

CONF-9510352--

**A METHOD FOR EFFICIENT FRACTIONAL SAMPLE DELAY GENERATION
FOR REAL-TIME FREQUENCY-DOMAIN BEAMFORMERS**

J. Eric Breeding
Thomas P. Karnowski
Oak Ridge National Laboratory*

RECEIVED
OCT 08 1997
OSTI

Paper to be presented at the International
Conference on Signal Processing
Applications and Technology
Boston, Mass.
October 24-26, 1995

MASTER

"The submitted manuscript has been
authored by a contractor of the U.S.
Government under contract No. DE-
AC05-84OR21400. Accordingly, the U.S.
Government retains a nonexclusive,
royalty-free license to publish or reproduce
the published form of this contribution, or
allow others to do so, for U.S. Government
purposes."

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED *ph*

19980407 073

* Managed by Lockheed Martin Energy Systems, Inc. for the U.S. Department of Energy
under contract DE-AC05-84OR21400.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

A METHOD FOR EFFICIENT FRACTIONAL SAMPLE DELAY GENERATION FOR REAL-TIME FREQUENCY-DOMAIN BEAMFORMERS

J. Eric Breeding and Thomas P. Karnowski
Oak Ridge National Laboratory*
P.O. Box 2008, Oak Ridge, Tennessee 37831-6007

Abstract

This paper presents an efficient method for fractional delay filter generation for frequency-domain beamformers. A common misunderstanding regarding frequency-domain beamforming is that any fractional time shift can be achieved using the delay property of the discrete Fourier transform (DFT). Blind application of the DFT delay property introduces circular convolution errors that may adversely affect the beam's time series. The method presented avoids these errors while enabling real-time processing.

I. Background

The beam pattern is the magnitude response of a spatial filter produced by summing the output signals from all sensors in an array. This filter response is characterized by a main lobe and several sidelobes. The main lobe is often referred to as the passband or maximum response axis (MRA) of the filter response [1]. Beamforming may be defined as the process of changing the MRA of the array response using the weighted delay-and-sum beamformer equation given by [2]

$$b(nT) = \sum_{i=1}^M w_i x_i(nT - n_i T), \quad (1)$$

where $b(nT)$ is the time series beam, $x_i(nT)$ is the output signal of the i -th sensor, M is the number of sensors, T is the temporal

sampling interval, and w_i denotes the shading coefficient for the i -th sensor. Each time shift, n_i , is selected on the basis of the desired MRA. Because the signals from the sensors are sampled, the achievable time shifts are limited to a set of quantized delays. Consequently, the number of achievable MRA steering directions is also limited.

The number of achievable steering directions is increased by increasing the sampling resolution [1]. Because of the large amount of data produced by a sensor array, the most efficient method of increasing the sampling resolution is the generation of fractional samples. Interpolation beamforming is a technique that uses multirate digital signal processing to effectively increase the physical sampling frequency by a factor of L in order to increase the number of achievable steering directions [1]. The interpolation beamformer equation is given by

$$b(nT) = \sum_{i=1}^M w_i x_i \left[nT - \left(n_i + \frac{m_i}{L} \right) T \right], \quad (2)$$

where m_i is an integer from 0 to $L - 1$ and m_i/L is the fractional delay required for the i -th sensor.

In the literature, the concept of fractional time shifts is traditionally approached from the time-domain beamforming perspective. Although the most intuitive approach, time-domain beamforming is not always the

* Managed by Lockheed Martin Energy Systems, Inc. for the U.S. Department of Energy under contract DE-AC05-84OR21400.

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. Regardless of the implementation, the need for fractional sample generation exists. The purpose of Section II is to examine the interpolation beamformer from a frequency-domain perspective.

II. Fractional Delay Generation Using DFT Delay Property

Frequency-domain beamforming is a term that describes algorithms that involve the use of the fast Fourier transform (FFT) as part of the beamforming process. Perhaps the most widely used algorithm for discrete time applications is the FFT beamformer. A diagram illustrating the production of a single beam using an FFT beamformer is given in Figure 1. Following the production of each sensor's discrete frequency spectrum, the weighted delay-and-sum beamforming operation defined by Eq. (1) is accomplished by a shift and sum operation given by

$$\hat{Y}(k) = \sum_{i=1}^M H_i(k) X_i(k), \quad (3)$$

$$k = 0, 1, \dots, N-1$$

where N is the length of the FFT. A common approach to FFT beamforming is to define $H_i(k)$ [3] as

$$H_i(k) = w_i \exp\left(-j 2\pi k \frac{\tau_i}{N}\right), \quad (4)$$

where τ_i is the time shift in samples required for the i -th sensor given by [1]

$$\tau_i = \frac{[(\mathbf{r}_i \cdot \mathbf{u}) - (\mathbf{r}_{ref} \cdot \mathbf{u})]}{f_s c}, \quad (5)$$

where \mathbf{r}_i is the position of the i -th sensor, \mathbf{r}_{ref} is the position of the reference sensor, \mathbf{u} is the beam steering direction, f_s is the sampling frequency, and c is the speed of propagation.

While apparently a straightforward application of the DFT delay property, Eq. (4) will produce errors if applied as shown in Figure 1. Consider the case in which τ_i is a purely integer delay α_i . In this case, substituting Eq. (4) into Eq. (3) circularly shifts the time series of the i -th sensor. As a result, the time series beam output from the inverse FFT will contain discontinuities. These discontinuities can be eliminated with the overlap-save method [1]. The amount of overlap required for the i -th sensor is determined by the length of the filter, $h_i(n)$. In Eq. (3), the multiplication of $X_i(k)$ by the linear phase term corresponds to convolution of $x_i(n)$ with a finite length filter having an impulse response defined by

$$h_i(n) = \delta [n - (\alpha_i \oplus N)], \quad (6)$$

where \oplus is the modulo operator. Thus, the amount of overlap required to eliminate circular convolution in the i -th sensor due to the integer shift is α_i .

Using Eqs. (3) and (4) coupled with overlap-save processing is equivalent to generating the integer time shift portion of the interpolation beamformer; however, the use of Eq. (4) for generating fractional sample delays causes distortion in $y(n)$. A

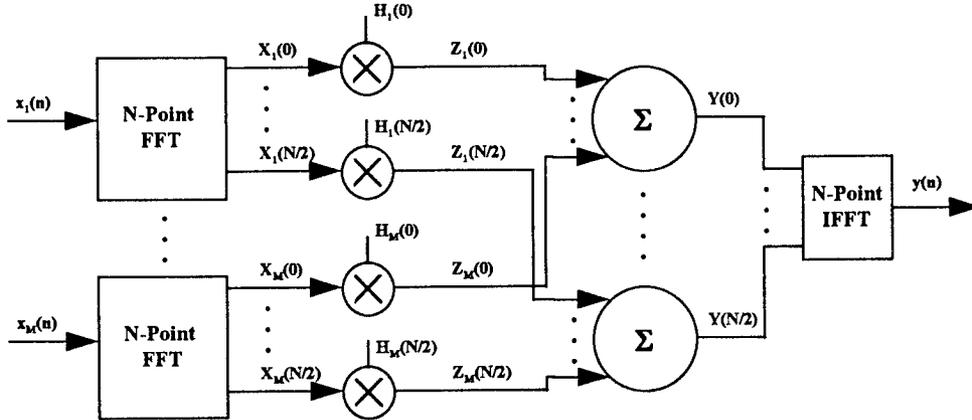


Figure 1. FFT beamformer structure for producing single beam [3].

fractional value for τ_i in Eq. (4) produces an impulse response given by [4]

$$h(n, \tau_i) = \exp\left[\frac{-j\pi(n - \tau_i)}{N}\right] \frac{\sin\left[\frac{\pi(n - \tau_i)}{N}\right]}{N \sin\left[\frac{\pi(n - \tau_i)}{N}\right]} \quad (7)$$

which is complex and nonzero for all values of n . Thus, circular convolution effects cannot be completely eliminated regardless of the amount of overlap [4]. Section III describes a method to properly generate fractional sample delays.

III. Delay Filter Generation

The time shift, τ_i , consists of an integer component and a fractional component. Based on these components, $H_i(k)$ can be synthesized using a combination of storage and computation. A diagram illustrating this algorithm is shown in Figure 2. In this scheme a time shift for the i -th sensor, τ_i , is computed using Eq. (5). This time shift is then partitioned into an integer and fractional component according to

$$\begin{aligned} Idelay_i &= \lfloor \tau_i fs + N \rfloor \\ Fdelay_i &= \tau_i fs - (Idelay_i - N) \end{aligned} \quad (8)$$

These components are used to compute pointers into large data stores containing the integer and fractional delay filters.

A. Fractional Delay Generation

The Fractional Delay Filters store contains the DFT representation of finite impulse response (FIR) filters that implement the fractional delays:

$$\Gamma_\rho = \frac{\rho}{L} \quad \rho = 0, 1, 2, \dots, L-1. \quad (9)$$

FIR filters eliminate the effects of circular convolution when used with overlap-save processing. The windowing method is used to design FIR filters with the delays given by Eq. (9). 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. (10):

$$h_{frac}(n, \Gamma_\rho) = w(n, \Gamma_\rho) h_{ideal}(n, \Gamma_\rho). \quad (10)$$

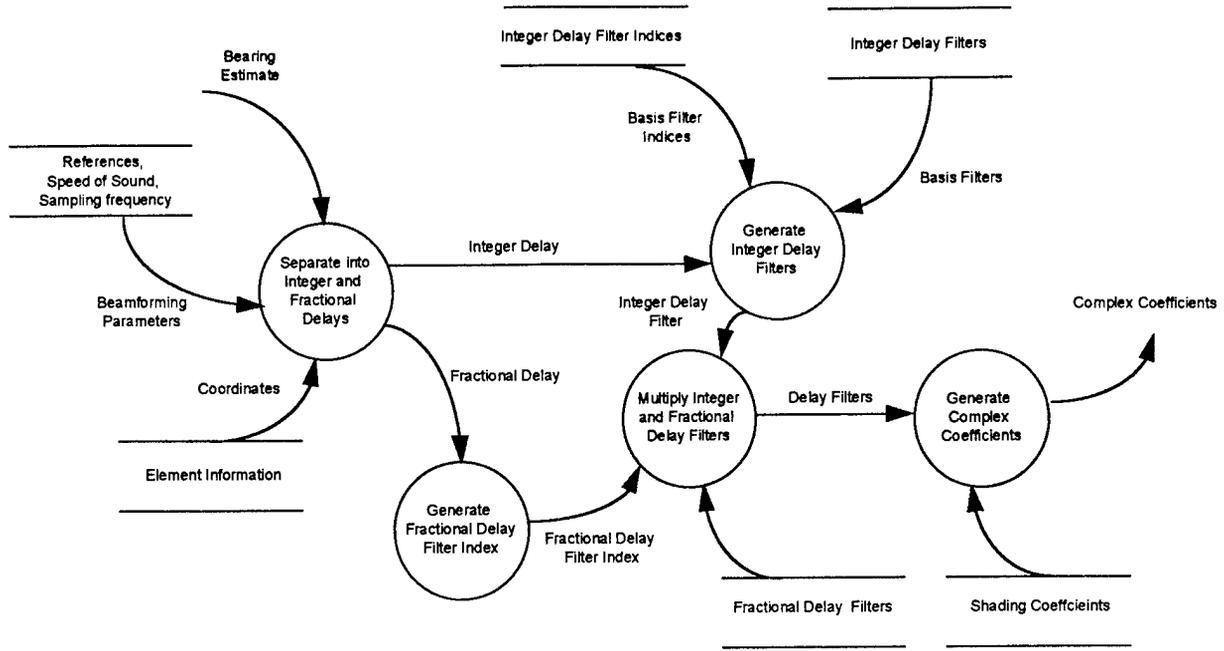


Figure 2. Algorithm for Generating Delay Filters.

The impulse response for the ideal filter that generates Γ_p is given by

$$h_{ideal}(n, \Gamma_p) = \frac{\sin[\pi(n - \Gamma_p)]}{\pi(n - \Gamma_p)}. \quad (11)$$

The window function, $w(n, \Gamma_p)$, is an l -point Hamming window given by

$$w(n, \Gamma_p) = \begin{cases} 0.54 + 0.46 \cos\left[\frac{\pi(n - \Gamma_p)}{l}\right] & l - \Gamma_p \leq n \leq l + \Gamma_p \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

B. Integer Delay Generation

Unlike the fractional delay filters, all integer delay filters required are not stored; instead, a set of basis integer delay filters given by

$$I(k)_p = \exp\left(\frac{j 2 \pi k p}{N}\right) \quad (13)$$

$$p = 0, 1, \dots, (p_{con} - 1), 2(p_{con} - 1), \dots, p_{max}$$

is stored. p_{con} is the number of consecutive delays and p_{max} is the number of the last filter required. The composite integer delay filter is given by

$$Hint(k, Idelay_i) = I(k)_{p1} I(k)_{p2}, \quad (14)$$

where

$$p1 + p2 = Idelay_i. \quad (15)$$

For $Idelay_i < 0$, the integer delay filter is given by

$$Hint(k, Idelay_i) = Hint(k, N + Idelay_i). \quad (16)$$

Therefore, the composite filter that best approximates the delay $\exp(j\omega\tau_i)$ is given by

$$H_i(k) = H_{int}(k, Idelay_i) H_{frac}(k, \Gamma_{\lfloor L F delay_i \rfloor}) \quad (17)$$

IV. Memory Requirements

The method presented in Section III provides significant improvement in required data storage. If every integer delay is stored, the amount of storage required for the integer and fractional delays is $(N + L)N$ complex words. This storage requirement would have exceeded the available 16 MB of memory in the array processor used for this application. Using the method in Section III, all possible integer delays are obtained by a single complex vector multiplication. Splitting the integer delays in this manner results in a substantial memory storage savings. This technique requires Q sets of filter coefficients, where Q is given by

$$Q = p_{con} + \left\lceil \frac{[p_{max} - (p_{con} - 1)]}{p_{con}} \right\rceil \quad (18)$$

Solving for the minimum value of Q yields

$$Q_{min} \approx \lceil 2\sqrt{N+1} - 1 \rceil. \quad (19)$$

Thus, the required storage space is reduced by a factor of F , where F is given by

$$F = \frac{(N + L)}{2\sqrt{N+1} - 1 + L}. \quad (20)$$

For $N \gg L$, the required storage space is reduced by approximately $\sqrt{N} / 2$.

V. Summary

This paper presents a method for generating phase shifts for frequency-domain

beamformers. The method utilizes the concept of fractional and integer delays to achieve the desired phase shift. The fractional delay component is produced by using FIR delay filters which are computed and stored prior to starting the beamformer. The use of FIR delay filters produces phase shifts without the circular convolution errors caused by blind application of the DFT delay property. The integer delays are produced efficiently by combining storage and computational resources. A base set of integer delays is generated and stored. From this set, all integer delays for a given FFT size are obtainable with a single complex vector multiplication. This compromise between memory storage and computation enables real-time frequency-domain beamforming operations.

VI. References

1. Neilson, Richard O. *Sonar Signal Processing*. Boston: Artech House, 1991.
2. Dudgeon, Dan E. and Russell M. Mersereau. *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1984.
3. Mucci, Ronald A. "A Comparison of Efficient Beamforming Algorithms", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-32, (1984).
4. Bomar, Bruce W. and L. Montgomery Smith. "Frequency-Domain Implementation of Fractional Sample Delays for Beamforming," *University of Tennessee Space Institute Report*, 1991.

M98000325



Report Number (14) ORNL/CP--94457
CONF-9510352--

Publ. Date (11) 1995 07 --

Sponsor Code (18) DOD, XF

UC Category (19) UC-000, DOE/ER

DOE