Journal of Computational and Applied Mathematics 477 (2026) 117152

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A low-complexity algorithm to digitally uncouple the mutual coupling effect in antenna arrays via symmetric Toeplitz matrices[☆]

Sirani M. Perera [a,*], Levi Lingsch [b], Arjuna Madanayake [c], Leonid Belostotski [d]

[a] Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL 32114, USA
[b] Department of Mathematics, ETH Zurich, Zurich, Switzerland
[c] Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174, USA
[d] Department of Electrical and Software Engineering, University of Calgary, Calgary, Alberta, T2N1N4, Canada

ARTICLE INFO

MSC:
15A23
15B05
15B10
65F50
65T50
65Y04
65Y05
65Y20
78A50
94A12

Keywords:
Mutual coupling effect
Structured matrices
Efficient and recursive algorithms
Complexity and performance of algorithms
Sparse matrices
S-parameters
Signal flow graphs

ABSTRACT

In this paper, we solve systems of linear equations having an $n \times n$ coefficient matrix as a symmetric Toeplitz matrix having elements found via the measured mutual coupling effects of electromagnetic fields caused by antenna array elements. This coefficient matrix is called the mutual coupling matrix. In general, these mutual coupling matrices are characterized as dense matrices. However,building on our prior work, we have introduced a symmetric Toeplitz structure, defining its elements through the self- and mutual coupling effects of antenna array elements. Thus, in this paper, we propose an algorithm to uncouple the mutual coupling effect of antenna arrays using $\mathcal{O}(n\log(n))$ as opposed to $\mathcal{O}(n^3)$ complexity while defining the mutual coupling matrix as a matrix defined by the structure, i.e., a symmetric Toeplitz matrix. The proposed mutually coupled systems will be solved using a sparse factorization of the uncoupling matrices consisting of diagonal and butterfly matrices. The proposed algorithm has low arithmetic complexity compared to brute-force computations in solving systems of linear equations associated with mutual coupling matrices. The proposed factorization also leads to an alternative method to solve the system of linear equations having symmetric Toeplitz matrices as coefficient matrices with $\mathcal{O}(n\log(n))$ as opposed to the $\mathcal{O}(n^3)$ complexity algorithm. To evaluate the accuracy and efficiency of the proposed Toeplitz solver, we have benchmarked our algorithm against highly optimized libraries such as *SciPy*, *NumPy*, and *PyTorch*, specifically focusing on operations involving Toeplitz system solvers and inversion. We show that the proposed Toeplitz solver achieves exceptional efficiency, especially when utilizing GPU acceleration in *PyTorch*, all while maintaining accuracy. For the demonstration of numerical results based on the proposed digital uncoupling algorithm and the effect of attenuation, we use S-parameters at 1.4 GHz of an 8-element sub-array and a 16-element sub-array. We show that the diagonal elements of the uncoupling matrices steadily decrease as one moves away from the main diagonal, highlighting the diminishing effect of mutual coupling and the predominance of self-coupling over mutual coupling. Finally, an 8-element signal flow graph will be presented to show the uncoupling of mutual coupling effects of antenna arrays in digital signal processing perspective.

[☆] This research was partially supported by the National Science Foundation's Division of Electrical, Communications, and Cyber Systems(ECCS), with award numbers 2229473 and 2229471.

[*] Corresponding author.
E-mail address: pereras2@erau.edu (S.M. Perera).

https://doi.org/10.1016/j.cam.2025.117152
Received 12 July 2023; Received in revised form 9 August 2025
Available online 14 October 2025

Nomenclature

In order to enhance the composition of the paper, we state nomenclature based on the frequently utilized terminologies.

Active reflection coefficient Γ_{act}	
ADC	Analog to digital converter
ASIC	Application-specific integrated circuit
BPF	Bandpass filter
Calibration matrix $\mathbf{A}_n$	
CMOS	Complementary metal-oxide semiconductor
Conjugate transpose of the mutual coupling matrix $\mathbf{C}_n^*$	
DFT	Discrete Fourier transform
FFT	Fast Fourier transform
FPGA	Field programmable gate array
Grid size of the antenna array $N \times N$, where $N^2 = n$	
Input vector $\mathbf{y}$	
LNA	Low noise amplifier
LPF	Low pass filter
MSE	mean-squared error
Mutual coupling matrix $\mathbf{C}_n$	
Number of elements in antenna array n	
Output vector $\mathbf{x}$	
RF	Radio frequency
RAM	Random access memory
SFG	Signal flow graph
TCA	Tightly-coupled array
TCDA	Tightly-coupled dipole array
ULA	Uniform linear array
Uncoupling matrix $\mathbf{Q}_n$	
VLSI	Very large scale integration
VNA	Vector network analyzer

1. Introduction

Uniformly spaced antenna arrays are extremely important and play a crucial role in all modern wireless communications, radar (national defense and weather monitoring), microwave imaging for biomedical applications, radio astronomy, location, and navigation. When antenna arrays are engineered, there exist electromagnetic fields that interact between antennas, causing unexpected and undesired distortions of the antenna array. These interactions also cause detrimental issues in system performance in terms of the deviation of the antenna beam patterns and an increase in the noise figure of the receivers. The mutual coupling effect between antenna elements in an array configuration is ever-present. It can be reduced by, for example, using low-scattering antennas, but it can never be removed as it arises due to physics. Near-field interaction of electric and magnetic fields ensures such mutual coupling, which in turn introduces distortions to element radiation patterns of individual antennas and subsequent array processing [1–5]. The impact of mutual coupling at a specific frequency can generally be represented using a complex-valued coupling matrix [1]. Our previous work shows that these mutual coupling matrices can be expressed as symmetric tridiagonal Toeplitz matrices [6]. This representation is particularly relevant for an antenna array comprising another antenna array on each side. In most cases, a digital receiver array consists of an $N \times N$ grid containing $n = N^2$ elements, each paired with direct-conversion receivers, as shown in Fig. 6, in Appendix. Thus, building on our work, we propose developing an understanding of mutual coupling matrices as symmetric Toeplitz matrices. Additionally, we propose an approach to solve systems of equations that utilize these symmetric Toeplitz matrices as coefficient matrices, achieving a reduction in computational complexity from $O(n^3)$ to $O(n \log n)$.

1.1. Mutual coupling effect of antenna-array elements via matrices

Mutual coupling is a function of antenna element design, array geometry, and inter-element spacing. Because electromagnetic fields attenuate rapidly with distance from an antenna, for a given antenna element, the effect of mutual coupling is dominant across its nearest neighbors [3,6]. The mutual coupling can be experimentally measured, using a vector network analyzer (VNA), and included as part of the multi-dimensional filter design problem to ensure the coupling effects are included in the final measured array response [7]. The mutual coupling is both frequency and polarization-dependent. The effect of mutual coupling at a particular frequency can typically be modeled by a complex-valued coupling matrix [1]. The elements in the coupling matrices can be estimated using parametric estimation methods [3,8,9] or Fourier decomposition of measured element patterns [1,4] or measured scattering

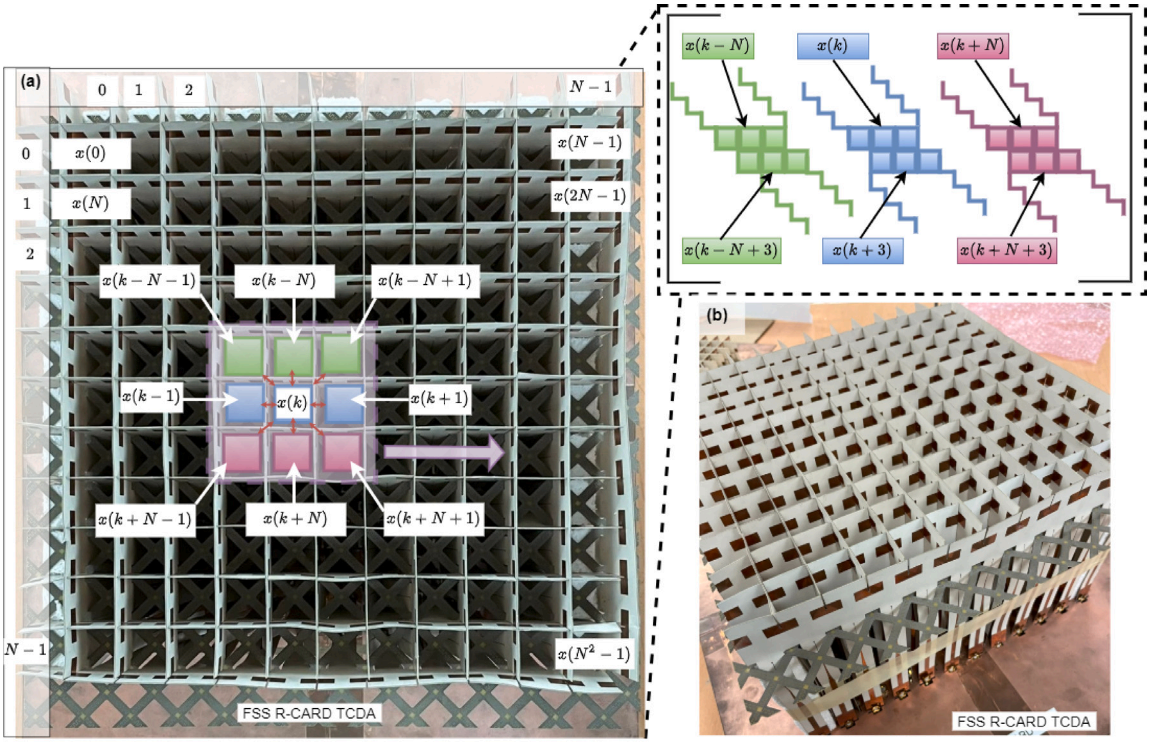


Fig. 1. An example of a tightly-coupled dipole array (TCDA) designed by Volakis et al. [12] consisting of a grid of $N = 11$ (dual polarized) wideband dipole elements each operating over the 130 MHz–6.0 GHz band. Each element in the array is electromagnetically coupled to every other element causing a frequency-dependent linear transform to appear between the incident radio waves and N^2 (for one polarization) radio frequency output port signals (ports at the bottom of the array, not shown here), where $N^2 = n$. It has been shown through measurements of S-parameter matrices that the electromagnetic coupling matrix can be closely approximated by a symmetric (due to the same mutual coupling effect on the array elements), banded(when the mutual coupling effects dominate across nearest neighbors), and unitary (due to passivity) Toeplitz matrix. The authors thank Satheesh Bojja Venkatakrishnan for images of the TCDA [12].

(S-) parameter-based formulations that describe the gain and phase of each coefficient [4,6,10]. The mutual coupling matrices are full and dense matrices. On the other hand, by following the radio-frequency (RF) measurements it was confirmed that the coupling is small for antenna arrays when the distance between antennas increases above a wavelength [6]. The S-parameters are not directly related to beamformer behavior due to the effects of the reflection coefficient at the LNA input port and the finite propagation delays due to transmission-line effects of the finite-length transmission lines that appear between antenna ports and LNAs [11].

Since an $N \times N$ array contains $n = N^2$ elements where every element couples to every other element, the S-parameters lead to a coupling matrix, which is an $(n \times n)$ square matrix. The paper [6] employed an S-parameter-based approach in a narrowband at a particular frequency and introduced a new structure for the mutual coupling matrices, so-called symmetric tridiagonal Toeplitz matrices, and stated a fast algorithm to furnish the uncoupling effect of the mutually coupled antenna elements for a low-order coupling function where each antenna is coupled to its nearest neighbors. In this paper, we extend the result in [6] while proposing a low-complexity system-solving algorithm having symmetric Toeplitz matrices as the coefficient matrix to facilitate higher-order coupling, albeit at greater system complexity.

The support for higher-order coupling is desired for extending the digital mutual coupling analysis to emerging high-bandwidth array geometries, such as magnetic photonic crystal-based tightly-coupled arrays (TCAs) that exhibit usable frequency ranges spanning several octaves at the price of high mutual coupling between elements [12]. Further, extension to square (or rectangular) array geometries from the original uniform linear array (ULA) necessitates the adoption of the Toeplitz matrix (banded when coupling effects get minor beyond k-neighbors) models for the mutual coupling coefficients when an $N \times N$ rectangular antenna array elements are represented by a vector containing $n = N^2$ elements. Fig. 1 shows how an $N \times N$ square array of uniformly spaced elements have a vector representation $\mathbf{x} = (x(0)\ x(1) \cdots x(n-1))^T$. Fig. 1 further shows a real-world wideband square-shaped TCA having a frequency range of 130 MHz–6 GHz and multiple applications in wireless communications, radar, microwave imaging, sensing, and multi-functional electronic warfare systems. We identify the dominant mutual coupling as those taking the form of banded Toeplitz matrices when the array signals are represented as an n-element column vector.

1.2. Banded Toeplitz and few other Toeplitz solvers

Many inversion techniques proposed for the banded and tridiagonal matrices are tested on banded Toeplitz matrices [13]. The manuscript [14] provides an explicit formula for the inverse of a tridiagonal Toeplitz matrix via the solution of finite difference equations. The formulas in [14] neither provide sparse factorization nor generator representation of semiseparable matrices [13]. The exact formula to compute the inverse of a specific covariance matrix in the symmetric tridiagonal Toeplitz matrix form was presented in [15]. The paper [15] uses the same technique as proposed in [14] but in a simpler way. It was identified in [13] that the inverse of covariance matrices can be seen as semiseparable plus diagonal matrices. It was proved in [16], that it is sufficient to calculate the band surrounding the diagonal to calculate the inverse of banded Toeplitz matrices. But the paper [16] does not provide the sparse factorization for the inverse of banded Toeplitz matrices. The inverse of the nonsymmetric tridiagonal matrix and its adaptation to the Toeplitz matrix was discussed in [13]. In [17], an inversion formula for tridiagonal 2-Toeplitz and tridiagonal 3-Toeplitz matrices is provided, with inversion of general tridiagonal k-Toeplitz matrices in [18]. The complexity of algorithms in [17] exceeds $\mathcal{O}(n)$ and does not have sparse factorization. In [19], inverses of tridiagonal and banded Toeplitz matrices are obtained by solving difference equations and provided with representation in terms of generators but did not include the sparse factorization for the inverse of matrices. Formulas in [19] are similar to those in [14]. The manuscript [20] presents a fast algorithm for the inversion of Toeplitz matrices (not necessarily banded) using displacement equations and Fourier representation of Toeplitz matrices. The proposed algorithm in [20] has the cost of 4 length n FFT algorithms having the complexity of order $\mathcal{O}(n\ \log\ n)$ plus $\mathcal{O}(n)$. The authors in [20] have compared their method with the $\mathcal{O}(n \log^2 n)$ algorithms in [21–23] to compute the inverse of Toeplitz matrices. The superfast approximation algorithm was proposed in [24] to compute the inverse of block Toeplitz matrices with Toeplitz blocks (not necessarily banded) by using displacement equations and a modified Newton method. But the proposed algorithm in [24] does not contain the sparse matrix factorization and is not an exact algorithm i.e. it is an approximation. The technical report [25] proposed a superfast divide and conquer algorithm to solve linear systems of equations having a coefficient matrix such as the Toeplitz or Hankel using displacement equations. The manuscript [26] proposed a fast algorithm to solve non-singular Toeplitz or Hankel system of linear equations with $\mathcal{O}(m(n)\, n\, \mu(\log\ n))$ where $m(n)$ and $\mu(d)$ bound the arithmetic and boolean cost of multiplying polynomials of degree n and integers modulo $2^d + 1$ using displacement equations. But algorithms in [25,26] do not provide sparse factorization to solve the non-singular Toeplitz system of linear equations. Apart from the algorithms to solve the system of equations having banded Toeplitz matrices, one can observe displacement approaches to compute the inverse of Toeplitz matrices in [27–30] with low arithmetic complexity. Moreover, there are fast algorithms that approximate the solution of Toeplitz and Toeplitz-like linear systems of equations by using a nearly linear number of arithmetic operations [31,32]. Furthermore, one could observe the spectrum of finite and infinite orthogonal and symmetric Toeplitz and the generation of bounded operators based on the spectrum in [33]. Additionally, it is important to highlight that the explicit inverses of Toeplitz and Hankel matrices are known as T or H Bezoutian matrices [34–37], and these algorithms have the complexity of $\mathcal{O}(n^2)$ operations. Concerning the extensive literature behind Toeplitz matrices, we finally refer to books [13,38] for more literature reviews on solving banded Toeplitz and Toeplitz systems. Furthermore, we highlight from [38] that solving a Toeplitz system cost a complexity of $\mathcal{O}(n^2)$. This reduction in complexity is referred to as a fast algorithm in [38]. The same source also notes that utilizing the preconditioned conjugate gradient method for approximating the solution of a Toeplitz system can achieve a lower complexity of $\mathcal{O}(n \log(n))$. Therefore, one could say the need to solve exact Toeplitz systems utilizing 2FFTs, especially in light of the extensive literature on Toeplitz system solvers. Consequently, our emphasis is to uncouple the mutual coupling effect of antenna arrays by employing the product of sparse matrices to efficiently solve exact symmetric Toeplitz systems using 2FFTs with $\mathcal{O}(n \log(n))$ as opposed to $\mathcal{O}(n^3)$ brute-force complexity.

1.3. Organization of the paper

The paper is organized as follows. Section 2 proposes structured matrix-based uncoupling architecture for the mutually coupled elements of antenna arrays, while Section 3 proposes sparse factorization for the uncoupled matrices leading to a low-complexity, exact, radix-2, and recursive algorithm using 2 FFTs. It is important to highlight that Section 3 does not suggest a method for computing the Toeplitz matrix–vector product using the FFT algorithm, as this is a widely recognized approach typically accomplished through standard 3FFTs, with a few exceptions where 2FFTs have been proposed for solving these systems [38–40]. Our contribution, however, lies in deriving an exact solution for Toeplitz systems via the sparse factorization of the uncoupling matrix. Section 4 shows the arithmetic complexity (quantified using the number of necessary adders and gains) and associated numerical results for the proposed algorithm and compares them with the brute-force system solver. In Section 5, We use S-parameters at 1.4 GHz of an 8-element sub-array and a 16-element sub-array for the demonstration of numerical results based on the proposed digital uncoupling algorithm and the effect of the attenuation. In Section 6, the proposed algorithm is implemented via an 8-element signal flow graph, while Section 7 concludes the paper.

2. Structured matrix interpretation of the uncoupling architecture

This section starts by defining mutual coupling matrices as a structured matrix, called a symmetric Toeplitz matrix, and ends by showing the existence of an uncoupling matrix to decouple the mutual coupling effects in antenna arrays. We note here that in radio receivers, the concept of active reflection coefficient, Γ_{act}, was introduced to explain the impact of mutual coupling on the sensitivity of the array [41,42]. However, Γ_{act} depends on the beamformer and differs from an isolated antenna element reflection coefficient due to mutual coupling [41]. Thus, to mitigate this beam dependence, decoupling networks have been investigated at the antenna ports [43–49].

2.1. Definition: Mutual coupling matrix

Let us consider a normalized mutual coupling of a uniform linear array of n elements. Then, the $n \times n$ mutual coupling matrix $\mathbf{C}_n$ can be described as a symmetric Toeplitz matrix of the form,

$$\mathbf{C}_{i,l} = \begin{cases} a, & i = l \\ b_{|i-l|}, & i \neq l \end{cases} = \begin{bmatrix} a & b_1 & b_2 & \cdots & b_{n-1} \\ b_1 & a & b_1 & \ddots & \vdots \\ b_2 & b_1 & a & \ddots & b_2 \\ \vdots & \ddots & \ddots & \ddots & b_1 \\ b_{n-1} & \cdots & b_2 & b_1 & a \end{bmatrix} \tag{1}$$

where $a = \alpha e^{-j\theta} \in \mathbb{C}$ correspondence to the self-coupling, $b_{|i-l|} \in \mathbb{C}$ correspondence to mutual coupling from the neighboring elements, $b_{i-l} = b_{l-i}$ due to the same mutual coupling effect on the array elements, $\alpha \leq 1$ and $|b_{|i-l|}| \leq 1$ due to passivity (all the antenna elements are connected to a single transmitter and/or receiver), $a \neq b_{|i-l|}$, $j^2 = -1$, $|b_{|i-l|}| > |b_{|i-l|+1}|$ because mutual coupling reduces monotonically with distance from the antenna to its neighbors, α captures the effect of the reflection coefficient combined with the antenna impedance, θ is the phase angle.

One could also simplify the structure of the mutual coupling matrix from Toeplitz into a $(k+1)$-banded Toeplitz matrix (i.e. $b_{|i-l|} = 0$ for $|i-l| > k/2$) due to the fact that the mutual coupling is dominant across its few nearest neighbors [3,6]. The mutual coupling matrix $\mathbf{C}_n$ embeds the frequency-dependent behavior of the measured electromagnetic effects between elements, antenna driving point impedance function, the characteristic impedance of the transmission lines, and frequency-dependent behavior of the LNA driving point impedance, in the Toeplitz matrix having complex coefficients(symmetric coefficients across the diagonal).

2.2. Existence of an uncoupling matrix

Acknowledging the passivity of the system, the inverse of the coupling matrix exists and can be represented as $\mathbf{Q}_n = \mathbf{C}_n^{-1}$ [50]. According to the mutual coupling matrix (1), the mutual coupling effect diminishes consistently as the distance between the antenna and its neighbors increases. Specifically, we observe that $|b_{|i-l|}| > |b_{|i-l|+1}|$ with $|b_{|i-l|}| \leq 1$, and $b_{|i-l|} = 0$ for $|i-l| > k/2$ because the mutual coupling primarily influences only a few of the nearest neighbors, i.e., self-coupling dominates over mutual coupling within the coupling matrix. Furthermore, the mutual coupling is a spatial linear transformation from $\mathbb{C}^n$ to $\mathbb{C}^n$ s.t. $\mathbf{x} \mapsto \mathbf{C}_n\mathbf{x}$ ensuring $\mathbf{x}_1 \neq \mathbf{x}_2 \mapsto \mathbf{C}_n\mathbf{x}_1 \neq \mathbf{C}_n\mathbf{x}_2$, where $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{C}^n$. Thus, by invertible matrix theorem, the existence of $\mathbf{Q}_n$ is guaranteed, and hence the coupling effect can be reversed (i.e., uncoupled) in the digital signal processor using a brute-force linear system solver at every time step, i.e. solving $\mathbf{C}_n\mathbf{x} = \mathbf{y}$ is equivalent in calculating $\mathbf{x} = \mathbf{C}_n^{-1}\mathbf{y} = \mathbf{Q}_n\mathbf{y}$. The existence of $\mathbf{Q}_n$ is significant because it is possible to digitally undo the mutual coupling effects of antenna arrays by multiplying the array outputs by the uncoupled matrix $\mathbf{Q}_n$ at each time step.

On the other hand, Proposition 3.1 followed by the Algorithms in Section 3 show that we have not computed the explicit inverse of the symmetric Toeplitz matrices $\mathbf{C}_n$, but rather computing the conjugate transpose of the mutual coupling matrix $\mathbf{C}_n^*$ followed by the calibration matrix $\mathbf{A}_n$. The calibration matrix $\mathbf{A}_n$ is diagonal and can be found numerically as stated in Remark 3.2, experimentally by illuminating the array with a plane-wave at broadside direction and shown numerically through Section 6. Finally, we recall from [51] that the explicit inverse of a matrix is hardly ever computed, especially for linear system solutions, because this is too expensive and numerically less accurate. We emphasize that we have not utilized explicit inverse to solve symmetric Toeplitz systems; rather, we have bypassed it through conjugate transpose of symmetric Toeplitz matrices followed by the calibration matrix.

The most straightforward realization would require measurement of S-parameters, one-time pre-computation of $\mathbf{C}_n^{-1}$ using $\mathbf{C}_n$ and realization of $\mathbf{x} = \mathbf{C}_n^{-1}\mathbf{y}$ where $\mathbf{y}$ is the n-element vector of samples at a given time step obtained from n-receivers in the array, and $\mathbf{x}$ is the n-element vector of samples corresponding to the n-elements albeit with mutual coupling effects removed using digital hardware. The complexity of the brute-force computation of $\mathbf{x}$ is $\mathcal{O}(n^3)$. Although perfectly reasonable for small arrays (for example, when $n = 8$) the arithmetic complexity grows in cubic powers with array size and becomes computationally difficult even for reasonably large n. In a typical phased-array radar containing hundreds of elements, the complexity of the brute-force system solver can become expensive in terms of both chip area and power consumption. To reduce the arithmetic complexity to a manageable level, we explore the structure of the mutual coupling matrices to facilitate a sparse matrix factorization that in turn allows the desired computation albeit at a significantly reduced complexity, i.e. $\mathcal{O}(n \log n)$, when compared to the brute-force approach as in the next section.

3. A low-complexity algorithm to solve symmetric Toeplitz systems: Decouple the mutual coupling effect

The realization of uncoupling caused by the mutual coupling effect can be obtained by solving a symmetric Toeplitz system of the form $\mathbf{C}_n\mathbf{x} = \mathbf{y}$, and is equivalent in calculating $\mathbf{x} = \mathbf{Q}_n\mathbf{y}$. For this realization, the coefficients defining the symmetric Toeplitz matrix $\mathbf{C}_n$, i.e. $a, b_{|i-l|} \in \mathbb{C}$ can be found experimentally via S-parameter measurements with a vector network analyzer(VNA). Let us first introduce all the notations before discussing the factorization for the uncoupling matrix $\mathbf{Q}_n$.

3.1. Other frequently used explicit matrices

Here, we introduce the remaining notations for sparse and orthogonal matrices which will frequently be used in this paper. We recall that $n \times n$ coupling matrix $\mathbf{C}_n$ has identified as a symmetric Toeplitz matrix (1), $\mathbf{Q}_n$ is the uncoupling matrix, and $\mathbf{C}_n^*$ is the conjugate transpose of $\mathbf{C}_n$. In 2.2, we have introduced $\mathbf{A}_n$ as the calibration matrix consisting only of diagonal elements. This matrix will be determined numerically or experimentally, as mentioned in Remark 3.2. Additionally, it will be numerically derived in Section 5 and illustrated in Section 6.

For a given vector $\mathbf{x} = [x_0, x_1, \dots, x_{n-1}]^T \in R^n$, let us introduce an even–odd permutation matrix $\mathbf{P}_n$ $(n \geq 3)$ by

$$\mathbf{P}_n\,\mathbf{x} = \begin{cases} \left[x_0, x_2, \dots, x_{n-2}, x_1, x_3, \dots, x_{n-1}\right]^T & \text{even } n, \\ \left[x_0, x_2, \dots, x_{n-1}, x_1, x_3, \dots, x_{n-2}\right]^T & \text{odd } n. \end{cases}$$

We define the DFT matrix by $\mathbf{F}_n = \frac{1}{\sqrt{n}}\,[w_n^{kl}]_{k,l=0}^{n-1}$ where $n = 2^t (t \geq 1)$, $w_n = e^{-\frac{2\pi j}{n}}$, a scaled DFT matrix by $\tilde{\mathbf{F}}_n = \sqrt{n}\,\mathbf{F}_n$ and its conjugate transpose by $\mathbf{F}_n^*$, a highly sparse matrix by $\mathbf{J}_{m\times n} = \begin{bmatrix} \mathbf{I}_n \\ \hline \mathbf{0}_n \end{bmatrix}$ where $m = 2n$, $\mathbf{I}_n$ is the identity matrix and $\mathbf{0}_n$ is the zero matrix, a scaled orthogonal matrix by $\mathbf{H}_n = \begin{bmatrix} \mathbf{I}_{\frac{n}{2}} & \mathbf{I}_{\frac{n}{2}} \\ \grave{\mathbf{D}}_{\frac{n}{2}} & -\grave{\mathbf{D}}_{\frac{n}{2}} \end{bmatrix}$ where $\grave{\mathbf{D}}_{\frac{n}{2}} = \text{diag}\left[w_n^l\right]_{l=0}^{\frac{n}{2}-1}$, and its conjugate transpose by $\mathbf{H}_n^*$, diagonal matrices by $\tilde{\mathbf{D}}_m = \text{diag}\left[\tilde{\mathbf{F}}_m\mathbf{r}\right] = \text{diag}\left[f_i\right]_{i=1}^m$ and $\mathbf{D}_m = \text{diag}\left[d_i\right]_{i=1}^m$, where $d_i = f_i^*$ (i.e. d_i is the conjugate of f_i^*) and $\mathbf{r}$ s.t. $\mathbf{r} = \left[a, b_1, b_2, \dots, b_{(n-1)}, a, b_{(n-1)}, \dots, b_2, b_1\right]^T$. is the first column of a circulant matrix $\mathbf{R}_m$.

3.2. Sparse and recursive factors for uncoupling matrices and to solve symmetric Toeplitz systems

This section proposes sparse and recursive factors to uncouple the mutual coupling effect, i.e. solve $\mathbf{C}_n\mathbf{x} = \mathbf{y}$ or equivalently computing $\mathbf{x} = \mathbf{Q}_n\mathbf{y}$ while proposing a sparse factorization for the uncoupling matrix $\mathbf{Q}_n$. We clarify that our primary objective is to develop an algorithm that effectively solves symmetric Toeplitz-structured systems, not to compute the Toeplitz-vector product or the explicit inverse of the Toeplitz matrices. We propose an approach to compute the uncoupling matrix $\mathbf{Q}_n$ and factor it into sparse matrices yielding an $\mathcal{O}(n \log n)$ algorithm, to decouple the mutual coupling effects via digital signal processing.

Proposition 3.1. *Let the mutual coupling matrix $\mathbf{C}_n$, where $(n = 2^t)$, is given via (1). Then the mutual coupling effect can be uncoupled, i.e. the linear system $\mathbf{C}_n\mathbf{x} = \mathbf{y}$ can be solved or equivalently $\mathbf{x} = \mathbf{Q}_n\mathbf{y}$ can be computed using*

$$\mathbf{x} = [\mathbf{J}_{m\times n}]^T \mathbf{F}_m^* \mathbf{D}_m \mathbf{F}_m \mathbf{J}_{m\times n} \mathbf{A}_n \mathbf{y}. \tag{2}$$

Proof. Identifying the mutual coupling matrix $\mathbf{C}_n$ as a symmetric Toeplitz matrix, it can be fully determined by the by its first column (also row) $\left[a, b_1, b_2, \dots, b_{n-1}\right]^T$. We use the matrix embedding to construct a circulant matrix using the Toeplitz matrices [38,39,52] s.t.

$$\mathbf{R}_m = \begin{bmatrix} \mathbf{C}_n & \hat{\mathbf{C}}_n \\ \hat{\mathbf{C}}_n & \mathbf{C}_n \end{bmatrix}, \tag{3}$$

where the symmetric Toeplitz matrix $\hat{\mathbf{C}}_n$ is defined by its first column (also row) $\left[a, b_{(n-1)}, \dots, b_2, b_1\right]^T$. We use the similarity transformation of the circulant matrix defined via the non-singular DFT matrices using 2-DFT matrices as in [52,53] s.t.

$$\mathbf{R}_m = \mathbf{F}_m^* \tilde{\mathbf{D}}_m \mathbf{F}_m. \tag{4}$$

Now, we scale the $\mathbf{R}_m$ matrix by rectangular sparse matrices to extract the mutual coupling matrix $\mathbf{C}_n$ s.t.

$$\mathbf{C}_n = \left[\begin{array}{c|c} \mathbf{I}_n & \mathbf{0}_n \end{array}\right] \mathbf{R}_m \begin{bmatrix} \mathbf{I}_n \\ \hline \mathbf{0}_n \end{bmatrix}. \tag{5}$$

Since $\mathbf{A}_n$ is the calibration matrix (a diagonal matrix) it can be found experimentally or numerically as stated in Remark 3.2 so that we can uncouple the mutual coupling effect using $\mathbf{A}_n$ followed by the sparse factorization of $\mathbf{C}_n^*$ leads to a parallel digital realization. Thus, we can solve $\mathbf{C}_n\mathbf{x} = \mathbf{y}$ or equivalent in calculating $\mathbf{x} = \mathbf{Q}_n\mathbf{y}$ via

$$\mathbf{x} = \mathbf{C}_n^* \mathbf{A}_n \mathbf{y} = \left[\begin{array}{c|c} \mathbf{I}_n & \mathbf{0}_n \end{array}\right] \mathbf{F}_m^* \mathbf{D}_m \mathbf{F}_m \begin{bmatrix} \mathbf{I}_n \\ \hline \mathbf{0}_n \end{bmatrix} \mathbf{A}_n \mathbf{y}. \tag{6}$$

□

Remark 3.2.

1. We can determine the matrix $\mathbf{A}_n$ numerically. The first step involves computing $\mathbf{C}_n\mathbf{C}_n^* := \mathbf{M}_n$ (say) followed by $\mathbf{M}_n^{-1}\mathbf{y} = \tilde{\mathbf{a}}$ (say). It is important to highlight that $\mathbf{M}_n$ is a Hermitian matrix, and consequently, $\mathbf{M}_n^{-1}$ also possesses this property. Additionally, referring to Section 2.2, since $\mathbf{C}_n$ is invertible, it follows that $\mathbf{M}_n$ is invertible. Thus, $\tilde{\mathbf{a}}$ exists. Upon obtaining $\tilde{\mathbf{a}}$, we compute the component-wise conjugate of $\tilde{\mathbf{a}}$. Subsequently, we divide each element by its squared magnitude to calculate $\mathbf{a} := [a_0, a_1, \dots, a_{n-1}]^T \in \mathbb{C}^n$. Once we have $\mathbf{a}$, we construct the diagonal matrix $\mathbf{A}_n$ with these elements. We then compute the matrix–vector product $\mathbf{A}_n\mathbf{y}$, and subsequently multiply the result by $\mathbf{C}_n^*$ to yield $\mathbf{x}$.

2. The calibration matrix $\mathbf{A}_n$ can be found experimentally by illuminating the array with a plane wave at a broadside direction, leading to a parallel digital realization.
3. We note from Proposition 3.1 that the uncoupling matrix can be given via $\mathbf{Q}_n = \mathbf{C}_n^* \mathbf{A}_n = [\mathbf{J}_{m\times n}]^T \mathbf{F}_m^* \mathbf{D}_m \mathbf{F}_m \mathbf{J}_{m\times n} \mathbf{A}_n$, and hence $\mathbf{Q}_n$ could be utilized to uncouple the mutual coupling effect of antenna arrays, rather than computing the explicit inverse of the mutual coupling matrix.
4. Our numerical results in Section 5 demonstrate that when the matrix $\mathbf{C}_n$ is symmetric, it is possible to incorporate a calibration matrix $\mathbf{A}_n$ for the input signals prior to performing the factorization of the uncoupling matrix $\mathbf{Q}_n$.
5. We have proposed solving a symmetric Toeplitz system to uncouple the mutual coupling effect of antenna arrays via a sparse factorization of the uncoupling matrix, but have not computed nor proposed the well-known Toeplitz-vector product using FFTs, in this paper. The linear transformation or the linear system solving problems based on Toeplitz matrices, such as computing the Toeplitz-vector product or banded Toeplitz-vector product or other Toeplitz solvers are in Section 1.2, and one could also find the inverse of a Toeplitz matrix via displacement structure in [52,54–57], and hence are not proposed in this paper.

3.3. Radix-2 and recursive algorithms to digitally uncouple the mutual coupling via symmetric Toeplitz systems

Following the factorization proposed for uncoupling the mutual coupling effect in Section 3.2, we present radix-2 algorithms for the uncoupling matrix $\mathbf{Q}_n$ which execute recursively with scaled 2 FFT algorithms (to reduce the multiplication count we have moved the scaling factor at the end of the computation). Hence in computing the uncoupled matrix–vector product, we have moved the factor $\frac{1}{\sqrt{m}}$ in $\mathbf{F}_m$ and $\mathbf{F}_m^*$ to the end of the computation, and computed $\mathbf{x} = m\mathbf{Q}_n\mathbf{y}$ for a given n, $a, b_{|i-l|} \in \mathbb{C}$, $\mathbf{y} \in \mathbb{C}^n$ and $\mathbf{r} \in \mathbb{C}^m$.

Algorithm 3.3.
Input: $n = 2^t$ $(t \geq 1)$, $m = 2n$, $a, b_{|i-l|} \in \mathbb{C}$, $\mathbf{y} \in \mathbb{C}^n$, *and* $\mathbf{r} \in \mathbb{C}^m$.
Output: $\mathbf{x} = m\mathbf{Q}_n\mathbf{y}$.
Function: $\mathbf{x}$=*uncoupl*($\mathbf{y}$, n).

```
1. if $n = 2$, then
     $\mathbf{x} \leftarrow \frac{1}{a^2-b_1^2}\begin{bmatrix} a & -b_1 \\ -b_1 & a \end{bmatrix}\mathbf{y}$,
2. end if
3. if $n \geq 4$, then
     $\mathbf{z} \leftarrow \mathbf{A}_n \cdot \mathbf{y}$,
     $\mathbf{u}_1 \leftarrow \mathbf{J}_{m\times n} \cdot \mathbf{z}$,
     $\mathbf{v}_1 \leftarrow dft(\mathbf{u}_1, m)$,
     $\mathbf{v}_2 \leftarrow \mathbf{D}_m \cdot \mathbf{v}_1$,
     $\mathbf{x}_1 \leftarrow idft(\mathbf{v}_2, m)$,
     $\mathbf{x} \leftarrow [\mathbf{J}_{m\times n}]^T \cdot \mathbf{x}_1$,
4. end if
5. return $\mathbf{x}$
```

The proposed *uncoupl* algorithm executes recursively with scaled FFTs in [58,59]. Let us refer to the scaled FFT and scaled inverse FFT algorithms by $\mathbf{v}_1 = dft(\mathbf{u}_1, m)$ and $\mathbf{x}_1 = idft(\mathbf{v}_2, m)$, respectively.

Algorithm 3.4.
Input: $m = 2^{t_1}$ $\left(t_1 \geq 1\right)$, $m_1 = m/2$, *and* $\mathbf{u}_1 \in \mathbb{C}^m$.
Output: $\mathbf{v}_1 = \tilde{\mathbf{F}}_m\mathbf{u}_1$.
Function: $\mathbf{v}_1$=*dft*($\mathbf{u}_1$, m).

```
1. if $m = 2$, then
     $\mathbf{v}_1 \leftarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\mathbf{u}_1$,
2. end if
3. if $m \geq 4$, then
     $\mathbf{p} \leftarrow \mathbf{H}_m \cdot \mathbf{u}_1$
     $\mathbf{s}_1 \leftarrow dft(\mathbf{p}(1 : m_1), m_1)$,
     $\mathbf{s}_2 \leftarrow dft(\mathbf{p}(m_1 + 1 : m), m_1)$,
     $\mathbf{v}_1 \leftarrow \mathbf{P}_m^T \cdot \left[\mathbf{s}_1^T \quad \mathbf{s}_2^T\right]^T$,
4. end if
5. return $\mathbf{v}_1$
```

Algorithm 3.5.
Input: $m = 2^{r_1}\ (r_1 \geq 1)$, $m_1 = m/2$, *and* $\mathbf{v}_2 \in \mathbb{C}^m$.
Output: $\mathbf{x}_1 = \tilde{\mathbf{F}}_m^* \mathbf{v}_2$.
Function: $\mathbf{x}_1$=*idft*($\mathbf{v}_2$, *m*).

```
1. if $m = 2$, then
     $\mathbf{x}_1 \leftarrow \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \mathbf{v}_2$,
2. end if
3. if $m \geq 4$, then
     $\mathbf{q} \leftarrow \mathbf{P}_m \cdot \mathbf{v}_2$
     $\mathbf{b}_1 \leftarrow idft(\mathbf{q}(1 : m_1), m_1)$,
     $\mathbf{b}_2 \leftarrow idft(\mathbf{q}(m_1 + 1 : m), m_1)$,
     $\mathbf{x}_1 \leftarrow \mathbf{H}_m^* \cdot \begin{bmatrix} \mathbf{b}_1^T & \mathbf{b}_2^T \end{bmatrix}^T$,
4. end if
5. return $\mathbf{x}_1$
```

Remark 3.6.

1. The sole condition for the algorithm's termination for $n = 2$ is the scenario where $a = b_1$. However, this situation does not occur in practice, as self-coupling fundamentally differs from mutual coupling, leading to $a \neq b_{|i-l|} \forall\, i, j$, as stated in Section 2.1. On the other hand, the condition $a = b_1$ becomes irrelevant for the mutual coupling matrices when $n \geq 4$. This is because the stipulation of $n = 2$ is not influenced by the algorithm's execution for $n \geq 4$.
2. We note that the proposed algorithms can be executed for the size of the mutual coupling matrices n such that $n = 2^t$ where $t \geq 1$.
3. The proposed algorithm to uncouple the mutual coupling effect executes recursively with the FFT and inverse FFT algorithm as of the paper [39]. But the proposed algorithm is different from the delay Vandermonde matrix (DVM) algorithm in [39] because the $\mathbf{x}$ = *uncoupl*($\mathbf{y}$, *n*) algorithm is based on a linear system solver having the mutual coupling matrix (which is a Toeplitz-structured matrix) as the coefficient matrix, i.e., we are not computing the Toeplitz matrix–vector product or a Vandermonde-structured matrix by a vector in [39], but rather solves a mutual coupling system (which is solving a Toeplitz system using 2FFTs).

4. Complexity of the algorithm

4.1. Complexity of systolic array processors

The proposed fast algorithms must be realized as real-time stream processors on an application-specific integrated circuit (ASIC) or field programmable gate array (FPGA) realization. The array outputs are typically bandlimited using suitable microwave filters before being amplified and down-converted to baseband using frequency mixer circuits before being converted to the discrete domain using dedicated analog to digital converters (ADCs) that time synchronously sample each of the down-converted outputs pertaining to the antenna elements in the array. For an $n-$ element array, there typically exists n number of parallel ADCs that feed the digital signal processor with sample values, with a new frame of $n-$ samples per clock cycle. The clock period of the system is equal to the ADC sample period and is the reciprocal of the clock frequency of the digital system.

The proposed algorithms for efficiently uncoupling the electromagnetic coupling between antenna elements albeit in the digital domain can be achieved by architecting a custom arithmetic processor that consists of a fully parallel systolic array realization of the proposed Toeplitz system solver algorithm using multiplier and adder/subtractor blocks realized as parallel processing digital logic circuits. Typically, the processors would operate based on fixed-point digital arithmetic conformant to the two's complement number representation in binary. A massively-parallel fully-pipelined systolic array processor that directly implements the fast algorithm for digital uncoupling of electromagnetic mutual coupling necessitates the parallel realization of very large scale integration (VLSI) integrated circuits using a standard cell library. Such standard cell libraries are available from chip foundries, such as the Taiwan Semiconductor Manufacturing Corporation (TSMC) 65 nm CMOS library, which can be employed by processor designers to realize custom algorithms as ASICs. The chip area and power consumption of the processors will be dominated by the arithmetic complexity of the core algorithm; to wit, it is imperative to minimize the number of multiplications per second to reduce power consumption, as well as the number of multipliers on the chip to reduce chip area and therefore implementation cost. The proposed fast algorithm aims to reduce the number of multiplications in the digital decoupling operation.

4.2. Arithmetic complexity of the algorithm

We determine the number of additions and multiplications, denoted as $\alpha(\mathbf{x}, n)$ and $\beta(\mathbf{x}, n)$, required to compute the length n vector $\mathbf{x}$ while uncoupling the mutual coupling effect. Consequently, the values of $\alpha(\mathbf{x}, n)$ and $\beta(\mathbf{x}, n)$ correspond to the adders and gains needed for the algorithms presented in Section 3.2. These counts are obtained to compute the uncoupling algorithm, i.e., $\mathbf{x} = uncoupl(\mathbf{y}, n)$, which executes recursively with the scaled FFTs in [58]. Thus, we take the number of additions and multiplications required to compute $\mathbf{v} = \tilde{\mathbf{F}}_n\mathbf{u}$, where $\mathbf{u} \in \mathbb{C}^n$, as nt and $\frac{1}{2}nt - \frac{3}{2}n + 2$, respectively, when the multiplications by ± 1 and $\pm j$ are not counted.

Lemma 4.1. *Let $n = 2^t (t \geq 1)$ be given. The uncoupling algorithm, i.e., $\mathbf{x} = uncoupl(\mathbf{y}, n)$, can recursively be computed using $\mathbf{v}_1 = dft(\mathbf{u}_1, m)$, and $\mathbf{x}_1 = idft(\mathbf{v}_2, m)$ algorithms with the following arithmetic complexities (respectively, adders and gains):*

$$\begin{aligned} &\alpha(\mathbf{x}, n) = 4nt + n, \\ &\beta(\mathbf{x}, n) = 2nt - n + 4. \end{aligned} \tag{7}$$

Proof. Referring to the uncoupling algorithm, we get

$$\alpha(\mathbf{x}, n) = 2\left(\alpha(\tilde{\mathbf{F}}_m), m\right) + 2\left(\alpha(\mathbf{J}_{m\times n}, m \times n)\right) + \alpha(\mathbf{D}_m, m) + \alpha(\mathbf{A}_n, n). \tag{8}$$

A similar equation holds for the number of multiplications as well. Following the structures of $\mathbf{A}_n$, $\mathbf{J}_{m\times n}$, and $\mathbf{D}_m$, and the multiplication of each matrix by a complex input, we have

$$\begin{aligned} \alpha\left(\mathbf{A}_n, n\right) &= 0, \quad \beta\left(\mathbf{A}_n, n\right) = n, \\ \alpha\left(\mathbf{J}_{m\times n}, m \times n\right) &= 0, \quad \beta\left(\mathbf{J}_{m\times n}, m \times n\right) = 0, \\ \alpha\left(\mathbf{D}_m, m\right) &= 0, \quad \beta\left(\mathbf{D}_m, m\right) = m. \end{aligned}$$

Multiplications of $\mathbf{J}_{m\times n}$ from the right and left of the DFT do not affect the multiplication counts, but do affect the addition counts (due to $\mathbf{0}_n$ matrix). Hence, we have to subtract $m+n$ addition counts (due to no m and n additions after and before $\mathbf{J}_{m\times n}$, respectively) from the total addition counts. By following nt additions and $\frac{1}{2}nt - \frac{3}{2}n + 2$ multiplications to compute $\mathbf{v} = \tilde{\mathbf{F}}_n\mathbf{u}$ with $\mathbf{u} \in \mathbb{C}^n$, we get addition and multiplication counts as in (7) based on the computation of the proposed uncoupling algorithm. □

5. Numerical results

To demonstrate the effectiveness of the decoupling approach, we take simulated data from [60] of a 71-element array. This array was developed for a Square Kilometer Array and the impact of mutual coupling on the array prototypes was investigated [11,61–64]. This array is a dual-polarization square array with 36 elements realizing vertical polarization and 35 elements implementing horizontal polarization. In this work, we use S-parameters at 1.4 GHz of an 8-element sub-array and a 16-element sub-array for the demonstration of the proposed digital uncoupling method. Both sub-arrays are horizontally polarized and consist of 2 rows of 4 antennas and 4 rows of 4 antennas. These 8-element and 16-element arrays are represented by 8 × 8 and 16 × 16 coupling matrices. These matrices are Toeplitz structured, although errors persist due to irregularities in the antenna array. Thus, the coupling matrices can be defined via the first row (or the first column) of the symmetric Toeplitz matrices of sizes 16 × 16 and 8 × 8, respectively as follows

$$\begin{bmatrix} 0.1701 + 0.1738j \\ 0.1517 + 0.0834j \\ -0.0376 + -0.0678j \\ -0.0017 + 0.0371j \\ 0.0660 + -0.0014j \\ -0.0373 + -0.0830j \\ -0.0207 + 0.0633j \\ 0.0347 + -0.0233j \\ -0.0516 + 0.0129j \\ 0.0208 + 0.0512j \\ 0.0216 + -0.0319j \\ -0.0221 + -0.0012j \\ 0.0288 + -0.0034j \\ -0.0082 + -0.0423j \\ -0.0216 + 0.0231j \\ 0.0142 + 0.0025j \end{bmatrix} \qquad \begin{bmatrix} 0.1537 + 0.0466j \\ 0.0295 + 0.1927j \\ 0.0502 + -0.0528j \\ -0.0279 + -0.0121j \\ 0.0874 + 0.0684j \\ 0.0465 + -0.1080j \\ -0.0624 + 0.0060j \\ 0.0122 + 0.0344j \end{bmatrix}. \tag{9}$$

In these numerical experiments, we compare the uncoupled vectors as calculated by the conjugate transpose of the coupling matrix by a given vector $\mathbf{C}_n^*\mathbf{y}$, the proposed sparse factorization for the uncoupled matrix by the vector $\mathbf{Q}_n\mathbf{y}$, and the brute-force calculation of the inverse of the coupling matrix by the vector $\mathbf{C}_n^{-1}\mathbf{y}$. The coupled vectors are initialized as $\mathbf{y} \sim \text{Unif}(0, 1) \in \mathbb{C}^8$ or $\mathbb{C}^{16}$. We present several of these vectors and the corresponding absolute error between the conjugate transpose, the proposed factorization to uncouple the mutual coupling effect, and the brute-force calculation of the inverse of the coupling matrix by $\mathbf{y}$ for the 8 × 8 and 16 × 16 coupling matrices in Tables 1 and 2. Furthermore, according to the simulated data presented in [60], the mutual coupling matrices of sizes 8 × 8 and 16 × 16 are not unitary; instead, they are characterized as symmetric Toeplitz matrices.

Table 1
Four sets of random input vectors, $\mathbf{y}$ are denoted in the first column. The absolute error between the conjugate transpose of the 8 × 8 coupling matrix by the vector $\mathbf{y}$ and the proposed sparse uncoupling factorization by the vector $\mathbf{y}$ is denoted in the second column. The first element of the second column generally has the lowest error, with a magnitude of 10^{-4} or 10^{-5}. The subsequent values have larger errors, on the order of magnitude of 10^{-2}. The absolute error between the brute-force calculation of the inverse of the coupling matrix by the vector $\mathbf{y}$ and the proposed sparse uncoupling factorization by the vector $\mathbf{y}$ is shown in the third column. The last column shows the elements of the calibration matrix $\mathbf{A}_n$ for the mutual coupling matrix which is not unitary.

$\mathbf{y}$	$\lvert\mathbf{Q}_n\mathbf{y} - \mathbf{C}_n^*\mathbf{y}\rvert$	$\lvert\mathbf{Q}_n\mathbf{y} - \mathbf{C}_n^{-1}\mathbf{y}\rvert$	$\mathbf{A}_n$
0.6407+0.0045j	0.0000+0.0001j	0.5400+0.0649j	1.8255−0.2194j
0.5986+0.2322j	0.0311+0.0913j	0.5483+0.0747j	1.7906−0.2440j
0.1655+0.1895j	0.1202+0.0292j	0.7473+0.3775j	1.0661−0.5385j
0.6948+0.7587j	0.1745+0.0331j	0.4150+0.0776j	2.3284−0.4351j
0.8495+0.2478j	0.1700+0.0187j	0.6049+0.2481j	1.4151−0.5803j
0.4349+0.9336j	0.0015+0.0519j	0.5727+0.1474j	1.6375−0.4216j
0.1612+0.4344j	0.0679+0.1064j	0.5800+0.3585j	1.2475−0.7711j
0.5435+0.3484j	0.0066+0.0202j	0.2457+0.2428j	2.0591−2.0348j
0.2115+0.2340j	0.0000+0.0001j	0.4684+0.2614j	1.6280−0.9086j
0.6933+0.5751j	0.0606+0.0429j	0.5066+0.2460j	1.5974−0.7757j
0.3751+0.1305j	0.1024+0.0111j	0.7301+0.7225j	0.6920−0.6848j
0.2441+0.8575j	0.1756+0.0678j	0.4625+0.2332j	1.7239−0.8694j
0.4610+0.5553j	0.1553+0.0503j	0.3476+0.3769j	1.3224−1.4337j
0.1269+0.8865j	0.0355+0.0007j	0.5705+0.4708j	1.0428−0.8605j
0.2463+0.9159j	0.0841+0.0534j	0.5091+0.5581j	0.8922−0.9781j
0.2608+0.7185j	0.0107+0.0161j	0.2035+0.4157j	0.9500−1.9404j
0.3371+0.0594j	0.0000+0.0001j	0.1951+0.1464j	3.2791−2.4602j
0.1288+0.0309j	0.0732+0.0370j	0.7695+0.0068j	1.2994−0.0114j
0.8776+0.0212j	0.0694+0.0390j	0.4748+0.3931j	1.2496−1.0347j
0.3734+0.9278j	0.1057+0.1104j	0.4711+0.1861j	1.8361−0.7252j
0.3254+0.5637j	0.1383+0.1568j	0.3872+0.3352j	1.4763−1.2781j
0.5335+0.6979j	0.0610+0.0501j	0.2920+0.1113j	2.9899−1.1401j
0.1442+0.4455j	0.1131+0.0720j	0.6262+0.3301j	1.2496−0.6587j
0.0027+0.6169j	0.0096+0.0122j	0.2304+0.3262j	1.4444−2.0451j
0.9762+0.9394j	0.0001+0.0001j	0.2643+0.2091j	2.3267−1.8412j
0.2968+0.8063j	0.0083+0.0284j	0.6361+0.7429j	0.6650−0.7767j
0.1864+0.9151j	0.0654+0.0232j	0.6240+0.4307j	1.0854−0.7492j
0.5483+0.1735j	0.1167+0.0174j	0.4413+0.2168j	1.8253−0.8969j
0.2232+0.1288j	0.1606+0.0137j	0.6331+0.3192j	1.2593−0.6349j
0.2413+0.6357j	0.1184+0.0059j	0.6039+0.5430j	0.9156−0.8233j
0.3995+0.7100j	0.0061+0.0576j	0.5229+0.2932j	1.4550−0.8157j
0.6310+0.0369j	0.0058+0.0107j	0.3849+0.2107j	1.9989−1.0941j

Thus, we demonstrate the elements of the calibration matrix $\mathbf{A}_n$ as the last column of Tables 1 and 2 so that one could calibrate the input signal before the uncoupling the mutual coupling effect as shown in Proposition 3.1 and signal flow graph 2(b).

We benchmark the performance of our algorithm in terms of execution time and mean-squared error in recovering the uncoupled input. For comparison, we employ several popular, highly optimized libraries in Python. This includes the *SciPy* library implementation of the Levinson–Durbin recursion algorithm [65,66], which is specifically designed for solving systems of Toeplitz matrices. This algorithm is faster than generic least-squares methods, though numerically less stable. Additionally, we employ the *NumPy* and *PyTorch* libraries, with optimized numerical inversion algorithms, and we investigate the benefits of GPU acceleration within *PyTorch*. Input sizes vary from 8 to 4096 points, with results averaged over 10 randomly initialized, complex-valued inputs.

The results from these experiments are organized in Figs. 3 and 4. As shown in Fig. 3, the proposed algorithm is generally faster than its direct counterpart. The Levinson–Durbin algorithm solves the Toeplitz system slightly faster than the *SciPy* implementation of the proposed algorithm; however, using GPU acceleration, the proposed algorithm achieves significantly better scaling properties than all other approaches. Results in Fig. 4 show that the errors are slightly higher than those of direct approaches, but are still in a similar order of magnitude. Likewise, the errors of the proposed algorithm are highly correlated. This slight increase in error is due to numerical errors introduced during the computation of the calibration matrix components, $\mathbf{A}_n$.

6. Signal flow graph for uncoupling architecture

Signal flow graphs depict the transformations performed at each stage of the algorithm. The input of coupled samples is represented by the elements $y(k)$ on the left side, and the output of decoupled samples by $x(k)$ on the right for $k = 0, 1, \ldots, n-1$. Likewise, stages of the algorithm follow successively from left to right. Two arrows converging upon the same node represent an

Table 2
Two sets of random input vectors, $\mathbf{y}$ are denoted in the first column. The absolute error between the conjugate transpose of the 16 × 16 coupling matrix by the vector $\mathbf{y}$ and the proposed sparse uncoupling factorization by the vector $\mathbf{y}$ is denoted in the second column. The first element of the second column generally has the lowest error, with a magnitude of 10^{-4} or 10^{-5}. The subsequent values have larger errors, on the order of magnitude of 10^{-2}. The absolute error between the brute-force calculation of the inverse of the coupling matrix by the vector $\mathbf{y}$ and the proposed sparse uncoupling factorization by the vector $\mathbf{y}$ is shown in the third column. The last column shows the elements of the calibration matrix $\mathbf{A}_n$ for the mutual coupling matrix which is not unitary.

$\mathbf{y}$	$\lvert\mathbf{Q}_n\mathbf{y} - \mathbf{C}_n^*\mathbf{y}\rvert$	$\lvert\mathbf{Q}_n\mathbf{y} - \mathbf{C}_n^{-1}\mathbf{y}\rvert$	$\mathbf{A}_n$
0.6449+0.6593j	0.0001+0.0002j	0.4050+0.2441j	1.8112−1.0917j
0.9392+0.5427j	0.0256+0.0348j	0.3115+0.2719j	1.8221−1.5904j
0.0321+0.0329j	0.0943+0.0084j	0.7978+0.3139j	1.0854−0.4270j
0.4639+0.6987j	0.0536+0.0233j	0.2094+0.0012j	4.7745+0.0274j
0.1754+0.1808j	0.1707+0.0404j	0.4226+0.2522j	1.7448−1.0412j
0.1420+0.6781j	0.0619+0.0398j	0.4678+0.0760j	2.0824−0.3385j
0.3299+0.3493j	0.1400+0.1395j	0.4636+0.2991j	1.5231−0.9828j
0.6412+0.6690j	0.1073+0.1579j	0.1925+0.3559j	1.1758−2.1740j
0.3070+0.0868j	0.2044+0.1565j	0.5598+0.1554j	1.6586−0.4605j
0.7124+0.6715j	0.0103+0.0907j	0.3220+0.1037j	2.8139+0.9060j
0.3825+0.8124j	0.0998+0.0966j	0.6187+0.3060j	1.2986−0.6422j
0.0579+0.8343j	0.1974+0.0232j	0.6491+0.2623j	1.3244−0.5351j
0.7555+0.2851j	0.0045+0.1426j	0.3974+0.1478j	2.2106−0.8221j
0.6373+0.0601j	0.0459+0.0347j	0.7746+0.2613j	1.1591−0.3910j
0.4733+0.4374j	0.1254+0.0445j	0.7544+0.3269j	1.1159−0.4836j
0.7038+0.2423j	0.0054+0.0291j	0.1792+0.3026j	1.4488−2.4462j
0.7491+0.5099j	0.0001+0.0001j	0.2319+0.2662j	1.8604−2.1354j
0.3049+0.2759j	0.0574+0.0100j	0.5769+0.4481j	1.0812−0.8398j
0.4161+0.5586j	0.0186+0.0343j	0.2688+0.2012j	2.3847−1.7848j
0.4000+0.3128j	0.0372+0.1058j	0.2913+0.3011j	1.6596−1.7156j
0.3899+0.2268j	0.1015+0.0287j	0.5011+0.1110j	1.9024−0.4212j
0.3674+0.6286j	0.1195+0.0680j	0.2343+0.2572j	1.9357+2.1248j
0.0036+0.1138j	0.0759+0.0050j	0.2331+0.3254j	1.4550−2.0310j
0.0440+0.4585j	0.1578+0.0482j	0.2637+0.2279j	2.1712−1.8759j
0.6635+0.4380j	0.0356+0.0577j	0.1898+0.0584j	4.8125−1.4795j
0.1295+0.0121j	0.0654+0.0219j	0.7655+0.2207j	1.2061−0.3477j
0.3164+0.9211j	0.0990+0.1044j	0.4021+0.0195j	2.4811+0.1200j
0.2909+0.7783j	0.0890+0.0681j	0.2740+0.3910j	1.2022−1.7152j
0.2578+0.5242j	0.0382+0.1240j	0.5445+0.3045j	1.3990−0.7822j
0.9911+0.5197j	0.0386+0.0279j	0.4549+0.2147j	1.7979−0.8484j
0.5935+0.2366j	0.0838+0.0008j	0.7356+0.5743j	0.8446−0.6594j
0.2793+0.5878j	0.0087+0.0158j	0.3635+0.3731j	1.3397−1.3750j

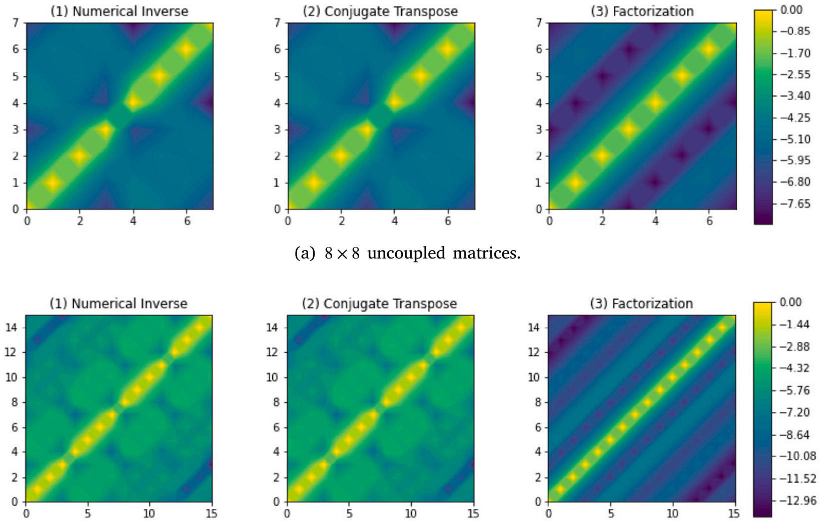
addition. The symbols in red above the arrows represent multiplication by the respective quantity, and a dashed line represents multiplication by −1. Although multiplications by −1 and $\pm j$ are depicted, these are not included in the evaluation of the computational complexity. Permutation matrices within the FFT are not depicted by the signal flow graph. Therefore, it is necessary to keep careful track of the elements between computational stages. Likewise, because the outputs of the FFT are in bit-reversed order, the elements of $\mathbf{D}_m$ are also depicted in bit-reversed order.

Following the uncoupling algorithm, i.e., $\mathbf{x} = uncoupl(\mathbf{y}, n)$ in Section 3.3, we present the 8-point signal flow graph for the scaled mutual decoupling algorithm in Fig. 5. Numerical simulation performed in MATLAB reveals that for a complex input of coupled values $\mathbf{y}$ of size $n = 8$, the ℓ^2-norm of the difference of uncoupled values $\mathbf{x}$ calculated by the proposed factorization and the brute-force method is on the order of 10^{-16}. Thus, the error is close to machine precision.

While the proposed signal flow graph shares design similarities with the one presented in [39], there are notable differences in the diagonal scalings and calibration weights employed in the proposed approach compared to the paper [39].

7. Conclusion

We have proposed a sparse factorization for the $n \times n$ uncoupled matrices followed by a fast, exact, and recursive algorithm to uncouple the mutual coupling effect of electromagnetic fields caused by antenna array elements while reducing the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n\log(n))$. The proposed factorization also shows an alternative method to solve the system of linear equations having symmetric Toeplitz matrices as coefficient matrices with $\mathcal{O}(n\log(n))$ as opposed to the $\mathcal{O}(n^3)$ complexity algorithm. We evaluated the proposed algorithm using S-parameters at 1.4 GHz, obtained from an 8-element sub-array and a 16-element sub-array. Our numerical results showed the effectiveness of the algorithm in mitigating the mutual coupling effect, considering the impact of attenuation. These findings exhibit promising outcomes, indicating the potential utilization of the proposed algorithm for uncoupling the mutual



(a) 8×8 uncoupled matrices.

(b) 16×16 uncoupled matrices.

Fig. 2. These plots represent the attenuation of signals corresponding to the uncoupled matrices based on the distance between antennas using the brute-force inverse of the coupling matrix as denoted by (1), the conjugate transpose of the coupling matrix as denoted by $\mathbf{C}_n^*$, and the proposed sparse factorization to uncouple the mutual coupling effect as denoted by (2). The results are plotted in a $\log_{10}$ scale to better illustrate the structure of the matrices. The elements with the largest magnitude of the 16×16 uncoupling matrix are along the main diagonal, i.e., elements corresponding to self-uncoupling, and these are generally normalized to have a magnitude of 1. Elements along the diagonal directions of these uncouple matrices show that the signals attenuate when antennas are far away. The numerical results clearly show that the diagonal elements of the uncoupling matrices steadily decrease as one moves away from the main diagonal, highlighting the diminishing effect of mutual coupling and the predominance of self-coupling over mutual coupling.

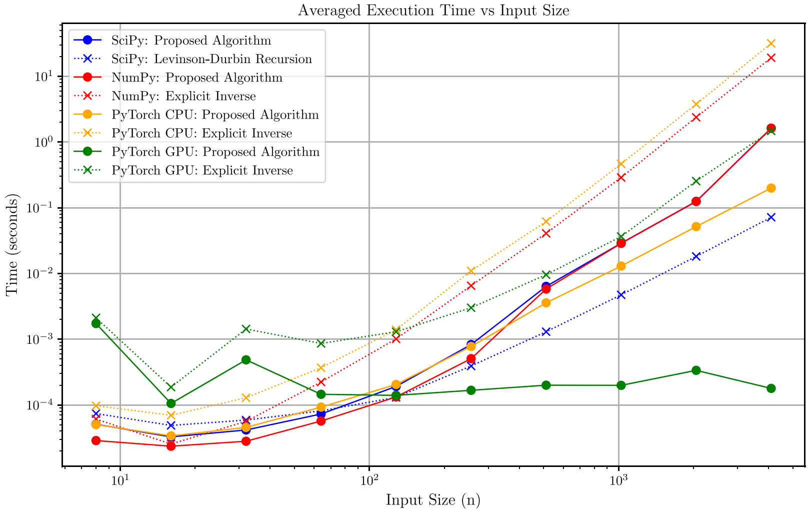


Fig. 3. Execution time trends, averaged over 10 randomly initialized inputs, for all implementations of the proposed algorithm and several benchmark algorithms in solving Toeplitz systems. The proposed algorithm gains substantial benefits from GPU acceleration compared to all the other Toeplitz system solvers while maintaining uniformity as the size of the Toeplitz systems increases. This assures that the proposed Toeplitz solver has a low-complexity even for larger Toeplitz systems.

coupling effect of the antenna arrays. We implemented the proposed algorithm alongside highly optimized libraries such as *SciPy*, *NumPy*, and *PyTorch*. These libraries present benchmarks due to their optimizations and hardware-specific improvements, especially in matrix inversion and recursive algorithms. Our algorithm has shown competitive performance, particularly when leveraging GPU acceleration in *PyTorch*. It has achieved superior scaling properties with larger input sizes, effectively navigating the complexities posed by these advanced baselines. The signal flow graph shows the connection of the algebraic operations in the linear system with

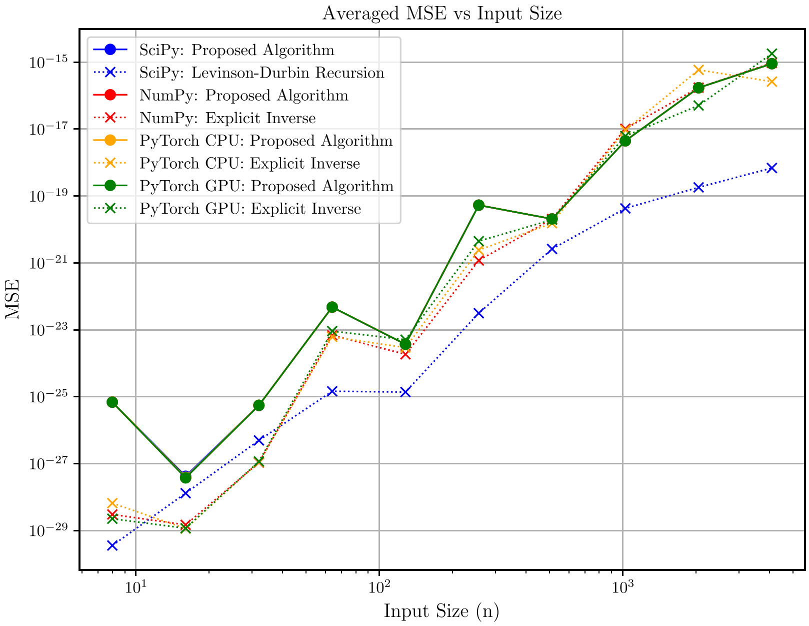


Fig. 4. Mean-squared error trends, averaged over 10 randomly initialized inputs, for all implementations of the proposed algorithm and several benchmark experiments. Although the proposed algorithm has a slightly higher error, than the other Toeplitz system solvers, the MSE is still significantly lower in magnitude, especially for larger Toeplitz systems.

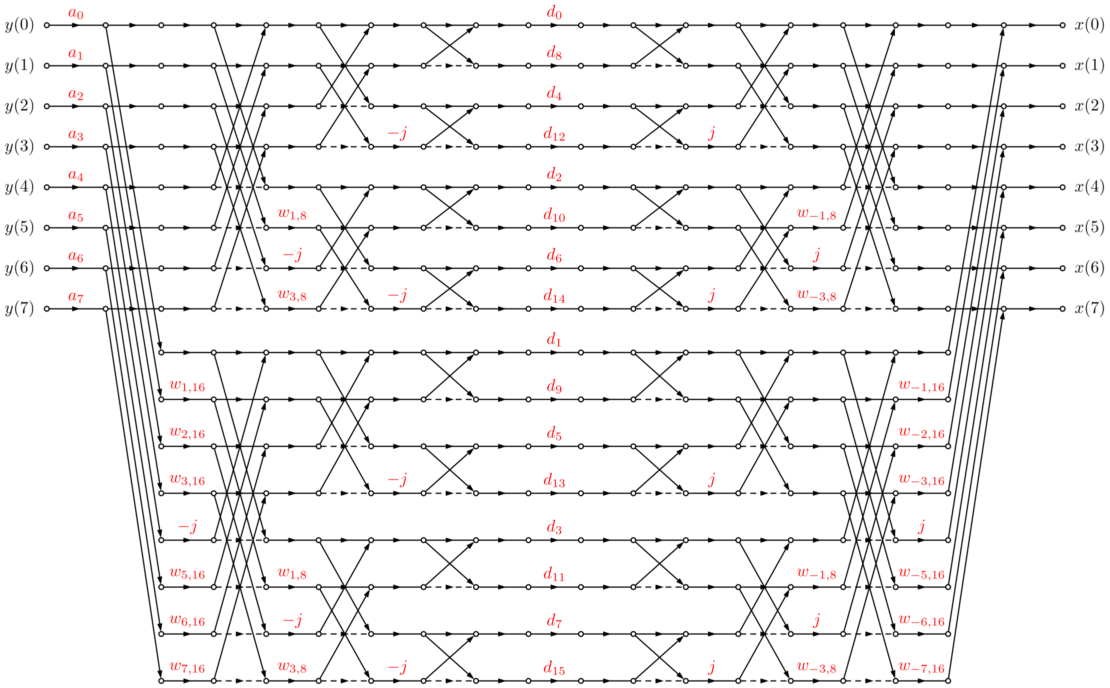


Fig. 5. Signal flow graph for 8-point scaled mutual decoupling algorithm, i.e., $\mathbf{x} = 16\mathbf{Q}_8\mathbf{y}$, where $\mathbf{Q}_8 = \mathbf{C}_8^*\mathbf{A}_8$, $\mathbf{A}_8 = \text{diag}\left[a_k\right]_{k=0}^{7}$, $w_{k,n} = e^{-i\frac{2\pi k}{n}}$, $m = 2n$, $j^2 = -1$, and d_k is the kth component of $\mathbf{D}_{16}$.

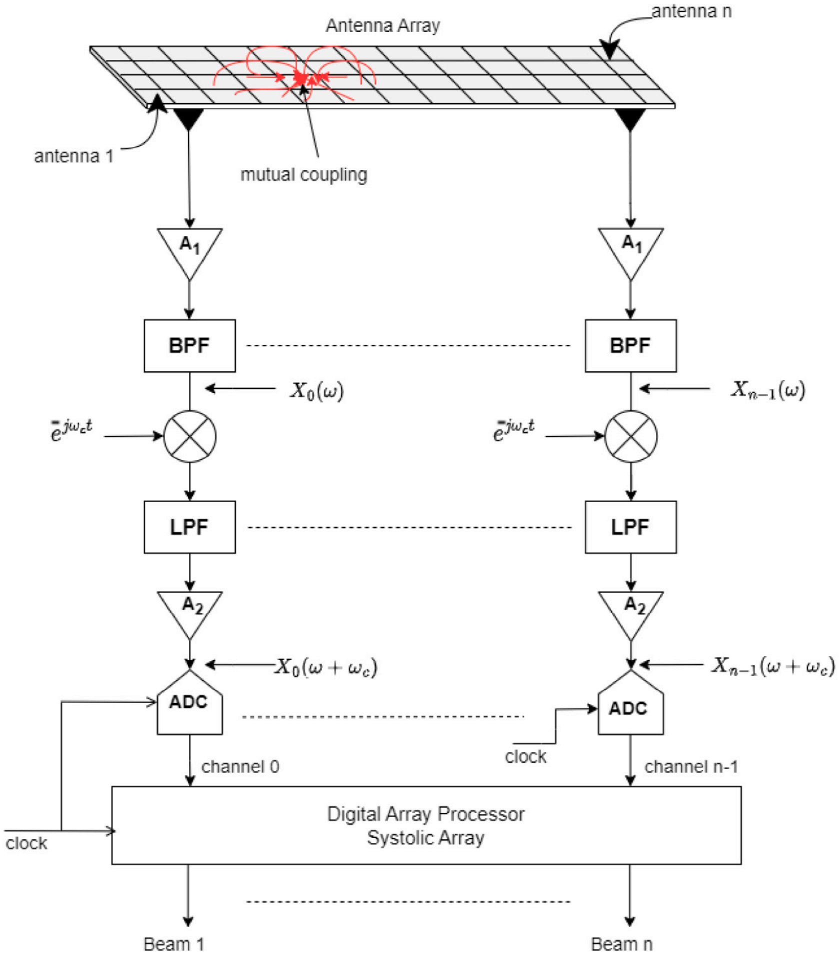


Fig. 6. Overview of a typical antenna array with receivers and digital signal processing hardware. This block diagram is representative of a wide range of wireless engineering applications including those in wireless communication systems such as 4G/LTE and 5G, radio astronomy imaging, radar, and electronic warfare. The mutual coupling between the antennas causes a linear mixing of signals that is undesirable as it causes smudging in imaging applications, loss of directivity in radar, and loss of signal-to-interference ratio in wireless communications. This paper discusses a digital signal processing-based fast algorithm that uncouples the mutual coupling using proposed linear algebra techniques that are realized as application-specific fast systolic-array processors operating in real-time on the digitized array signals.

the fundamental building blocks of the flow graph, the simplicity of the proposed algorithm, and the architecture for the VLSI. In contrast to the brute-force uncoupling algorithms, the proposed simplified uncoupled algorithm is expected to reduce the required size of the digital signal processing hardware, thereby reducing the cost, power consumption, and complexity of the mutually coupled system.

Appendix

Fig. 6 shows a typical digital receiver array consisting of an $N \times N$ grid of $n = N^2$ elements, with n number of dedicated direct-conversion receivers. Each antenna in the array has a dedicated receiver, where each receiver consists of a low-noise amplifier (LNA) (shown as A_1) and bandpass filter (BPF) followed by a direct down-conversion stage (mixers) that implement the Fourier transform relation $x(t)e^{-j\omega_c t} \Longleftrightarrow X(\omega+\omega_c)$, where ω is the circular frequency related to temporal frequency $f\ Hz$ via $\omega = 2\pi f$, and $\omega_c = 2\pi f_c$ where $f_c\ Hz$ is a carrier frequency. The down-conversion of the bandlimited radio signals causes them to be replicated at the lower-frequency range (baseband) while mutual coupling functions between array elements remain the same albeit frequency translated down to baseband. Subsequent amplification stages (shown as A_2) followed by image-rejection low-pass filters (LPFs) lead to the desired analog signals that are sampled and quantized by the analog to digital converters (ADCs) to yield the digital baseband signal applied to the array processor chip. The array processor implements fast algorithms for digital uncoupling of electromagnetic mutual coupling, which occurs in real-time through fast massively-parallel systolic array realization of the proposed algorithm, followed by digital multi-beam and adaptive beamformers (not discussed in this paper) that yield multiple RF beams in the digital signal processor.

Data availability

Data will be made available on request.

References

[1] H. Steyskal, J.S. Herd, Mutual coupling compensation in small array antennas, IEEE Trans. Antennas and Propagation 38 (12) (1990) 1971–1975.
[2] Z. Zheng, W. Wang, Y. Kong, Y.D. Zhang, MISC array: A new sparse array design achieving increased degrees of freedom and reduced mutual coupling effect, IEEE Trans. Signal Process. 67 (7) (2019) 1728–1741.
[3] Z. Zheng, et al., Robust adaptive beamforming against mutual coupling based on mutual coupling coefficients estimation, IEEE Trans. Veh. Technol. 66 (10) (2017) 9124–9133.
[4] H.T. Hui, Decoupling methods for the mutual coupling effect in antenna arrays: A review, Recent. Pat. Eng. 1 (2007) 187–193.
[5] E.M. Friel, K.M. Pasala, Effects of mutual coupling on the performance of STAP antenna arrays, IEEE Trans. Aerosp. Electron. Syst. 36 (2) (2000) 518–527.
[6] S. Pulipati, V. Arivarathna, S.M. Perera, C. Wijenayake, L. Belostotski, A. Madanayake, Digital uncoupling of coupled multi-beam arrays, in: 2021 Int. Appl. Comput. Electromagn. Soc. Symp., ACES, 2021, pp. 1–4.
[7] S. Pulipati, V. Ariyarathna, A.L. Jayaweera, C.U.S. Edussooriya, C. Wijenayake, L. Belostotski, A. Madanayake, FPGA-based 2-D FIR frost beamformers with digital mutual coupling compensation, in: 2020 IEEE/MTT-S International Microwave Symposium, IMS, 2020, pp. 1077–1080, http://dx.doi.org/10.1109/IMS30576.2020.9224004.
[8] T.T. Zhang, Y.L. Lu, H.T. Hui, Simultaneous estimation of mutual coupling matrix and DOAs for UCA and ULA, in: Proc. 17th Int. Zurich Symp. on EMC, 2006.
[9] T. Svantesson, Mutual coupling compensation using subspace fitting, in: Proc. IEEE Sens. Arr. and Multichannel Sig. Process. Workshop, 2000.
[10] J. Kota, et al., A 2-D signal processing model to predict the effect of mutual coupling on array factor, IEEE AWP Lett. 12 (2013) 1264–1267.
[11] R. Ali, L. Belostotski, G.G. Messier, A. Madanayake, A.T. Sutinjo, Impact of bandwidth on antenna-array noise matching, Electron. Lett. 57 (4) (2021) 158–160.
[12] A.D. Johnson, J. Zhong, S.B. Venkatakrishnan, E.A. Alwan, J.L. Volakis, Phased array with low-angle scanning and 46:1 bandwidth, IEEE Trans. Antennas and Propagation 68 (12) (2020) 7833–7841, http://dx.doi.org/10.1109/TAP.2020.2998869.
[13] R. Vandebril, M.V. Barel, N. Mastronardi, Matrix Computations and Semiseparable Matrices: Linear Systems, The Johns Hopkins University Press, Baltimore, USA, 2008.
[14] E.G. Kounias, An inversion technique for certain patterned matrices, J. Math. Anal. Appl. 21 (695–698) (1968).
[15] V.R.R. Uppuluri, J.A. Carpenter, The inverse of a matrix occurring in first-order moving-average models, Sankhya: Indian J. Stat. Ser. A 31 (79–82) (1969).
[16] E.L. Allgower, Exact inverse of certain band matrices, Numer. Math. 21 (279–284) (1973).
[17] C.M.D. Fonseca, J. Petronilho, Explicit inverses of some tridiagonal matrices, Linear Algebra Appl. 325 (7–21) (2001).
[18] C.M.D. Fonseca, J. Petronilho, Explicit inverses of a tridiagonal k-Toeplitz matrix, Numer. Math. 100 (2005) 457–482.
[19] M. Dow, Explicit inverses of Toeplitz and associated matrices, ANZIAM J. 44 (E) (2003) E185–E215, [Online] http://anziamj.austms.org.au/V44/E019 (23 January 2003).
[20] P.G. Martinsson, V. Rokhlin, M. Tygert, A fast algorithm for the inversion of general Toeplitz matrices, Comput. Math. Appl. 50 (741–752) (2005).
[21] G. Ammar, Classical foundations of algorithms for solving positive definite Toeplitz equations, Calcolo 33 (1996) 99–113.
[22] M. Van Barel, G. Heinig, P. Kravanja, A stabilized superfast solver for nonsymmetric Toeplitz systems, SIAM J. Matrix Anal. Appl. 23 (2) (2001) 494–510, http://dx.doi.org/10.1137/S0895479899362302.
[23] M. Stewart, A superfast Toeplitz solver with improved numerical stability, SIAM J. Matrix Anal. Appl. 25 (3) (2003) 669–693.
[24] V. Olshevsky, I. Oseledets, E. Tyrtyshnikov, Superfast inversion of two-level Toeplitz matrices using Newton iteration and tensor-displacement structure, in: J.a. Ball, Y. Eidelman, J.W. Helton, V. Olshevsky, J. Rovnyak (Eds.), Recent Adv. Mat. Oper. Theory. Operator Theory: Advances and Applications 179. Birkhäuser Basel, 2007.
[25] V.Y. Pan, TR-2003004: Superfast algorithms for singular Toeplitz/Hankel-like matrices, in: CUNY Acad. Works, 2003.
[26] V. Pan, Can we optimize Toeplitz/Hankel computations? in: E.W. Mayr, V.G. Ganzha, E.V. Vorozhtzov (Eds.), Proc. the 5th International Workshop on Computer Algebra in Scientific Computing, CASC, 02, Technische Univ. of München, Germany, 2002, pp. 253–264.
[27] T. Kailath, V. V. Olshevsky, Displacement structure approach to polynomial Vandermonde and related matrices, Linear Algebra Appl. 261 (1997) 49–90.
[28] V.Y. Pan, Structured Matrices and Polynomials: Unified Superfast Algorithms, Birkhauser/Springer, Boston/New York, 2001.
[29] V. Olshevsky, Fast Algorithms for Structured Matrices: Theory and Applications, in: Contemp. Math., vol. 323, American Mathematical Society, USA, 2003.
[30] G. Heinig, K. Rost, Fast algorithm for Toeplitz and Hankel matrices, Linear Algebra Appl. 435 (2011) 1–59.
[31] V.Y. Pan, Transformations of matrix structures work again, Linear Algebra Appl. 465 (1–32) (2015).
[32] V.Y. Pan, Fast approximate computations with Cauchy matrices and polynomials, Math. Comp. 86 (2799–2826) (2017).
[33] A. Böttcher, Orthogonal symmetric Toeplitz matrices, CAOT 2 (2008) 285–298.
[34] G. Heinig, K. Rost, Algebraic Methods for Toeplitz-Like Matrices and Operators, Akademie-Verlag, Berlin, and Birkhauser Basel, Boston, MA, 1984.
[35] T. Kailath, Linear Systems, Pearson, India, 2016.
[36] S.M. Perera, Quasiseparable Approach to Matrices of Vandermonde Type, ProQuest Publication, Ann Arbor, USA, 2012.
[37] S.M. Perera, V. Olshevsky, A fast Schur-euclid-type algorithm for quasiseparable polynomials, in: I.S. Kotsireas, E. Martinez-Moro (Eds.), Springer Proceedings in Mathematics and Statistics 198, 2017, pp. 341–383.
[38] M. Benzi, V. Simoncini(eds), Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications, in: Lect. Notes Math., Springer, Cham, 2016.
[39] S.M. Perera, L. Lingsch, A. Madanayake, S. Mandal, N. Mastronardi, Fast DVM algorithm for wideband time-delay multi-beam beamformers, IEEE Trans. Signal Process. 70 (5913–5925) (2022) http://dx.doi.org/10.1109/TSP.2022.3231182.
[40] S.M. Perera, I.S. Kotsireas, A low-complexity algorithm to search for Legendre pairs, Linear Algebra Appl. (2025) http://dx.doi.org/10.1016/j.laa.2025.01.010, URL https://www.sciencedirect.com/science/article/pii/S0024379525000102.
[41] L. Belostotski, B. Veidt, K.F. Warnick, A. Madanayake, Low-noise amplifier design considerations for use in antenna arrays, IEEE Trans. Antennas and Propagation 63 (6) (2015) 2508–2520.
[42] D. Jones, et al., Modeled and measured mutual impedances, element patterns, and sensitivity for a 19 element focal plane array, in: IEEE Int. Symp. Antenn. Propog., 2008.
[43] J. Andersen, H. Rasmussen, Decoupling and descattering networks for antennas, IEEE Trans. Antennas and Propagation 24 (6) (1976) 841–846.

[44] H.J. Chaloupka, X. Wang, Novel approach for diversity and MIMO antennas at small mobile platforms, in: Int. Symp. Pers., Indoor, Mobile Radio Commun., 2004, pp. 637–642, http://dx.doi.org/10.1109/PIMRC.2004.1370948.

[45] J. Weber, C. Volmer, K. Blau, R. Stephan, M.A. Hein, Miniaturized antenna arrays using decoupling networks with realistic elements, IEEE T-MTT 54 (6) (2006) 2733–2740.

[46] A. Diallo, C. Luxey, P. Le Thuc, R. Staraj, G. Kossiavas, Study and reduction of the mutual coupling between two mobile phone PIFAs operating in the DCS1800 and UMTS bands, IEEE Trans. Antennas and Propagation 54 (11) (2006) 3063–3074.

[47] C. Volmer, J. Weber, R. Stephan, K. Blau, M.A. Hein, An eigen-analysis of compact antenna arrays and its application to port decoupling, IEEE Trans. Antennas and Propagation 56 (2) (2008) 360–370, http://dx.doi.org/10.1109/TAP.2007.915450.

[48] K. Warnick, M. Jensen, Optimal noise matching for mutually coupled arrays, IEEE Trans. Antennas and Propagation 55 (6) (2007) 1726–1731, http://dx.doi.org/10.1109/TAP.2007.898596.

[49] L. Belostotski, A. Sutinjo, R. Subrahmanyan, S. Mandal, A. Madanayake, General framework for array noise analysis and noise performance of a two-element interferometer with a mutual-coupling canceler, IEEE Trans. Antennas and Propagation 70 (9) (2022) 8059–8068, http://dx.doi.org/10.1109/TAP.2022.3165540.

[50] K. Kurokawa, Power waves and the scattering matrix, IEEE T-MTT 13 (2) (1965) 194–202, http://dx.doi.org/10.1109/TMTT.1965.1125964.

[51] N.J. Higham, Accuracy and Stability of Numerical Algorithms, SIAM, Philadelphia, USA, 1996.

[52] T. Kailath, A. Sayed, Fast Reliable Algorithms for Matrices with Structure, SIAM Publications, Philadelphia, USA, 1999.

[53] D.A. Bini, Matrix structures in queuing models, in: M. Benzi, V. Simoncini (Eds.), Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications, in: Lect. Notes in Mathematics, vol. 2173, 2016, pp. 65–160.

[54] V. Olshevsky, I. Oseledets, E. Tyrtyshnikov, Superfast inversion of two-level Toeplitz matrices using Newton iteration and tensor-displacement structure, in: J.A. Ball, Y. Eidelman, J.W. Helton, V. Olshevsky, J. Rovnyak (Eds.), Recent Advances in Matrix and Operator Theory, Birkhäuser Basel, Basel, 2008, pp. 229–240.

[55] I. Gohberg, V. Olshevsky, Complexity of multiplication with vectors for structured matrices, Linear Algebra Appl. 202 (1994) 163–192.

[56] I. Gohberg, V. Olshevsky, Fast algorithms with preprocessing for matrix-vector multiplication problems. Journal of complexity, Linear Algebra Appl. 10 (4) (1994) 411–427.

[57] G. Golub, C.V. Loan, Matrix Computations, fourth ed., The Johns Hopkins University Press, Baltimore, MD, 2013.

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

[59] R. Yavne, An economical method for calculating the discrete fourier transform, in: Proc. AFIPS Fall Joint Computer Conf. 33, 1968, pp. 115–125.

[60] R. Sarkis, B. Veidt, C. Craeye, Fast numerical method for focal plane array simulation of 3D Vivaldi antennas, in: International Conference on Electromagnetics in Advanced Applications, Torino, Italy, 2012, pp. 772–775, http://dx.doi.org/10.1109/ICEAA.2012.6328733.

[61] T. Burgess, B. Veidt, L. Belostotski, A.J. Beaulieu, E. Zailer, J.W. Haslett, A large phased array feed with CMOS low-noise amplifiers, in: European Conference on Antennas and Propagation, EuCAP, 2018, pp. 1–3, http://dx.doi.org/10.1049/cp.2018.0913.

[62] A.J. Beaulieu, L. Belostotski, T. Burgess, B. Veidt, J.W. Haslett, Noise performance of a phased-array feed with CMOS low-noise amplifiers, IEEE Antennas Wirel. Propag. Lett. 15 (2016) 1719–1722, http://dx.doi.org/10.1109/LAWP.2016.2528818.

[63] A.J. Beaulieu, G. Wu, L. Belostotski, J.W. Haslett, T. Burgess, B. Veidt, Development of a CMOS receiver for a radio-telescope phased-array feed, in: IEEE International Microwave Symposium, IMS, 2016, pp. 1–3, http://dx.doi.org/10.1109/MWSYM.2016.7540346.

[64] L. Belostotski, A.J. Beaulieu, T. Burgess, B. Veidt, J.W. Haslett, Low noise phased-array feed with CMOS LNAs, in: United States National Committee of URSI National Radio Science Meeting, USNC-URSI NRSM, 2016, pp. 1–2, http://dx.doi.org/10.1109/USNC-URSI-NRSM.2016.7436230.

[65] N. Levinson, The Wiener RMS error criterion in filter design and prediction, J. Math. Phys. 25 (1947) 261–278.

[66] J. Durbin, The fitting of time series models, Rev. L’Institut Int. Stat. 28 (3) (1960) 233–244.