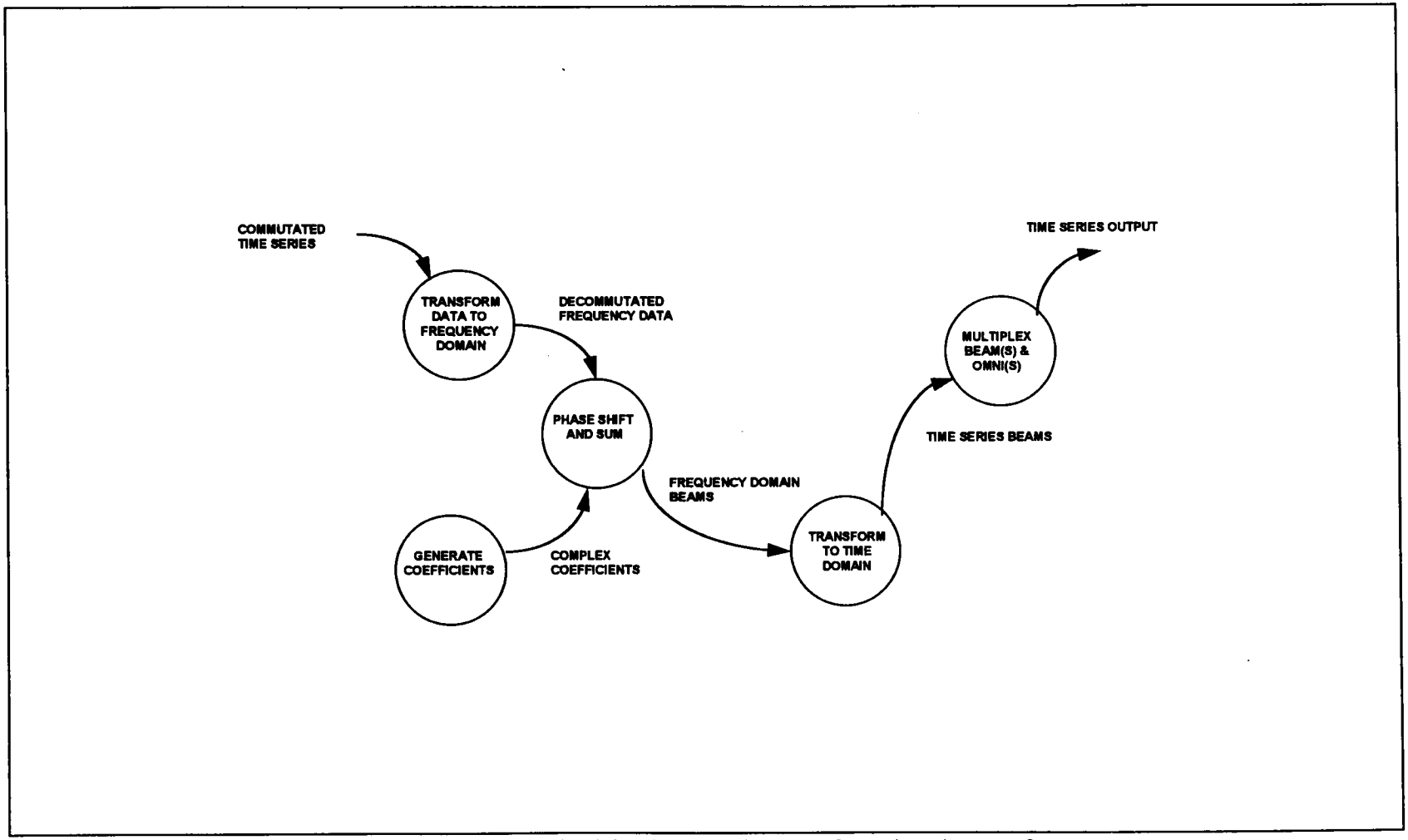

Figure 5.8. Tasks required for fast Fourier transform (FFT) Beamformer.

instance, N must be large enough to accommodate the maximum delay as defined by Eq. (3.22). Practically, the number of valid samples of the beam should be greater than or equal to the amount of overlap. Preferably, the amount of overlap should be minimized so as to obtain high throughput. This small percentage of overlap is achieved with a large FFT size; however, the FFT size is also a function of the required update rate of the steering angle. This is because the steering angle may only be updated on FFT boundaries. Thus, if too large an FFT size is used, the beam may not update its steering angle quickly enough. A 2048-point FFT is chosen for the development which follows.

From a system design perspective, the execution time required for the FFT operations is essentially optimized. Thus, any computational performance improvements needed to meet real-time processing goals must be obtained in the coefficient generation and beamforming operations. Execution time requirements on the beamforming operation, i.e., shift and sum, and the coefficient generation are based on the sampling frequency, overlap, FFT size, and number of sensors. Generation of $\mathbf{V}$ is based on a far-field model for τ as given by

$$\tau_i = \frac{\left[(\mathbf{r}_i \cdot \mathbf{u}) - (\mathbf{r}_{ref} \cdot \mathbf{u})\right]}{c}, \tag{5.19}$$

where $\mathbf{r}_i$ is the position of the i-th sensor, $\mathbf{u}$ is the steering direction, and $\mathbf{r}_{ref}$ is the position of the reference sensor. An ST-50 routine for generating $\mathbf{V}$ using the DFT delay property exceeds the allotted time for coefficient generation by 60 milliseconds. The inefficiency of this routine is due primarily to the 9 cycles per sample required by the ST-50 to generate a trigonometric function. Since it requires the same type of trigonometric function generation, the windowing technique does not offer an improvement on this

execution time. The polyphase filter generation is not feasible for this application due to the large amount of computation required for equiripple delay filters [14].

Coefficient generation for arbitrary values of τ does not satisfy real-time processing requirements; however, performance goals for the beamformer are realizable with an upsampling factor, L, of 60. To achieve this factor in real-time, a combination of storage and computation based on the concept of integer and fractional shifts developed in Chapter 3 is used to generate **V**. A diagram illustrating this algorithm is shown in Figure 5.9. In this scheme a time shift for the i-th sensor, τ_i, is computed using Eq. (5.19). This time shift is then partitioned into an integer and fractional component according to

$$\begin{aligned} \text{IDLY} &= \lfloor \tau_i f_s + N \rfloor \\ \text{FDLY} &= \tau_i f_s - (IDLY - N) \end{aligned} \quad . \tag{5.20}$$

The integer and fractional components correspond to coarse and fine shifts, respectively. These shifts are used to compute pointers into the stores containing the integer and fractional delay filters. The FRACTIONAL DELAY FILTERS store shown in Figure 5.9 contains the DFT representation of the FIR filters which implement the fractional delays:

$$\tau_\rho = \frac{\rho}{L} \quad \rho = 0,1,2 \ldots L-1 \, . \tag{5.21}$$

By storing these filters contiguously, the pointer to the required filter's starting location is computed from the fine delay according to

$$FPTR = \lfloor L|FDLY| + 0.5 \rfloor (N+2) + 1 \, , \tag{5.22}$$

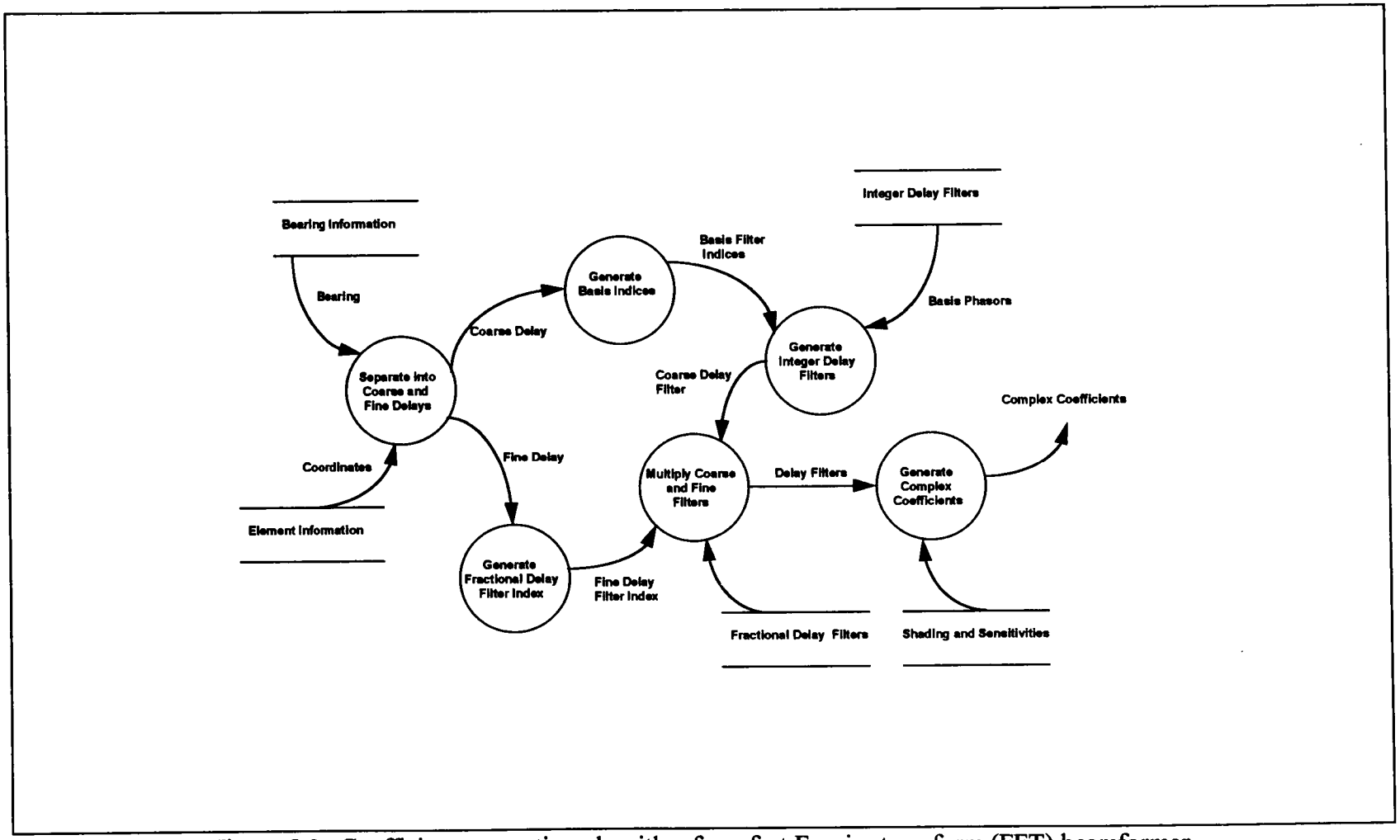

Figure 5.9. Coefficient generation algorithm for a fast Fourier transform (FFT) beamformer.

where N is the size of the FFT.

The technique for obtaining integer delay filters is similar to that used for the fractional delay filters; however, because the ST-50 is limited to 4 MWords of memory, all possible integer delay filters cannot be stored for an FFT size of 2K. A solution to this limitation is to establish a set of integer delay filters from which all possible integer delays may be obtained by a single complex vector multiplication. For a 2K FFT, one set of basis integer delay filters from which all integer delay filters may be realized is given by

$$H(k)_p = \exp\left(\frac{j2\pi k\, p}{N}\right) \quad . \qquad (5.23)$$
$$p = 0,1,2\ldots48,96,144,\ldots2016$$

For instance, an integer delay of 382 samples is achieved by multiplying the appropriate basis vectors as shown in Eq. (5.24):

$$H(k)_{382} = H(k)_{46} H(k)_{336} \quad . \qquad (5.24)$$

Note that the circular shift characteristic of the DFT's delay property may be used for generating advances from this basis set. For instance, an integer delay of -1 is obtained by using Eq. (5.25):

$$H(k)_{-1} = H(k)_{2047} = H(k)_{31} H(k)_{2016} \quad . \qquad (5.25)$$

Assuming contiguous storage of the basis integer delay filters, the appropriate pair of basis filters are obtained by generating a pointer into the INTEGER DELAY FILTER STORE

$$INPTR = 2(IDLY \oplus N) + 1 \,. \tag{5.26}$$

Having obtained the delay filters for both integer and fractional shifts, the two filters are multiplied.

Summary

By using the delay property of the DFT to achieve fractional time delays, circular convolution errors are introduced in the beam. By applying the concept of integer and fractional samples developed for the polyphase filter-based algorithm in Chapter 3, fractional samples can be properly obtained for an FFT beamformer without the circular convolution errors. The fractional samples are produced by using FIR delay filters which are computed and stored prior to starting the beamformer. For the ST-50 array processor, an efficient means of producing integer delays involves storing a base set of integer delays. From this set, all possible integer delays for given FFT size are obtainable with a single complex vector multiplication.

CHAPTER 6

RESULTS

Testing of the interpolation beamforming algorithms is accomplished by using simulated hydrophone array data to generate beam power patterns. Specifically, the data represent a single sinusoid, swept from one end of the array to the other, as it appears to each element in the array. Once generated, the data are downloaded into the ST-50 main memory arrays which are used to buffer the real-time data during normal operation. The beamformer is steered to a single direction while processing the swept source. For each source location, the energy in the beam is computed using Parseval's relation [15]

$$Energy = \sum_{n=0}^{N} b^2(n) \ . \tag{6.1}$$

Having computed the energy for each source location, a normalized beam power pattern is produced. Figures 6.1 – 6.8 show beam power patterns obtained for a 15-element line array using the interpolation beamformer described in Chapter 4. Note that for the near-field patterns of Figures 6.6 – 6.8, the source radius is held constant. Figures 6.9 – 6.11 demonstrate beam power patterns obtained for a 50-element line array using the FFT beamformer of Chapter 5.

These beam power patterns are used to validate that the algorithms are correctly implemented; however, other testing is required to guarantee correct operation of the beamformer. These tests include spectral analysis of the time-domain beam to determine

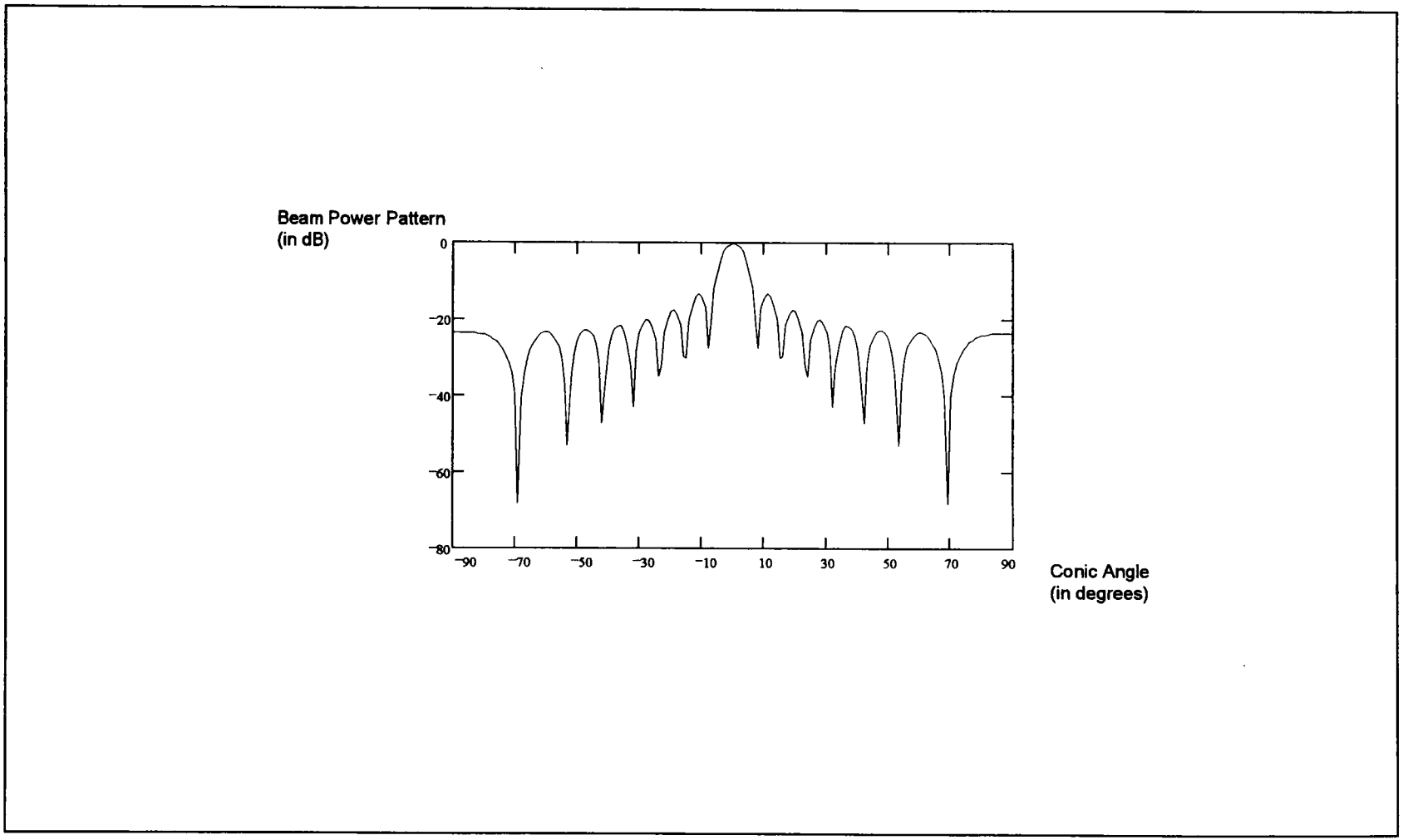

Figure 6.1. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°.

(c = Speed of sound in water)

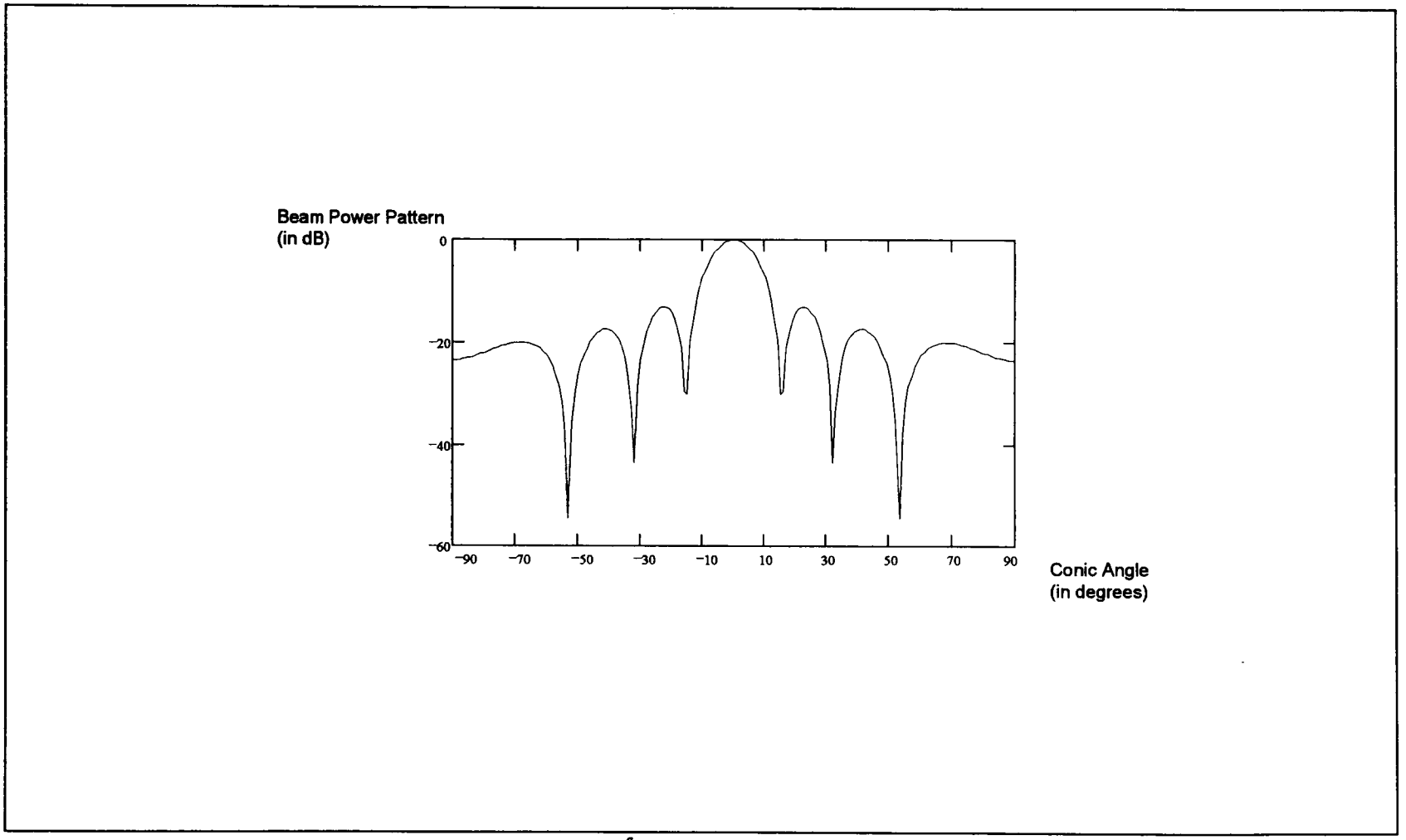

Figure 6.2. Far-field pattern: source frequency = $\frac{f}{2}$; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°.
(c = Speed of sound in water)

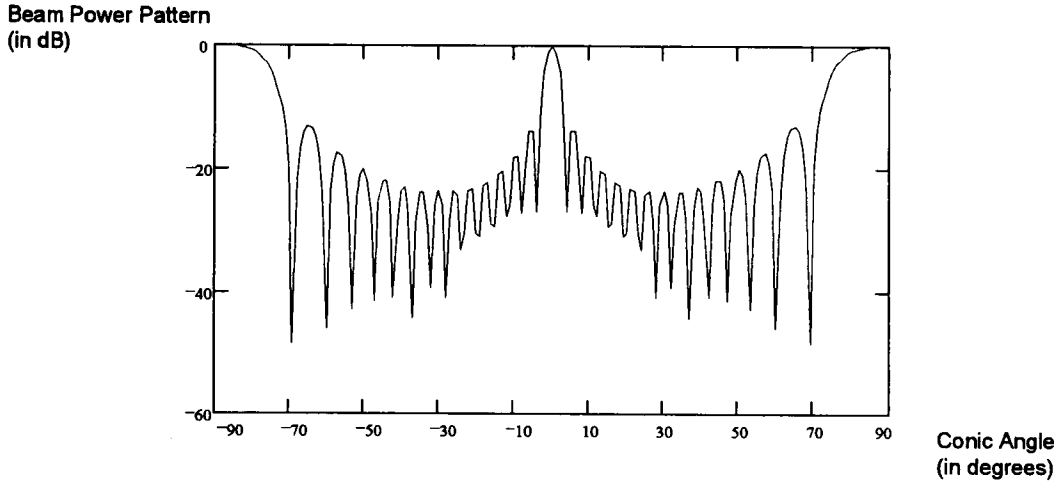

Figure 6.3. Far-field pattern: source frequency = $2f$; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°.

(c = Speed of sound in water)

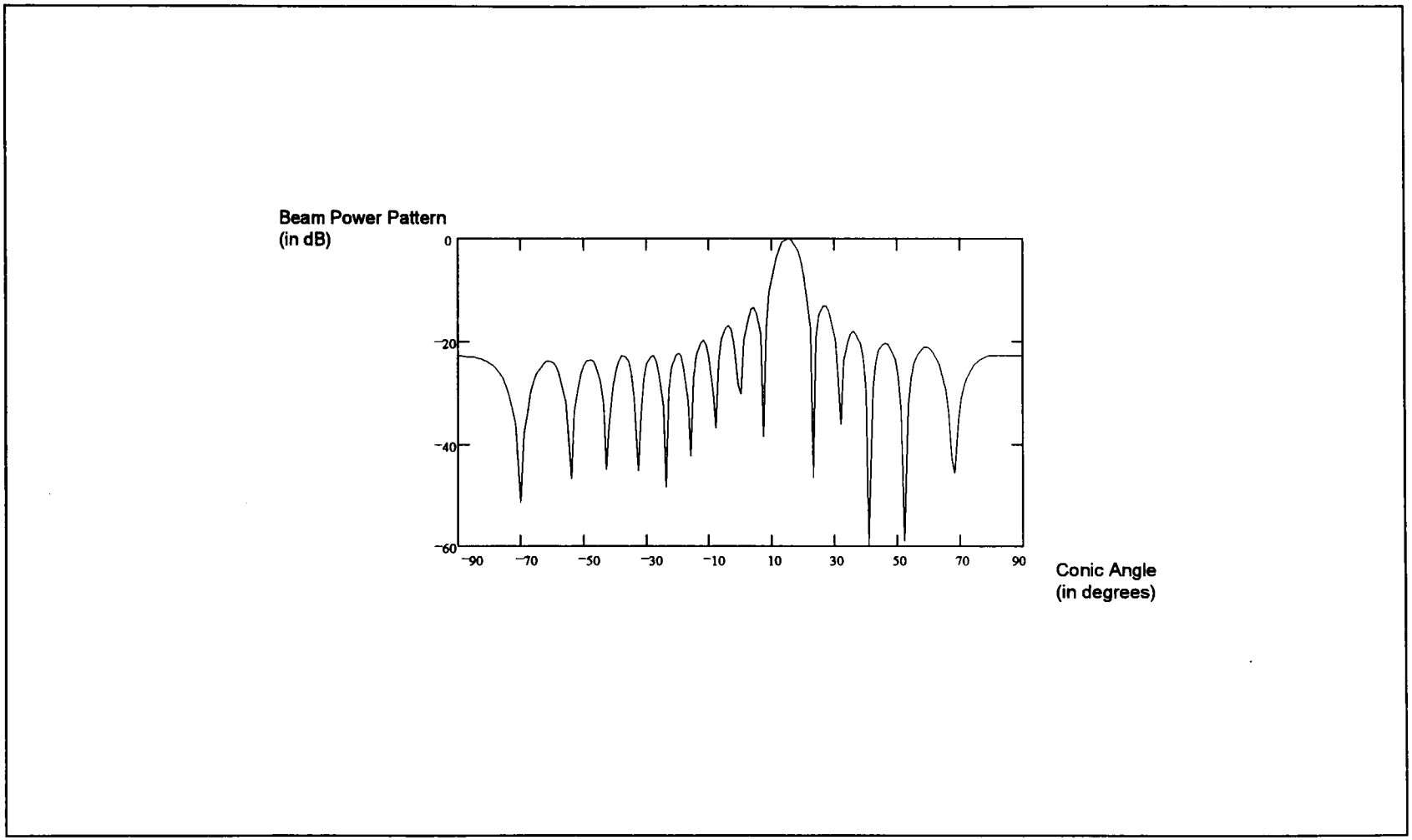

Figure 6.4. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 15.0°.

(c = Speed of sound in water)

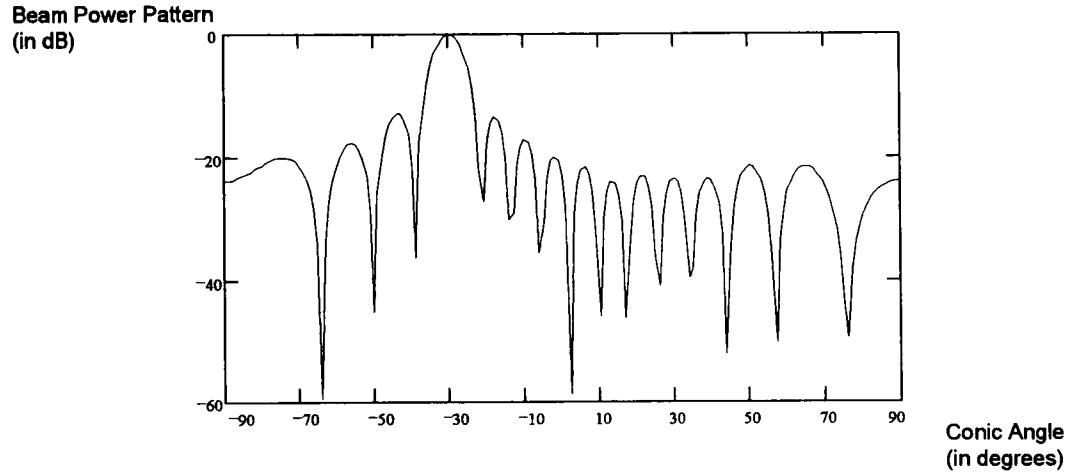

Figure 6.5. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = –30.0°.

(c = Speed of sound in water)

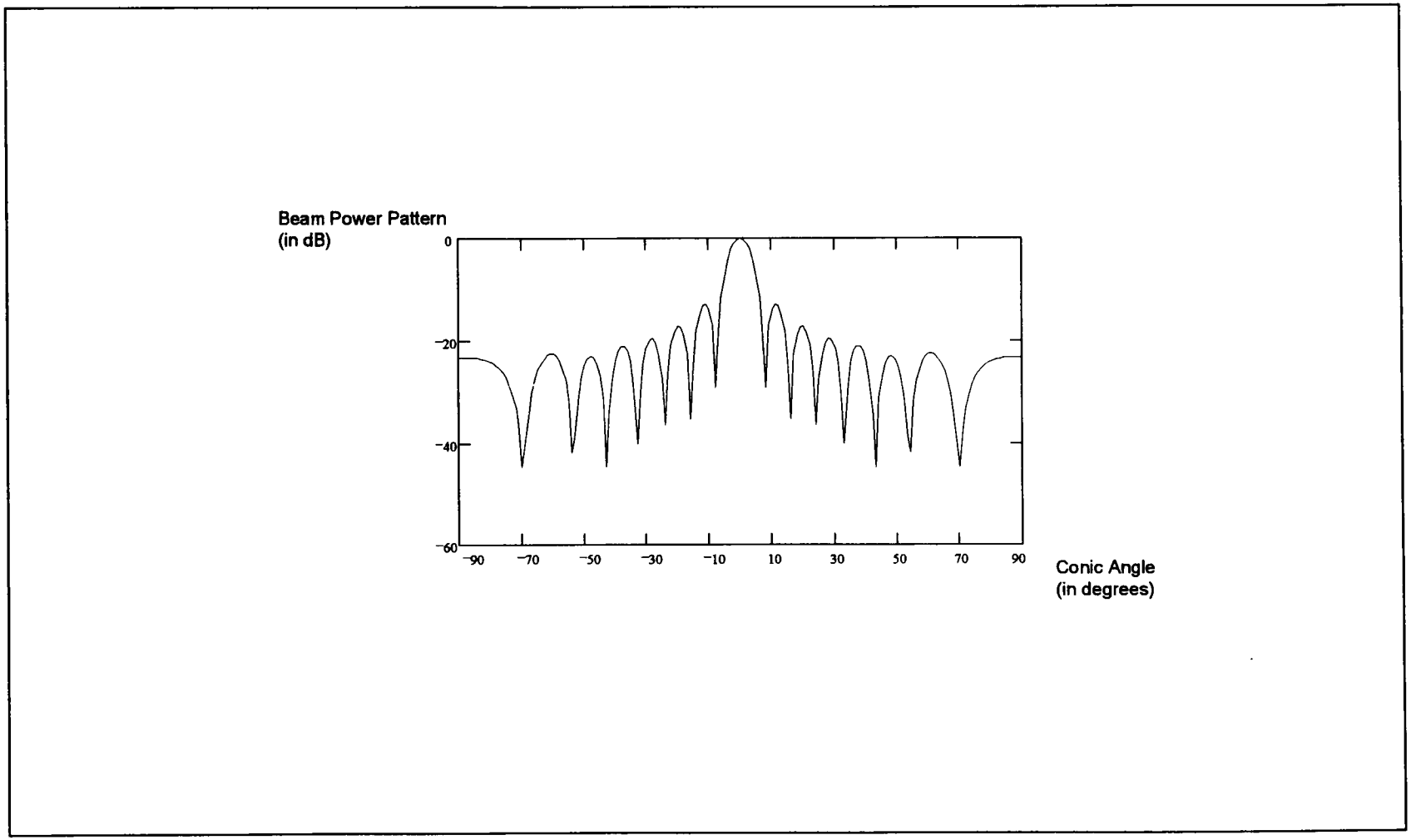

Figure 6.6. Near-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; focal point = 1.52 m at 0.0°.

(c = Speed of sound in water)

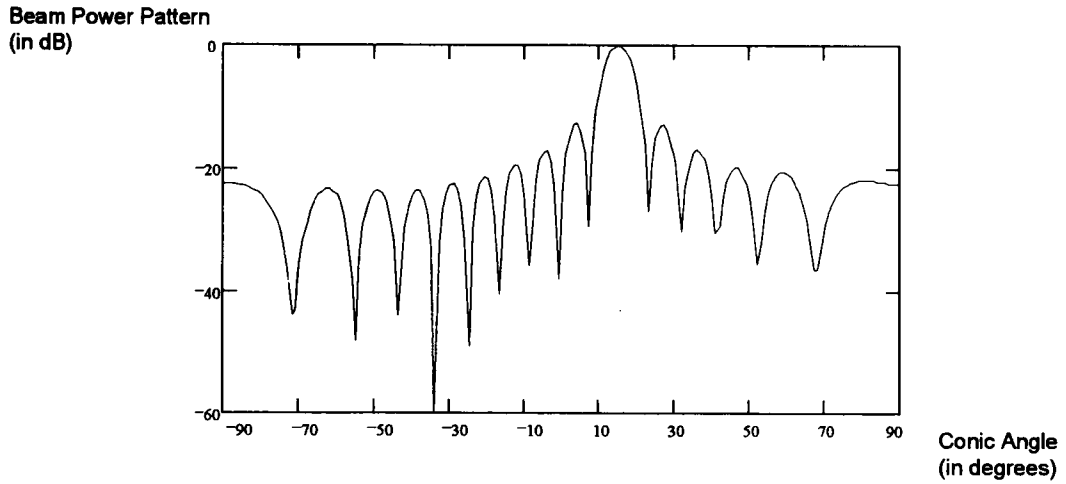

Figure 6.7. Near-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; focal point = 1.52 m at 15.0°.
(c = Speed of sound in water)

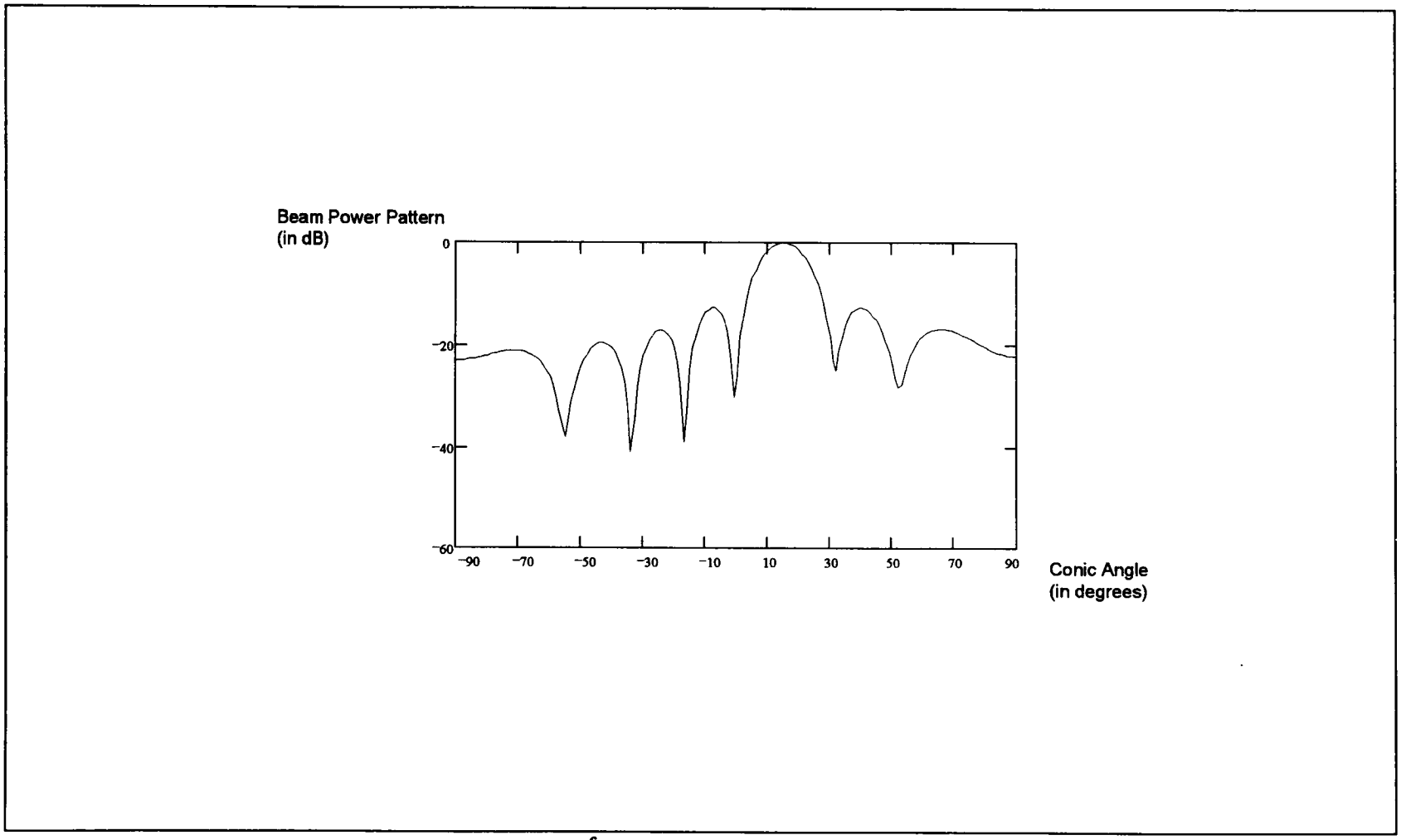

Figure 6.8. Near-field pattern: source frequency = $\frac{f}{2}$; element spacing = $\frac{c}{2f}$; uniform shading; focal point = 1.52 m at 15.0°.
(c = Speed of sound in water)

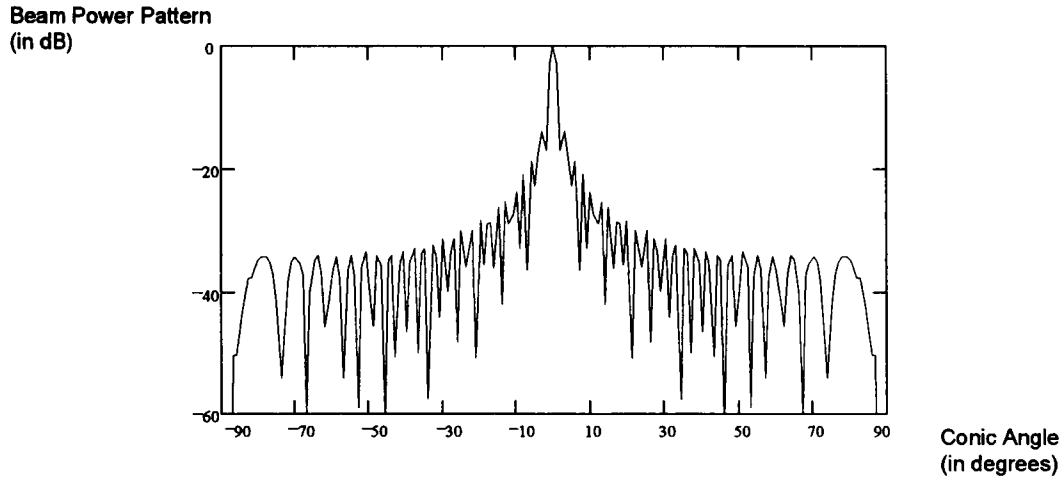

Figure 6.9. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = 0.0°.

(c = Speed of sound in water)

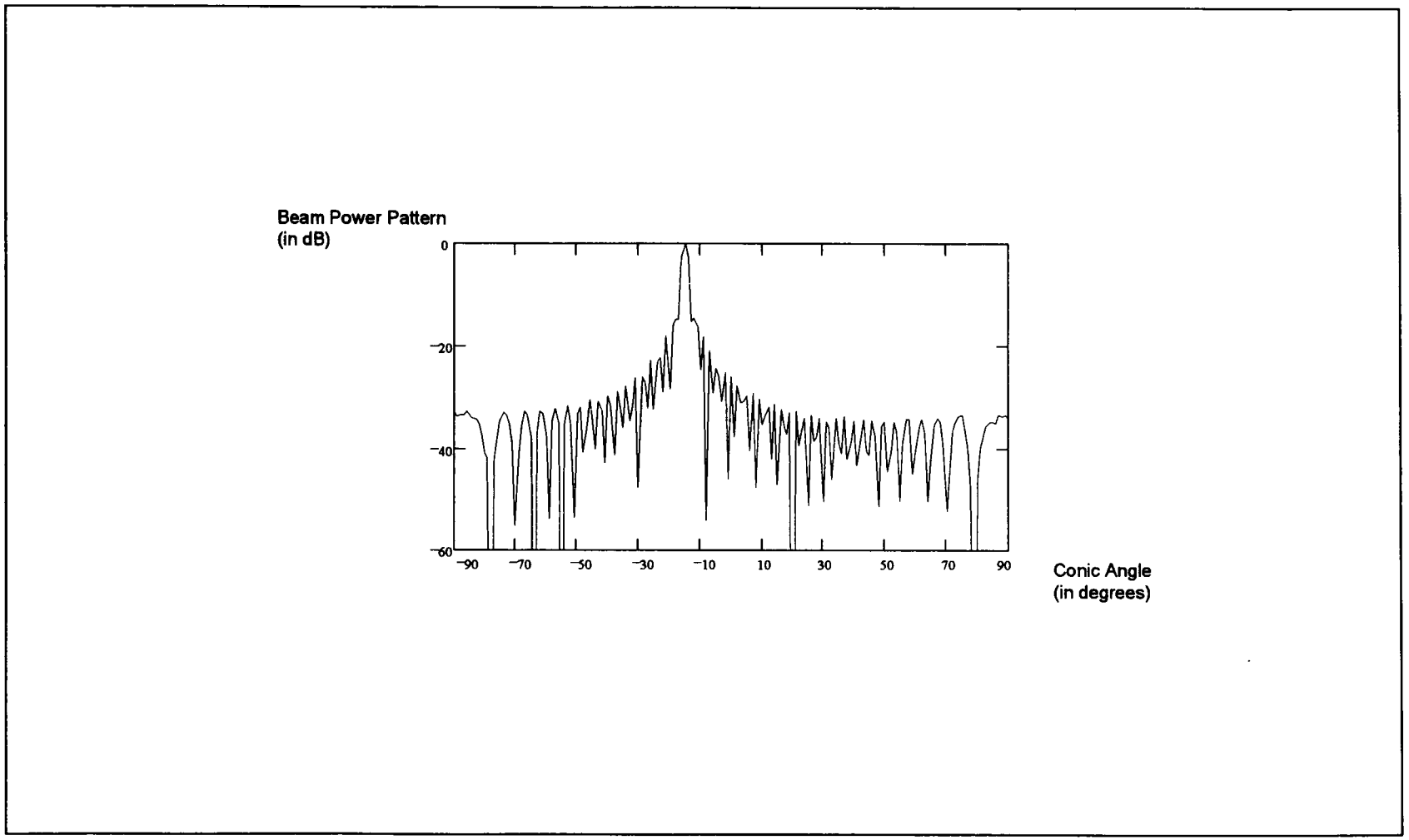

Figure 6.10. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; uniform shading; steering angle = −15.0°.

(c = Speed of sound in water)

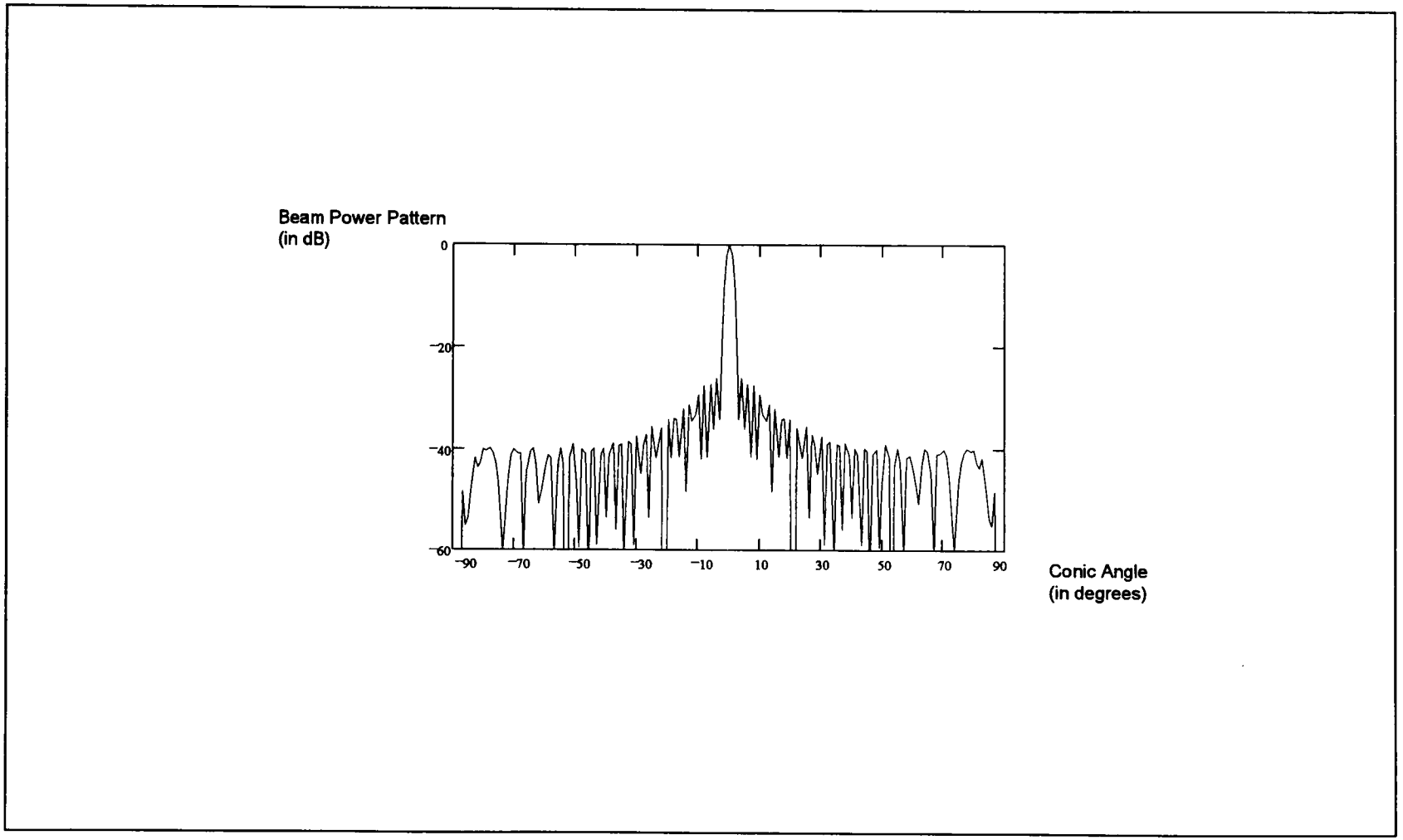

Figure 6.11. Far-field pattern: source frequency = f; element spacing = $\frac{c}{2f}$; Taylor shading; steering angle = 0.0°.

(c = Speed of sound in water)

that the overlap processing and buffer design is functioning properly. By taking the FFT of the time-domain beam, discontinuities caused by an improper overlap factor or software executing too slowly are exposed. The interpolation beamformer of Chapter 4 has been thoroughly tested in this respect and, as of this writing, is fully operational at the test facility.

CHAPTER 7

CONCLUSIONS

Beamforming is a spatial filtering operation that improves the performance of acoustic measurement systems by increasing the SNR. In modern systems, the signals from each hydrophone in the array are digitized, thus enabling the use of digital beamforming algorithms. A disadvantage to most digital beamforming algorithms is the limitation on steering direction due to the fact that achievable time shifts are quantized. Although increasing the sampling frequency used to digitize the hydrophone array provides more achievable time shifts, this approach is not practical because a cost-effective solution using current ADC technology cannot achieve the sampling frequencies that meet the beam steering requirements of most applications. Even if a solution could be found, the data management and interface requirements would make digital beamforming feasible only for systems with very large computational capacity.

As a result, a multirate digital signal processing technique known as interpolation is used to effectively increase the frequency at which a signal is sampled without changing the physical sampling rate of the ADCs. If an application calls for the production of a single beam, a counterclockwise commutator model of an interpolator using polyphase filters provides an efficient implementation because only the interpolated sample needed for shifting a sensor is required. Thus, significant computational savings are realized by filtering the signal with a polyphase filter rather than the higher order prototype filter.

The polyphase filter-based beamformer is actually a subset of an interpolation beamforming algorithm that relies on coarse and fine time shifts. This scheme is useful in

both time domain and frequency domain applications. Furthermore, polyphase filters can be replaced by other FIR delay filters such as those designed using the windowing technique. This algorithm enables real-time processing because filters are generated and stored prior to execution of the beamformer.

Finally, an efficient implementation of a real-time digital beamformer enables its application to real-world problems. Perhaps the most obvious of these applications is the implementation of a focused beamformer for a test facility in which conventional far-field techniques do not apply.

LIST OF REFERENCES

LIST OF REFERENCES

1. Burdic, William S. *Underwater Acoustic System Analysis.* Englewood Cliffs, NJ: Prentice Hall, 1984.
2. Neilson, Richard O. *Sonar Signal Processing.* Boston: Artech House, 1991.
3. Clay, Clarence S. and Herman Medwin. *Acoustical Oceanography: Principles and Applications.* New York: John Wiley & Sons, 1977.
4. Johnson, Don H. and Dan E. Dudgeon. *Array Signal Processing: Concepts and Techniques.* Englewood Cliffs, NJ: Prentice Hall, 1993.
5. Sklar, Bernard. *Digital Communications: Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice Hall, 1988.
6. Neff, Herbert P. *Basic Electromagnetic Fields.* New York: Harper & Row, 1981.
7. Dudgeon, Dan E. and Russell M. Mersereau. *Multidimensional Digital Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1984.
8. Harris, Fredric J. "Windows, Harmonic Analysis, and the Discrete Fourier Transform," Naval Undersea Center Report, NUC/TP-532, 1976.
9. Taylor, T. T. "Design of Line-Source Antennas for Narrow Beamwidth and Low Sidelobes," IRE Trans., Vol. AP-3, (1955).
10. Pridam, Roger G. and Ronald A. Mucci. "A Novel Approach to Digital Beamforming," J. Acoust. Soc. Am. , Vol. 63,(1978).
11. Crochiere Ronald E. and Lawerence R. Rabiner. *Multirate Digital Signal Processing.* Englewood CLiffs, NJ: Prentice Hall, 1983.
12. Oppenheim Alan V. and Ronald W. Schafer. *Discrete Time Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1989.

13. Mucci, Ronald A. "A Comparison of Efficient Beamforming Algorithms", IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-32, (1984).

14. Bomar, Bruce W. and L. Montogomery Smith. "Frequency-Domain Implementation of Fractional Sample Delays for Beamforming," University of Tennessee Space Institute Report, 1991.

15. Jackson, Leland B. *Digital Filters and Signal Processing.* Boston: Kluwer Academic Publishers, 1989.

VITA

John Eric Breeding was born on September 13, 1965, in Knoxville, Tennessee. Upon graduation from Central High School (Knoxville) in 1983, he entered The University of Tennessee, Knoxville. While attending the university, he received a Chancellor's Citation for Extraordinary Academic Achievement and was initiated into Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi. In August 1987, he received a Bachelor of Science degree in Electrical Engineering and was named top graduate in the College of Engineering.

From 1987 to 1989, he was employed as a hardware design engineer for Martin Marietta Electronic Systems in Orlando, Florida. In 1989, he joined Oak Ridge National Laboratory in Oak Ridge, Tennessee as a development engineer in the Real-Time Computer Systems group of the Instrumentation and Controls Division. His areas of concentration include digital signal processing, real-time systems development, and high-speed interface design.