









Wideband N -Beam Arrays Using Low-Complexity Algorithms and Mixed-Signal Integrated Circuits

Sirani M. Perera , Viduneth Ariyaratna , *Student Member, IEEE*, Nilan Udayanga, *Student Member, IEEE*, Arjuna Madanayake , *Member, IEEE*, Ge Wu, *Member, IEEE*, Leonid Belostotski , *Senior Member, IEEE*, Yingying Wang , *Student Member, IEEE*, Soumyajit Mandal , *Senior Member, IEEE*, Renato J. Cintra , *Senior Member, IEEE*, and Theodore S. Rappaport , *Fellow, IEEE*

Abstract—This paper proposes a low-complexity wideband beamforming subarray for millimeter wave (mmW) 5G wireless communications. The multibeam subarray is based on using a novel delay Vandermonde matrix (DVM) algorithm to efficiently generate analog true-time-delay beams that have no beam squint. A factorization for the DVM leading to low-complexity analog realizations is provided and complexity analysis for real and complex inputs is derived. The DVM is a special case of a Vandermonde matrix but with complex nodes that lack any special properties (unlike the discrete Fourier transform matrix). Error bounds for the DVM are established and then analyzed for numerical stability. Mixed-signal CMOS integrated circuits designs are proposed for the implementation of DVM multibeam algorithms along with low-complexity digital realizations to achieve hybrid beamforming for mmW applications. Analog–digital hybrid mmW multibeam beamforming circuits and systems are designed, for example, with eight beams at 28 GHz and simulated in cadence for functional verification.

Index Terms—Delay Vandermonde matrix, wideband beamforming, low-complexity algorithm, 5G multibeam arrays.

I. INTRODUCTION

FIFTH generation (5G) wireless communications are based on millimeter-wave (mmW) channels, which are dominated by scattering and reflection from objects that are larger

Manuscript received September 14, 2017; revised January 24, 2018; accepted March 19, 2018. Date of publication April 9, 2018; date of current version May 22, 2018. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Constantinos B. Papadias. (*Corresponding author: Viduneth Ariyaratna.*)

S. M. Perera is with the Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114-3900 USA (e-mail: pereras2@erau.edu).

V. Ariyaratna, N. Udayanga, and A. Madanayake are with the Department of Electrical and Computer Engineering, University of Akron, Akron, OH 44325 USA (e-mail: bpv1@zips.uakron.edu; gnu1@zips.uakron.edu; arjuna@uakron.edu).

G. Wu and L. Belostotski are with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: gwu@ucalgary.ca; lbelosto@ucalgary.ca).

Y. Wang and S. Mandal are with the Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106 USA (e-mail: yxw788@case.edu; sxm833@case.edu).

R. J. Cintra is with the Signal Processing Group, Departamento de Estatística, UFPE, Recife 50670-901, Brazil, and also with the Department of Electrical and Computer Engineering, University of Calgary, Calgary, AB, Canada (e-mail: rjds@de.ufpe.br).

T. S. Rappaport is with NYU WIRELESS in the Department of Electrical and Computer Engineering, New York University, New York, NY 10003 USA (e-mail: tsr@nyu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTSP.2018.2822940

than the wavelength (e.g., vehicles, people, buildings) and are thus more directional than the microwave channels used in existing sub-6 GHz cellular bands [1]. Moreover, mmW channels have high free-space path loss in the first meter of propagation (which increases as the square of frequency as predicted by the Friis formula) and also suffer from additional heavy attenuation due to weather (water droplets from rain, fog/hail, etc.). Such attenuation can be compensated by increasing antenna gain; in particular, by using arrays that form mmW radio frequency (RF) beams. The resulting increase in signal-to-noise ratio (SNR) improves channel capacity and throughput without requiring more transmit power [2]. Beamforming also allows mmW multi-input multi-output (MIMO) communication systems to further increase link capacity by exploiting the spatial multiplexing and/or diversity provided by multiple propagation paths (due to scattering, reflection, and wave-guiding) present in indoor and urban environments. Thus, the ability to form multiple sharp mmW beams is essential for taking advantage of typical mmW channels [3], [4].

Here we propose a low-complexity hybrid (analog-digital) mmW multi-beam beamforming architecture that does not suffer from the well-known frequency-dependent beam angle problem (also known as “squinting”) encountered in phased-array apertures. As a result, the proposed architecture can be used for beamforming of wideband 5G signals. The architecture is based on a sparse factorization of the N -beam true-time delay (TTD) beamformer, which takes the form of a DVM. This allows it to be efficiently realized at mmW using analog circuits; in particular, CMOS all-pass filters (APFs).

The rest of the paper is organized as follows. Section II introduces multi-beam beamforming networks and the beam squint problem. Section III defines the proposed DVM algorithm, while Sections IV and V analyze its arithmetic complexity, error bounds, and stability. An analog IC design of the proposed architecture is described in Section VI, while Sections VII and VIII focus on implementations of the level-1 analog beamformer and level-2 digital beamformer, respectively. Finally, Section IX concludes the paper.

II. MULTI-BEAM BEAMFORMERS

A. FFT Beamforming and the Beam Squint Problem

Performing a spatial fast Fourier transform (FFT) across an N -element uniform linear array leads to N orthogonal RF beams

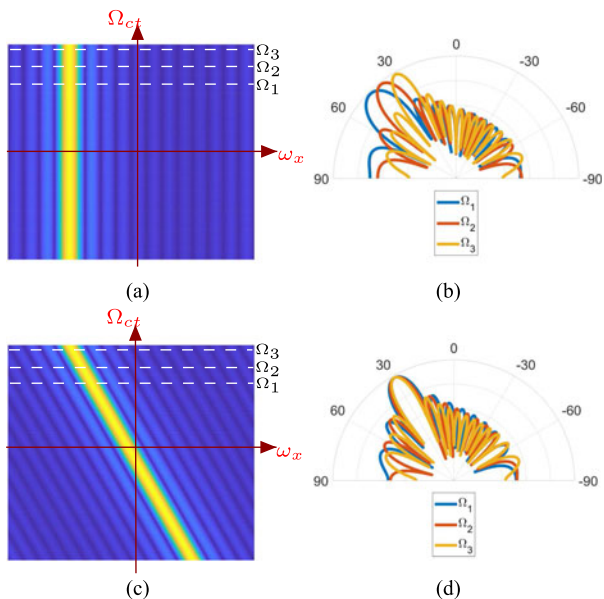


Fig. 1. (a) Frequency response of a 16-point FFT-based beamformer (bin 4), and (b) its array factors, showing squint; (c) DVM beam for bin 8, and (d) its squint-free array factors. In (a) and (b), Ω_{ct} is the wave speed normalized temporal (continuous-time) frequency and ω_x refer to sampled spatial frequency.

at an upper bound of $\mathcal{O}(N \log N)$ computational complexity. However, FFT-based beams suffer from “beam squint”, i.e., the beam directions are strongly dependent on the temporal frequency of operation (see Fig. 1(a)). This phenomenon is fundamental; it occurs because the twiddle factors (complex weights) are narrow-band frequency independent and do not represent the TTDs required for wideband beamforming. Such frequency-independent complex valued weights are simpler to implement than TTDs (e.g., using phase-shifters) but cause the look direction to become strongly frequency-dependent over wideband channels.

The “24 GHz” (24.25–27.5 GHz) and “28 GHz” (26.5–29.5 GHz) portions of the mmW spectrum have recently become available for licensed communications on several continents [5], and are likely to be the initial targets for 5G deployment. Thus, 5G radios should support real-time bandwidths of > 1 GHz in order to take full advantage of mmW spectrum availability. In particular, they should use wideband (i.e., squint-free) multi-beamformers. However, wideband TTD beamformers do not have the $\mathcal{O}(N \log N)$ complexity typical of FFT-based beamformers. In fact, TTD N -beam networks are of complexity $\mathcal{O}(N^2)$. Thus, such beamformers are not feasible even for moderately large N , thus entailing the development of FFT-like wideband multi-beam beamforming algorithms.

B. Hybrid Beamforming Systems

Digital beamforming (DBF) delivers maximum flexibility including multiple beams [6], high dynamic range, and high accuracy using digital calibration [7], [8]. However, DBF requires one RF chain and two analog-to-digital converters (ADCs) per antenna element (assuming I-Q downconversion), i.e., P RF chains and $2P$ ADCs for P antenna elements. This results in high power consumption because of the large number of ADCs,

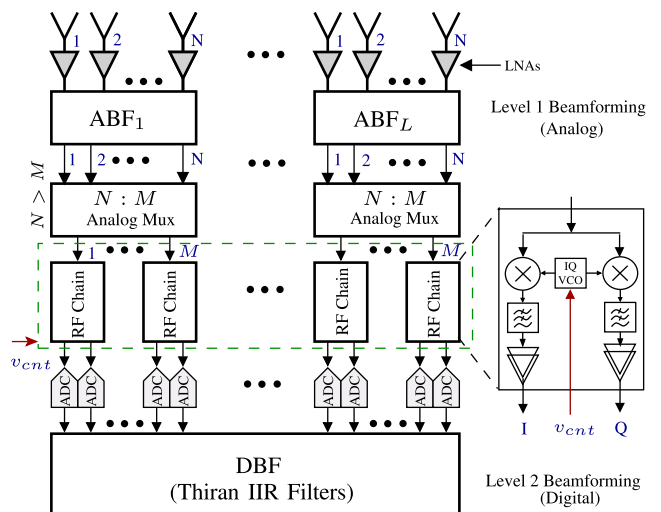


Fig. 2. System architecture of the proposed hybrid squint-free multi-beam network.

which are usually the most power-hungry blocks in mmW receivers [9]. Moreover, the real-time signal processing required to generate beams from the digitized data consumes a large amount of additional power, making large-scale DBF implementations (e.g., for $P = 64$ or 128 elements) impractical at mmW [10], [12], and [13].

Hybrid-beamforming [11] addresses this challenge by combining low-dimensional digital beamformers (at baseband) with analog beamformers (at RF). Such architectures can achieve performance similar to fully-digital schemes at lower cost and power. They typically use RF phase-shifters, TTDs, or lenses for level-1 analog beamforming [11]–[13] and baseband digital processing for level-2 beamforming. We propose the optimized hybrid beamforming architecture shown in Fig. 2 for 5G mmW base stations.

Consider a P -element antenna array consisting of L N -element sub-arrays where $P = LN$ and inter-element spacing Δx is set to $\frac{\lambda_{\min}}{2}$. Here λ_{\min} is the wavelength corresponding to the maximum operating frequency, i.e., $f_0 + B/2$ where f_0 is the center frequency and B is the signal bandwidth. Each antenna output is followed by i) a low-noise amplifier (LNA) to minimize system noise figure (NF), and ii) level-1 analog beamformers (ABFs). The ABFs use a wideband squint-free multi-beam algorithm realized at RF to generate level-1 beams. Analog $N : M$ multiplexers then select $M \leq N$ outputs from each ABF. These signals are I-Q downconverted and amplified by LM parallel RF chains, and digitized by $2LM$ ADCs. The digitized baseband signals are further processed by a digital beamformer (DBF) to generate narrow level-2 beams. Since $M < N$, the proposed hybrid architecture reduces the total number of RF chains and ADCs by a factor of M/N compared to a fully-digital beamformer. This is because the digital system picks M analog beams from the N available from each subarray for subsequent processing. The choice of how many are needed is proportional to the capacity of the system. If the application only requires a single channel, then M can be as small as unity. On the other hand, for maximum capacity, the system needs to exploit all available beams (for example, if it is used in a

base station or access point). In such a situation, one may digitize all N beams, so $M = N$. Therefore, the ratio lies between $1 \geq M/N \geq 1/N$ where M and N are both integers. When a large number of beams is not required, the number of ADC channels are $2LM \ll 2LN$.

C. Squint-Free Multi-Beam Beamformers

Consider a wideband analog TTD N -beam scheme having N antennas that form beams at θ_l , $l = 1, 2, \dots, N$, from the array broadside. If θ_l s are chosen such that $\theta_l = \sin^{-1} \frac{c\tau l}{\Delta x}$, where τ is the smallest TTD delay implemented in the system and c is the wave speed, we can express the l th beamformer as $H_l(e^{j\omega_x}, \Omega_t, \theta_l) = \sum_{k=0}^{N-1} e^{-jk(\omega_x + \Omega_t \tau l)}$ where $j^2 = -1$ and Ω_t and ω_x represent temporal (continuous time) and spatial frequency variables, respectively. Further, the N -beam multi-beam beamformer arising from $H_l(e^{j\omega_x}, \Omega_t, \theta_l)$ can be expressed as a product of two matrices, such that $\mathbf{y}_m = \mathbf{A}_N \mathbf{D}_N \mathbf{w}_m$. Here, $\mathbf{w}_m = [w_m(1, j\Omega_t) \ w_m(2, j\Omega_t) \ \dots \ w_m(N, j\Omega_t)]^T$ is the input vector (from N antennas) and $\mathbf{y}_m = [y_m(1, j\Omega_t) \ y_m(2, j\Omega_t) \ \dots \ y_m(N, j\Omega_t)]^T$ is the output vector in the Fourier domain. Also \mathbf{D}_N is the $N \times N$ diagonal matrix having elements $b_{kk} = e^{-jk\omega_x}$, and \mathbf{A}_N is a Vandermonde matrix having elements $m_{kl} = \alpha^{kl}$ for $\alpha = e^{-j\Omega_t \tau}$. The term α accounts for the phase rotation associated with the delay τ at frequency f , where $\Omega_t = 2\pi f$.

The transform matrix \mathbf{A}_N is henceforth denoted as the DVM. Moreover, the multi-beam system denoted by $\mathbf{A}_N \mathbf{D}_N$ provides wideband squint-free beams. Note that the DVM is a Vandermonde matrix having nodes $\{\alpha, \alpha^2, \dots, \alpha^N\}$, where $\alpha \in \mathbb{C}$. The DFT matrix is also a Vandermonde matrix having N nodes that are the N th roots of unity. Although the DVM is also a complex Vandermonde-structured matrix, unlike the DFT it has no nice properties like unitarity, periodicity, symmetry, or circular shift.

The DFT matrix, whose elements are given by ω_N^{kn} , $n, k = 0, 1, \dots, N-1$, where $\omega_N = \exp(-j\frac{2\pi}{N})$ is the N th root of unity, is a special case of the DVM. The DFT matrix follows from DVM by setting $f = \frac{1}{\Delta T}$ leading to $\alpha = e^{-j2\pi \frac{1}{\Delta T} \frac{\Delta T}{N}} = e^{-j\frac{2\pi}{N}}$ which is the N th primitive root of unity. However, the use of the DFT matrix instead of the DVM requires replacing the wideband delays $e^{-j\Omega_t \tau}$ with frequency-independent complex multiplications $e^{-j\frac{2\pi kn}{N}}$. This is the root cause of the frequency-dependent beam shape (i.e., the beam squint problem) associated with wideband spatial FFT beamformers.

III. FACTORIZED DVM ALGORITHMS

In general let $\mathbf{M} \in \mathbb{C}^{N \times N}$ be a matrix determined by N^2 entries. Computation of $\mathbf{M}\mathbf{z}$ for the input vector $\mathbf{z} \in \mathbb{C}^N$ costs $\mathcal{O}(N^2)$ operations, with N^2 complex multiplications and complex $N(N-1)$ additions. Fortunately, the structure of \mathbf{M} , e.g., banded, Toeplitz, Hankel, Bezoutian, Cauchy, Vandermonde, Quasiseparable, DFT, etc., can be exploited to reduce the computational cost. The DVM is a Vandermonde structured matrix with complex entries. Thus, the structure of the DVM can also be used to state a sparse factorization and also to derive a ‘‘novel’’ fast and stable DVM algorithm while reducing the arithmetic complexity leading to lower hardware complexity in circuit

realizations. Moreover, as shown in Sections VI, VII, and VIII the proposed DVM algorithm leads to reduce the hardware cost and to solve the longstanding ‘‘beam squint’’ problem.

From [14], [15], if a Vandermonde matrix \mathbf{V} is of the form $\mathbf{V} = [x_i^k]_{i,k=0}^N$ where $x_0, x_1, \dots, x_N \in \mathbb{R}$ (real entries), then it can be factored into the product of 1-banded upper and lower triangular matrices with division entries by using complete symmetric functions. Later, [16] established a simple approach for the factorization of \mathbf{V}^T into the product of 1-banded upper and lower triangular matrices without implementing division entries. Works in [14]–[16] have no information about the complexity, error bounds, and stability of the proposed algorithms. Nevertheless one can extend the results in [14]–[16] to derive a low complexity algorithm using bidiagonal factorizations of a Vandermonde matrix. A $\mathcal{O}(N \log N)$ complexity algorithm to compute Vandermonde matrices having distinct prime nodes via interpolation was shown in [17], while [18] showed that the product of a complex Vandermonde matrix and a vector can be computed using $\mathcal{O}(N \log N)$ complex arithmetic operations by i) using its low rank displacement structure, and ii) pre-computing the generators. Moreover, [19] showed that the cost of multiplying a Vandermonde matrix by a vector is nearly linear in time by approximating the former with semiseparable matrices, i.e., the subclass of quasiseparable matrices. All the results in [14]–[19] are established for the real nodes such that $x_i \in \mathbb{R}$ of the matrix \mathbf{V} except [14], but even [14] has restriction for a generator of the complex Vandermonde matrix. Here we propose a fast algorithm that is based on complex nodes without considering quasiseparability and displacement equations. Rather, our approach is based on reducing the $\mathcal{O}(N^2)$ arithmetic complexity.

In the following, we provide a sparse factorization for the DVM \mathbf{A}_N over complex nodes $\alpha, \alpha^2, \dots, \alpha^N$.

Proposition 3.1: The Delay Vandermonde Matrix $\mathbf{A}_N = [\alpha^{kl}]_{k=1, l=0}^{N, N-1}$, where $N (\geq 4)$ is an integer, having $\{\alpha, \alpha^2, \dots, \alpha^N\} \in \mathbb{C}$ can be factored into

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

where for $1 \leq m \leq N-1$

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

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

and

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

Algorithm III.3: $\text{dvm}(N, \alpha, \mathbf{z})$.

Input: $N(\geq 4)$ -integer, $\alpha \in \mathbb{C}$, and $\mathbf{z} \in \mathbb{R}^n$ or \mathbb{C}^n .

- 1) Set $\mathbf{x} = [\alpha, \alpha^2, \dots, \alpha^N]^T$, $\mathfrak{L}^{(1)} = \mathbf{I}_N$, and $\mathfrak{U}^{(1)} = \mathbf{I}_N$
- 2) For $1 \leq m \leq N - 1$, $1 \leq l \leq N$, and $1 \leq k \leq N$
 - a) Calculate $\mathbf{L}^{(m)}$'s
 - If $l = k$
 - If $l \leq N - m$; $\mathbf{L}^{(m)}(l, k) = 1$
 - Else; $\mathbf{L}^{(m)}(l, k) = \mathbf{x}(l) - \mathbf{x}(n - m)$
 - If $l = k + 1$
 - If $l \geq N - m + 1$; $\mathbf{L}^{(m)}(l, k) = 1$
 - Else; $\mathbf{L}^{(m)}(l, k) = 0$
 - Else
 - $\mathbf{L}^{(m)}(l, k) = 0$
 - b) Calculate $\mathbf{U}^{(m)}$'s
 - If $l = k$
 - $\mathbf{U}^{(m)}(l, k) = 1$
 - If $l + 1 = k$
 - If $k \geq N - m + 1$; $\mathbf{U}^{(m)}(l, k) = \mathbf{x}(k - N + m)$
 - Else; $\mathbf{U}^{(m)}(l, k) = 0$
 - Else
 - $\mathbf{U}^{(m)}(l, k) = 0$
- 3) For $1 \leq m \leq N - 1$
 - a) Calculate the product of $\mathbf{L}^{(m)}$'s
 - $\mathfrak{L}^{(m+1)}(l, k) = \mathfrak{L}^{(m)}(l, k) \cdot \mathbf{L}^{(m)}(l, k)$
 - b) Calculate the product of $\mathbf{U}^{(m)}$'s
 - $\mathfrak{U}^{(m+1)}(l, k) = \mathbf{U}^{(m)}(l, k) \cdot \mathfrak{U}^{(m)}(l, k)$
- 4) Take $\mathbf{L}^{(N)} \cdot \mathbf{U}^{(N)}$

Output: $\mathbf{A}_N \mathbf{z} = \mathfrak{L}^{(N)} \cdot \mathfrak{U}^{(N)} \mathbf{z}$.

Proof: This follows immediately from [16, Th. 3] when real nodes x_i are replaced with complex nodes α^{i+1} and swapping i and j . ■

Remark 3.2: Although the factorization for the DVM can be stated as in Proposition 3.1, we should recall here that the classical Vandermonde matrix \mathbf{V} is extremely ill-conditioned. In fact, the conditional number of the matrix \mathbf{V} grows exponentially with the size [20]–[22]. In Section V, we show how the error bound of the proposed DVM algorithm can be changed in terms of the choices for nodes.

Following the factorization for the DVM, we provide a novel DVM algorithm to compute $\mathbf{A}_N \mathbf{z}$ as stated next.

Remark 3.3: In general, to compute the product of a DVM with a vector, one has to use N^2 complex entries of the DVM $\mathbf{A}_N = [\alpha^{kl}]_{k=1, l=0}^{N, N-1}$. However, Algorithm III.3 uses only the second column of the matrix \mathbf{A}_N , i.e., N complex entries as opposed to N^2 complex entries. This significant reduction in complexity plays a key role in the stability of the algorithm, as we show in Section V.

IV. ARITHMETIC COMPLEXITY OF THE DVM ALGORITHM

The number of additions and multiplications required to carry out a computation is called the arithmetic complexity. In this section, the proposed DVM Algorithm III.3 is used to establish the arithmetic complexity with real and complex inputs. Before

deriving addition and multiplication counts of the proposed algorithm, we derive the complexity of the direct computation of DVM by a vector \mathbf{z} where $\mathbf{z} \in \mathbb{R}^n$ or \mathbb{C}^n . For analog circuit implementation, the direct computation of $\mathbf{A}_N \mathbf{z}$ where $\mathbf{z} \in \mathbb{R}^n$, costs $N(N-1)^2(N+2)$ real multiplications and $\frac{N^2(N^2-1)}{2}$ real additions. Similarly, when $\mathbf{z} \in \mathbb{C}^n$, the analog circuit implementation costs $\frac{N(N^3+3N-4)}{4}$ complex multiplications (counted as multipliers) and $N(N-1)$ complex additions (counted as adders) to compute $\mathbf{A}_N \mathbf{z}$. Unlike in a direct product of a matrix (having complex entries) and a vector (having real or complex entries), these counts are obtained by considering α 's with the corresponding multiplicities.

When the direct computation of DVM by a real vector is considered, the multiplications of two α 's are implemented with four real multiplications and two real additions and we continue the process going through Steps 1–4 with respect to the multiplicities of α in Algorithm III.3. Thus, constructing entries of the DVM directly costs $N(N-1)(N^2+N-4)$ real multiplications. This is calculated by using four times the sum of $N+2$ different arithmetic series having first terms a_i and common differences d_i : $N-2$ terms with $\{a_1=1, d_1=1\}$, $N-1$ terms with $\{a_2=1, d_2=2\}$, $\{a_3=2, d_3=3\}, \dots, \{a_N=N-1, d_N=N\}$, and N terms with $\{a_{N+1}=1, d_{N+1}=1\}$ and $\{a_{N+2}=-2, d_{N+2}=0\}$. Once the entries of DVM are calculated, multiplying the DVM by a real vector costs $2N(N-1)$ real multiplications. Hence the direct computation of the DVM by a vector (with real entries) costs a total of $N(N-1)^2(N+2)$ real multiplications. Since we consider α 's with multiplicities and taking two additions for the multiplications of two α 's into account, construction of the DVM costs $\frac{N(N-1)(N^2+N-4)}{2}$ real additions. This is calculated by using twice the sum of $N+2$ different arithmetic series having first terms a_i and common differences d_i : $N-2$ terms with $\{a_1=1, d_1=1\}$, $N-1$ terms with $\{a_2=1, d_2=2\}$, $\{a_3=2, d_3=3\}, \dots, \{a_N=N-1, d_N=N\}$, and N terms with $\{a_{N+1}=1, d_{N+1}=1\}$ and $\{a_{N+2}=-2, d_{N+2}=0\}$. Once these entries are calculated, to multiply DVM by a real vector costs $2N(N-1)$ real additions. Hence the direct computation of the DVM by a real vector costs a total of $\frac{N^2(N^2-1)}{2}$ real additions.

When the direct computation of DVM by a complex vector is considered, each α is counted as a multiplier (simply counting multiplicities of each α). Thus, to construct entries of the DVM requires $\frac{N^2(N^2-1)}{4}$ multipliers (calculated by using the product of two arithmetic series having the first term as 1 and common difference as 1 with terms N and $N-1$ respectively). Once these multipliers are implemented in hardware circuits, to compute the DVM by a complex input requires $N(N-1)$ multipliers. Hence the direct computation of the DVM by a vector (with complex entries) requires a total of $\frac{N(N^3+3N-4)}{4}$ multipliers. There are no adders needed to construct DVM (as each α is counted as a multiplier, there are no complex addition entries). Hence the direct computation of the DVM by a complex vector requires a total of $N(N-1)$ adders.

A. Complexity Computation of the Proposed Algorithm

We now obtain the arithmetic complexity of the proposed DVM algorithm to compute $\mathbf{A}_N \mathbf{z}$ with arbitrary $\mathbf{z} \in$

\mathbb{R}^n (having real input) and $\mathbf{z} \in \mathbb{C}^n$ (having complex input). Let us denote the real/complex number of additions and multiplications required to compute a length N DVM Algorithm III.3 by $\#a\mathbb{R}(DVM, N)/\#a\mathbb{C}(DVM, N)$ and $\#m\mathbb{R}(DVM, N)/\#m\mathbb{C}(DVM, N)$, respectively. Note that i) we do not count multiplication by ± 1 , and ii) the complex counts are in terms of adders and multipliers.

Lemma 4.1: For a given integer $N(\geq 4)$, the arithmetic complexity (counting the multiplicities for α 's) for computing the DVM algorithm $\text{dvm}(N, \alpha, \mathbf{z})$ using Algorithm III.3 having real input ($\mathbf{z} \in \mathbb{R}^n$) is given by

$$\begin{aligned}\#a\mathbb{R}(DVM, N) &= 3N(N-1), \\ \#m\mathbb{R}(DVM, N) &= 4N(N-1).\end{aligned}\quad (2)$$

Proof: To calculate the vector \mathbf{x} in Step 1 of the DVM Algorithm III.3 requires $N(N-1)$ real addition counts and $2N(N-1)$ real multiplication counts (counts are computed based on the multiplication of α 's having four real multiplications and two real additions). In Step 2 of the DVM algorithm, we have computed $\mathbf{L}^{(m)}$'s and $\mathbf{U}^{(m)}$'s for each $m = 1, 2, \dots, N-1$. In Step 2, there is no real multiplication count to compute any $\mathbf{L}^{(m)}$ and $\mathbf{U}^{(m)}$ (we only access the computed entries in \mathbf{x}) but only real addition counts to compute each $\mathbf{L}^{(m)}$. Computing each $L^{(m)}$ costs $2m$ real additions. Thus to compute $\mathbf{L}^{(m)}$'s for $m = 1, 2, \dots, N-1$ costs $N(N-1)$ real additions. After precomputing the entries of $\mathbf{L}^{(m)}$'s and $\mathbf{U}^{(m)}$'s (as in 1, 2(a), and 2(b) of Algorithm III.3), we calculate the products of $\mathbf{L}^{(m)}$'s and $\mathbf{U}^{(m)}$'s at $\mathbf{z} \in \mathbb{R}^n$ to obtain:

$$\begin{aligned}\#a\mathbb{R}(DVM, N) &= \#a\mathbb{R}\left(\sum_{m=1}^{N-1} \mathbf{L}^{(m)}\right) + \#a\mathbb{R}\left(\sum_{m=1}^{N-1} \mathbf{U}^{(m)}\right), \\ \#m\mathbb{R}(DVM, N) &= \#m\mathbb{R}\left(\sum_{m=1}^{N-1} \mathbf{L}^{(m)}\right) + \#m\mathbb{R}\left(\sum_{m=1}^{N-1} \mathbf{U}^{(m)}\right).\end{aligned}\quad (3)$$

Using the structures of $\mathbf{L}^{(m)}$ and $\mathbf{U}^{(m)}$ for $\mathbf{z} \in \mathbb{R}^n$ yields:

$$\begin{aligned}\#a\mathbb{R}\left(\mathbf{L}^{(m)}\right) &= m, & \#m\mathbb{R}\left(\mathbf{L}^{(m)}\right) &= 2m, \\ \#a\mathbb{R}\left(\mathbf{U}^{(m)}\right) &= m, & \#m\mathbb{R}\left(\mathbf{U}^{(m)}\right) &= 2m.\end{aligned}\quad (4)$$

Substituting (4) into (3) and simplifying by precomputing real multiplication and addition counts in Steps 1 and 2 of Algorithm III.3 yields the number of real additions and multiplications for real inputs, as shown in (2). ■

Lemma 4.2: For a given integer $N(\geq 4)$, the arithmetic complexity (counting the multiplicities for α 's) for computing the DVM algorithm $\text{dvm}(N, \alpha, \mathbf{z})$ using Algorithm III.3 with complex input ($\mathbf{z} \in \mathbb{C}^n$) is given by

$$\begin{aligned}\#a\mathbb{C}(DVM, N) &= 3N(N-1)/2, \\ \#m\mathbb{C}(DVM, N) &= N(3N-1)/2.\end{aligned}\quad (5)$$

Proof: To calculate the vector \mathbf{x} in Step 1 of the DVM Algorithm III.3, one has to use $\frac{N(N+1)}{2}$ multipliers but no

TABLE I
ADDITIONS REQUIRED BY THE DVM ALGORITHM

N	Real Input		Complex Input	
	Direct	$\#a\mathbb{R}(DVM, N)$	Direct	$\#a\mathbb{C}(DVM, N)$
4	120	36	12	18
8	2016	168	56	84
16	32640	720	240	360
32	523776	2976	992	1488

TABLE II
MULTIPLICATIONS REQUIRED BY THE DVM ALGORITHM

N	Real Input		Complex Input	
	Direct	$\#m\mathbb{R}(DVM, N)$	Direct	$\#m\mathbb{C}(DVM, N)$
4	216	48	72	22
8	3920	224	1064	92
16	64800	960	16560	376
32	1045568	3968	262880	1520

adders (we have computed each α as a multiplier with the corresponding multiplicity of its power). In Step 2 of the DVM Algorithm III.3, $\mathbf{L}^{(m)}$'s and $\mathbf{U}^{(m)}$'s have to be computed for each $m = 1, 2, \dots, N-1$. In Step 2, there is no multiplier required (we only access the computed entries of \mathbf{x}) but we use adders to construct each $\mathbf{L}^{(m)}$. To compute each $\mathbf{L}^{(m)}$ one has to use m adders. Thus to compute $\mathbf{L}^{(m)}$'s for $m = 1, 2, \dots, N-1$ requires $\frac{N(N-1)}{2}$ adders. After precomputing the entries of $\mathbf{L}^{(m)}$'s and $\mathbf{U}^{(m)}$'s (as in 1, 2(a), and 2(b) of Algorithm III.3), we calculate the products of $\mathbf{L}^{(m)}$'s and $\mathbf{U}^{(m)}$'s at $\mathbf{z} \in \mathbb{C}^n$ to obtain:

$$\begin{aligned}\#a\mathbb{C}(DVM, N) &= \#a\mathbb{C}\left(\sum_{m=1}^{N-1} \mathbf{L}^{(m)}\right) + \#a\mathbb{C}\left(\sum_{m=1}^{N-1} \mathbf{U}^{(m)}\right), \\ \#m\mathbb{C}(DVM, N) &= \#m\mathbb{C}\left(\sum_{m=1}^{N-1} \mathbf{L}^{(m)}\right) + \#m\mathbb{C}\left(\sum_{m=1}^{N-1} \mathbf{U}^{(m)}\right).\end{aligned}\quad (6)$$

Using the structures of $\mathbf{L}^{(m)}$ and $\mathbf{U}^{(m)}$ for $\mathbf{z} \in \mathbb{C}^n$ yields:

$$\begin{aligned}\#a\mathbb{C}\left(\mathbf{L}^{(m)}\right) &= m, & \#m\mathbb{C}\left(\mathbf{L}^{(m)}\right) &= m, \\ \#a\mathbb{C}\left(\mathbf{U}^{(m)}\right) &= m, & \#m\mathbb{C}\left(\mathbf{U}^{(m)}\right) &= m.\end{aligned}\quad (7)$$

Substituting (7) into (6) and simplifying by precomputing multipliers and adders in Steps 1 and 2 of Algorithm II.3 yields the number of adders and multipliers in computing the DVM Algorithm III.3 with complex input as shown in (5). ■

B. Numerical Results for the Complexity Algorithm II.3

Here we provide numerical results for the addition and multiplication complexity of the proposed DVM algorithm $\text{dvm}(N, \alpha, \mathbf{z})$ having real and complex inputs as opposed to a direct computation (see Direct in Tables I and II). Although the algorithm is valid for any even $N(\geq 4)$, for brevity we chose the number of antenna elements to be 2^k ; $k = 2, \dots, 5$ to summarize numerical results as given in Tables I and II.

As shown in Table I, the addition counts for complex inputs are not improved by the proposed Algorithm III.3 when compared to the direct computation of DVM. This is because the addition complexity of the proposed DVM Algorithm III.3 by a complex vector is $3N(N-1)/2$ and the direct computation of the DVM by a complex vector is $N(N-1)$. However, there is significant reduction of adders for real inputs. Also, Table II shows that both the real and complex multiplication counts of the algorithm are much lower than the direct computation.

V. ERROR BOUND AND STABILITY OF THE ALGORITHM III.3

A. Theoretical Analysis

On-chip process, voltage, and temperature (PVT) variations invariably lead to numerical errors while implementing fast algorithms using analog circuits and are a major concern at mmW frequencies. The DVM factorization should be numerically stable enough to obtain acceptable performance with such variations. Hence we explore the numerical stability of the algorithm under perturbation of the coefficients.

We use the perturbation of the product of matrices (stated in [23]) to compute the error bound of the DVM Algorithm III.3. As in Algorithm III.3, we have to compute weights α^k for $k = 1, 2, \dots, N$. The values of α^k affect the accuracy of the DVM algorithm. Thus, we will assume that perturbed computed weights $\hat{\alpha}^k$ are used, and satisfy for all $k = 1, 2, \dots, N$

$$\hat{\alpha}^k = \alpha^k + \epsilon_k, \quad |\epsilon_k| \leq \mu, \quad (8)$$

where $\mu = cu$, $\mu = cu \log k$, and $\mu = cuk$ where u is the unit roundoff and c is a constant based on the method specified in [24].

Consider the perturbation of the product of matrices stated in [23, Lemma 3.7] i.e., if $\mathbf{A}_k + \Delta \mathbf{A}_k \in \mathbb{R}^{N \times N}$ satisfies $|\Delta \mathbf{A}_k| \leq \delta_k |\mathbf{A}_k|$ for all k , then

$$\left| \prod_{k=0}^m (\mathbf{A}_k + \Delta \mathbf{A}_k) - \prod_{k=0}^m \mathbf{A}_k \right| \leq \left(\prod_{k=0}^m (1 + \delta_k) - 1 \right) \prod_{k=0}^m |\mathbf{A}_k|,$$

where $|\delta_k| < u$. Moreover, recall $\prod_{k=1}^N (1 + \delta_k)^{\pm 1} = 1 + \theta_N$ where $|\theta_N| \leq \frac{Nu}{1-Nu} =: \gamma_N$ and $\gamma_k + u \leq \gamma_{k+1}$, $\gamma_k + \gamma_j + \gamma_k \gamma_j \leq \gamma_{k+j}$ from [23, Lemma 3.1 and 3.3]. If the floating point number of $x \in \mathbb{C}$ is represented by $fl(x)$ and similarly for $y \in \mathbb{C}$, then $fl(x \pm y) = (x + y)(1 + \delta)$ where $|\delta| \leq u$ and $fl(xy) = (xy)(1 + \delta)$ where $|\delta| \leq \sqrt{2}\gamma_2$ from [23, Lemma 3.5].

We now prove the error bound on computing the Algorithm III.3.

Theorem 5.1: Let $\hat{\mathbf{y}} = fl(\mathbf{A}_N \mathbf{z})$, where $N \geq 4$ is an integer, be computed using Algorithm III.3, and assume (8) holds, then

$$|\mathbf{y} - \hat{\mathbf{y}}| \leq \frac{2(N-1)\eta}{1-2(N-1)\eta} |\mathbf{L}^{(1)}| |\mathbf{L}^{(2)}| \dots |\mathbf{L}^{(N-1)}| |\mathbf{U}^{(N-1)}| \dots |\mathbf{U}^{(2)}| |\mathbf{U}^{(1)}| |\mathbf{z}|, \quad (9)$$

where $\eta = \mu + \gamma_5(1 + \mu)$.

Proof: Let $\hat{\mathbf{L}}^{(m)}$ and $\hat{\mathbf{U}}^{(m)}$ be defined in terms of the computed weights $\hat{\alpha}^k$ for $m = 1, 2, \dots, N-1$. Then

$$\begin{aligned} \hat{\mathbf{y}} &= fl \left(\hat{\mathbf{L}}^{(1)} \hat{\mathbf{L}}^{(2)} \dots \hat{\mathbf{L}}^{(N-1)} \hat{\mathbf{U}}^{(N-1)} \dots \hat{\mathbf{U}}^{(2)} \hat{\mathbf{U}}^{(1)} \mathbf{z} \right) \\ &= (\hat{\mathbf{L}}^{(1)} + \Delta \hat{\mathbf{L}}^{(1)}) (\hat{\mathbf{L}}^{(2)} + \Delta \hat{\mathbf{L}}^{(2)}) \dots (\hat{\mathbf{L}}^{(N-1)} + \Delta \hat{\mathbf{L}}^{(N-1)}) \\ &\quad (\hat{\mathbf{U}}^{(N-1)} + \Delta \hat{\mathbf{U}}^{(N-1)}) \dots (\hat{\mathbf{U}}^{(2)} + \Delta \hat{\mathbf{U}}^{(2)}) \\ &\quad (\hat{\mathbf{U}}^{(1)} + \Delta \hat{\mathbf{U}}^{(1)}) \mathbf{z}. \end{aligned}$$

Each $\mathbf{L}^{(m)}$ and $\mathbf{U}^{(m)}$ has only two nonzero entries per row and we are using complex arithmetic. Thus $|\Delta \hat{\mathbf{L}}^{(m)}| \leq \gamma_5 |\hat{\mathbf{L}}^{(m)}|$, $|\Delta \hat{\mathbf{U}}^{(m)}| \leq \gamma_4 |\hat{\mathbf{U}}^{(m)}|$ for $m = 1, 2, \dots, N-1$. By considering computed weights i.e., in view of (8), $\hat{\mathbf{L}}^{(m)} = \mathbf{L}^{(m)} + \Delta \mathbf{L}^{(m)}$, $|\Delta \mathbf{L}^{(m)}| \leq \mu |\mathbf{L}^{(m)}|$ and $\hat{\mathbf{U}}^{(m)} = \mathbf{U}^{(m)} + \Delta \mathbf{U}^{(m)}$, $|\Delta \mathbf{U}^{(m)}| \leq \mu |\mathbf{U}^{(m)}|$. Thus,

$$\begin{aligned} \hat{\mathbf{y}} &= (\mathbf{L}^{(1)} + \mathbf{E}^{(1)}) (\mathbf{L}^{(2)} + \mathbf{E}^{(2)}) \dots (\mathbf{L}^{(N-1)} + \mathbf{E}^{(N-1)}) \\ &\quad (\mathbf{U}^{(N-1)} + \tilde{\mathbf{E}}^{(N-1)}) \dots (\mathbf{U}^{(2)} + \tilde{\mathbf{E}}^{(2)}) \\ &\quad (\mathbf{U}^{(1)} + \tilde{\mathbf{E}}^{(1)}) \mathbf{z}, \end{aligned}$$

where $|\mathbf{E}^{(m)}| \leq (\mu + \gamma_5(1 + \mu)) |\mathbf{L}^{(m)}|$ and $|\tilde{\mathbf{E}}^{(m)}| \leq (\mu + \gamma_4(1 + \mu)) |\mathbf{U}^{(m)}|$. Let $\eta = \mu + \gamma_5(1 + \mu)$. Hence we find the relative error in computing DVM by a vector

$$\begin{aligned} |\mathbf{y} - \hat{\mathbf{y}}| &\leq [(1 + \eta)^{2(N-1)} - 1] |\mathbf{L}^{(1)}| |\mathbf{L}^{(2)}| \dots |\mathbf{L}^{(N-1)}| \\ &\quad |\mathbf{U}^{(N-1)}| \dots |\mathbf{U}^{(2)}| |\mathbf{U}^{(1)}| |\mathbf{z}| \\ &\leq \frac{2(N-1)\eta}{1-2(N-1)\eta} |\mathbf{L}^{(1)}| |\mathbf{L}^{(2)}| \dots |\mathbf{L}^{(N-1)}| \\ &\quad |\mathbf{U}^{(N-1)}| \dots |\mathbf{U}^{(2)}| |\mathbf{U}^{(1)}| |\mathbf{z}|. \end{aligned}$$

Thus the result in (9). \blacksquare

If \mathbf{z} is a random input in the interval $(0, 1)$, $\mu = u = 2.2204e-16$ (using machine precision in MATLAB), and the size of \mathbf{A}_N varies from 4×4 to 32×32 , (9) simplifies to the componentwise theoretical relative error bound (upper) of order $t_N \cdot 10^{-14}$ where $t_N = |\mathbf{L}^{(1)}| |\mathbf{L}^{(2)}| \dots |\mathbf{L}^{(N-1)}| |\mathbf{U}^{(N-1)}| \dots |\mathbf{U}^{(2)}| |\mathbf{U}^{(1)}|$.

B. Numerical Results

We will now state numerical results in connection to the stability of the proposed Algorithm III.3 using MATLAB (R2014a version) with machine precision 2.2204e-16. Forward error analysis results are presented by taking the exact solutions as the output of Algorithm III.3 computed with double precision and the computed value as the output of Algorithm III.3 with single precision. Although the DVM algorithm is proposed for any even $N(\geq 4)$, for practical implementation purposes (see Remark 3.2) and brevity, we will show numerical results for some matrix sizes from 4×4 to 32×32 .

Node reordering of polynomial Vandermonde matrices using row permutations can affect the accuracy of algorithms. More

TABLE III
RELATIVE FORWARD ERROR IN CALCULATING THE DVM ALGORITHM III.3
WITH $\alpha = \frac{5}{7} + \frac{j}{\sqrt{3}}$ SUCH THAT $|\alpha| < 1$

N	EDVMA with \mathbf{z}_1	EDVMA-Leja with \mathbf{z}_1	EDVMA with \mathbf{z}_2	EDVMA-Leja with \mathbf{z}_2
4	2.3592e-08	1.7509e-08	2.4889e-08	2.9024e-08
8	2.4755e-08	2.2365e-08	2.9003e-08	1.9791e-08
16	2.4848e-08	2.8113e-08	2.8770e-08	2.2507e-08
32	2.4054e-08	2.1726e-08	2.7280e-08	2.6668e-08

TABLE IV
RELATIVE FORWARD ERROR IN CALCULATING THE DVM ALGORITHM III.3
WITH $\alpha = \frac{1}{\sqrt{2}} + \frac{j}{\sqrt{2}}$ SUCH THAT $|\alpha| = 1$

N	EDVMA with \mathbf{z}_1	EDVMA-Leja with \mathbf{z}_1	EDVMA with \mathbf{z}_2	EDVMA-Leja with \mathbf{z}_2
4	3.0071e-08	2.0753e-08	1.7276e-08	2.5564e-08
8	2.9827e-08	3.0176e-08	3.2228e-08	1.6343e-08
16	2.5757e-08	2.8992e-08	3.3992e-08	2.9307e-08
32	2.7735e-08	2.3493e-08	3.6240e-08	2.2134e-08

TABLE V
RELATIVE FORWARD ERROR IN CALCULATING THE DVM ALGORITHM III.3
WITH $\alpha = 1 + \frac{j}{4}$ SUCH THAT $|\alpha| > 1$

N	EDVMA with \mathbf{z}_1	EDVMA-Leja with \mathbf{z}_1	EDVMA with \mathbf{z}_2	EDVMA-Leja with \mathbf{z}_2
4	2.1082e-08	2.3318e-08	2.4548e-08	3.1017e-08
8	3.2268e-08	3.2268e-08	2.4237e-08	2.1985e-08
16	2.2324e-08	2.5521e-08	9.9889e-09	2.1048e-08
32	1.0413e-08	2.7010e-08	2.2268e-08	4.1812e-08

importantly, Leja ordering (see [25], [26]) defined via:

$$|\alpha| = \max_{1 \leq i \leq N} |\alpha^i|, \prod_{l=1}^{k-1} |\alpha^k - \alpha^l| = \max_{k \leq i \leq N} \prod_{l=1}^{k-1} |\alpha^i - \alpha^l|,$$

$$k = 2, 3, \dots, N$$

can greatly improve the accuracy of similar algorithms. Therefore, we show numerical results for the relative error of the proposed algorithm with and without Leja ordering.

We compare the relative forward error e , with and without Leja ordering, of Algorithm III.3 defined by

$$e = \|\mathbf{y} - \hat{\mathbf{y}}\|_2 / \|\mathbf{y}\|_2$$

where $\mathbf{y} = \mathbf{A}_N \mathbf{z}$ is the exact solution computed using Algorithm III.3 with double precision and $\hat{\mathbf{y}}$ is the computed solution of Algorithm III.3 with single precision. Tables III, IV, and V show numerical results for the forward error of the proposed Algorithm III.3 with α such that $|\alpha| < 1$, $|\alpha| = 1$, and $|\alpha| > 1$ respectively, and random real and complex inputs \mathbf{z}_1 and \mathbf{z}_2 respectively, without and with Leja ordering of the nodes $\{\alpha, \alpha^2, \dots, \alpha^N\}$ of the DVM, say EDVMA and EDVMA-Leja respectively. We have assumed uniformly distributed random input \mathbf{z} in the interval (0, 1) for each N , say \mathbf{z}_1 , and uniformly distributed random real and imaginary parts of the input \mathbf{z} in the interval (0, 1) for each N , say \mathbf{z}_2 .

For any $\alpha \in \mathbb{C}$ such that $|\alpha| = 1$, one can show that the forward error of the proposed algorithm with Leja ordering

preserves the stability of the system even with the large $N = 256$ having an error of order 10^{-8} . This is because all nodes of DVM lies on the unit circle and hence the DVM is perfectly conditioned (see e.g., [23]). For some $\alpha \in \mathbb{C}$ such that $|\alpha| < 1$, the forward error of the proposed algorithm with and without Leja ordering will have the same numerical values, i.e., of order 10^{-8} . This is because we have taken the computed values $\hat{\mathbf{y}}$ with single precision. For some $\alpha \in \mathbb{C}$ such that $|\alpha| > 1$, the system becomes unstable even for $N < 64$ without Leja ordering of nodes. This is because the conditional number of the DVM grows exponentially (see e.g., [20]–[23]). Finally, even for $N = 64$ Leja ordering of nodes $\{\alpha, \alpha^2, \dots, \alpha^N\}$ of the DVM leads to uniform error order 10^{-8} e.g., when $\alpha = \frac{5}{7} + \frac{j}{\sqrt{3}}$: EDVMA-Leja with \mathbf{z}_1 is 2.3501e-08 and with \mathbf{z}_2 is 1.988e-08, when $\alpha = \frac{1}{\sqrt{2}} + \frac{j}{\sqrt{2}}$: EDVMA-Leja with \mathbf{z}_1 is 1.0636e-08 and with \mathbf{z}_2 is 3.2138e-08, and when $\alpha = 1 + \frac{j}{4}$: EDVMA-Leja with \mathbf{z}_1 is 2.7525e-08 and with \mathbf{z}_2 is 2.4679e-08. On the other hand when $N = 64$, for the same α e.g., $\alpha = 1 + \frac{j}{4}$, the forward error of the proposed algorithm without Leja ordering overflows. Hence for $|\alpha| > 1$, Leja ordering of nodes produces better forward error and provides good forward stability even for some ill-conditioned DVM matrices.

VI. CIRCUIT DESIGN FOR MULTI-BEAM REALIZATION

A block diagram of the overall receiver architecture is shown in Fig. 3 (a) and (b). This section focuses on the LNAs and RF chains; the design of ABF block is discussed in Section VII. Each RF chain consists of i) a quadrature frequency downconverter realized using two mixers and a quadrature local oscillator (LO), and ii) intermediate frequency (IF) amplifiers and low-pass filters. The outputs of the IF amplifiers are fed into parallel ADCs for digitization. Independent RF chains are used to process the M parallel outputs of the analog multiplexer. However, the LOs of these M receiver channels should be synchronized in order to obtain coherent outputs for digital beamforming. Thus, we need to control LO duty cycle, skew, and jitter (both random and deterministic) across the array in the presence of variable routing length (μm to mm) in the power divider network. In particular, the LO distribution network often dominates out-of-band phase noise (i.e., at large offset frequencies), making its design critical for broadband systems [27]. There are three main design approaches: i) distributing a single LO generated by a central phase-locked loop (PLL); ii) injection-locking individual LOs in each channel to that of a central PLL, resulting in a distributed PLL; and iii) distributing a low-frequency reference signal to local PLLs at each channel. Each approach results in a different trade-off between the power consumed by LO generation and distribution [28]. The local PLL option provides good performance, flexibility, and scalability at the cost of the highest power and area consumption. Thus, the central and distributed PLL options are more suitable for smaller arrays. Of these two, the central PLL is favored for broadband systems, for which injection locking becomes less power-efficient. Further discussion and detailed design of the LO distribution network for this particular system is deferred to future papers.

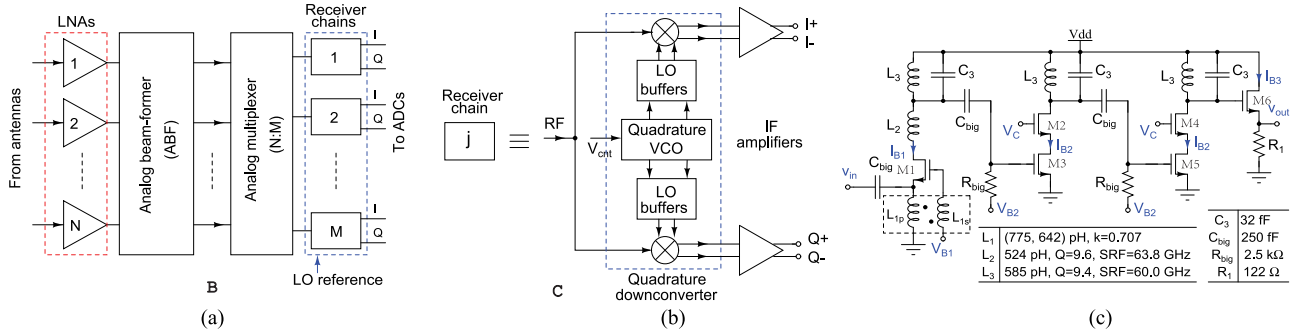


Fig. 3. (a) High-level block diagram of the overall RF front-end architecture, (b) a more detailed block diagram of a single RF chain, (c) simplified schematic of the proposed LNA.

All circuits were designed in the UMC 65 nm RF-CMOS process, which has 8 metal layers (including a “ultra-thick” 3.25 μm top layer for inductors) and both lateral- and vertical-field metal-insulator-metal (MIM) capacitors.

A. Low-Noise Amplifier (LNA)

A simplified schematic of the proposed LNA is shown in Fig. 3(c). It consists of four stages. The first stage uses a common-gate (CG) topology to provide broadband impedance matching, followed by a resonant load (L_3 - C_3) to define a band-pass frequency response. The feedback transformer L_{1p} - L_{1s} provides passive voltage gain between the gate and source terminals of the input transistor M_1 , thus reducing the noise figure (NF) and power consumption [29]–[31]. In addition, L_{1p} acts as a high-impedance element (RF choke) at the source of M_1 . This stage also uses an additional inductor L_2 that acts as a “peaking” element [32] and reduces the impact of M_1 ’s gate-drain capacitance C_{gd} on bandwidth and NF. The second and third stages are cascoded common-source stages with resonant loads. They provide voltage gain and additional band-pass filtering. The final stage is a resistively-loaded source-follower that acts as a unity-gain buffer. It provides a low output impedance suitable for driving later signal-processing stages.

Lumped equivalent parameters (at 28 GHz) of the main passive components used within the LNA are summarized in Fig. 3(c). Note that the transformer L_1 has a self-resonance frequency (SRF) slightly below 28 GHz, resulting in a large impedance over the operating frequency range but a capacitive lumped equivalent model. In order to provide design insight, the figure shows low-frequency (2 GHz) parameters for L_1 .

B. Quadrature Voltage-Controlled Oscillator (VCO)

A simplified schematic of the proposed quadrature VCO is shown in Fig. 4(a). It consists of two differential LC oscillators that are coupled together via additional differential pairs. The bias voltages V_{B1} , V_{B2} , and V_C control the tail currents of the main and coupling differential pairs, respectively. Part of the load capacitance is realized using MOS varactors (C_2), thus allowing the output frequency to be adjusted via the DC control voltage V_{cnt} . In addition, RC polyphase networks (R_3 - C_3) are used as phase shifters in the coupling paths. It is known

that such a 90° phase shift i) desensitizes the oscillator’s output phases to component mismatches, and ii) minimizes phase noise [33]. The transfer function of the chosen network is given by $H(s) = (1 - sR_3C_3) / [1 + sR_3(C_3 + C_L)]$ where C_L is the load capacitance seen by the network and is dominated by the gate-source capacitance C_{gs} and the gate-drain capacitance C_{gd} of the coupling transistors. In our case $C_3 \approx C_L$, resulting in a phase shift of $\approx 72^\circ$ and gain of ≈ 0.8 when $\omega R_3(C_3 + C_L) = 1$. This is close enough to the ideal behavior (90° phase shift and unity gain) to be of benefit. In addition, R_3 should be large enough to ensure that the lossy RC phase shifter does not significantly load the LC tank, since this will worsen phase noise. Lumped equivalent parameters (at 28 GHz) of the main passive components used within the VCO are summarized in Fig. 4(a). Each VCO output is buffered using the same resistively-loaded source-follower used in the LNA, with $R_L = 122 \Omega$ (not shown in Fig. 4(a)).

C. Mixer and IF Amplifiers

A simplified schematic of the proposed downconverting mixer is shown in Fig. 4(b). It uses a single-balanced topology in which the RF signal is capacitively fed into the source terminal of a differential pair switched by the LO [34]. This isolates the bias current of the input transconductor from that of the LO pair, allowing their ratio I_T/I_B to be reduced. As a result, (i) the LO pair switches more abruptly, which reduces NF and increases conversion gain; and (ii) LO feedthrough to the IF port also decreases.

The value of L_1 is chosen to resonate with parasitic capacitance at the drain of the input transistor at the nominal RF frequency. As a result, most of the RF current flows through C_c , as desired. In addition, the values of R_L and C_L are selected to optimize the trade-off between low-frequency conversion gain ($\propto R_L$) and IF bandwidth ($B_{IF} = 1/(2\pi R_L C_L)$). We target $B_{IF} > 850$ MHz to enable the system to cover the entire U.S. “28 GHz” licensed band (27.5–28.35 GHz) as defined by the Federal Communications Commission. The lumped equivalent parameters (at 28 GHz) for the passive components used within the mixer are summarized in Fig. 4(b).

The differential IF signals generated by the mixers are further amplified and low-pass filtered. Each IF amplifier uses a simple resistively-loaded differential pair topology. Cascoding reduces

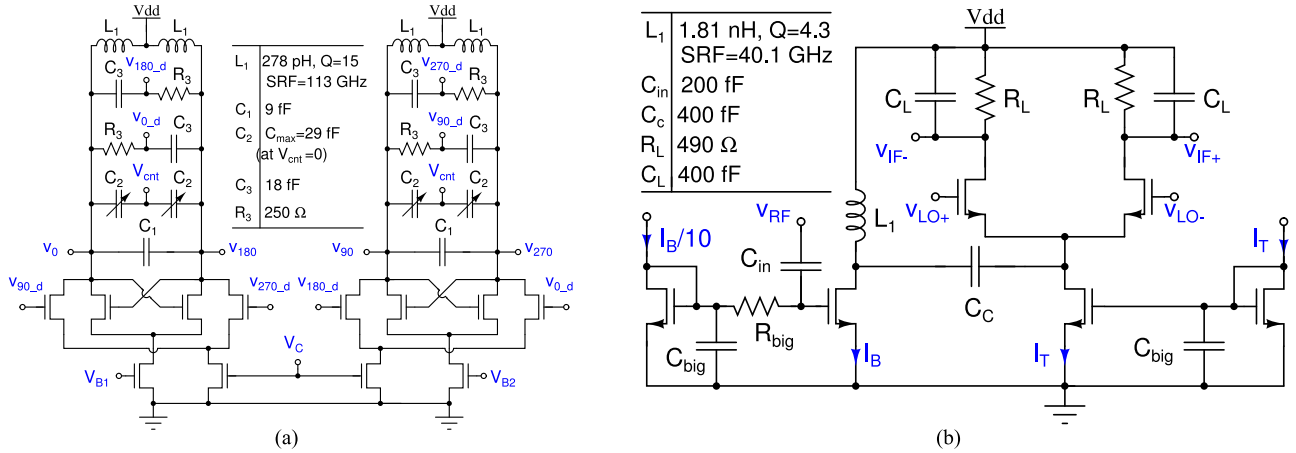


Fig. 4. (a) Simplified schematic of the proposed quadrature VCO (output buffers are not shown), (b) simplified schematic of the proposed single-balanced mixer.

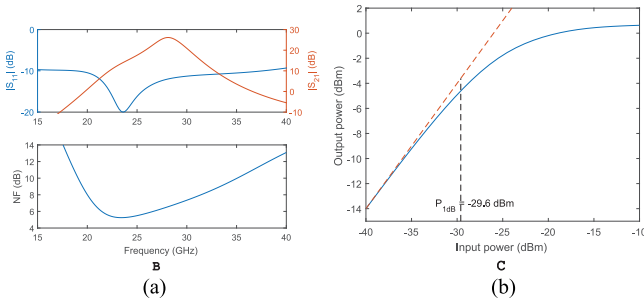


Fig. 5. (a) Reflection coefficient ($|S_{11}|$), gain ($|S_{21}|$), and NF of the LNA as a function of frequency. (b) Compression curve of the LNA, with the input-referred P_{1dB} curve labeled.

capacitive loading on previous stages and improves input-output isolation. The tail current source is programmed via a current DAC to set the desired voltage gain.

D. Simulation Results

The circuits described in the previous section were simulated using the Cadence SpectreRF simulator. All components were modeled using foundry-supplied parametric RF macromodels that include layout parasitics.

1) *LNA*: The nominal bias currents for the LNA were set to $I_{B1} = 2.4$ mA, $I_{B2} = 2.8$ mA, and $I_{B3} = 7.1$ mA, resulting in a total power consumption of 18.1 mW from a 1.2 V power supply. Note that the output buffer (biased at I_{B3}) consumes a significant fraction of the total power. Fig. 5(a) shows the simulated input reflection coefficient, gain, and NF of the LNA as a function of frequency. The CG topology provides a broadband input impedance match ($|S_{11}| < -10$ dB at frequencies greater than 19 GHz). The amplifier also has a peak small-signal gain of 26.2 dB and a -3 dB bandwidth of 2.6 GHz centered around 28 GHz, which is sufficient for 5G applications. Finally, we simulate an NF of 6.5 dB at 28 GHz.

Fig. 5(b) shows the simulated gain compression curve of the LNA. The estimated input-referred 1 dB compression point is $P_{1dB} = -29.6$ dBm. In future work, we will reduce NF for the

same power consumption by modifying the first stage to use i) improved feedback transformer design, and ii) thermal noise cancellation [35]–[37]. We will also focus on various linearization methods to increase P_{1dB} .

2) *VCO*: The nominal bias currents for the VCO were set using V_{B1} , V_{B2} , and V_C to 3 mA, 1.5 mA, and 750 μ A, respectively, resulting in a total power consumption of 7.2 mW. These values of V_{B1} and V_{B2} minimize amplitude mismatch between the I and Q paths. In addition, each LO buffer was biased at 3.5 mA to ensure low output impedance, resulting in a total power consumption of 16.9 mW. Note that the buffers consume significantly more power than the VCO core.

Fig. 6(a) shows typical simulated time-domain VCO output waveforms after the LO buffers. The control voltage V_{cnt} was adjusted to obtain oscillations near 28 GHz. Nearly ideal quadrature oscillations were observed, with low amplitude and phase imbalance between the I and Q paths (≈ 0.3 dB and 3.3° , respectively). The simulated tuning curve of the VCO versus V_{cnt} is shown in Fig. 6(b). The output frequency can be adjusted by $\pm 2.5\%$, i.e., a 1.4 GHz span around 28 GHz.

The phase noise of the VCO is shown in Fig. 6(c). The simulated power spectrum is $\propto 1/f^3$ at offset frequencies < 35.9 kHz, and $\propto 1/f^2$ for larger offset frequencies. Thus, the so-called “ $1/f^3$ corner frequency” is ~ 35.9 kHz. The phase noise in the $\propto 1/f^2$ region reaches -90.6 dBc/Hz at an offset of 1 MHz.

3) *Mixer*: The nominal bias currents for the mixer were set to $I_T = 500$ μ A and $I_B = 6.5$ mA, which satisfies $I_T \ll I_B$. Fig. 6(d) shows the resulting RF-IF conversion gain for RF and LO frequencies around 28 GHz as a function of LO drive level. LO amplitudes of 200–300 mV maximize the conversion gain, which has a peak value of 7.7 dB. Fig. 6(e) shows the simulated NF of the mixer as a function of IF frequency for several LO drive levels. The NF increases at low IF frequencies (< 10 MHz) because of $1/f$ noise within the mixer, which increases as the frequency decreases. In addition, it decreases as the LO drive level increases before saturating at 12.0 dB for levels > 200 mV. This behavior is in agreement with the conversion gain results (see Fig. 6(d)).

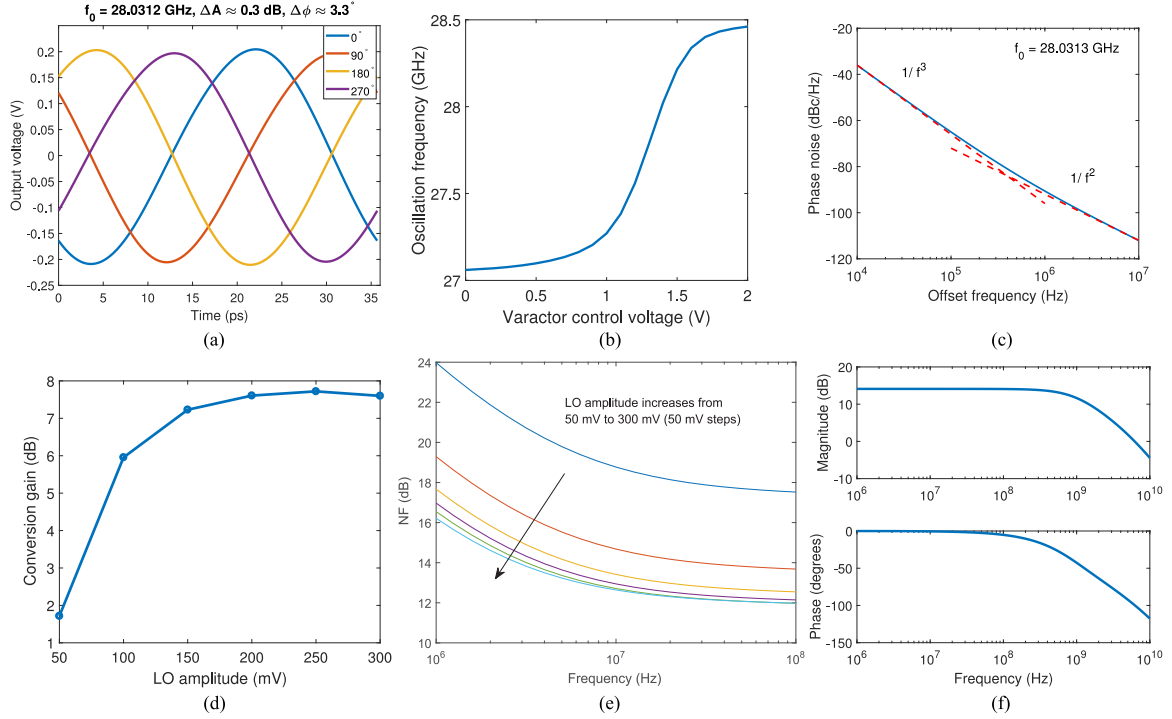


Fig. 6. (a) Typical time-domain VCO output voltages at the outputs of the LO buffers (DC components removed for clarity). The steady-state oscillation frequency was 28.0312 GHz. (b) VCO tuning curve versus V_{cnt} (the varactor control voltage). (c) Phase noise power spectrum of the VCO for an oscillation frequency of 28.0313 GHz (the $1/f^3$ and $1/f^2$ regions are labeled). (d) RF-IF conversion gain of the mixer as a function of LO drive level; the LO and RF frequencies was kept fixed at 28.1 GHz and 28.0 GHz, respectively. (e) Mixer NF as a function of IF frequency for several LO drive levels. (f) Small-signal differential voltage gain of the IF amplifier.

4) *IF Amplifier*: The IF amplifier's tail current and load were set to 1 mA and 960 Ω , respectively. Fig. 6(f) shows the resulting small-signal voltage gain, which has a low-frequency value of 14.1 dB and a -3 dB bandwidth of 1.15 GHz. The latter satisfies our IF bandwidth requirement of 850 MHz.

5) *Overall Receiver*: The entire receiver (including the LNA, but without the ABF and multiplexer) consumes 81.5 mA from a 1.2 V supply. Fig. 7 shows simulated signals at various points in the receiver chain for an input RF amplitude of 500 μ V, which corresponds to a power level of -56.0 dBm for 50 Ω loads. The VCO control voltage V_{cnt} was set to 1.5 V, resulting in LO and IF frequencies of 28.12 GHz and 120 MHz, respectively. The plot shows (from top to bottom) the RF input, LNA output, I - and Q -branch outputs of the quadrature downconverter (VCO, LO buffers, and two mixers), and I - and Q -branch outputs of the IF amplifier. In addition to providing voltage gain, the IF amplifier also acts as a low-pass filter that removes LO feedthrough and upconverted RF signals from its outputs. The phase imbalance between the two IF outputs is $\approx 3^\circ$, which is in agreement with VCO simulation results (see Fig. 6(a)). The overall voltage gain of the receiver is 50.2 dB, of which 29.3 dBm is contributed by the LNA, 7.0 dB by the downconverter, and 13.9 dB by the IF amplifier. These results are in excellent agreement with simulations of the individual components.

VII. LEVEL-1 ANALOG MULTI-BEAM BEAMFORMER

The linear phase shift $e^{-j\Omega_i\tau}$ associated with \mathbf{A}_N (or its factorization) can be efficiently approximated on-chip by CMOS

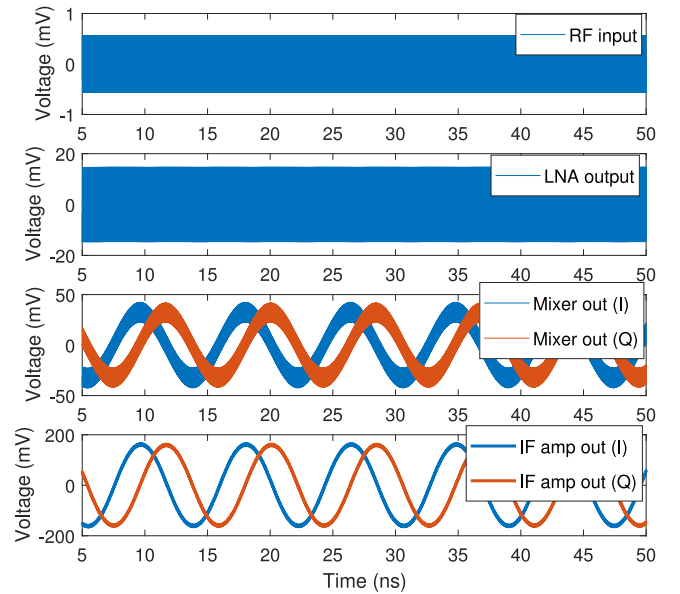


Fig. 7. Signals at various points in the receiver for a VCO control voltage of $V_{cnt} = 1.5$ V and input RF amplitude of 500 μ V. The RF, LO, and IF frequencies were 28.0 GHz, 28.12 GHz, and 120 MHz, respectively.

APFs [38], [39]. Thus, \mathbf{A}_N can be approximately realized by using them as building blocks. APFs approximate the time delay τ using the fact that $e^{-j\omega\tau} \approx \left(\frac{1-j\omega\tau/2M}{1+j\omega\tau/2M}\right)^M$, $M \in \mathbb{Z}^+$ and ω is the frequency variable in general. Typically, $M = 3$ is sufficient [38].

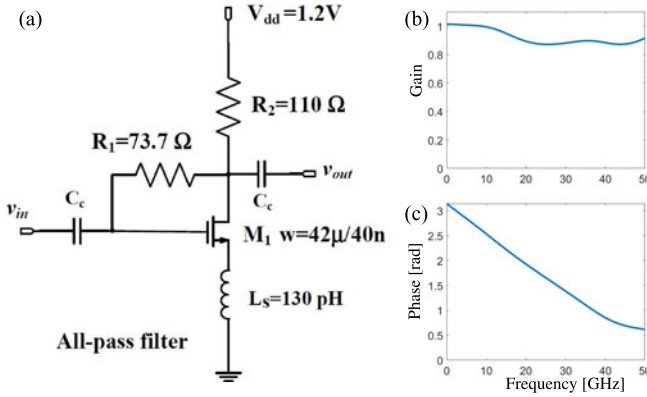


Fig. 8. (a) APF schematic; its (b) gain and (c) phase profile.

If $\psi(s) = \left(\frac{1 - j\omega\tau/2M}{1 + j\omega\tau/2M} \right)^M$ is the transfer function of an APF approximation of the delay τ , then the beamforming matrix from sparse factorization of \mathbf{A}_N can be approximated by the APF network matrix $\Psi_N \approx \mathbf{A}_N$. Here, (l, k) -th element of Ψ_N takes the form $\psi^{lk}(j\omega)$ as the corresponding element in \mathbf{A}_N is $e^{-j\omega(\tau \cdot lk)}$ where $l, k \in \{0, 1, \dots, N-1\}$. Here we use an APF implemented using 45 nm CMOS technology to realize Ψ_N . A direct approach to implementing Ψ_N would require cascading of $l \cdot k$ such APFs for realizing each (l, k) -th node. Subsequently, we analyze and compare example realizations of level-1 beamforming networks for $N = 4$ and 8 using the proposed DVM factorization.

A. 28-GHz Voltage-Mode CMOS All-Pass Filter

Our voltage-mode CMOS APF is a single-transistor circuit (see Fig. 8) [40]. Ignoring parasitic poles and zeros due to C_{gs} and C_{gd} and assuming $g_m \gg g_{ds}$, the transfer function is

$$H(s) = \frac{v_{out}}{v_{in}} = -\frac{R_2(R_1 g_m - 1)}{R_1 + R_2} \cdot \frac{1 - s \frac{L_s g_m}{R_1 g_m - 1}}{1 + s L_s g_m}, \quad (10)$$

where g_m is the transconductance of M_1 , L_s is a source degeneration inductor, R_1 is a feedback resistor between gate and drain of M_1 , and R_2 is a load resistor to convert current to voltage. For (10) to represent an APF, the left-plane pole and right-side zero should have the same frequencies. This can be achieved by setting $R_1 g_m - 1 = 1$, resulting in

$$\frac{v_{out}}{v_{in}} = -\frac{R_2}{R_1 + R_2} \cdot \frac{1 - s L_s g_m}{1 + s L_s g_m}. \quad (11)$$

The pole and zero frequencies are $\omega_{p1} = -\omega_z = -(L_s g_m)^{-1}$, the phase response is $\phi(\omega) = -2 \tan^{-1}(\omega/|\omega_{p1,z}|)$ and the group delay $t_d = \frac{2}{\omega_{p1,z}} \cdot \frac{\omega_{p1,z}^2}{\omega_{p1,z}^2 + \omega^2}$ where ω is the angular frequency and $\omega_{p1,z}$ denotes the pole or the zero frequency. At low frequencies $t_d \approx 2/\omega_{p1,z}$, and the term $\frac{\omega_{p1,z}^2}{\omega_{p1,z}^2 + \omega^2}$ captures the high-frequency t_d dispersion associated with all first-order APFs. Once the parasitic parameters of M_1 , which mainly include C_{gd} and C_{gs} , are considered, high frequency poles will be generated. Small-signal analysis confirms a dominant parasitic pole at $\omega_{p2} = -(R_1 C_{gd})^{-1}$. To reduce the effect of ω_{p2} on the

APF, R_1 is kept small and large g_m is used to ensure $\omega_{p1} = -\omega_z$. Large g_m is achieved by providing a large overdrive voltage to M_1 while keeping M_1 's size fixed, which increases ω_{p2} while keeping C_{gd} constant.

B. 4-Beam and 8-Beam Networks

The DVM for the 4-beam case (\mathbf{A}_4) is given by

$$\mathbf{A}_4 = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 \\ 1 & \alpha^4 & \alpha^8 & \alpha^{12} \end{bmatrix}. \quad (12)$$

Following the Proposition 3.1, \mathbf{A}_4 can be factorized as $\mathbf{A}_4 = \mathbf{L}^{(1)} \mathbf{L}^{(2)} \mathbf{L}^{(3)} \mathbf{U}^{(3)} \mathbf{U}^{(2)} \mathbf{U}^{(1)}$ where,

$$\mathbf{U}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \alpha \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{U}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \alpha & 0 \\ 0 & 0 & 1 & \alpha^2 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{U}^{(3)} = \begin{bmatrix} 1 & \alpha & 0 & 0 \\ 0 & 1 & \alpha^2 & 0 \\ 0 & 0 & 1 & \alpha^3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{L}^{(3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \alpha(\alpha - 1) & 0 & 0 \\ 0 & 1 & \alpha(\alpha^2 - 1) & 0 \\ 0 & 0 & 1 & \alpha(\alpha^3 - 1) \end{bmatrix},$$

$$\mathbf{L}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & \alpha^2(\alpha - 1) & 0 \\ 0 & 0 & 1 & \alpha^2(\alpha^2 - 1) \end{bmatrix},$$

$$\text{and } \mathbf{L}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & \alpha^3(\alpha - 1) \end{bmatrix}.$$

Here α represents the phase delay $e^{-j\Omega\tau}$. An implementation of the factorized stages of \mathbf{A}_4 is shown in Fig. 9. Here, $\psi^i(s)$ where $i \in \{2, 3, 4\}$ can be realized by cascading $i \cdot M$ identical APFs. The meeting of two arrow heads indicates addition unless otherwise specified. The number of APFs required in the branch originating from $x[3]$ (in Fig. 9) have been optimized to reduce hardware complexity. The unique look-directions θ_k s of this beamformer where $1 \leq k \leq N$, are a function of both APF group delay t_d and antenna spacing Δx . For a given t_d and Δx , the look-directions of the beams are

$$\theta_k = \sin^{-1} \frac{c \cdot k \cdot t_d}{\Delta x}, \quad 1 \leq k \leq N. \quad (13)$$

The signal flow graph (SFG) in Fig. 9 uses only 24 APF blocks. A similar 4-beam network using a direct TTD phased array would involve 60 such blocks. Thus the proposed approach achieves a 60% reduction in hardware complexity.

To simulate the SFG shown in Fig. 9, we assume a 2 GHz signal bandwidth around a $f_0 = 28$ GHz carrier and an antenna spacing $\Delta x = \lambda_{\min}/2$ to eliminate grating lobes. The

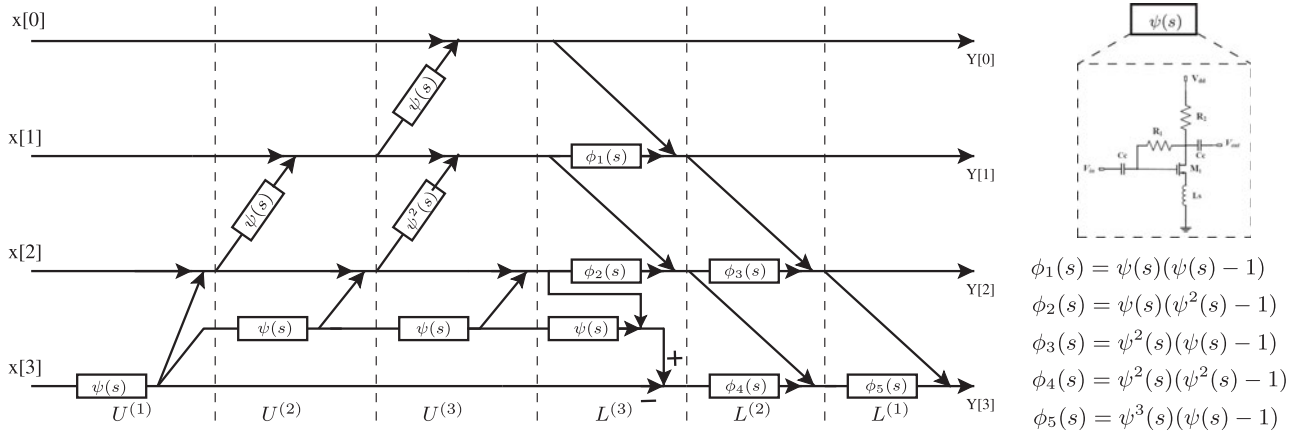


Fig. 9. Signal flow graph (SFG) for the proposed low-complexity 4-beam wideband beamformer.

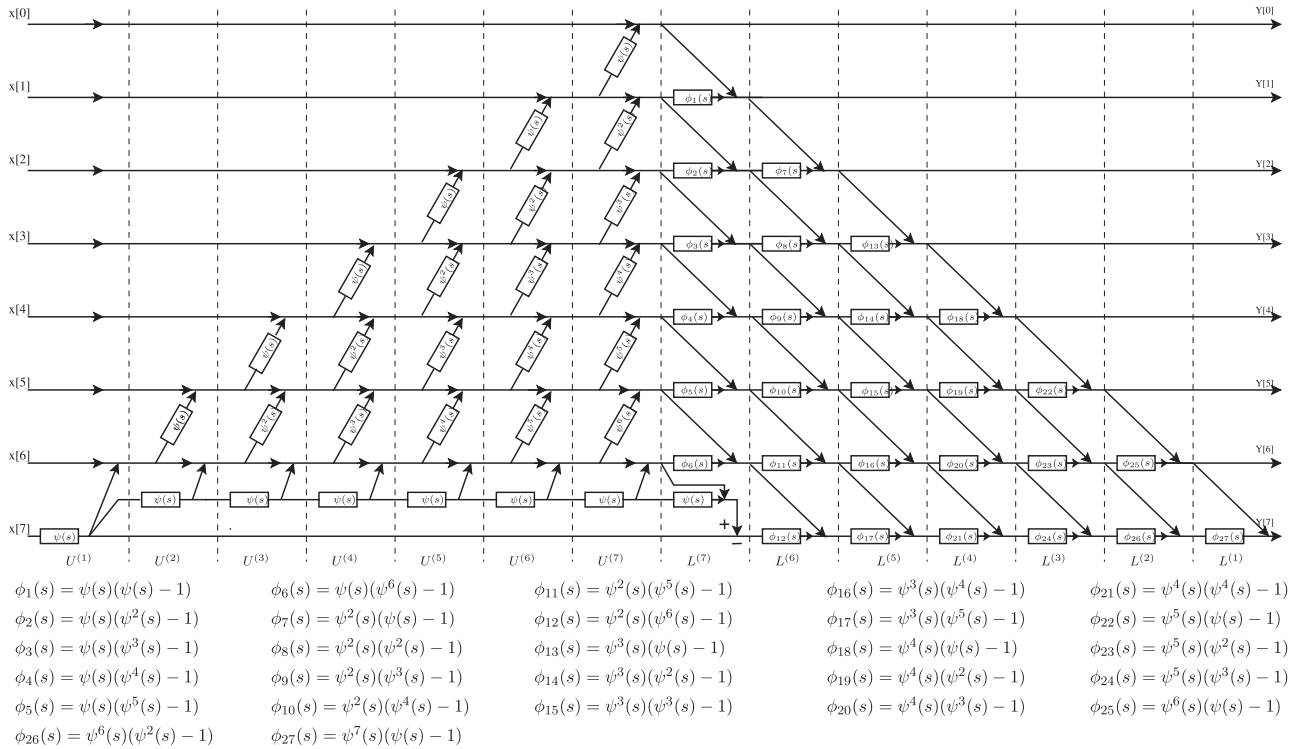


Fig. 10. Signal flow graph (SFG) for the proposed low-complexity 8-beam wideband beamformer.

APG group delay required for the beams to be within ϕ° from array broadside is $t_d = \frac{\Delta x \sin^{-1}\{\phi\}}{c \cdot N}$. The required delay was first estimated using this relationship taking $\phi = 50^\circ$, which leads to $t_d \approx 3.2$ ps for $N = 4$, resulting in four beams directed at 11.1° , 22.6° , 35.2° , and 50.2° . Next the APF design in Section VII-A was tuned in Cadence to obtain the desired group delay over the 26–30 GHz range. Average group delay was calculated as 3.202 ps using the Cadence simulated data. The simulated APF frequency response was exported to MATLAB to simulate beam responses for the SFG in Fig. 9. For comparison, beams were also generated using ideal delays, i.e., $e^{-j\omega t_d}$ where t_d is the desired group delay. The two sets of beams are in good agreement, but are not shown due to space constraints.

The SFG of the factorized stages for \mathbf{A}_8 is shown in Fig. 10 (same conventions used as in Fig. 9). This network only requires 224 APF blocks, whereas a direct 8-beam network would require 1008 blocks. As for the $N = 4$ case, t_d of the APF was selected such that all 8-beams are within 50° from array broadside; this required $t_d \approx 1.6$ ps. The tuned APF circuit as simulated in Cadence had an average group delay of 1.606 ps which ideally should generate beams at 5.5° , 11.1° , 16.7° , 22.6° , 28.7° , 35.2° , 42.2° , and 50.2° . The simulated gain and phase values of the APF tuned to $t_d = 1.60$ ps are shown in Fig. 8(b) and (c) respectively. The same procedure as described for $N = 4$ was followed. Fig. 11 shows the simulated results for beams 2, 4, 6, and 8 in two

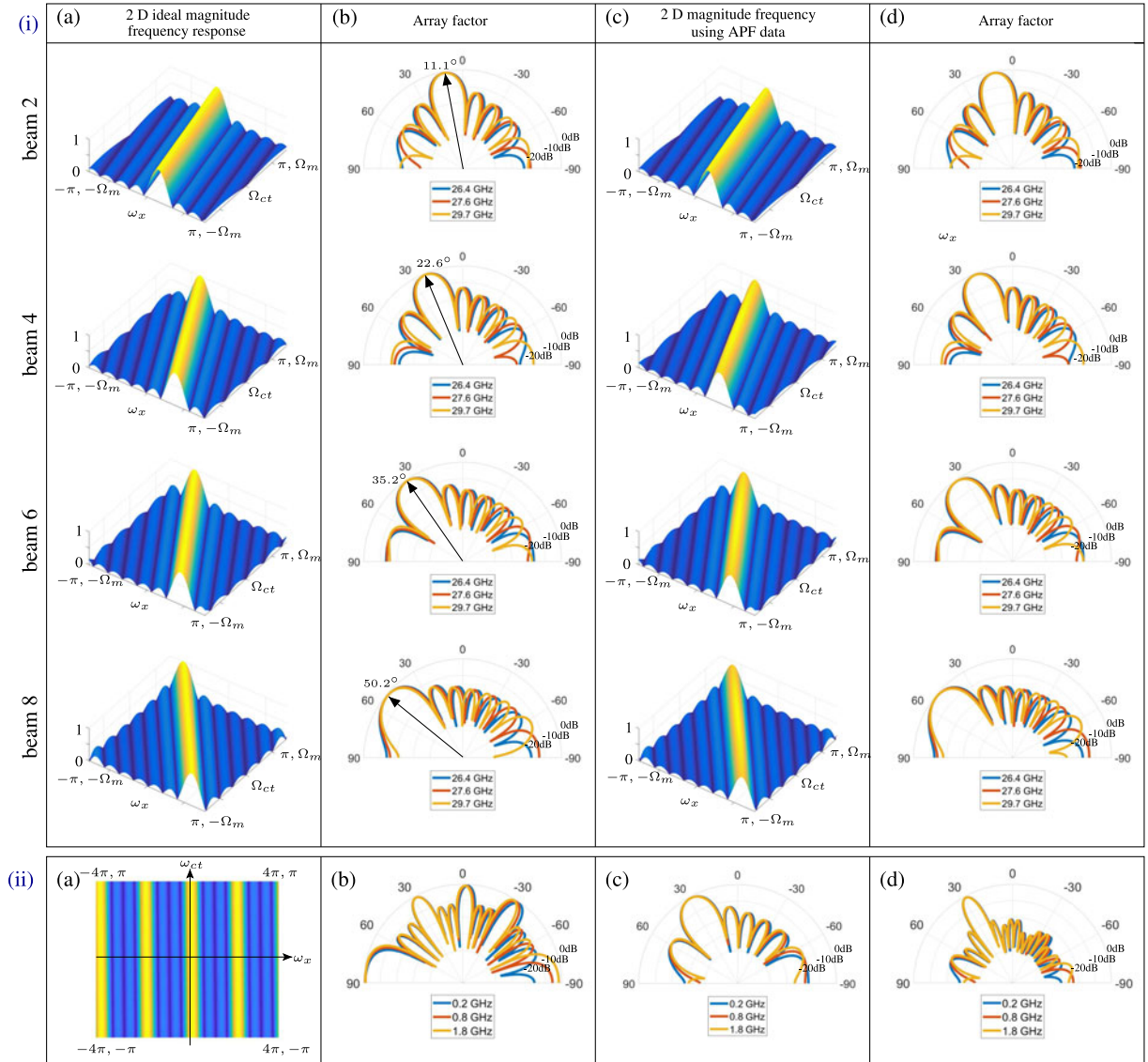


Fig. 11. (i-a) 2-D magnitude frequency domain plots of $H_l(e^{j\omega_x}, \Omega_t)$ for $l = 2, 4, 6,$ and 8 generated by the proposed 8-beam wideband beamformer assuming ideal time delays, and (i-b) the corresponding AFs. (i-c) and (i-d): Same as (i-a) and (i-b) but using simulated APF data. 2-D magnitudes (ii-a) 2-D magnitude frequency domain plot of the digital filter transfer function tuned to $\theta_5 = 28.7^\circ$; (ii-b) AF of the DBF in (ii-a) for different IF frequencies; (ii-c) AF of the 5th beam of the ABF; (ii-d) composite AF obtained by combing the ABF and DBF.

cases: Fig. 11(a) and (b) show 2-D frequency magnitudes and array factors (AFs) using ideal delays, while Fig. 11(c) and (d) show similar plots using simulated APF data. The two sets of simulated beams are in good agreement where the maximum beam deviation was $< 0.8^\circ$ with respect to the direction corresponding to the use of ideal delay. Note that while the algorithm provides N beams, the direct sum beam (look-direction to array broadside) only requires summing circuits. Also, beams pointing to $-\theta_k$ s can use an identical network with reversed antenna inputs.

C. Analysis of Variations in Beams Due to Circuit Imperfections

Depending on the technology, all APF blocks that are used to construct the beamforming network will not be identical due

to PVT variations; in practice, gain mismatches of up to 10% are expected between them. A probabilistic approach was taken to analyze the impact of such mismatch on the final beam outputs. The gain variation of the APF blocks was assumed to be distributed uniformly between unity and the maximum error margin. Monte-Carlo simulations were conducted using these randomly-distributed gains to see how the beam shapes were affected. Fig. 12 shows that the simulated beam patterns of the 4-beam network are relatively robust to the assumed amount of mismatch.

VIII. LEVEL-2 DIGITAL BEAMFORMING STAGE

The goal of the level-2 DBF is to generate narrow beams that maximize the link budget. The $N : M$ analog multiplexers after the ABF dynamically select the level-1 beam(s) fed into

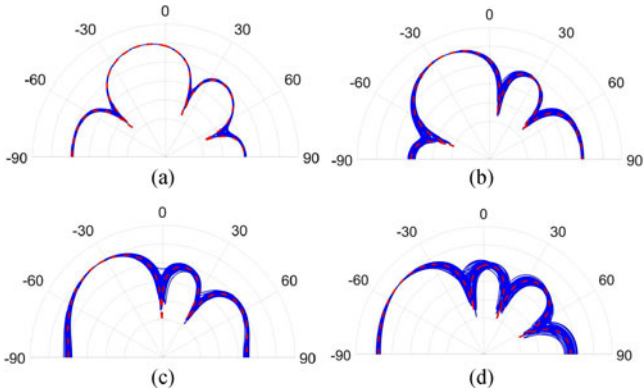


Fig. 12. Monte-Carlo simulations for each beam of the $N = 4$ network (50 runs). The red curve represents the beam pattern for the nominal value of APF gain (unity).

the DBF. Since each output beam from the L sub-array ABFs creates a spatially undersampled input to the DBF, its array factor will contain grating lobes. The latter are removed by the ABF array factors, thus allowing the hybrid system to generate sharp output beams. In addition, the proposed DBF uses TTDs in order to maintain squint-free beams. Different digital filter structures, including finite impulse response (FIR) fractional delay approximation filters and their infinite impulse response (IIR) counterparts, can be used. For example, wideband Thiran APFs can be used to approximate the required TTD [41].

Thiran filters are a special class of low-complexity IIR filters that have similar numerator and denominator coefficients, but in reverse order. They are the recursive counterpart of the FIR Lagrange interpolation method, which provides maximally flat group delay at zero frequency [42]. Thiran fractional delay filters thus realize APFs with maximally flat group delay compared to other fractional delay implementation methods [43]. A significant gain in hardware complexity can be achieved without compromising accuracy by replacing FIR fractional delays with Thiran APFs [41]. Their transfer function is

$$H(z_{ct}) = \frac{z_{ct}^{-\beta} Q(1/z_{ct})}{Q(z_{ct})}, \quad (14)$$

where $Q(z_{ct}) = \sum_{i=1}^{\beta} \alpha_i z_{ct}^{-i}$ and $z_{ct} = e^{-j\omega_{ct}}$ is the temporal z domain variable, and β is the filter order. Thiran APF coefficients can be expressed in closed form as

$$\alpha_i = (-1)^i \binom{\beta}{i} \prod_{n=0}^{\beta} \frac{D - \beta + n}{D - \beta + i + n}; \quad i = 0, 1, \dots, \beta \quad (15)$$

which approximates the delay D , where β is the filter order and $\binom{\beta}{i} = \frac{\beta!}{i!(\beta-i)!}$ are binomial coefficients. Also $\alpha_0 = 1$, and stability requires $D > \beta - 1$ [42], [43]. The 2-D \mathbf{z} domain transfer function of the Thiran beamformer is $H_T(z_x, z_{ct}) = \sum_{k=0}^{N_x-1} H_T^k(z_{ct}) z_x^{-(N_x-k)} z_{ct}^{-ik}$ where N_x is the number of spatial channels, z_x is the spatial \mathbf{z} domain variable and $H_T^k(z_{ct})$ is the Thiran filter transfer function at the k th element, which realizes the required fractional delay τ_k (i.e., the actual delay in the system is $(\beta - 1)T_s + \tau_k$ where T_s is the sampling period

of the digital system). 1) *DBF Simulation*: We assume a 32-element ($P = 32$) system with four 8-element ABFs ($N = 8$, $L = 4$). The effective DBF element spacing is $N \times \Delta x$, and its 2-D magnitude frequency response using a 3rd order Thiran filter (connected to the 5th ABF beam, which points at $\theta_5 = 28.7^\circ$) is shown in Fig. 11(ii-a). Fig. 11(ii-b) shows the DBF array factor for different IF frequencies. Fig. 11(ii-c) shows the array factor of the 5th beam generated by the level-1 sub-array ABF, while Fig. 11(ii-d) shows the resultant hybrid array factor, i.e., the combination of level-1 and level-2 beamformers.

IX. CONCLUSION

We propose a low-complexity DVM algorithm having sparse factors. Arithmetic complexities of the proposed algorithm show that it is much more efficient than the direct computation of DVM-vector multiplication. Theoretical error bounds on computing the algorithm are established, and numerical results of the forward relative error are used to analyze its stability for higher-dimensional DVMs. Proposed algorithm is used to realize a squint-free wideband mmW multi-beam beamforming architecture using mixed-signal CMOS integrated circuits. Extensive simulation results verify the operation of the hybrid architecture. The proposed architecture is useful for emerging 5G wireless communication systems requiring a variable number of sharp steerable mmW beams. The beam sharpness is achieved in a power- and circuit-optimal way using the hybrid combination of both mmW-analog as well as digital beamforming via sub-arrays. The analog sub-arrays support from 1 to N fixed TTD wideband mmW beams, depending on required capacity, which makes the architecture flexible for use in both mobiles and base stations.

The proposed hybrid beamformer contains several analog circuit components that are subject to on-chip PVT variations. The resulting delay and gain shifts in each receiver cause errors in the spatial orientation of the beams. Calibration methods to compensate for such shifts include variable-gain amplifiers for gain, tunable APFs for delay, and reference input signals for training the algorithm. Development of such methods and associated circuitry are beyond the scope of this work.

ACKNOWLEDGEMENT

Authors thank the National Science Foundation (NSF) (award #1711625 and award #1711395) for financial support.

REFERENCES

- [1] T. S. Rappaport *et al.*, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.
- [2] S. Sun, T. S. Rappaport, R. W. Heath, A. Nix, and S. Rangan, "Mimo for millimeter-wave wireless communications: Beamforming, spatial multiplexing, or both?" *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 110–121, Dec. 2014.
- [3] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.
- [4] R. Méndez-Rial, C. Rusu, N. González-Prelcic, A. Alkhateeb, and R. W. Heath, "Hybrid MIMO architectures for mm-wave communications: Phase shifters or switches?" *IEEE Access*, vol. 4, pp. 247–267, 2016.

- [5] "Federal communications commission FCC 16-89," 2016. [Online]. Available: https://apps.fcc.gov/edocs_public/attachmatch/FCC-16-89A1.pdf
- [6] S. W. Ellingson and W. Cazemier, "Efficient multibeam synthesis with interference nulling for large arrays," *IEEE Trans. Antennas Propag.*, vol. 51, no. 3, pp. 503–511, Mar. 2003.
- [7] C. Fulton, M. Yeary, D. Thompson, J. Lake, and A. Mitchell, "Digital phased arrays: Challenges and opportunities," *Proc. IEEE*, vol. 104, no. 3, pp. 487–503, Mar. 2016.
- [8] S. E. Bankov, I. Y. Bredihin, A. G. Davydov, and M. D. Safin, "Digital beam-forming super wideband multi-beam hybrid antenna system," in *Proc. 24th Int. Crimean Conf. Microw. Telecommun. Technol.*, Sep. 2014, pp. 449–450.
- [9] W. B. Abbas and M. Zorzi, "Towards an appropriate receiver beamforming scheme for millimeter wave communication: A power consumption based comparison," in *Proc. 22th Eur. Wireless Conf.*, May 2016, pp. 1–6.
- [10] S. Han, C. I. I. Z. Xu, and C. Rowell, "Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 186–194, Jan. 2015.
- [11] F. Sahrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501–513, Apr. 2016.
- [12] J. Lota, S. Sun, T. S. Rappaport, and A. Demosthenous, "5G uniform linear arrays with beamforming and spatial multiplexing at 28 GHz, 37 GHz, 64 GHz and 71 GHz for outdoor urban communication: A two-level approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9972–9985, Nov. 2017.
- [13] S. Sun, T. S. Rappaport, and M. Shafi, "Hybrid beamforming for 5G millimeter-wave multi-cell networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Honolulu, HI, USA, Apr. 2018.
- [14] H. Oruç and G. M. Phillips, "Explicit factorization of the Vandermonde matrix," *Linear Algebra Appl.*, vol. 315, pp. 113–123, 2000.
- [15] H. Oruç and H. K. Akmaz, "Symmetric functions and the Vandermonde matrix," *J. Comput. Appl. Math.*, vol. 172, no. 1, pp. 49–64, Nov. 2004.
- [16] S.-L. Yang, "On the LU factorization of the Vandermonde matrix," *Discrete Appl. Math.*, vol. 146, no. 1, pp. 102–105, 2005.
- [17] J. F. Canny, E. Kaltofen, and L. Yagati, "Solving systems of nonlinear polynomial equations faster," in *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation*, ser. ISSAC '89. New York, NY, USA: ACM, 1989, pp. 121–128.
- [18] I. Gohberg and V. Olshevsky, "Complexity of multiplication with vectors for structured matrices," *Linear Algebra Appl.*, vol. 202, pp. 163–192, 1994.
- [19] V. Y. Pan, *Fast Approximate Computations With Cauchy Matrices, Polynomials and Rational Functions*. Cham, Switzerland: Springer International Publishing, 2014, pp. 287–299.
- [20] V. Y. Pan, "How bad are Vandermonde matrices?" *SIAM J. Matrix Anal. Appl.*, vol. 37, no. 2, pp. 676–694, 2016.
- [21] E. E. Tyrtyshnikov, "How bad are Hankel matrices?" *Numerische Mathematik*, vol. 67, no. 2, pp. 261–269, Mar. 1994.
- [22] W. Gautschi and G. Inglese, "Lower bounds for the condition number of Vandermonde matrices," *Numerische Mathematik*, vol. 52, no. 3, pp. 241–250, May 1987.
- [23] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*. Philadelphia, PA, USA: SIAM, 1996.
- [24] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*. Philadelphia, PA, USA: SIAM, 1992.
- [25] N. J. Higham, "Stability analysis of algorithms for solving confluent Vandermonde-like systems," *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 1, pp. 23–41, 1990.
- [26] L. Reichel and G. Opfer, "Chebyshev-Vandermonde systems," *Math. Comput.*, vol. 57, pp. 703–721, 1991.
- [27] A. Fahim, "mmWave LO distribution insights," in *Proc. IEEE Custom Integr. Circuits Conf.*, Apr. 2017, pp. 1–72.
- [28] C. Marcu, "LO generation and distribution for 60 GHz phased array transceivers," Ph.D. dissertation, Dept. Elect. Eng. Comp. Sci., Univ. California, Berkeley, Berkeley, CA, 2011.
- [29] M. Khanpour, K. W. Tang, P. Garcia, and S. P. Voinigescu, "A wideband W-band receiver front-end in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 8, pp. 1717–1730, Aug. 2008.
- [30] P. Y. Chang, S. H. Su, S. S. H. Hsu, W. H. Cho, and J. D. Jin, "An ultra-low-power transformer-feedback 60 GHz low-noise amplifier in 90 nm CMOS," *IEEE Microw. Wireless Compon. Lett.*, vol. 22, no. 4, pp. 197–199, Apr. 2012.
- [31] P. Sakian, E. Janssen, A. H. M. van Roermund, and R. Mahmoudi, "Analysis and design of a 60 GHz wideband voltage-voltage transformer feedback LNA," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 3, pp. 702–713, Mar. 2012.
- [32] S. Shekhar, J. S. Walling, and D. J. Allstot, "Bandwidth extension techniques for CMOS amplifiers," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2424–2439, Nov. 2006.
- [33] A. Mirzaei, M. E. Heidari, R. Bagheri, S. Chehrizi, and A. A. Abidi, "The quadrature LC oscillator: A complete portrait based on injection locking," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 1916–1932, Sep. 2007.
- [34] B. Razavi, "Design of millimeter-wave CMOS radios: A tutorial," *IEEE Trans. Circuits Syst. I, Regular Papers*, vol. 56, no. 1, pp. 4–16, Jan. 2009.
- [35] F. Bruccoleri, E. A. Klumperink, and B. Nauta, "Wide-band CMOS low-noise amplifier exploiting thermal noise canceling," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 275–282, Feb. 2004.
- [36] C.-F. Liao and S.-I. Liu, "A broadband noise-canceling CMOS LNA for 3.1–10.6 GHz UWB receivers," *IEEE J. Solid-State Circuits*, vol. 42, no. 2, pp. 329–339, Feb. 2007.
- [37] S. C. Blaakmeer, E. A. Klumperink, D. M. Leenaerts, and B. Nauta, "Wide-band balun-LNA with simultaneous output balancing, noise-canceling and distortion-canceling," *IEEE J. Solid-State Circuits*, vol. 43, no. 6, pp. 1341–1350, Jun. 2008.
- [38] C. Wijenayake, A. Madanayake, Y. Xu, L. Belostotski, and L. T. Bruton, "A steerable DC-1 GHz all-pass filter-sum RF space-time 2-D beam filter in 65 nm CMOS," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 1276–1279.
- [39] P. Ahmadi, B. Maundy, A. S. Elwakil, L. Belostotski, and A. Madanayake, "A new second-order all-pass filter in 130-nm CMOS," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 63, no. 3, pp. 249–253, Mar. 2016.
- [40] P. Ahmadi, M. H. Taghavi, L. Belostotski, and A. Madanayake, "A 0.13- μm CMOS current-mode all-pass filter for multi-GHz operation," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 12, pp. 2813–2818, Dec. 2015.
- [41] A. Madanayake, N. Udayanga, and V. Ariyaratna, "Wideband delay-sum digital aperture using Thiran all-pass fractional delay filters," in *Proc. 2016 IEEE Radar Conf.*, May 2016, pp. 1–5.
- [42] J.-P. Thiran, "Recursive digital filters with maximally flat group delay," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 6, pp. 659–664, Nov. 1971.
- [43] T. Laakso, V. Valimaki, M. Karjalainen, and U. Laine, "Splitting the unit delay [FIR/all pass filters design]," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, Jan. 1996.

Authors' biographies and photos not available at the time of publication.