

## Article

# Thiran Filters for Wideband DSP-Based Multi-Beam True Time Delay RF Sensing Applications

Sirani M. Perera <sup>1,\*</sup>  , Gayani Rathnasekara <sup>2</sup> and Arjuna Madanayake <sup>2</sup> 

<sup>1</sup> Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA

<sup>2</sup> Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33199, USA; grath003@fiu.edu (G.R.); amadanay@fiu.edu (A.M.)

\* Correspondence: pereras2@erau.edu

**Abstract:** The ability to sense propagating electromagnetic plane waves based on their directions of arrival (DOAs) is fundamental to a range of radio frequency (RF) sensing, communications, and imaging applications. This paper introduces an algorithm for the wideband true time delay digital delay Vandermonde matrix (DVM), utilizing Thiran fractional delays that are useful for realizing RF sensors having multiple look DOA support. The digital DVM algorithm leverages sparse matrix factorization to yield multiple simultaneous RF beams for low-complexity sensing applications. Consequently, the proposed algorithm offers a reduction in circuit complexity for multi-beam digital wideband beamforming systems employing Thiran fractional delays. Unlike finite impulse response filter-based approaches which are wideband but of a high filter order, the Thiran filters trade usable bandwidth in favor of low-complexity circuits. The phase and group delay responses of Thiran filters with delays of a fractional sampling period will be demonstrated. Thiran filters show favorable results for sample delay blocks with a temporal oversampling factor of three. Thiran fractional delays of orders three and four are mapped to Xilinx FPGA RF-SoC technologies for evaluation. The preliminary results of the APF-based Thiran fractional delays on FPGA can potentially be used to realize DVM factorizations using application-specific integrated circuit (ASIC) technology.



**Citation:** Perera, S.M.; Rathnasekara, G.; Madanayake, A. Thiran Filters for Wideband DSP-Based Multi-Beam True Time Delay RF Sensing Applications. *Sensors* **2024**, *24*, 576. <https://doi.org/10.3390/s24020576>

Academic Editors: Roman Sotner and Giuseppe Ferri

Received: 7 November 2023

Revised: 8 December 2023

Accepted: 29 December 2023

Published: 17 January 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** complexity and performance of algorithm; structured matrices; signal processing and analysis; TTD wideband multi-beam beamforming; Thiran fractional delays; array processing; wireless communication

## 1. Introduction

The directional enhancement of propagating RF plane waves is a key requirement for electromagnetic sensing using antenna array processors. Linear arrays of uniformly spaced wideband antenna elements receive far-field propagating electromagnetic plane waves, which must be directionally enhanced along the desired look direction using a beamformer. A TTD beamformer enhances a wideband plane-wave signal along a particular DOA. To wit, as the instantaneous bandwidth of modern antenna apertures increases, it leads to wideband systems. The underlying beamforming schemes for such an aperture must be realized with support for squint-free operation across the bands of interest [1]. The TTD beamformer is the optimal beamformer when the signals of interest are contaminated by wide sense stationary AWGN [2–5]; the case of the sensing of RF plane waves along multiple directions is a crucially important application for antenna array-based RF sensing and direction-of-arrival estimation [6] with use cases in RF imaging, RF communications, and RF location.

For an aperture with  $N$  elements, an  $N$ -th order TTD beamformer can be realized using analog, mixed-signal, or direct-digital methods for realizing such a TTD aperture beam. A TTD beamformer can be steered by the appropriate selection of the progressive time delay from one element to the next in a uniform aperture configuration. In cases where

a large number of wideband beams are required, a dedicated TTD beam must be created through its own TTD beamsteering network [5]. Typically, for  $N$  elements,  $N$ -independent beams can be realized by considering a short duration time delay  $\tau_0 = T/N$ , where  $T$  is the time taken for the radio signal to propagate between two neighboring elements in the aperture. The parallel realization of  $N$  beams using a network of TTDs with durations set to integer multiples of  $\tau_0$  leads to an  $N$ -beam TTD multi-beam beamformer with an underlying mathematical framework given by the DVM [7–11]. The DVM is similar in concept to a spatial DFT; however, unlike the case of the DFT which uses a fixed complex coefficient in the matrix–vector operation, the DVM uses frequency-dependent weights defined as integer multiples of the smallest TTD component  $\tau_0$ .

### 1.1. Review: Fractional Delay Filters

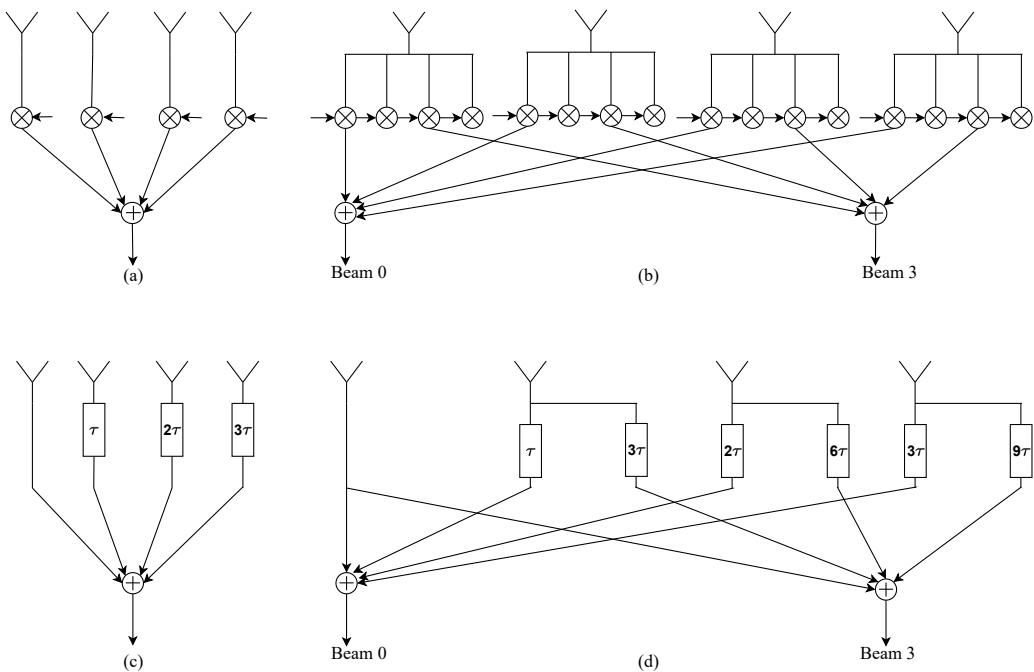
Although FFT-type algorithms utilizing Thiran fractional delays are presently not available, one could observe other methods based on fractional delay filters. The authors in [12] proposed different methodologies and parameters for designing allpass fractional delay filters. The authors claimed that the employment of Lagrange FIR and Thiran fractional delay filters are the best choices for achieving desirable phase delay responses. The paper [13] evaluated designs for FIR fractional filters, including sinc, frequency sampling, polynomial interpolation, and least integral-square error. In designing fractional delay filters, the paper [14] addressed the utilization of Taylor series expansions of  $e^{-j\omega\tau}$  when  $\tau$  represents a combination of integer delay and a fractional delay within the interval of  $[-0.5, 0.5]$ . This approach was utilized to design fractional delay filters and gauge the efficacy of their structure in relation to the Farrow structure, primarily in terms of the storage of filter coefficients in [14]. The realization of allpass fractional delay filters was also addressed through the truncation of the power series expansion based on the denominator of the transfer function of the variable  $z$  at  $z^{-1} = 1$  [15]. The paper [16] addressed the reconstruction techniques for nonuniformly sampled band-limited signals using digital fractional delay filters using a Vandermonde structured matrix followed by a diagonal scaling. One can also find the factorization of the transfer function of the infinite impulse response (IIR) Simpson integrator for designing fractional delay filters in [17]. On the other hand, authors in [18] presented the introduction of the Hilbert transform operator, which was developed using a half-sample delay operator created through the B-spline transform interpolation and decimation. However, there is a lack of computationally efficient algorithms in the literature, despite the fact that the methods for designing Thiran fractional delay filters have been established.

### 1.2. Wideband Beamformer Structure

Figure 1 shows a single narrow-beam phased-array beamforming structure, a DFT-based multi-beam phased-array structure, a single TTD wideband beamformer structure, and an  $N$ -beam TTD wideband multi-beam structure, respectively. We first define the DVM by

$$[\mathbf{A}_N]_{kl} := \mathbf{A}_N = [\alpha^{kl}]_{k=1,l=0}^{N,N-1},$$

where  $\alpha = e^{-j\omega\tau_0}$ . The term  $\alpha$  accounts for the phase rotation associated with the smallest fractional delay  $\tau_0$  (necessary for realizing the DVM  $N$ -beam system) at frequency  $f$ , where the temporal frequency  $\omega = 2\pi f$ . Unlike the DFT-based  $N$ -beam phased array, which can be realized using self-contained factorization of the DFT followed by the FFT algorithm, the corresponding  $N$ -beam TTD structure has no direct equivalent of the FFT. We address this problem using sparse factorization followed by a DVM algorithm while replacing complex coefficients with frequency-dependent weights realized through linear filters.



**Figure 1.** (a) Narrowband phased-array single-beam beamforming structure; (b) narrowband DFT-based multi-beam phased-array beamforming structure corresponding to a matrix–vector operation using the DFT matrix, in turn achieved via FFT based on the self-contained factorization of DFT matrix; (c) wideband single-beam TTD beamformer structure; (d) unsimplified wideband  $N$ -beam multi-beam TTD beamforming structure corresponding to a matrix–vector operation using the DVM. The proposed work shows how the above-mentioned architecture corresponds to a sparse factorization of DVM followed by a DVM algorithm that is realized as butterfly-like signal flow graphs (SFGs), where twiddle factors are realized in the discrete-time domain by the proposed twiddle filters based on a Thiran filter building block.

The analog realization of low-complexity and/or radix-2 DVM algorithms was previously discussed in references [7–11]. The structure of the DVM offers a framework for analyzing TTD-based wideband multi-beam beamforming [1]. The DVM overcomes the limitations associated with employing the DFT matrix. For example, wideband signals applied to DFT-based multi-beams lead to an undesirable “beam squint” effect. The DVM entries, referred to as  $\mathbf{A}_N$  in [7–11], serve as the foundation for a TTD-based beamformer. It is important to note that the DFT matrix equals the matrix  $\mathbf{A}_N$  at a single temporal frequency because the DVM covers a wide range of multiple temporal frequencies. Thus, the DVM is a superclass of the DFT matrix. However, unlike the DFT matrix, the DVM matrix lacks both properties of unitary and periodicity. Therefore, unlike the case of the DFT that can be self-factored to obtain FFT algorithms [19–21], we are generally unable to factorize the DVM to obtain a self-contained and radix-2 factorization, leading to an FFT-like algorithm.

### 1.3. Contribution of the Paper

To develop an efficient digital DVM algorithm that incorporates fractional delays, we will utilize the bidiagonal factorization technique in [7]. By implementing an array representation of the DVM factorization by a vector, we present a DVM algorithm that efficiently reduces the complexity associated with the matrix–matrix multiplication step of the DVM algorithm described in [7]. Furthermore, we provide SFGs specifically for small-array elements, i.e.,  $N = 4, 8$  incorporating Thiran fractional delays. Specifically, the fractional part associated with each node in the SFGs is implemented using Thiran fractional delay filters. Additionally, the integer part is realized using FIFO buffers implemented using D-flop pipelines and/or realized as the integer delay component of a Thiran filter. This

approach allows for a precise and efficient representation of the SFGs, facilitating enhanced performance and accuracy in TTD-based wideband multi-beam digital beamforming.

#### 1.4. Organization of the Paper

We provide an overview of the analog DVM algorithms proposed for TTD wideband multibeam beamforms in Section 2. In Section 3, we give an overview of digital DVM beamformers using Thiran filters. Next, in Section 4, we recall the DVM factorization in [7] and present an array implementation of the factorization using a novel DVM algorithm. The proposed algorithm is more reliable for large  $N$  than that for the matrix–matrix multiplication-based DVM algorithm in [7,8]. Section 4.1 shows the arithmetic complexity (quantified using the number of necessary adders and gain–delay blocks) and numerical results for the DVM algorithm, and compares the arithmetic complexity with the matrix–vector product with the previous work. Computing the DVM–vector product with fractional delays has low arithmetic complexity compared to the brute force calculation. Section 5 presents a brief overview of analog and digital DVM beamforming using APFs. Section 6 shows the simulation for the linear phase response of Thiran filters. Next, Section 7 demonstrates Thiran filters for realizing each fractional sample delay based on the proposed DVM algorithm followed by the SFGs. Section 8 discusses future work on utilizing the APF-based Thiran fractional delay units to realize the full DVM factorization. Finally, Section 9 concludes the paper.

## 2. Review: Analog DVM Beamformers

In prior work, we proposed a sparse factorization of the  $N$ -beam DVM leading to 60% reduction in integrated circuit complexity [7]. This low-complexity  $N$ -beam DVM algorithm is based on the product of complex 1-band upper and lower matrices. In this paper, we expand on the algorithmic results in [22–24] by utilizing complex nodes instead of real nodes. The 1-band upper and lower matrix factorization of  $\mathbf{A}_N$  in [7] is expressed without utilizing quasi separability, displacement equations, and the factorization of Hankel/Toeplitz matrices [25–31]. Moreover, the paper addressed error bounds and stability by filling the gaps in [22–24] to compute DVM by a vector.

The paper [10] proposed an exact, efficient (i.e., more efficient than brute force multiplication of the DVM by a vector but not radix-2), and self-recursive DVM algorithm by using a matrix factorization of the DVM and a polynomial evaluation associated with its nodes to analyze multi-beam antenna arrays. The exact calculation of the DVM vector product via the DVM algorithm in [10] can be used to reduce the complexity of RF  $N$ -beam analog beamforming systems. Although the algorithms proposed in [8,10] are much more efficient than brute-force calculation, their order of arithmetic complexity is not  $\mathcal{O}(N \log N)$ . We recall here that the stable (well conditioned) and  $\mathcal{O}(N \log N)$  algorithms for Vandermonde matrices proposed in [9] can be used for the narrowband communication system. Derivations have been established for radix-2 and split-radix FFT algorithms in [19–21,32–36]. Despite the derivation of size- $N$  DFT into two size- $\frac{N}{2}$  DFTs being quite straightforward, its extension to the DVM is cumbersome because useful DFT matrix properties (periodicity and unitary) are *not* present in the DVM.

To overcome the above challenge, we proposed an  $\mathcal{O}(N \log N)$  DVM algorithm having sparse factors to achieve multi-directional TTD wideband beamforming to solve the longstanding beam squint problem in [11]. In [11], the “multiplication counts” in the frequency domain—albeit in the analog domain—are a combined result of analog gains (amplifiers) and time delays in the SFG when realized in a time-domain circuit.

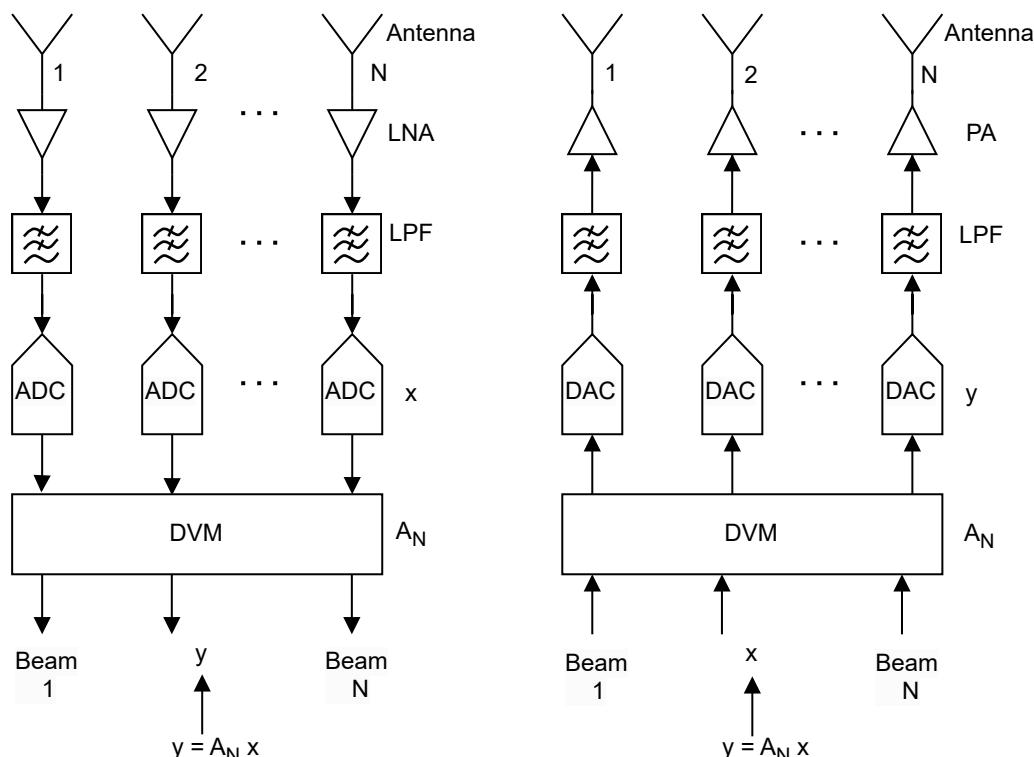
## 3. Digital DVM Beamformers Using Thiran Filters

Figure 2 shows a TTD multi-beam beamforming aperture for receive (left) and transmit (right) applications. In both cases, the beamformers are realized in the digital signal processing (DSP) unit that implements the DVM–vector operation once every sampling period. The TTD digital beamformers can be realized as the DVM–vector product s.t.  $\mathbf{A}_N \mathbf{x}$ ,

where  $\mathbf{x}$  is the  $N$ -point vector of input signals obtained from  $N$  receivers (or feeding  $N$  transmitters after digital-to-analog conversion). In the DVM  $\mathbf{A}_N$ , the  $k^{th}$ -row and  $l^{th}$ -column element is of the form  $e^{-j\omega\tau_0 kl}$ , and hence the corresponding delays take the form  $T(l) = \tau_0 kl$ , where  $\tau_0$  is the smallest fractional delay necessary for realizing the DVM  $N$ -beam system, and  $\omega$  is the temporal frequency. Typically,  $\tau_0 = T_s/N$ . We note here that the factorization of the  $N$ -beam DVM leads to a range of delays  $T_i = \tau_0 P_i$ , where  $i$  is the index to the  $i^{th}$ -node of the signal flow graph (which is, in fact, the butterfly diagram for the radix case) corresponding to the factored version of  $\mathbf{A}_N$  and  $P_i \in \mathbb{Z}^+$ , and  $P_i \leq (N-1)^2$  are the different time delays across the nodes of the signal flow graph (butterfly for the radix case) pertaining to the proposed fast algorithm for  $\mathbf{y} = \mathbf{A}_N \mathbf{x}$ . It is important to note that at node  $i$ , the delay can be decomposed to the form  $\tau_0 P_i = n_i T_s + \tau_i$ , where  $\tau_i < T_s$  and where  $n_i \in \mathbb{Z}^+$  is an integer delay realized through FIFO register buffers located at point  $i$  of the signal flow graph (butterfly for the radix case). Each delay operation at node  $i$  consists of an integer component and a fractional component. The fractional delays  $\tau_i \leq T_s$  will be realized as fast interpolation filters based on finite impulse response (FIR) and/or all-pass IIR Thiran filters. We start with the  $n_i^{th}$ -order Thiran fractional delay filter of the form [12]:

$$H_i(z) = z^{-n_i} \frac{p_i(z)}{p_i(z^{-1})}, \quad (1)$$

where  $p_i(z) = 1 + a_1 z + \dots + a_{n_i} z^{n_i-1}$ ,  $a_k = (-1)^k \binom{n_i-1}{k} \prod_{l=0}^{n_i-1} \frac{D_i - n_i + l}{D_i - n_i + k + l}$ ,  $D_i = \frac{\tau_i}{T_s}$ , fractional sample delay  $\tau_i$ , temporal sample period  $T_s$ , and  $k = 1, 2, \dots, n_i$ . Here, each set of coefficients  $a_k$  is unique to node  $i$ ; however, we dropped the dependence on  $i$  for readability. We obtain an approximated transfer function for the digital all-pass Thiran fractional delay



**Figure 2.** Overview architecture. RF antenna array used for optimal electromagnetic sensing of wideband plane waves based on their DOAs. (Left)  $N$ -beam TTD receiver aperture using DVM beamforming, and, (right)  $N$ -beam TTD transmit aperture using DVM beamforming.

#### 4. A Fast DVM Algorithm for Thiran Fractional Delays

We begin this section by recalling the sparse factorization of the DVM in [7]. Subsequently, an algorithm is presented based on the array implementation as opposed to the matrix–matrix implementation associated with the phase rotation  $\alpha = e^{-j\omega\tau_0}$  followed by the smallest fractional delay  $\tau_0$ . We note that the primary objective of this study is to propose a low-complexity DVM algorithm yielding to compute DVM–vector multiplication using the Thiran fractional delays. Our work builds upon the studies [7–11]. It is important to note that this paper does not address related issues, such as the computation of Vandermonde structured matrices by a vector, inverse Vandermonde structured matrices (leading to inversion formulas or inversion algorithms), or the solution of the linear systems associated with Vandermonde structured matrices as the coefficient matrices as in [37–47].

**Proposition 1** (Recalling from [7]). *Let the DVM, denoted by  $\mathbf{A}_N$ , be defined via the nodes  $\{\alpha, \alpha^2, \dots, \alpha^N\} \in \mathbb{C}$  for integers  $N$  s.t.  $N \geq 4$ . Then, the DVM can be factored into the product of bidiagonal matrices s.t.*

$$\mathbf{A}_N = \mathbf{L}^{(1)} \mathbf{L}^{(2)} \dots \mathbf{L}^{(N-1)} \mathbf{U}^{(N-1)} \dots \mathbf{U}^{(2)} \mathbf{U}^{(1)}, \quad (2)$$

where

$$\mathbf{L}^{(m)} =$$

$$\left[ \begin{array}{c|ccccc} \mathbf{I}_{N-m-1} & & & & & \\ \hline & 1 & & & & \\ & 1 & \alpha^{N-m}(\alpha - 1) & & & \\ & & 1 & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & 1 & \alpha^{N-m}(\alpha^m - 1) \end{array} \right],$$

$$\mathbf{U}^{(m)} = \left[ \begin{array}{c|ccccc} \mathbf{I}_{N-m-1} & & & & & \\ \hline & 1 & \alpha & & & \\ & 1 & \alpha^2 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & \alpha^m & \\ & & & & & 1 \end{array} \right],$$

$m = 1, 2, \dots, N - 1$ , and  $\mathbf{I}_k$  is the  $k \times k$  identity matrix.

**Remark 1.** We note here that the above factorization can be used to propose the ddvm algorithm (Algorithm 1) based on Thiran fractional delays when the nodes of the DVM (i.e.,  $\alpha^k$ ) are associated with the phase rotation  $\alpha = e^{-j\omega\tau_0}$  followed by the smallest fractional delay  $\tau_0$ . Additionally, in Section 5, we utilize the above factorization to present the signal flow graphs for  $\mathbf{y} = \mathbf{A}_N \mathbf{x}$ , where the  $(k, l)$  element of the matrix  $\mathbf{A}_N$  is determined by the fractional delay component associated with  $e^{-j\omega\tau_0 kl}$ .

##### 4.1. Arithmetic Complexity of the DVM Algorithm

We show that the arithmetic complexity of the proposed DVM algorithm is less than the standard quadratic complexity. Furthermore, when the algorithm executes, it uses array implementation to avoid matrix–matrix multiplication in [7]. The number of additions and multiplications used to execute the DVM algorithm correspond to the adders and gain delay blocks, respectively.

The digital DVM using Thiran fractional delays describes a DSP system where each fractional true time delay is realized using an IIR all-pass fractional delay filter. The Thiran filter is one type of digital all-pass filter that maintains nearly constant group delay in the frequency domain  $-\pi/2 \leq \omega \leq \pi/2$ . The Thiran filters in this work will be realized using the direct-form type-II SFG so that the amount of D-flop FIFO memory is minimized.

**Algorithm 1** *ddvm*

**Input:** Integer  $N$  s.t.  $N \geq 4$ ,  $\alpha = e^{-j\omega\tau_0} \in \mathbb{C}$  having the smallest fractional delay  $\tau_0$ , and  $\mathbf{x} \in \mathbb{R}^N$  or  $\mathbb{C}^N$

**Output:**  $\mathbf{y} = \mathbf{A}_N \cdot \mathbf{x}$

**Function:**  $\mathbf{y} = ddvm(\mathbf{x}, N, \alpha)$

**Steps:**

1. **Set**  $s = 2N$ ,  $\mathbf{u} = [\alpha \quad \alpha^2 \quad \dots \quad \alpha^N]^T$ , and  $\mathfrak{A} = [\mathbf{x} \quad \mathbf{0}_{N \times (s-1)}]$ .
2. **for**  $l = 1, 2, \dots, N-1$ ,  
**for**  $k = 1, 2, \dots, N$ ,  
**if**  $((k < (N-l))$  **or**  $(k = N))$   
 $\mathfrak{A}(k, l+1) = \mathfrak{A}(k, l)$ ,  
**else**  
 $\mathfrak{A}(k, l+1) = \mathfrak{A}(k, l)$   
 $+ \mathfrak{A}(k+1, l) \cdot \mathbf{u}(k-N+l+1)$ ,  
**end if.**  
**end for,**  
**end for.**
3. **for**  $l = N, N+1, \dots, s-1$ ,  
**for**  $k = 1, 2, \dots, N$ ,  
**if**  $(k \leq (l-N+1))$   
 $\mathfrak{A}(k, l+1) = \mathfrak{A}(k, l)$ ,  
**else**  
 $\mathfrak{A}(k, l+1) = \mathfrak{A}(k-1, l)$   
 $+ \mathfrak{A}(k, l) \cdot (\mathbf{u}(k) - \mathbf{u}(l-N+1))$ ,  
**end if.**  
**end for,**  
**end for.**
4. **return**  $\mathfrak{A}(:, s)$ , and **Set**  $\mathbf{y} = \mathfrak{A}(:, s)$ .

The number of complex additions (say  $\#a$ ) and the number of complex multiplications (say  $\#m$ ), i.e., adders and gain–delay blocks, respectively, are required to compute the DVM algorithm, i.e., *ddvm*, stated next.

**Lemma 1.** Let  $N \geq 4$  be a given integer and  $\mathbf{x} \in \mathbb{C}^N$ . The DVM algorithm, i.e., *ddvm*, can be computed with the following arithmetic complexities (respectively, adders and gain–delay blocks):

$$\begin{aligned} \#a(ddvm, N) &= \frac{3}{2}N(N-1), \\ \#m(ddvm, N) &= (N-1)^2. \end{aligned} \quad (3)$$

**Proof.** To pre-calculate the vector  $\mathbf{u}$  in Step 1 of the algorithm, one has to use  $\frac{N(N+1)}{2}$  gain–delay blocks but no adders. These counts are based on pre-computation and correlated with the design time computations. Also, the aim of Step 1 is to compute the output of the algorithm effectively. In Step 2 of the algorithm, there are no adders or gain–delay counts involved in the *if* part. For the *else* part, when  $(N-k) \leq i \leq (N-1)$ , we have one adder and gain–delay for each  $k$ , so in total, we have  $\frac{N(N-1)}{2}$  adders and also gain–delay blocks in Step 2. In Step 3 of the algorithm, there are no adders or gain–delay involved in the *if* part. For the *else* part, when  $(k-N+1) < i \leq N$ , we have two adders and one gain–delay for each  $k$ , so in total we have  $N(N-1)$  adders and  $\frac{N(N-1)}{2}$  gain–delay blocks in Step 3. In Step 4 of the algorithm, we extract the last column of  $\mathfrak{A}$  and assign it to the vector  $\mathbf{y}$ ; hence, no cost is involved. Hence, the proposed algorithm requires  $\frac{3}{2}N(N-1)$  adders

(respectively, complex additions) and  $(N - 1)^2$  gain–delay blocks (respectively, complex multiplications).  $\square$

**Remark 2.** We recall here that an array implementation of a DVM algorithm was used to solve the DVM system for the phased array digital receivers in [45], and hence were able to reduce the cost of solving the DVM system from  $\mathcal{O}(N^3)$  to  $\mathcal{O}(N^2)$ . Unlike in [45], we do not solve the systems of equations but rather compute the DVM–vector product.

#### 4.2. Comparison Results for the Arithmetic Complexities of DVM Algorithms

In this section, we compare the arithmetic complexity of the proposed DVM algorithm with the related work of DVM algorithms.

The gain–delay blocks of the proposed DVM algorithm (i.e.,  $(N - 1)^2$ ) show favorable results in comparison to [7] (i.e.,  $\frac{N(3N-1)}{2}$ ). This advantage is gained based on the pre-computation of nodes followed by the array implementation of the proposed algorithm. If we consider the gain–delay blocks in the pre-computation stage of the *ddvm* algorithm (step 1), the number of gain–delay blocks in the proposed algorithm and the algorithm in [7] is the same, i.e.,  $\frac{3}{2}N^2 + \mathcal{O}(N)$ . Furthermore, the number of adders remains consistent between the *ddvm* algorithm and the algorithm in [7]. The *ddvm* algorithm exhibits a higher count of adders and gain–delay blocks compared to the algorithms presented in [10,11]. This is because the algorithms in those papers have adders and gain–delay block counts of the order  $\mathcal{O}(N \log N)$ , whereas the *ddvm* algorithm maintains these counts of less than  $\mathcal{O}(N^2)$ . But, on the other hand, it is important to note that the paper [10] introduces additional criteria for the arrangement of nodes. Specifically, the nodes are evenly distributed on a circle centered at the origin with a radius  $r$ , where  $r \geq 1$ . This criterion is not imposed in this paper. Despite the fact that the algorithms presented in [11] exhibit a complexity of  $\mathcal{O}(N \log N)$ , this algorithm does not yield favorable outcomes when applied to antenna arrays with a small number of elements, specifically 4 and 8. However, the DVM algorithm in [11] shows significant efficiency, surpassing 90%, for DVM sizes of  $N \geq 256$ . On the other hand, the gain–delay blocks of the *ddvm* algorithm consistently exhibit better performance compared to those in [9] for both  $N = 4$  and  $N = 8$  configurations.

However, the objective of this paper is to utilize the DVM factorization, as outlined in Proposition 1 from [7], for the implementation of multi-beam digital wideband beam-forming systems that employ Thiran fractional delays. Thus, we will utilize the proposed algorithm to realize the signal flow graphs as shown in the next section.

#### 5. Thiran Fractional Delays for Twiddle Filter Realization

In this section, we will provide a brief overview and formulation of how analog and digital DVM beamforming can be realized using APFs.

##### 5.1. Analog RC Lattice APFs

The linear phase shift associated with analog DVM beamforming is  $e^{-j\omega\tau}$ , where  $\tau > \tau_0$ . The factor  $e^{-j\omega\tau}$  is associated with the factorization of the  $\mathbf{A}_N$  and can be efficiently approximated on chip by CMOS APFs [7–11,48,49].

##### 5.2. Digital Fractional Delay APFs

Upon digitization, the time domain becomes discrete due to the sampling operation in the ADCs. Hence, the linear phase fractional sample delay blocks corresponding to the  $(k, l)$  element of  $\mathbf{A}_N$ , i.e.,  $e^{-j\omega\tau_0 kl}$  where  $\tau_0$  is the smallest fractional delay necessary for realizing the DVM  $N$ -beam DSP system. The matrix factorization of  $\mathbf{A}_N$  for digital DVM can be realized using discrete-domain Thiran fractional delay filters while approximating the fractional time delay, i.e.,  $\tau_i < T_s$  where  $T_s = \tau_0 N$ .

### 5.3. Thiran IIR Filter Blocks

Unlike the analog DVM case, which uses the fact that  $e^{-j\omega\tau_i} \approx \left(\frac{1-j\omega\tau_i/2M}{1+j\omega\tau_i/2M}\right)^M$ ,  $M \in \mathbb{Z}^+$  and  $\omega$  is the frequency variable in general, the digital DVM requires Thiran fractional delay approximations given by the digital infinite impulse response (IIR) filter function  $H_i(z)$ , having a filter order  $n_i \in \mathbb{Z}^+$ . The purely fractional delay component  $\psi(z)$  is related to the Thiran filter via  $H_i(z) = z^{-n_i}(\psi(z))^{n_i}$ , and  $\psi(z)$  is the fractional delay associated with  $\tau_0$ .

### 5.4. Higher-Order Thiran APFs

We note that the fractional delay  $(\psi(z))^p$  can be realized either by cascading  $p$  units of first-order (or low-order) digital  $\psi(z)$  all-pass filters or by designing a single APF with the desired level of fractional delay so that it fits the model of  $H_i(z)$ . The former approach leads to  $p$ -times replicated computational hardware structures of a Thiran filter when realized as a VLSI implementation. We adopt the latter approach, where a unique Thiran filter of order  $n_i$  is realized as a hardware core for the implementation of  $\psi(z)^p$ .

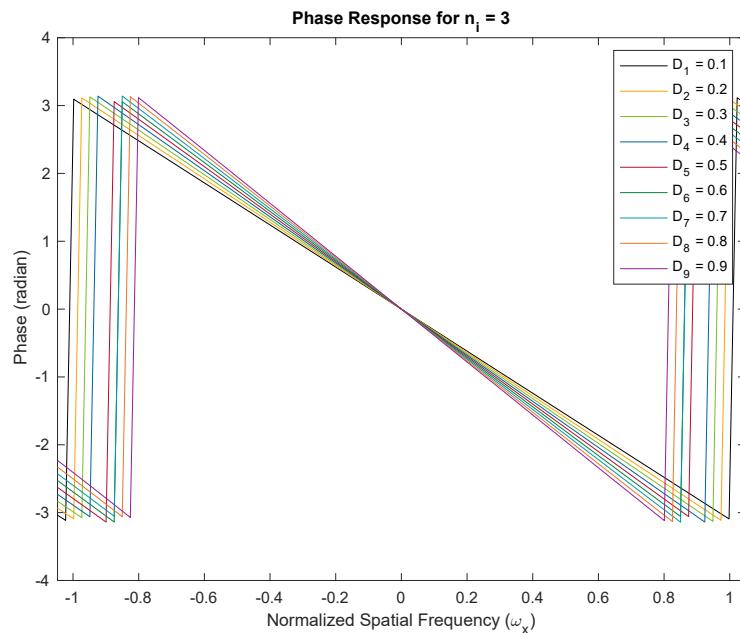
## 6. Simulation of Thiran Fractional Delays

We demonstrate the approximately linear phase response of Thiran filters. We show two examples with fractional delays of  $D_i = 0.1\text{--}0.9$  fractions of the temporal sampling period. We used Thiran IIR APFs of order  $n_i = 3$  and  $n_i = 4$ , respectively, with corresponding integer delay components of three and four clock periods, and the desired fractional delay components. Both filters are usable as approximately linear-phase fractional sample delay blocks better than the first 50% of the Nyquist period corresponding to a temporal oversampling factor of about two. All plots of the phase and group delay vs. frequency are in the domain  $-\pi/3 \leq \omega \leq \pi/3$ . In Figure 3, we show the phase response of the Thiran all-pass filter with  $n_i = 3$  and  $D_i = 0.1\text{--}0.9$ . The negative linear phase behavior of the filter within the first half of the Nyquist period is clearly apparent from the simulation. The approximately linear phase response results in approximately constant group delays as shown in Figure 4. The experiments are repeated for  $n_i = 4$  in Figure 5 for phase responses and Figure 6 for group delay, respectively.

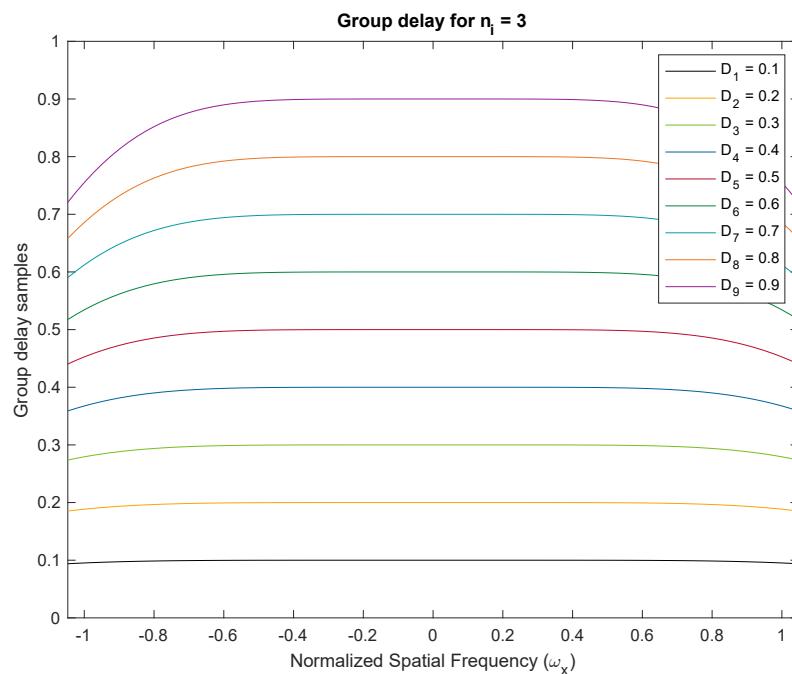
### Phase Responses and Group Delay Profiles of Thiran Filters

Figures 3 and 5 illustrate phase responses for order  $n_i = 3$  and order  $n_i = 4$  for delay variations  $D_i = 0.1\text{--}0.9$ , where  $D_i = \tau_i/T_s$  respectively. Figures 4 and 6 illustrate the group delay profiles for order  $n_i = 3$  and order  $n_i = 4$  for delay variations  $D_i = 0.1\text{--}0.9$ , where  $D_i = \tau_i/T_s$  respectively.

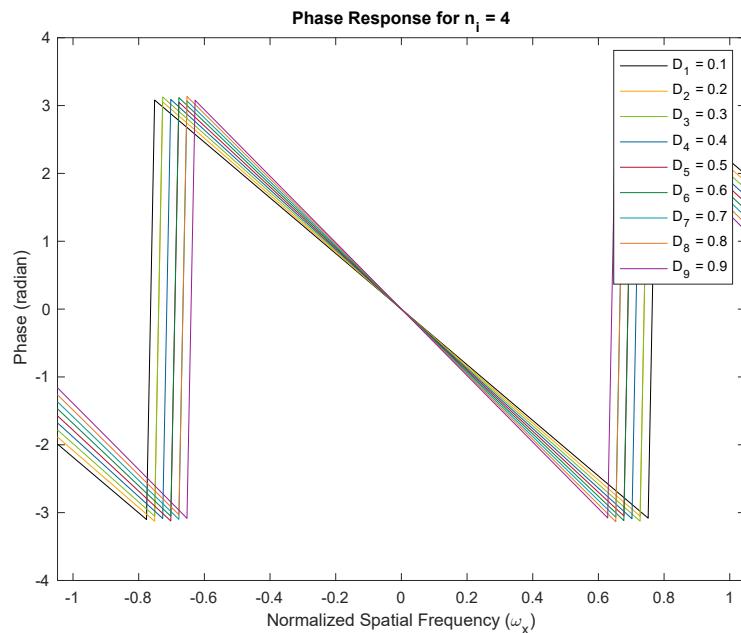
The TTD beam shapes of the multi-beam beamformer remain unchanged regardless of the method used to achieve the TTDs. The main requirement is to have a constant group delay that matches the desired TTD value and a response with a unit magnitude. These conditions are satisfied within the usable frequency range. The group delay of the Thiran filters enables the desired TTD value to be achieved within the operating band, assuming temporal over-sampling up to  $3\times$ . If the group delay of the TTD at a specific frequency matches the desired TTD, the aperture will produce the same beam. When there is an insufficient amount of temporal oversampling, the Thiran filters may not perform at their best, making a strong case for deviation. In practice, we can disregard this due to the use of significant over-sampling. Our paper concludes that the Thiran filter-based approach is exclusively suitable for temporal over-sampling factors greater than approximately  $3\times$  and summarizes this using Figure 7.



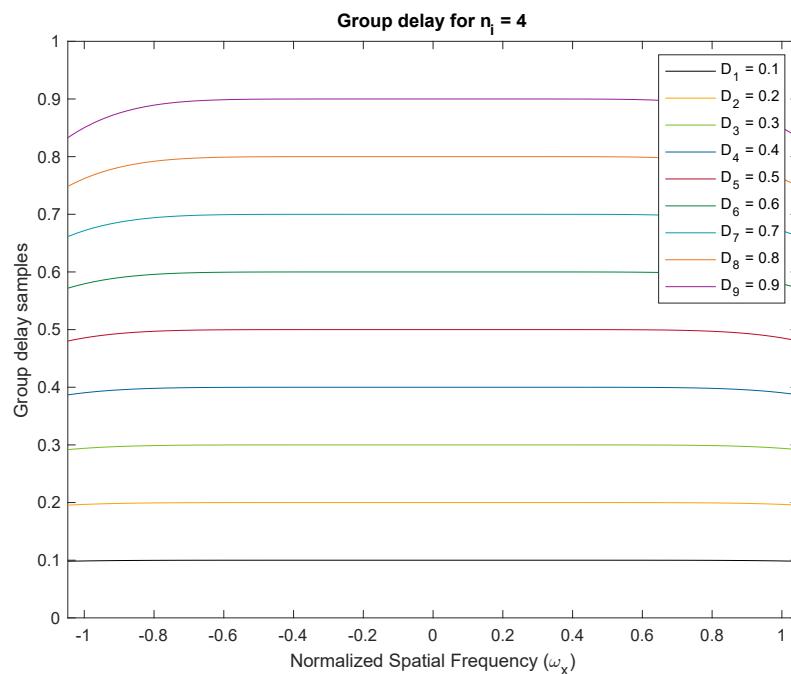
**Figure 3.** Phase responses for order  $n_i = 3$  and delay variations  $D_i = 0.1\text{--}0.9$ , where  $D_i = \tau_i/T_s$ . With temporal over-sampling of  $3\times$ , the usable range of the Thiran filter response is contained within the normalized  $-0.33$  to  $0.33$  frequency band. The linear phase response establishes the role of Thiran filters in realizing fractional sample delays.



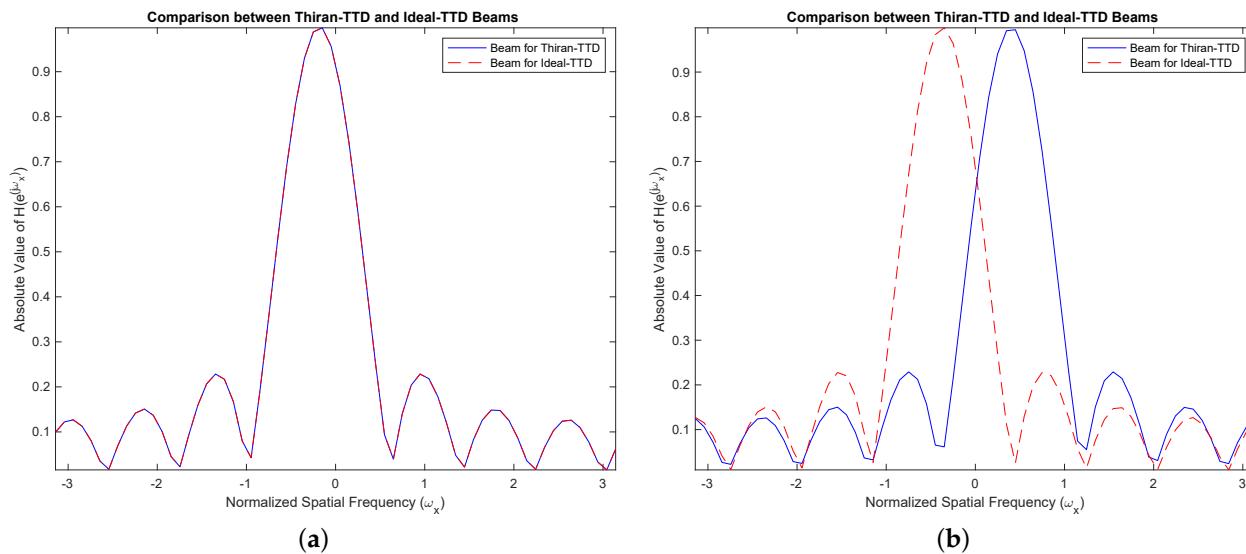
**Figure 4.** Group delay profiles for order  $n_i = 3$  and delay variations  $D_i = 0.1\text{--}0.9$ , where  $D_i = \tau_i/T_s$ . Constant group delay can be observed over 50% of the Nyquist interval. With temporal over-sampling of  $3\times$ , the usable range of the Thiran filter response is contained within the normalized  $-0.33$  to  $0.33$  frequency band. The constant group delay establishes the role of Thiran filters for realizing fractional sample delays.



**Figure 5.** Phase responses for order  $n_i = 4$  and delay variations  $D_i = 0.1\text{--}0.9$ , where  $D_i = \tau_i/T_s$ . With temporal over-sampling of  $3\times$ , the usable range of the Thiran filter response is contained within the normalized  $-0.33$  to  $0.33$  frequency band. The linear phase response establishes the role of Thiran filters in realizing fractional sample delays.



**Figure 6.** Group delay profiles for order  $n_i = 4$  and delay variations  $D_i = 0.1\text{--}0.9$ . Constant group delay can be observed over 50% of the Nyquist interval. With temporal over-sampling of  $3\times$ , the usable range of the Thiran filter response is contained within the normalized  $-0.33$  to  $0.33$  frequency band. The constant group delay establishes the role of Thiran filters in realizing fractional sample delays.



**Figure 7.** (a) shows a TTD beam for  $3 \times$  temporal over-sampling ( $\omega = 0.33\pi$ ) which is the worst-case performance under the recommended scheme. The Thiran filter yields a beam that is identical to the perfect delay case. Furthermore, we show in (b) what would happen when the frequency is out of the recommended range, i.e., when  $\omega = 0.8\pi$ , where there is a significant deviation of the beam axis. We only show this for information. In practice, by keeping  $\omega < 0.33\pi$  by selecting  $3 \times$  temporal over-sampling, we can avoid this issue.

## 7. SFGs for the Fast Digital DVM Algorithm Using Thiran Fractional Delays

In this section, we will utilize SFGs to elaborate on Thiran fractional delays for twiddle filter realization. Although a direct-form realization of the digital DVM beamformer [50] can be realized using Thiran filters for realizing each fractional sample delay, we proposed to reduce the computational complexity of the multi-beam beamformer by exploiting sparse factorization of the DVM in [7] followed by the fast DVM algorithm, i.e., *ddvm*. However, the SFGs in [7] are continuous-time and realized as analog microwave circuits.

In this paper, we are operating in the discrete domain, and the resulting SFGs are discrete-time and realizable using DSP systems; to wit, the sparse factorization of the DVM leads to discrete-time SFGs having twiddle filters consisting of digital fractional delay lines that can be approximated using Thiran filter building blocks.

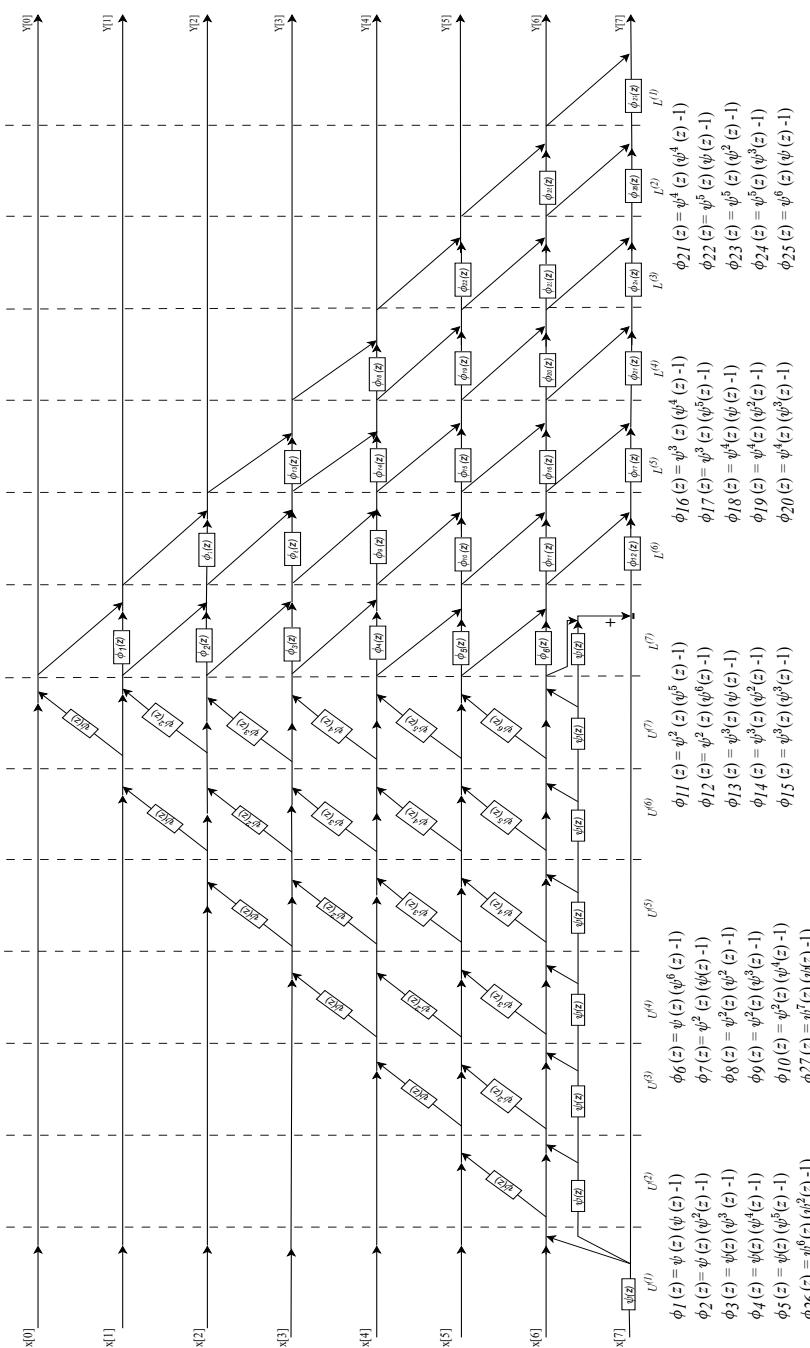
### 7.1. An $N = 4$ Point DVM with Thiran Twiddle-Filters

Figure 8 shows the  $N = 4$  point DVM factorization, where fractional true time delays of the form  $\psi^p(z)$  are realized as a single Thiran filter block, having delay  $p\tau$  where  $\tau = 1/N$ . Figure 9 shows the direct form II SFG of a typical Thiran all-pass filter that approximates a fractional true time delay within the first 60% of the Nyquist period. A temporal over-sampling factor of two easily allows the signals of interest to be located within the usable band of the Thiran filters of the transfer function order  $n_i = 3$  and  $n_i = 4$ , respectively. The twiddle filter-based DVM factorizations also require several special filters denoted as  $\phi_k(z), k = 1, 2, \dots, 5$ , where each special filter is realized using  $\psi(z)$  Thiran filters as a building block. Therefore, once the Thiran block  $\psi(z)$  is available, it can be re-used to create the special twiddle filters  $\phi_k(z)$  as shown in Figure 10.



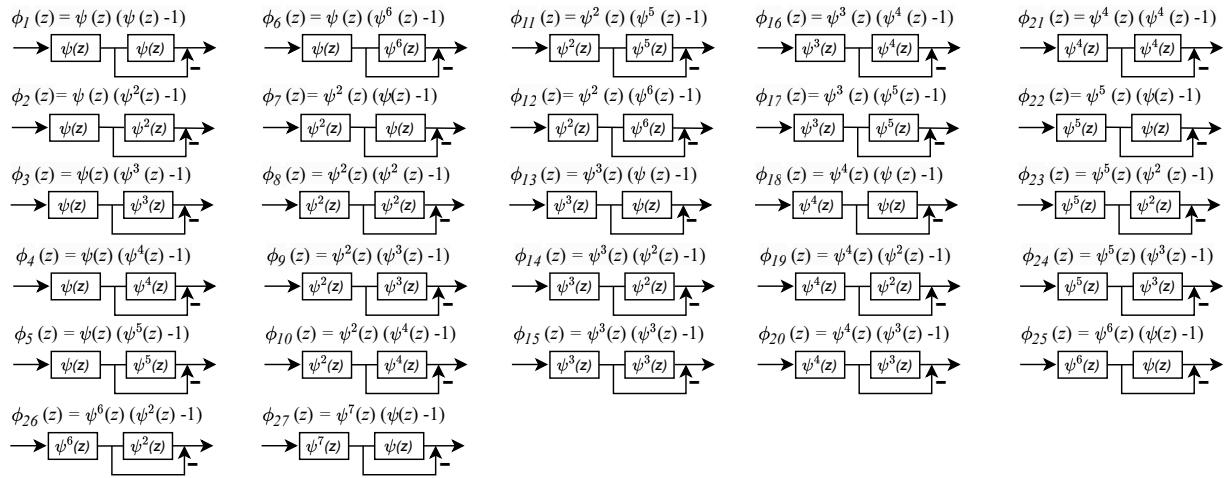
## 7.2. An $N = 8$ Point DVM with Thiran Twiddle-Filters

Figure 11 shows the  $N = 8$  point DVM factorization, where fractional true time delays are of the form  $\psi^p(z)$  as was the case for  $N = 4$ . For  $N = 8$ , the twiddle filters require special filters that also use  $\psi(z)$  Thiran filters as the primary building block. The twiddle filters  $\phi_k(z)$  are realized using  $\psi(z)$  Thiran fractional delay filters as the building blocks. Once a reconfigurable Thiran filter is realized in a hardware design, it can be re-used at scale.



**Figure 11.** SFG for the  $N = 8$ –beam wideband beamformer using the DVM factorization followed by the *ddvm* algorithm having the fractional delays  $\psi(z)$  associated with  $\tau_0$ .

**Figure 10:** Signal flow graph (SFG) for the proposed low-complexity 8-beam wideband beamformer.



**Figure 12.**  $\phi_k(z)$  function architectures for the order  $N = 8$  filter. The twiddle filters are realized using the  $\psi(z)$  Thiran filter hardware core as a building block.

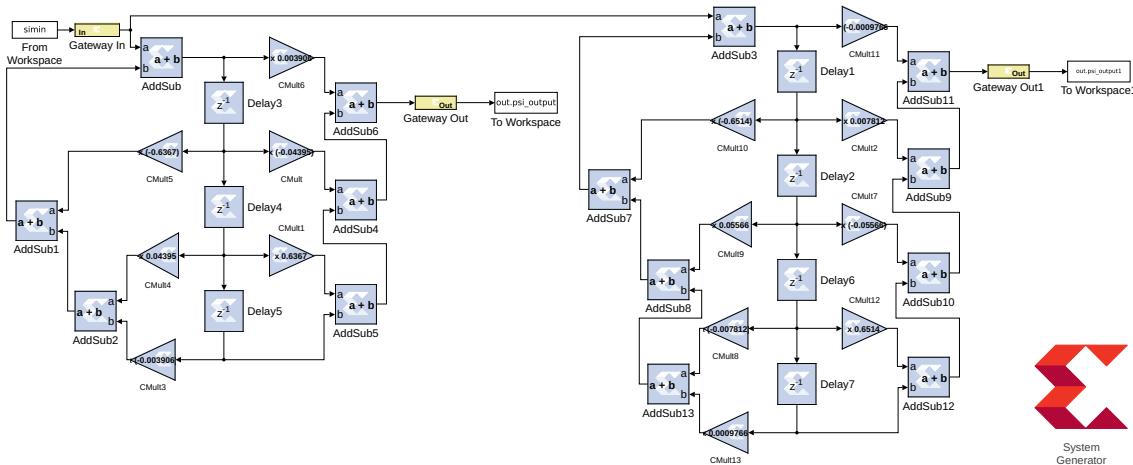
### 7.3. Preliminary FPGA Digital Hardware Architectures

The full realization of the DVM architecture on a digital integrated circuit (IC) is a complex task and is beyond the scope of this paper. Nevertheless, we provide preliminary results toward the eventual realization of the DVM beamformers on digital IC using application-specific integrated circuit (ASIC) technology.

#### 7.3.1. FPGA Realization of Thiran Filter Blocks

We prototyped a single software reconfigurable and delay-tunable digital Thiran filter using field programmable gate array (FPGA) technology. Using Xilinx FPGA design tools, including System Generator and PlanAhead, we realized a Thiran filter as a direct form II

implementation of red with an associated



**Figure 13.** (Left) The direct form II computer architecture for realizing the Thiran filter block realizing  $\psi(z)$  for the order  $n_i = 3$ . (Right) The direct form II computer architecture for realizing the Thiran filter block realizing  $\psi(z)$  for order  $n_i = 4$ . The digital FPGA design uses the Xilinx System Generator FPGA blockset, which is a plug-in to Matlab/Simulink that provides a software interface to the Xilinx Vivado toolset.

### 7.3.2. Synthesis and Mapping to FPGA Fabric

The RF-SoC realization of a Thiran filter for  $n_i = 3, 4$  order APFs are given in Table 1. The designs were ported from Matlab System Generator to Xilinx Vivado for synthesis and post place-and-route timing analysis. The design for order  $n_i = 3$  filter consumes 418 configurable logic block look-up tables (CLB LUTs), 158 CLB registers, and 51 CARRY8 on the Xilinx FPGA reconfigurable logic fabric. The design for the order  $n_i = 4$  Thiran consumes 487 CLB LUTs, 182 CLB registers, and 70 CARRY8 on FPGA logic fabric. The target FPGA device was an RF System-On-Chip (SoC) ZCU-111.

The hardware consumption data in Table 1 indicate the scalability of the Thiran filter building blocks for large apertures having many TTD units in the SFG.

**Table 1.** AMD-Xilinx ZCU-111 RF-SoC realizations of Thiran filters for orders  $n_i = 3$  and  $n_i = 4$ , respectively.

Order	Site Type	Used	Available	Util %
3	CLB LUTs	418	425,280	0.10
	LUT as Logic	410	425,280	0.10
	LUT as Memory	8	213,600	<0.01
	LUT as Distributed RAM	0		
	LUT as Shift Register	8		
	CLB Registers	158	850,560	0.02
4	Registers as Flip Flop	158	850,560	0.02
	Registers as Latch	0	850,560	0.00
	CARRY8	51	53,160	0.10
	CLB LUTs	487	425,280	0.11
	LUT as Logic	471	425,280	0.11
	LUT as Memory	16	213,600	<0.01
4	LUT as Distributed RAM	0		
	LUT as Shift Register	16		
	CLB Registers	182	850,560	0.02
	Registers as Flip Flop	182	850,560	0.02
	Registers as Latch	0	850,560	0.00
	CARRY8	70	53,160	0.13

### 7.3.3. Post Place Route Timing

Designs were pipelined to obtain a critical path delay (CPD) of  $T_{CPD} = n_i T_{A/S} + T_M$ , where  $n_i \in \mathbb{Z}^+$  is the order,  $T_{A/S}$  is the latency for an adder/subtractor, having 14 bits of precision, and  $T_M$  is the latency for a multiplier, having a coefficient precision of 14 bits and requantized output precision of 14 bits, selected to match the precision of the analog-to-digital converters (ADCs) on the RF-SoC. The maximum clock frequency is  $F_{CLK} = 1/T_{CPD}$ . Post place and route timing analysis on Vivado indicates maximum clock speeds of 220 MHz and 213 MHz for Thiran filter orders of  $n_i = 3$  and  $n_i = 4$ , respectively.

## 8. Future Work

We will utilize the APF-based Thiran fractional delay units to realize the full DVM factorizations provided in the paper. The twiddle filters will be realized using the Thiran building blocks. Thereafter, the factorized DVMs will be realized following the respective SFGs. The DVM algorithms will be pipelined using clocked D-flop stages inserted within the feed-forward parts of the signal flow. The building block Thiran filters are IIR filters that cannot be pipelined further without specialized computer architecture techniques; for example, we will explore the scattered/structured look-ahead pipelining [51] for digital processors, which are based on complex pole-zero cancellations. The computer architectures arrived at by mapping the SFGs to pipelined parallel processors with Thiran twiddle filters will be implemented on an AMD-Xilinx RF-SoC platform, such as ZCU-1285/1275/111,

and operated with RF front ends realized off chip to evaluate the multi-beam beamforming performance in a hardware-in-the-loop emulation [52–54].

## 9. Conclusions

The ability to directionally enhance a propagating wideband RF plane-wave signal is a crucial requirement for RF sensing applications, as well as RF imaging, RF communications, and RF-based location systems. The application domains of RF sensing include radar, radio astronomy, and joint-communication and sensing applications that are emerging for next-generation wireless systems (e.g., 6G networks). A multi-beam TTD beamformer allows optimal sensing along a set of pre-determined DOAs for wideband waveforms. We proposed an algorithm for the efficient computation of wideband true time delay digital DVM using Thiran fractional delays. By adopting an array implementation of the DVM algorithm followed by the bidiagonal and diagonal matrices, the complexity of the digital DVM is significantly reduced. By applying this algorithm, the digital DVM becomes more efficient and optimized. Consequently, the algorithm yields a reduction in circuit complexity for multi-beam digital wideband beamforming systems that implement Thiran fractional delays. We implemented the DVM algorithm to effectively demonstrate the phase and group delay responses of Thiran filters that possess fractional sampling period delays of both the third and fourth orders. These filters can be integrated into the Xilinx FPGA RF-SoC technologies. To validate the feasibility of implementing the DVM algorithm, we utilized signal flow graphs as the basis for realizing digital application-specific integrated circuits (ASICs).

**Author Contributions:** Conceptualization, S.M.P. and A.M.; theory, S.M.P.; software, S.M.P. and G.R.; validation, S.M.P., A.M. and G.R.; formal analysis, S.M.P., A.M. and G.R.; investigation, S.M.P. and A.M.; resources, S.M.P. and A.M.; data curation, S.M.P., A.M. and G.R.; writing—original draft preparation, S.M.P., A.M. and G.R.; digital design, A.M. and G.R.; writing—review and editing, S.M.P., A.M. and G.R.; supervision, S.M.P. and A.M.; project administration, S.M.P. and A.M.; funding acquisition, S.M.P. and A.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Science Foundation award numbers 2229471 and 2229473.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Nomenclature

$\mathbf{A}_N$	$N \times N$ DVM
ADC	Analog to digital converter
APF	All-pass filter
ASIC	Application-specific integrated circuit
AWGN	Additive white Gaussian noise
CMOS	Complementary metal-oxide semiconductor
Complex Laplace variable	$s \in \mathbb{C}$
Complex $z$ -variable	$z \in \mathbb{C}$
Complex polynomial of order $n_i$	$p_i(z)$
CLB	Configurable logic block
Critical path delay	$T_{CPD}$
DAC	Digital to analog converter
Delay (integer and fractional parts)	$\tau$

DFT	Discrete Fourier transform
Digital integer delays at node $i$	$P_i$
Digital fractional delay at node $i$	$D_i$
DSP	Digital signal processing
DOA	Directions of Arrival
Discrete-time APF	$\psi(z)$
DVM	Delay Vandemonde matrix
FFT	Fast Fourier transform
FPGA	Field programmable gate array
FIFO	first in first out
FIR	Finite impulse response
IIR	Infinite impulse response
Input vector or signals to DVM	$\mathbf{x}$
Latency of addition/subtraction core	$T_A/D$
Latency of multiplier core	$T_M$
LNA	Low noise amplifier
LPF	Low pass filter
LUT	Look-up table
Nodes of DVM	$\alpha^k = e^{-j\omega\tau_0 k}$
Number of APFs in cascade	$M$
Order of $i$ -th APF	$n_i$
Order of DVM	$N \times N$
Output vectors or beamformed signals	$\mathbf{y}$
Phase function at $(p, q)$	$\alpha_{p,q}$
Phase rotation	$\alpha = e^{-j\omega\tau_0}$
RF	Radio frequency
RAM	Random access memory
Sampling period	$T_s$
SFG	Signal flow graph
SNR	Signal-to-noise ratio
Smallest fractional delay	$\tau_0$
SoC	System on chip
Temporal frequency	$\omega$
The $i$ -th node in the SFG	$i$
Thiran APF at node $i$	$H_i(z)$
Time delay at node $i$	$\tau_i$
TTD	True time delay
Twiddle Filter at node $i$	$\phi_i(z)$
VLSI	Very large scale integration

## References

1. Dalrymple, T.; Buck, J.; Buxa, P.; McCann, J.; Neidhard, R.; Scalzi, G.; Shreffler, C.; Spendley, D.; Watson, P. Transformational Element Level Arrays (TELA) Testbed (Preprint). 2007; p. 27. Available online: <https://api.semanticscholar.org/CorpusID:107691946> (accessed on 28 December 2023).
2. Fulton, C.; Yeary, M.; Thompson, D.; Lake, J.; Mitchell, A. Digital Phased Arrays: Challenges and Opportunities. *Proc. IEEE* **2016**, *104*, 487–503. [[CrossRef](#)]
3. Hoffmann, T.; Fulton, C.; Yeary, M.; Saunders, A.; Thompson, D.; Murmann, B.; Chen, B.; Guo, A. IMPACT—A common building block to enable next generation radar arrays. In Proceedings of the 2016 IEEE Radar Conference (RadarConf), Philadelphia, PA, USA, 2–6 May 2016; pp. 1–4. [[CrossRef](#)]
4. Hu, Z.; Guo, N.; Qiu, R.; Bonior, J.; Liou, L.; Lin, D.; Longbrake, M.; Buxa, P.; Dalrymple, T.; Hary, S.; et al. Robust wideband beamforming. In Proceedings of the IEEE 2010 National Aerospace & Electronics Conference, Dayton, OH, USA, 14–16 July 2010; pp. 60–64. [[CrossRef](#)]
5. Hu, Z.; Guo, N.; Qiu, R.; Bonior, J.; Liou, L.; Lin, D.; Longbrake, M.; Buxa, P.; Dalrymple, T.; Hong, S.; et al. Design of look-up table based architecture for wideband beamforming. In Proceedings of the 2010 International Waveform Diversity and Design Conference, Niagara Falls, ON, Canada, 8–13 August 2010; pp. 000001–000005. [[CrossRef](#)]
6. Zhang, Z.; Wen, F.; Shi, J.; He, J.; Truong, T.K. 2D-DOA Estimation for Coherent Signals via a Polarized Uniform Rectangular Array. *IEEE Signal Process. Lett.* **2023**, *30*, 893–897. [[CrossRef](#)]

7. Perera, S.; Ariyarathna, V.; Udayanga, N.; Madanayake, A.; Wu, G.; Belostotski, L.; Cintra, R.; Rappaport, T. Wideband N-beam Arrays with Low-Complexity Algorithms and Mixed-Signal Integrated Circuits. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 368–382. [\[CrossRef\]](#)

8. Ariyarathna, V.; Udayanga, N.; Madanayake, A.; Perera, S.M.; Belostotski, L.; Cintra, R. Design methodology of an analog 9-beam squint-free wideband IF multi-beamformer for mmW applications. In Proceedings of the IEEE 2017 Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 29–31 May 2017; pp. 236–241. [\[CrossRef\]](#)

9. Perera, S.M.; Madanayake, A.; Cintra, R. Efficient and Self-Recursive Delay Vandermonde Algorithm for Multi-beam Antenna Arrays. *IEEE Open J. Signal Process.* **2020**, *1*, 64–76. [\[CrossRef\]](#)

10. Perera, S.M.; Madanayake, A.; Cintra, R. Radix-2 Self-recursive Algorithms for Vandermonde-type Matrices and True-Time-Delay Multi-Beam Antenna Arrays. *IEEE Access* **2020**, *8*, 25498–25508. [\[CrossRef\]](#)

11. Perera, S.M.; Lingsch, L.; Madanayake, A.; Mandal, S.; Mastronardi, N. A Fast DVM Algorithm for Wideband Time-Delay Multi-Beam Beamformers. *IEEE Trans. Signal Process.* **2022**, *70*, 5913–5925. [\[CrossRef\]](#)

12. Laakso, T.; Valimaki, V.; Karjalainen, M.; Laine, U. Splitting the unit delay [FIR/all pass filters design]. *IEEE Signal Process. Mag.* **1996**, *13*, 30–60. [\[CrossRef\]](#)

13. Cain, G.; Murphy, N.; Tarczynski, A. Evaluation of several variable FIR fractional-sample delay filters. In Proceedings of the ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing, Adelaide, Australia, 19–22 April 1994; Volume 3, pp. III/621–III/624. [\[CrossRef\]](#)

14. Pei, S.C.; Tseng, C.C. An efficient design of a variable fractional delay filter using a first-order differentiator. *IEEE Signal Process. Lett.* **2003**, *10*, 307–310. [\[CrossRef\]](#)

15. Pei, S.C.; Wang, P.H. Closed-form design of allpass fractional delay filters. In Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, FL, USA, 13–17 May 2002; Volume 2, pp. II–1561–II–1564. [\[CrossRef\]](#)

16. Johansson, H.; Lowenborg, P. Reconstruction of nonuniformly sampled bandlimited signals by means of digital fractional delay filters. *IEEE Trans. Signal Process.* **2002**, *50*, 2757–2767. [\[CrossRef\]](#)

17. Tseng, C.C. Design of FIR and IIR Fractional Order Simpson Digital Integrators. *Signal Process.* **2007**, *87*, 1045–1057. [\[CrossRef\]](#)

18. Olkkonen, T.J.; Olkkonen, H. Complex Hilbert Transform Filter. *J. Signal Inf. Process.* **2011**, *2*, 112–116. [\[CrossRef\]](#)

19. Cooley, J.W.; Tukey, J.W. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.* **1965**, *19*, 297–301. [\[CrossRef\]](#)

20. Yavne, R. An economical method for calculating the discrete fourier transform. In AFIPS '68 (Fall, Part I): Proceedings of the December 9–11, 1968, Fall Joint Computer Conference, Part I; Association for Computing Machinery: New York, NY, USA, 1968; pp. 115–125. [\[CrossRef\]](#)

21. Johnson, S.G.; Frigo, M. A modified split-radix fft with fewer arithmetic operations. *IEEE Trans.* **2007**, *55*, 111–119. [\[CrossRef\]](#)

22. Oruc, H.; Phillips, G.M. Explicit factorization of the Vandermonde matrix. *Linear Algebra Its Appl.* **2000**, *315*, 113–123. [\[CrossRef\]](#)

23. Oruc, H.; Akmaz, H.K. Symmetric functions and the Vandermonde matrix. *J. Comput. Appl. Math.* **2004**, *172*, 49–64. [\[CrossRef\]](#)

24. Yang, S.L. On the LU factorization of the Vandermonde matrix. *Discret. Appl. Math.* **2005**, *146*, 102–105. [\[CrossRef\]](#)

25. Gohberg, I.; Olshevsky, V. Complexity of multiplication with vectors for structured matrices. *Linear Algebra Appl.* **1994**, *202*, 163–192. [\[CrossRef\]](#)

26. Gohberg, I.; Olshevsky, V. Fast algorithms with preprocessing for matrix-vector multiplication problems. *J. Complex.* **1994**, *10*, 411–427. [\[CrossRef\]](#)

27. Pan, V.Y. Fast approximate computations with Cauchy matrices and polynomials. *Math. Comput.* **2017**, *86*, 2799–2826. [\[CrossRef\]](#)

28. Boley, D.L.; Luk, F.T.; Vandevoorde, D. A fast method to diagonalize a Hankel matrix. *Linear Algebra Its Appl.* **1998**, *284*, 41–52. [\[CrossRef\]](#)

29. Backstrom, T. Vandermonde factorization of Toeplitz matrices and applications in filtering and warping. *IEEE Trans. Signal Process.* **2013**, *61*, 6257–6263. [\[CrossRef\]](#)

30. Stoica, P.; Moses, R.L. *Spectral Analysis of Signals*; Pearson/Prentice Hall Upper: Saddle River, NJ, USA, 2005.

31. Backstrom, T.; Fischer, J.; Boley, D. Implementation and evaluation of the Vandermonde transform. In Proceedings of the in 2014 22nd European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, 1–5 September 2014.

32. Strang, G. *Introduction to Applied Mathematics*; SIAM, Wesley-Cambridge Press: Philadelphia, PA, USA, 1986.

33. Loan, C.V. *Computational Frameworks for the Fast Fourier Transform*; SIAM Publications: Philadelphia, PA, USA, 1992.

34. Golub, G.; Loan, C.V. *Matrix Computations*, 4th ed.; The Johns Hopkins University Press: Baltimore, MD, USA, 2013.

35. Rao, K.R.; Kim, D.; Hwang, J.J. *Fast Fourier Transform: Algorithm and Applications*; Springer: New York, NY, USA, 2010.

36. Blahut, R.E. *Fast Algorithms for Signal Processing*; Cambridge University Press: Cambridge, UK, 2010.

37. Kailath, T.; Olshevsky, V. Displacement structure approach to polynomial Vandermonde and related matrices. *Linear Algebra Appl.* **1997**, *261*, 49–90. [\[CrossRef\]](#)

38. Pan, V.Y. *Structured Matrices and Polynomials: Unified Superfast Algorithms*; Birkhauser: Boston, MA, USA, 2001.

39. Parker, F. Inverses of Vandermonde matrices. *Am. Math. Mon.* **1964**, *71*, 410–411. [\[CrossRef\]](#)

40. Björck, A.; Pereyra, V. Solution of Vandermonde systems of equations. *Math. Comp.* **1970**, *24*, 893–903. [\[CrossRef\]](#)

41. Reichel, L.; Opfer, G. Chebyshev-Vandermonde systems. *Math. Comput.* **1991**, *57*, 703–721. [\[CrossRef\]](#)

42. Bella, T.; Eidelman, Y.; Gohberg, I.; Koltracht, I.; Olshevsky, V. A Björck-Pereyra-type algorithm for Szegö-Vandermonde matrices based on properties of unitary Hessenberg matrices. *Linear Algebra Appl.* **2007**, *420*, 634–647. [\[CrossRef\]](#)

43. Bella, T.; Eidelman, Y.; Gohberg, I.; Koltracht, I.; Olshevsky, V. A fast bjorck-pereyra-type algorithm for solving hessenberg-quasiseparable-vandermonde systems. *SIAM J. Matrix Anal. Appl.* **2009**, *31*, 790–815. [\[CrossRef\]](#)

44. Perera, S.M. Quasiseparable Approach to Matrices of Vandermonde Type. Ph.D. Thesis, University of Connecticut, Storrs, CT, USA, 2012.

45. Perera, S.M.; Madanayake, A.; Ogle, A.; Silverio, D.; Qi, J. A Fast Algorithm to Solve Delay Vandermonde Systems in Phased-Array Digital Receivers. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2288–2297. [\[CrossRef\]](#)

46. Demmel, J.; Koev, P. The Accurate and Efficient Solution of a Totally Positive Generalized Vandermonde Linear System. *SIAM J. Matrix Anal. Appl.* **2005**, *27*, 142–152. [\[CrossRef\]](#)

47. Olshevsky, V. Fast Algorithms for Structured Matrices: Theory and Applications. In *Contemporary Mathematics*; SIAM: Philadelphia, PA, USA, 2003; Volume 323.

48. Wijenayake, C.; Madanayake, A.; Xu, Y.; Belostotski, L.; Bruton, L.T. A Steerable DC-1 GHz all-pass filter-Sum RF space-time 2-D beam filter in 65 nm CMOS. In Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS 2013), Beijing, China, 19–23 May 2013; pp. 1276–1279. [\[CrossRef\]](#)

49. Ahmadi, P.; Maundy, B.; Elwakil, A.S.; Belostotski, L.; Madanayake, A. A New Second-Order All-Pass Filter in 130-nm CMOS. *IEEE Trans. Circuits Syst. II Express Briefs* **2016**, *63*, 249–253. [\[CrossRef\]](#)

50. Steiner, K.A.; Yeary, M.B. A 1.6-GHz Sub-Nyquist-Sampled Wideband Beamformer on an RFSoC. *IEEE Trans. Radar Syst.* **2023**, *1*, 308–317. [\[CrossRef\]](#)

51. Parhi, K.K. *VLSI Digital Signal Processing Systems: Design and Implementation*; Wiley-Interscience: Hoboken, NJ, USA, 1999.

52. Pulipati, S.; Ariyathenna, V.; Silva, U.D.; Akram, N.; Alwan, E.; Madanayake, A.; Mandal, S.; Rappaport, T.S. A Direct-Conversion Digital Beamforming Array Receiver with 800 MHz Channel Bandwidth at 28 GHz using Xilinx RF SoC. In Proceedings of the 2019 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), Tel Aviv, Israel, 4–6 November 2019; pp. 1–5. [\[CrossRef\]](#)

53. Akram, N.; Madanayake, A.; Venkatakrishnan, S.B.; Volakis, J.L.; Psychogiou, D.; Marzetta, T.L.; Rappaport, T.S. Massive-MIMO and Digital mm-Wave Arrays on RF-SoCs using FDM for M-Fold Increase in Antennas per ADC/DAC. In Proceedings of the 2021 IEEE Space Hardware and Radio Conference (SHaRC), San Diego, CA, USA, 17–22 January 2021; pp. 25–27. [\[CrossRef\]](#)

54. Akram, N.; Madanayake, A.; Pulipati, S.; Ariyathenna, V.; Venkatakrishnan, S.B.; Psychogiou, D.; Volakis, J.; Rappaport, T.S.; Marzetta, T.L. Frequency-Multiplexed Array Digitization for MIMO Receivers: 4-Antennas/ADC at 28 GHz on Xilinx ZCU-1285 RF SoC. *IEEE Access* **2021**, *9*, 142743–142753. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.