ABSTRACT In this paper, a set of sequences is developed with a minimized mean sidelobe level (MSL). The problem is formulated for cognitive radars using the $l_1$-norm, where the target cognition determines which sidelobes should be suppressed. The cognitive radar configuration requires fast waveform regeneration in each cognition cycle. In this light, the computational burden of the algorithms developed here is revealed to be situated on the singular value decomposition (SVD) operation. The randomization method is adopted to speed up the proposed algorithms. The obtained fast generation of sequences with the desired autocorrelation is key to its utilization in cognitive radars. We also consider two practically important cases and accommodate the proposed approach to them: unimodular and finite-alphabet sequences. The superiority of the developed algorithms is confirmed both in suppression level and speed through extensive numerical simulations.

INDEX TERMS Cyclic algorithm, cognitive radar, mean sidelobe level, radar waveform design.

I. INTRODUCTION
Sequence design with low autocorrelation sidelobes has emerged as an important topic in signal processing. This is because transmitting such a sequence can result in low false detections at the receiver. Radar waveform design comprises a large number of studies, targeting different aspects of the problem. Traditional studies have focused on binary and polyphase methods. Baker [1], Golomb [2], Frank [3] codes, and their contemporaries [4] are a few to enumerate in this category. Two major weaknesses can be observed in traditional approaches. Firstly, the suppression level has equal emphasis along all sidelobes. New developments unfold that considering unequal suppression may result in significantly more suppression levels [5]. This approach is even more beneficial when also considering cognition in radar systems. There, only part of the sidelobe is required to be suppressed since the whereabouts of the target are known to some extent through cognitive perception. Secondly, traditional approaches do not consider increasing the number of allowed phases. Besides, increasing the number of phases and assuming arbitrary complex sequences, i.e., continuous phase, can result in much lower sidelobes (see also [5]–[8]).

Generally, the suppression degree is measured by a number of metrics including integrated sidelobe level (ISL) [5], [9]–[11], peak sidelobe level (PSL) [12], [13] and mean sidelobe level (MSL) [14]–[18]. These metrics can be considered for both autocorrelation [19] and ambiguity function (AF) [20], [21]. The ISL and PSL have already attracted considerable attention. In this regard, researchers have investigated various methods to generate sequences with low ISL and PSL, including chaotic [22], [23] and optimization [24] methods. Meanwhile, the MSL can be considered among the most important metrics [14]–[17] and its investigation is advocated in [25], [26]. Furthermore, low MSL results in low intercept range for low probability of intercept (LPI) radars [27]. However, no investigation has been conducted on MSL minimization. Besides, minimizing MSL puts less emphasis on outliers, i.e., higher sidelobes, among considered sidelobes. This is due to the Taxicab norm’s intrinsic property, applied extensively in other fields [28]–[30]. The ISL is the most studied metric mainly because it is based on the $l_2$-norm, which is the most canonical and established norm. Several authors have considered minimizing the ISL based on cyclic algorithm (CA) [5], [31] and majorization minimization (MM) [6], [7], [32].

Inspired by nature, cognitive radar introduces a new way of thinking about the radar systems [33], [34]. Many authors
investigated various aspects of this idea [35], [36]. Particularly, its waveform design is mainly investigated based on several types of cognitions [34]: range, Doppler, and Doppler-range. In the latter, gathered cognition about the target can prioritize some regions of the Doppler-range plane. Ambiguity function (AF) is designed by some authors to utilize this cognition [20], [37]–[40]. The autocorrelation [41], [42] and spectral power [43] are the focus for the first two types. Here, we focus on the latter type, where we observe such systems require two features: 1) the ability to design autocorrelation based on the desired pattern and 2) high speed to regenerate the waveform in each cognition cycle. This necessitates developing new algorithms to arbitrarily and quickly design autocorrelation. In this regard, the fast algorithm with the ability to arbitrarily suppress autocorrelation sidelobes would be a suitable waveform design candidate for a cognitive radar system. An in-depth analysis of computational complexity would also constitute a cornerstone to reach faster algorithms. In this article, we investigate using range cognition for designing radar waveform. Solicited approaches like considering MSL and using CA are adopted to reach superior suppression levels and low time/computational complexity compared to high-performing benchmark algorithms.

Compared with the literature, the major characteristics of the methods developed here are:

- MSL minimization is formulated and solved through the CA. Its corresponding generic problem is revealed to be a geometric median problem, where Weiszfeld’s algorithm can be applied. Furthermore, the unimodularity constraint is separated from the problem under study, resulting in an algorithm able to suppress more than half of the sidelobes to almost zero. This rare ability is particularly desirable when the cognitive radar is detecting several targets and needs a wide range of sidelobes to be suppressed.

- The computational burden of the proposed algorithms is determined to rest upon two points: the embedded SVD and the Weiszfeld’s algorithm. Accordingly, fast-randomized SVD and loose criterion selection are applied to speed up the proposed algorithms. The presented algorithms’ computational complexity is also obtained mathematically and revealed to be proportional to the sequence length and the desired number of suppression lags. The significant speed up in the generation process also is confirmed through numerical simulations. The fast generation waveform is particularly appealing to cognitive radars, where the waveform should be regenerated every time the target cognition is gathered. Besides, the in-depth computational complexity analysis can be a basis for future alike algorithms.

- We further considered restricting the developed algorithms to finite-alphabet constellations. Having a finite-alphabet waveform would significantly simplify the implementation of the radar system.

In this paper, regular and bold lower cases are used to represent a scalar and vector variable, respectively; bold uppercase is reserved for matrices. The notations \((\cdot)^*\) and \((\cdot)^H\) denote complex conjugate and Hermitian (i.e., conjugate transpose), respectively. The \(|\cdot|\) and \(|\cdot|_1\) represent absolute value (i.e., Euclidean norm) and Taxicab norm for a complex-valued scalar variable. That is, \(|a|_1 = |\Re\{a\}| + |\Im\{a\}|\), where \(\Re\{a\}\) and \(\Im\{a\}\) represent the real and imaginary parts of the complex-valued \(a\). The \(|a|_1 = \sum |a_i|\) is the \(l_1\) or Taxicab norm of the vector \(a\). Also, \(|A|_{1,1}\) represents the entry-wise Taxicab or \(l_1\)-norm for matrix \(A\), defined by (see [44], Eq. 1.4.47)

\[
|A|_{1,1} = \|\text{vec}(A)\|_1 = \sum_{i,j} |a_{i,j}|. 
\] (1)

where \(\text{vec}(A)\) is the vectorized format of \(A\). This notation is consistent with the \(L_{p,q}\) entrywise norm of the matrix \(A\) defined by:

\[
|A|_{p,q} = \left( \sum_j \left( \sum_i |a_{ij}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}}. 
\] (2)

Finally, \(e_i\) represents the \(i\)-th canonical vector, that is, a vector with all elements equal to zero except for the \(i\)-th, being equal to one.

The paper is organized as follows. In Section II, the problem is formulated in a general form and solved through the CA. In Section III, some improvements are discussed, and the algorithms are analyzed. In particular, the imposition of the unimodularity constraint is argued in subsection III-A, and subsection III-C is devoted to developing a speedup solution for a long sequence design. Consequently, the simulation results are presented in Section IV. Finally, conclusions are given in Section V.

II. PROBLEM FORMULATION

Consider a complex-valued sequence \(\{x_n\}_{n=1}^N\). This sequence can constitute the waveform transmitted from cognitive radar or more generally a mono-static radar. The autocorrelation for this sequence is

\[
r_k = \sum_{n=k+1}^N x_n^* x_{n-k} = r_{x_k}^*, \quad k = 0, \ldots, N - 1. 
\] (3)

Suitability of the autocorrelation function can be described through several metrics including ISL, PSL and MSL. The first is defined by

\[
\text{ISL} = \sum_{k=1}^{N-1} |r_k|^2. 
\] (4)

and can be considered as Euclidean norm on autocorrelation sidelobes. Since this norm is the most canonical, minimizing ISL is the most popular approach for sequence design [5], [45]. Some studies show that minimizing other metrics like
PSL can suppress a large part of autocorrelation [8]. It is defined by

$$PSL = \max_{k \neq 0} |r_k|.$$  \hfill (5)

The MSL is the least investigated metric and is defined by

$$MSL = \sum_{k=1}^{N-1} |r_k|.$$  \hfill (6)

Akin to ISL, PSL and MSL can be thought of as the Chebyshev and Taxicab norms on the autocorrelation side-lobes, respectively. The PSL minimization is studied in [13]; but to the best of our knowledge, no study has been reported on the MSL minimization. We consider the weighted MSL (WMSL) given by

$$WMSL = \sum_{k=1}^{N-1} \lambda_k |r_k|,$$  \hfill (7)

where $\{\lambda_k\}_{k=1}^{N-1}$ are real-valued weights. The choice of this problem is based on two considerations. First, any cognition about the target position can be prioritize suppression in some of the autocorrelation lags. Using (7), such a cognition can be accommodated in $\lambda_k$ coefficients. Second, rapid sequence regeneration is required for cognitive radars since the sequence should be redesigned for gathered cognition.

To design sequences with minimized MSL, let

$$X = \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ x_N & \cdots & x_1 \\ 0 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & x_N \end{bmatrix}_{(2N-1) \times N}.$$  \hfill (8)

be the Toeplitz matrix defined by values of target sequence $\{x_n\}_{n=1}^N$. The autocorrelation is obtained by

$$X^H X = \begin{bmatrix} r_0 & r_1^* & \cdots & r_{N-1}^* \\ r_1 & r_0 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ r_{N-1} & \cdots & r_1^* & r_0 \end{bmatrix}.$$  \hfill (9)

Putting (6) and (9) together, the MSL minimization problem can be formulated as

$$\min_{\{x_n\}_{n=1}^N} \|X^H X - r_0 I\|_{1,1},$$  \hfill (10)

where $r_0$ is autocorrelation peak and can be set to $r_0 = N$ if unimodularity or almost unimodularity is expected. Note that (10) and (6) are not equivalent but related. The relation is clarified in Remark 1. Due to the quadratic nature and non-differentiable behavior of the $\|\cdot\|_{1,1}$, a solution to this minimization problem cannot be found directly. Hence, similar to the methods developed in [5], [13], [45] for the ISL and PSL, minimizing the following metric is contemplated here

$$\|X - \sqrt{N}L\|_{1,1}.$$  \hfill (11)

where $L^H L = I$. Therefore, the optimization problem in (10) is changed to the following constrained optimization,

$$\min_{L, \{s_i\}_{i=1}^Q} \|X - \sqrt{N}L\|_{1,1}$$  \hfill s.t. $L^H L = I.$  \hfill (12)

The problems in (10) and (12) are equivalent, in the sense that a sequence converging to the solution of approximate problem converges to the solution of the main problem as well. This statement is rigorously proved in [46] in a general scenario. Next, we consider a case where suppressing only part of the autocorrelation is desired. In particular, consider the case in which for some $Q$ and $k_0$ ($Q \leq k_0 \leq N - 1$), suppression of $r_k$, $i = 1, \ldots, Q$ is contemplated, where $\{k_i\}_{i=1}^Q$ is an ordered sequence in the interval $[1, k_0]$. In this case, the matrix $X$ should be altered properly so that only the important autocorrelation coefficients are created in (9). Accordingly, we define the new matrix $\tilde{X}$ as follows (see [13], [45] for similar approaches),

$$\tilde{X} = \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ x_N & \cdots & x_1 \\ 0 & \cdots & \cdots & \cdots \\ \vdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & x_N \end{bmatrix}_{(N+k_0-1) \times k_0}.$$  \hfill (13)

and, let $T = [e_{k_i}]_{k_i=1}^Q$, $i = 1, 2 \ldots, Q$, be columns of the identity matrix corresponding to the desired coefficients. Then, we have

$$\tilde{X} = X T.$$  \hfill (14)

Substituting $\tilde{X}$ instead of $X$ in (9) results in the almost desired autocorrelation coefficients. Suppressing the predetermined part of the autocorrelation in the MSL sense, we have

$$\min_{\{s_i\}_{i=1}^Q} \|\tilde{X} - \sqrt{N}L\|_{1,1}$$  \hfill s.t. $L^H L = I.$  \hfill (15)

Figure 1 illustrates the regions of influence for $X$, $\tilde{X}$, and $\bar{X}$. In particular, we observe that $X$ considers all sidelobes while $\tilde{X}$ is only dedicated to specific lags of interest.

Remark 1: (See the Remark 1 in [5]) The above discussion is not completely equivalent to minimizing MSL but the weighted MSL for specific weights. In fact, when $k_Q = Q,$
Interestingly, (19) is a geometric median problem in one where,

\[ \min_x \sum_k |x - \mu_k|, \]

we have

\[ \tilde{X}^H\tilde{X} = \begin{bmatrix} \tilde{r}_0 & \tilde{r}_1^* & \ldots & \tilde{r}_Q^* \\ \tilde{r}_1 & \tilde{r}_0 & \ldots & \vdots \\ \vdots & \vdots & \ddots & \tilde{r}_1^* \\ \tilde{r}_Q & \ldots & \tilde{r}_1 & \tilde{r}_0 \end{bmatrix}. \quad (16) \]

When entrywise \( l_1 \)-norm is computed, the number of \( r_k \) appearing in the summation is \( 2(k_Q - k) \). Hence, (15) is equivalent to the WMSL minimization when \( \lambda_k = 2(k_Q - k) \). When \( Q < k_Q \), the weights cannot be computed directly, but can be regarded as proportional to cumulative occurrence of \( r_k \) and \( r_k^* \) in \( \tilde{X}^H\tilde{X} \).

Henceforth, the CA is adopted to solve (15). It is solved with respect to \( \{x_n\}_{n=1}^N \) and \( \mathbf{L} \) cyclically. That is, given \( \{x_n\}_{n=1}^N \), (15) is solved to achieve \( \mathbf{L} \), and vice versa. However, for the entrywise Taxicab norm, (15) does not possess a closed-form solution. Since just an improvement is required in each iteration, an approximate solution of \( \mathbf{L} \) suffices, indicating that the solution of \( \mathbf{L} \) can be obtained by solving the Euclidean version of (15) instead (see [5], [45] and references therein). Let

\[ \tilde{X}^H = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{U}_1^H \]  

be the economy-sized or thin SVD [47] for \( \tilde{X}^H \), where \( (\cdot)^H \) denotes the conjugate transpose. Then, the solution of the Euclidean norm version of (15) is as follows

\[ \mathbf{L} = \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{U}_2^H. \quad (18) \]

The other step in the cycle is to find the solution of (15) with respect to \( \{x_n\}_{n=1}^N \), with the assumption that \( \mathbf{L} \) is a known constant. In order to achieve this goal, consider a specific element of \( \{x_n\}_{n=1}^N \), say \( x \), and let \( \{\mu_k\} \) be the sequence constituted by the elements of \( \sqrt{N} \mathbf{L} \) with the same position of \( x \) in \( \tilde{X} \). Then, the generic form of the optimization of (15) is given by

\[ \min_x \sum_k |x - \mu_k|, \]

where, \(|\cdot|\) denotes the absolute value of complex numbers. Interestingly, (19) is a geometric median problem in one dimension and can be solved by applying Weiszfeld’s algorithm [48]. Accordingly, the optimal solution to this problem is where the following equilibrium is guaranteed,

\[ 0 = \sum_{k=1}^Q \frac{\mu_k - x}{|\mu_k - x|}. \quad (20) \]

Equation (20) can be restated as

\[ x = \frac{\sum_{k=1}^Q \frac{\mu_k}{|\mu_k - x|}}{\sum_{k=1}^Q \frac{1}{|\mu_k - x|}}. \quad (21) \]

Subsequently, by applying the following iteration, the optimal solution of (19) can be reached for almost all initial conditions (see [48] and references therein)

\[ x_{i+1} = \frac{\sum_{k=1}^Q \frac{\mu_k}{|\mu_k - x_i|}}{\sum_{k=1}^Q \frac{1}{|\mu_k - x_i|}}. \quad (22) \]

Here, the superscript \( i \) denotes the iteration index. Consequently, the algorithm for minimizing MSL is given in Table 1.

**Remark 2:** If \(|\cdot| \) is replaced by the \( l_1 \)-norm on complex numbers in (19), then it will be a completely different problem, known as sum-of-absolute-value problem, which can be solved by linear programming. The resulted algorithm can be applied to generate sequences with properly shaped autocorrelation as well. Accordingly, the Appendix is devoted to solving for the \( l_1 \)-norm version of (19). Although this method is not developed here completely to result in a final algorithm, the same approach can be adopted and developed for MIMO configuration, which is a good topic for future research.

### III. DISCUSSIONS AND IMPROVEMENTS

#### A. IMPOSING UNIMODULARITY CONDITION

Without loss of generality, designing sequences with unit modulus (a.k.a. unimodular) can be performed by the introduced method. Accordingly, the sequence \( \{x_n\} \) is unimodular.

| Step | Description |
|------|-------------|
| 1    | Set \( \{x_n\}_{n=1}^N \) to some initial sequence; set \( k = 0 \) (Initialization) |
| 2    | Constitute the matrix \( \tilde{X} \) and compute \( \mathbf{L} \) according to (18) |
| 3    | For each \( x_n \), constitute the sequence \( \{\mu_k\} \), i.e., find the elements of \( \sqrt{N} \mathbf{L} \) whose positions are the same as \( x_n \) \( \tilde{X} \) |
| 4    | For the sequences \( \{\mu_k\} \) and for some initial value (e.g., \( x_0^0 = 0 \)), compute the iteration in (22) until some stop criterion is satisfied (e.g., \( |x_n^{i+1} - x_n^i| < \delta \)) |
| 5    | Go to step 2 till some stop criterion is satisfied (e.g., \( \|X_{\text{new}} - X_{\text{old}}\|_{1,1} < \epsilon \)) |

**TABLE 1.** Mean sidelobe cyclic algorithm (MeSiCA).
if and only if
\[ |x_n| = 1, \quad n = 1, \ldots, N. \]  

(23)

Note that no deficiency is imposed on the suitability of sequences. On the contrary, the unimodularity condition is equivalent to the minimum possible peak-to-average power ratio (PAPR), which is a suitability index for the designed sequences (see [13] and references therein). However, this restriction is revealed to translate to lower computing speed. Our preference is, therefore, to consider unimodularity separately. It is required to impose unimodularity condition on (10), (15), and (19) in the above-mentioned argument To design unimodular sequences. Therefore, the following models are obtained.

\[
\begin{align*}
\min_{\{x_n\}_{n=1}^N} \text{MSL} &= \|X^H X - NI\|_{1,1} \\
\text{s.t.} \quad & |x_n| = 1, \quad n = 1, \ldots, N, \\
\min \quad & \|X - \sqrt{NL}\|_{1,1} \\
\text{s.t.} \quad & L^*L = I \\
\text{and for some initial value } x^0_n, \quad & |x_n| = 1, \quad n = 1, \ldots, N, \\
\min_x \quad & \sum_k |x - \mu_k| \\
\text{s.t.} \quad & |x| = 1.
\end{align*}
\]

(24)

(25)

(26)

To design unimodular sequences, the only required alteration is that the constrained minimization in (26) is solved instead of (19). Accordingly, equation (26) can be restated as,

\[
\begin{align*}
\min_x \quad & \sum_k \sqrt{(x_1 - \Re\{\mu_k\})^2 + (x_2 - \Im\{\mu_k\})^2} \\
\text{s.t.} \quad & x_1^2 + x_2^2 = 1,
\end{align*}
\]

(27)

where, \(x_1\) and \(x_2\) are real and imaginary parts of the generic \(x\). Observe that the feasible set \(\{(x_1, x_2)|x_1^2 + x_2^2 = 1\}\) is not convex. Therefore, the Karush-Kuhn-Tucker (KKT) cannot guarantee the optimality of the its solution for this problem. The preference here is rather to apply a nearly optimal unimodular solution. To this end, it is adequate to find the solution of \(\min_x \sum_k |x - \mu_k|\) (as in (19)), then to project it onto the unit circle. Consequently, the algorithm for designing a unimodular sequence with minimized MSL can be derived. Step 4 of MeSiCA in Table 1 becomes:

| Step | Description |
|------|-------------|
| 4a   | For the sequences \(\{\mu_k\}\) and for some initial value (e.g., \(x_n^0 = 0\)), compute the iteration in (22) till some stop criterion is satisfied (e.g., \(|x_{n+1}^t - x_n^t| < \delta\)) and the solution to (20), \(y^*\), is achieved |
| 4b   | Compute \(x\) according to \(x = f(y^*, k)\) |
| 4c   | Set \(k = k + 1\) |

### B. Restricting to a finite-alphabet constellation

With the advancement of digital hardware, it is unequivocally plausible to consider the sequences with arbitrary phases. However, there are applications where restricting the designed sequences into a certain finite-alphabet constellation results in substantial implementation simplification. In this regard, MeSiCA can be generalized to the fixed-constellation scenario, where sequence elements are restricted to a fixed set of points. Bi-phase and polyphase signals are examples of such generalization. In this case, the design problem includes an additional constraint restricting the space from which sequence elements are selected:

\[ x_n \in S, \quad n = 1, \ldots, N. \]

(28)

Here, \(S\) is the constellation. For instance, we have \(S = \{1, -1\}\) and \(S = \{1, -j, -1, j\}\) for bi-phase and quad-phase constellations, respectively. In order to cope the finite-element problem, the converging set notation [49] is applied. Consider the function \(f(x, k) : \mathbb{C} \times (\mathbb{N} \cup \{0\}) \rightarrow \mathbb{C}\), the set of complex numbers \(\mathbb{C}\) converges under the function \(f \) to the set \(S\), if \(\lim_{k\rightarrow \infty} f(x, k) \in S\). This concept allows an iterative algorithm for designing continuous phase sequences to be converted to an algorithm with finite-alphabet outputs. Here, we restrict the argument to binary case. For more details on the theory of converging sets and other constellations see [49]. A suitable \(f\) can be constructed for binary case as

\[ f(x, k) = \text{sgn}(\Re\{x\})|x|e^{-ik}, \]

(29)

where \(v\) is a parameter determining the speed of convergence. In this light, the finite-alphabet MeSiCA (FAMEsiCA) is different from the original MeSiCA in step 4, given below. We name special case of \(S = \{-1, 1\}\) as BiMeSiCA for binary MeSiCA.

### C. Randomization

The computational burden of the MeSiCA comes from two sources: the SVD (i.e., step 2 of each algorithm) and the iteration in step 4. For the latter, it is sufficient to choose a loose criterion. For the former, a method should be adopted to improve the computation speed of SVD. Active research exists on the fast computation of SVD [50], where randomization is applied. Consider the SVD of \(X\) in (18), choose \(S \ll Q < N\), and let \(\Omega\) denote a Gaussian random matrix of the size \((N + kQ - 1) \times S\), we have

\[ \Gamma = X^H \Omega. \]

(30)

where \(\Gamma\) is a \(Q \times S\) matrix. By applying the QR factorization, we have

\[ \Gamma = A R, \]

(31)

such that \(A\) be the \(Q \times S\) orthogonal matrix. Consequently, define \(E\) as the following

\[ E = A^H X^H. \]

(32)
defining its stopping criteria as loop. Note that the inner loop could also be implemented by the number of operations needed for convergence of the inner loop’s final cost is $O(NQK)$ for all $N$ members of the sequence $\{x_n\}$. Therefore, the inner-loop’s final cost is $O(2NQK)$, where $K$ is the minimum number of operations needed for convergence of the inner loop. Note that the inner loop could also be implemented by defining its stopping criteria as $i \leq K$. In brief, computational complexity of MeSiCA and RaMeSiCA in each iteration are $O((N + KQ - 1)Q^2 + 2NQK)$ and $O((N + KQ - 1)Q^2 + 2NQK)$, respectively. Table 2 depicts complexity comparison between MeSiCA, MWISL [6], and WeCAN [5]. In fact, MWISL has a vector-matrix multiplication in step 9 which dominates its complexity with $O(N^2)$; $O(N \log N)$ and $O(N)$ comes from FFT and vector-scalar operations in other steps. Furthermore, per-iteration complexity is not improved by applying SQUAREM. The dominant part of the complexity of WeCAN comes from the matrix-vector multiplication in step 9 which dominates its complexity with $O(N^2)$; $O(N \log N)$ and $O(N)$ comes from FFT and vector-scalar operations in other steps. Moreover, since it is lower-bounded by zero, it should converge. Due to the similarity of the argument, we avoid repeating it here to preserve brevity. Empirical simulations also confirm the convergence of the CA (alternating minimization) in both the waveform design [45] and dictionary learning [54].

The computational complexity of the algorithms presented here is a function of the desired number of suppressed auto-correlation lags. In MeSiCA, complexity in the outer loop is dominated by computing the SVD in (17). Considering the structure of the matrix $\tilde{X}^H$, the usual SVD requires $O((N + KQ - 1)Q^2)$ operations. Applying the low-rank randomized SVD reduces this amount to $O((N + KQ - 1)Q \log(k))$. Besides, the randomized SVD applied here requires no random access to the matrix, indicating that it even performs better for very large matrices [51], [55]. Beside SVD, the inner-loop given in (22) is a major source of time consumption in both MeSiCA and RaMeSiCA. In each iteration of the inner loop, two summations of length $Q$ are computed for all $N$ members of the sequence $\{x_n\}$. Therefore, the inner-loop’s final cost is $O(2NQK)$, where $K$ is the minimum number of operations needed for convergence of the inner loop. Note that the inner loop could also be implemented by defining its stopping criteria as $i \leq K$. In brief, computational

### D. CONVERGENCE AND COMPLEXITY ANALYSIS

Convergence analysis of the algorithms developed here falls under the convergence properties of the general CA methods. Specifically, consider the real-valued target function (15). The two-step CA method divides the optimization problem into alterations between two separate minimization problems. This approach is reported in other applications with the name of alternating minimization [52], which its convergence is studied with more depth [53]. The essence of the convergence is based on the fact that solving the resulted sub-problem decreases the target function in each step. Moreover, since it is lower-bounded by zero, it should converge. Due to the similarity of the argument, we avoid repeating it here to preserve brevity. Empirical simulations also confirm the convergence of the CA (alternating minimization) in both the waveform design [45] and dictionary learning [54].

The computational complexity of the algorithms presented here is a function of the desired number of suppressed auto-correlation lags. In MeSiCA, complexity in the outer loop is dominated by computing the SVD in (17). Considering the structure of the matrix $\tilde{X}^H$, the usual SVD requires $O((N + KQ - 1)Q^2)$ operations. Applying the low-rank randomized SVD reduces this amount to $O((N + KQ - 1)Q \log(k))$. Besides, the randomized SVD applied here requires no random access to the matrix, indicating that it even performs better for very large matrices [51], [55]. Beside SVD, the inner-loop given in (22) is a major source of time consumption in both MeSiCA and RaMeSiCA. In each iteration of the inner loop, two summations of length $Q$ are computed for all $N$ members of the sequence $\{x_n\}$. Therefore, the inner-loop’s final cost is $O(2NQK)$, where $K$ is the minimum number of operations needed for convergence of the inner loop. Note that the inner loop could also be implemented by defining its stopping criteria as $i \leq K$. In brief, computational

### TABLE 2. Comparison for per iteration complexity among MeSiCA, MWISL, WeCAN, and their alternative implementations.

| Algorithm       | MWISL       | WeCAN       | MeSiCA  |
|-----------------|-------------|-------------|---------|
| Computational Complexity | $O(N^2 + \log N + N)$ | $O(N^2 + N)$ | $O((N + KQ - 1)Q^2 + 2NQK)$ |
| Alternative Algorithm | MWISL+SQUAREM | WeCAN+SQUAREM way | RaMeSiCA |
| Computational Complexity | $O(N^2 + \log N + N)$ | $O(N^2 + \log N + N)$ | $O((N + KQ - 1)Q \log(k) + 2NQK)$ |

### TABLE 3. Randomized mean sidelobe cyclic algorithm (RaMeSiCA).

1. Set $\{x_n\}_{n=1}^N$ to some initial sequence, choose $S \ll k_Q < N$ and set $\Omega$ to a Gaussian random matrix of the size $S \times N$ (Initialization).

2. Constitute the matrix $X$ (13); calculate the low-rank SVD of this matrix using equations (30) to (34) to find $U_2$ and $U_2^H$, then constitute $L$ according to (18).

3. For each $x_n$ constitute the sequence $\{\mu_k\}$, i.e., find the elements of $\sqrt{N}L$ whose positions are the same as $x_n$ in $X$.

4. For the sequences $\{\mu_k\}$ and for some initial value (e.g., $x_0 = 0$), compute the iteration in (22) until some stop criteria is satisfied (e.g., $|x_n^{k+1} - x_n^k| < \delta$).

5. Go to step 2 until some stop criterion is satisfied (e.g., $\|X_{new} - X_{old}\|_{1,1} < \varepsilon$).

### IV. SIMULATION RESULTS

In this section, the performance of the proposed algorithms is evaluated. The Barker [1] sequence is chosen here as a result of its prevalence. In order to have a fair evaluation, comparisons with the recent majorization-minimization (MM) method [6] are performed. All the simulations in this section are run on a Corr i5 machine with 16 GB of random access memory (RAM) and 8 MByte of cache. In the following simulations, all algorithms start from the Golomb sequence unless explicitly specified. The following formula defines the Golomb sequence:

$$G(n) = e^{i(n-1)\pi N}, \quad n = 1, \ldots, N.$$ (35)
This sequence is, by definition, unimodular and belongs to the family of polyphase sequences. It is worth mentioning that one of the advantages of the Golomb sequence is that it can be produced for any specified length of \( N \). On the contrary, not all members of the polyphase family have this feature. For instance, the Frank sequence, another member of this family, can be just produced for square lengths. Nevertheless, in order to visualize the results of algorithms, some parameters should be defined here. Accordingly, we define normalized autocorrelation as

\[
\text{Normalized Autocorrelation} = 20 \log_{10} \left( \frac{|r_k|}{r_0} \right), \quad k = 1, \ldots, N. \quad (36)
\]

This definition is essentially the same as the correlation level given in [5]. However, the name normalized autocorrelation is preferred in order to avoid confusion. Furthermore, we define mean autocorrelation sidelobe level (MASL) as

\[
\text{MASL} = 20 \log_{10} \left( \frac{\sum_{k=1}^{N-1} |r_k|}{r_0} \right). \quad (37)
\]

Note that this criterion cannot measure the algorithm performance whenever suppressing a part of the autocorrelation is contemplated. In such a situation, we can apply the modified mean autocorrelation sidelobe level (MMASL), defined according to

\[
\text{MMASL} = 20 \log_{10} \left( \frac{\sum_{i=1}^{Q} |r_k|}{r_0} \right), \quad (38)
\]

can be applied. We also compare the algorithms based on the PSL. In this regard, we define the modified peak sidelobe level (MPSL) by:

\[
\text{MPSL} = 20 \log_{10} \left( \max_{i=1}^{Q} \frac{|r_k|}{r_0} \right). \quad (39)
\]

This metric measures the PSL over the lags of interest: \( r_{k_1}, \ldots, r_{k_Q} \). All measures mentioned above are appropriate to describe the performance.

### A. SIDELOBE SUPPRESSION ABILITY

In this section, the performance of newly introduced algorithms is compared with several benchmarking algorithms. The comparison with methods like Barker or Golomb is performed because they have been applied to many applications. The MeSiCA and RaMeSiCA are also compared with the monotonic minimizer for weighted ISL (MWISL) [6]. This algorithm exploits the MM to minimize a weighted ISL metric.

1) **COMPARISON WITH BARKER (SUPPRESSING MORE THAN HALF OF THE SIDELOBES)**

Barker is the most prevalent binary code desired for its autocorrelation properties. The maximum length of this bi-phase code is 13, where the corresponding code is

\[
x = [1 \ 1 \ 1 \ 1 \ 1 - 1 - 1 - 1 - 1 - 1 - 1]. \quad (40)
\]

It has the lowest possible autocorrelation level in the set of bi-phase codes with the same length. Figure 2 illustrates the comparison of normalized autocorrelation among the MeSiCA, RaMeSiCA, WISL, WeCAN and Barker 13. Here, the length is \( N = 13 \) while \( q_0 = Q = N - 2 = 11 \) for all benchmark algorithms. The termination point for WeCAN is \( \varepsilon = 10^{-8} \), while the MWISL is run for 50000 iterations. Both of these values ensure the convergence of the algorithms. However, these two benchmark algorithms can only suppress the sidelobes down to almost zero if the lags of interest covers less than half of the side lobes, as embraced by their authors [5], [6]. On the other hand, both MeSiCA and RaMeSiCA can suppress beyond half of the side lobes except for the all-side lobes scenario. To illustrate this, we first set \( \varepsilon = 3 \times 10^{-11} \) for both RaMeSiCA and MeSiCA. Interestingly, both algorithms generate the same autocorrelation. It is because both solve the same problem. However, the RaMeSiCA reaches the termination point in slightly fewer iterations: 54410, compared to 55919 for MeSiCA. It is worth mentioning that we didn’t expect significant speedup at this stage since the length is low, while the RaMeSiCA is specifically designed for lengthy sequence speedups. We also included the MeSiCA and RaMeSiCA waveforms for \( \varepsilon = 3 \times 10^{-4} \) and \( \varepsilon = 3 \times 10^{-5} \), respectively, to showcase the suppression for various \( \varepsilon \)-s. It is observed that by setting a lower value of \( \varepsilon \), stronger suppression can be guaranteed at the expense of longer running time. The ultimate limit of \( \varepsilon \), which is in the order of \( 10^{-15} \), is defined by MatLab for handling small numbers.

Meanwhile, it is worth noting that most of the akin algorithms developed in the literature, including those in [5], [6], are not able to suppress more than half of the side lobes. The ability of MeSiCA rests on the fact that this algorithm is produced by relaxing the unimodularity constraint, compared with unimodular sequence design algorithms; this constraint confines the degree of freedom for the target sequence to the half as given in [5] and [6]. Since the difference between MeSiCA and RaMeSiCA is in the SVD part, the same behavior from RaMeSiCA is expected. Here, we further showcase this ability of MeSiCA compared to MWISL. Consider a case where \( N = 20 \), while suppressing the first \( L \) side lobes are desired. We consider 5, 9, 11, and 15 values for \( L \) to particularly include values less, about, and more than the margin of half of the side lobes or 10. Figure 3 illustrates the MWISL and MeSiCA performance for this example. Here, all MWISL scenarios are run for 50000 iterations. For MeSiCA, the \( \varepsilon \) is set to \( \varepsilon = 10^{-12} \) except for the last \( L = 15 \), which is set to \( \varepsilon = 10^{-11} \). It is evident that the MWISL cannot suppress side lobes down to almost zero for \( L = 11 \) and \( L = 15 \), while the MeSiCA can do the task for all scenarios.
2) suppressing apart coefficients

The MeSiCA can suppress apart coefficients as well. However, in order to suppress a specified part of autocorrelation, either \( k_i \) should be calculated such that the cumulative occurrence of ones rests more on the desired coefficient lags in the \( \mathbf{X}\mathbf{X}^H \) or proper values for \( \{k_i\} \) should be found out through trial and error. In this regard, consider a situation where suppression of the first five autocorrelation lags and the lags from \( k = 40 \) to \( k = 50 \) is desired. It is sufficient to choose \( k_0 = 50 \) and \( \{k_i\} = \{1, \ldots, 5\} \cup \{45, \ldots, 50\} \). Each coefficient in \( r_1, \ldots, r_5 \) gets cumulative occurrence of two and each coefficient in \( r_{40}, \ldots, r_{50} \) gets a single occurrence. The result of this configuration is demonstrated in Fig. 4 in blue, where the suppression level of the \( r_1, \ldots, r_5 \) is more than that of \( r_{40}, \ldots, r_{50} \) as expected. To reach this depiction, \( \varepsilon \) is set to \( 0.3 \times 10^{-10} \). This example
also provides an opportunity to demonstrate the applicability of the finite-alphabet scenario. In this regard, MeSiCA and BiMeSiCA are compared for a certain number of iterations \( K = 2000 \) with the same configuration as above. It can be observed that BiMeSiCA is also able to suppress the desired coefficients. However, it cannot compete with MeSiCA in terms of suppression level. This is because it is restricted to the binary constellation.

**3) SUPPRESSION ABILITY OF THE ALGORITHMS**

Here, an example is contemplated for various iteration numbers to evaluate the performance of the proposed algorithms in minimizing the MMASL. Consider a problem where the suppression of the first 24 lags for the sequences of length 100 is contemplated. That is,

\[
\begin{align*}
k_Q = Q & = 24 \\
N & = 100
\end{align*}
\]

For such a scenario, the MMASL metric is computed for the MeSiCA and RaMeSiCA. The result is compared with that of the MWISL algorithm [6]. Fig. 5 shows that an improvement of approximately 10 to 20 dB is achieved. The parameter \( \delta \) is set to 0.01 for both of the proposed algorithms.

**4) TIME PERFORMANCE OF THE ALGORITHMS**

Although depicting the MMASL versus the number of iterations represents some aspects of the convergence of the developed algorithms, it cannot completely characterize the algorithms’ speed. The speed of some steps in each iteration can be improved with special care. This is the motivation behind developing the RaMeSiCA. In order to compare the speed of the algorithms, suppression of \( r_0, \ldots, r_{120} \) is contemplated for a sequence of the length \( N = 256 \), that is,

\[
\begin{align*}
k_Q = Q & = 120 \\
N & = 256
\end{align*}
\]

The suppression level of this scenario measured through the MMASL is depicted in Fig. 6, where the x-axis represents the time in seconds. The comparison among the MeSiCA and RaMeSiCA and the benchmarking algorithm, i.e., MWISL [6], reveals that an advantage of several orders of magnitudes is attainable through the newly developed algorithms. The advantage of RaMeSiCA comes from two facts: the fast-SVD and a deliberate choice of the parameter \( \delta \). Generally, trial-and-error shows that choosing a loose value for the \( \delta \) results in better performance for RaMeSiCA. For instance, in the scenario depicted in Fig. 6 this parameter is chosen to be \( \delta = 0.1 \), for RaMeSiCA versus \( \delta = 0.0001 \) for MeSiCA. Here, the \( \delta \) for both algorithms are tuned by trial and error to get the best performance of the algorithm. Finally, the parameter \( k \), i.e., number of rows in the matrix \( \mathbf{Q} \), is set to 4 for this simulation.

The MeSiCA and RaMeSiCA can also beat the benchmark algorithm in other metrics like PSL, although they are specifically designed for minimizing the MSL and should
be expected to outperform other benchmarks in MSL only. To illustrate their superiority, we compare the MPSL performance among MeSiCA, RaMeSiCA, and MWISL for the same scenario as above. Figure 7 depicts this comparison, where it is observed that both MeSiCA and RaMeSiCA can significantly outperform MWISL in MPSL metric.

V. CONCLUSION

This article deals with designing sequences with a minimized mean sidelobe level (MSL). The MSL minimization is formulated and solved through the cyclic algorithm, where the Weiszfeld algorithm is applied to solve the resulting generic optimization problem. Additionally, imposing the practically important constraint of unimodularity criteria and restricting to finite-element constellations are argued. The computational burden of the developed algorithms is revealed to rest on the embedded SVD. The randomization method is proposed to speed up the SVD to improve the developed algorithms, particularly for generating long sequences. Simulation results affirm the superiority of the proposed methods in suppression level and consumed time. The supremacy in speed and its ability to suppress the autocorrelation’s desired part makes the presented methods suitable for cognitive radars.

APPENDIX

The problem in (19), when $|$ denotes the $l_1$-norm on complex numbers, is a sum-of-absolute-value problem. This appendix is devoted to solving this problem, establishing some different algorithms for minimizing the MSL. Accordingly, the following problem should be solved,

$$\min_{x} \sum_{k} |x - \mu_{ik}| = \min_{x} \sum_{k} |\Re \{x \} - \Re \{\mu_k \}|$$

$$+ |\Im \{x \} - \Im \{\mu_k \}|.$$  \hspace{1cm} (43)

To this end, define

$$y_k = \Re \{x \} - \Re \{\mu_k \},$$  \hspace{1cm} (44)

and

$$z_k = \Re \{x \} - \Im \{\mu_k \}.$$  \hspace{1cm} (45)

By these definitions, (19) is equivalent to

$$\min_{x} \sum_{k} |y_k + z_k|$$

$$y_k + \Re \{x \} - \Re \{\mu_k \} = 0$$

$$z_k + \Im \{x \} - \Im \{\mu_k \} = 0.$$  \hspace{1cm} (46)

Although (46) is not a linear programming (LP) problem, it can be transformed to an LP directly. First, write each of the $y_k$ and $z_k$ as two nonnegative variables,

$$y_k = y_k^+ - y_k^-,$$  \hspace{1cm} (47)

$$z_k = z_k^+ - z_k^-,$$  \hspace{1cm} (48)

where $y_k^+ \geq 0$, $y_k^- \geq 0$ and $z_k^+ \geq 0$, $z_k^- \geq 0$. Hence, the resulting problem is

$$\min_{x} \sum_{k} |y_k^+ - y_k^-| + |z_k^+ - z_k^-|$$

$$y_k^+ + y_k^- + \Re \{x \} - \Re \{\mu_k \} = 0$$

$$z_k^+ + z_k^- + \Im \{x \} - \Im \{\mu_k \} = 0.$$  \hspace{1cm} (49)

This problem is still nonlinear. However, note that in the objective function, $|y_k^+ - y_k^-|$ can be simplified whenever either $y_k^+$ or $y_k^-$ is zero, and similarly $|z_k^+ - z_k^-|$ can be simplified whenever either $z_k^+$ or $z_k^-$ is zero. Hence, by imposing additional restrictions, i.e., $y_k^+ \times y_k^- = 0$ and $z_k^+ \times z_k^- = 0$, the optimization problem in (49) converts into the following problem:

$$\min_{x} \sum_{k} |y_k^+ + y_k^- + z_k^+ + z_k^-|$$

$$y_k^+ + y_k^- + \Re \{x \} - \Re \{\mu_k \} = 0$$

$$z_k^+ + z_k^- + \Im \{x \} - \Im \{\mu_k \} = 0.$$  \hspace{1cm} (50)

Note that $|y_k^+ - y_k^-| = y_k^+ + y_k^-$, whenever $y_k^+ \times y_k^- = 0$ and $|z_k^+ - z_k^-| = z_k^+ + z_k^-$ whenever $z_k^+ \times z_k^- = 0$ are applied. The resulting problem in (50) is an LP, whose exact solution can be derived by applying one of the relevant methods.

REFERENCES

[1] R. H. Barker, “Group synchronizing of binary digital sequences,” in Communication Theory. London, U.K.: Butterworth, 1953, pp. 273–287.

[2] S. W. Golomb, “Two-valued sequences with perfect periodic autocorrelation,” IEEE Trans. Aerosp. Electron. Syst., vol. 28, no. 2, pp. 383–386, Apr. 1992.

[3] R. Heimiller, “Phase shift pulse codes with good periodic correlation properties,” IEEE Trans. Inf. Theory, vol. IT-7, no. 4, pp. 254–257, Oct. 1961.

[4] E. García, J. A. Paredes, F. J. Álvarez, M. C. Pérez, and J. J. García, “Spreading sequences in active sensing: A review,” Signal Process., vol. 106, pp. 88–105, Jan. 2015.
[5] P. Stoica, H. He, and J. Li, “New algorithms for designing unimodular sequences with good correlation properties,” IEEE Trans. Signal Process., vol. 57, no. 4, pp. 1415–1425, Apr. 2009.

[6] J. Song, P. Babu, and D. P. Palomar, “Sequence design to minimize the weighted integrated and peak sidelobe levels,” IEEE Trans. Signal Process., vol. 64, no. 1, pp. 2051–2064, Apr. 2016.

[7] L. Zhao, J. Song, P. Babu, and D. P. Palomar, “A unified framework for low autocorrelation sequence design via majorization-minimization,” IEEE Trans. Signal Process., vol. 65, no. 2, pp. 438–453, Jan. 2017.

[8] H. Esmaeili-Najafabadi, “MIMO radars waveform design using chaos theory,” Ph.D. dissertation, Dept. Elect. Eng., Univ. Isfahan, Isfahan, Iran, 2015.

[9] Z. Cheng, B. Liao, Z. He, J. Li, and C. Han, “A nonlinear-ADMM method for designing MIMO radar constant modulus waveform with low correlation side lobes,” Signal Process., vol. 159, pp. 93–103, Jun. 2019.

[10] M. Soltanalian, M. M. Naghsh, and P. Stoica, “A fast algorithm for designing complementary sets of sequences,” Signal Process., vol. 93, no. 7, pp. 2096–2102, Jul. 2013.

[11] M. B. Alae and S. A. Olamaei, “Waveform design for TDM-MIMO radar systems,” Signal Process., vol. 167, Feb. 2020, Art. no. 107307.

[12] W. Fan, J. Liang, G. Yu, H. C. So, and G. Lu, “Minimum local peak side-lobe level waveform design with correlation and/or spectral constraints,” Signal Process., vol. 171, Jun. 2020, Art. no. 107450.

[13] H. Esmaeili-Najafabadi, M. Ataei, and M. F. Sabahi, “Designing sequence with minimum PSL using chebychev distance and its application for chaotic MIMO radar waveform design,” IEEE Trans. Signal Process., vol. 65, no. 3, pp. 690–704, Feb. 2017.

[14] A. Ramadan, M. El-hoshy, and A. E.-D. S. Hafez, “A new biphase coding LFM for pulse compression radar,” Int. J. Sci. Eng. Res., vol. 5, no. 4, pp. 732–736, 2014.

[15] M. Abdel-Latif, E.-S.-A. Youssef, A. Mokhtar, and M. Madkour, “A new minimum peak side lobe codes in radar systems,” in Proc. IEEE Antennas Propag. Int. Symp., Jun. 2007, pp. 3812–3815.

[16] E.-S.-A. Youssef, A. Mokhtar, M. Madkour, and M. Abdel-Latif, “Simulation of pulse compression performance under influence of background noise in modern surveillance radar,” in Proc. Int. Conf. Comput. Eng. Syst., Nov. 2007, pp. 205–208.

[17] E.-S. A. Youssef, A. Mokhtar, M. Madkour, and M. Abdel-Latif, “A novel multilevel biphase pulse compression codes,” in Proc. ICCCIA, 2007.

[18] C. Ma, T. S. Yeo, C. S. Tan, Y. Qiang, and T. Zhang, “Receiver design for MIMO radar range side lobes suppression,” IEEE Trans. Signal Process., vol. 58, no. 10, pp. 5469–5474, Oct. 2010.

[19] M. Alae-Kerahroodi, M. Modarres-Hashemi, and M. M. Naghsh, “Designing sets of binary sequences for MIMO radar systems,” Signal Process., vol. 67, no. 13, pp. 3347–3360, Jul. 2019.

[20] H. Esmaeili-Najafabadi, H. Leung, and P. W. Moo, “Unimodular waveform design with desired ambiguity function for cognitive radar,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 3, pp. 2489–2496, Jun. 2020.

[21] J. Yang, G. Cui, X. Yu, and L. Kong, “Dual-use signal design for radar and communication via ambiguity function sidelobe control,” IEEE Trans. Veh. Technol., vol. 69, no. 9, pp. 9781–9794, Sep. 2020.

[22] H. Esmaeili-Najafabadi, M. Ataei, and M. F. Sabahi, “Chebychev chaotic polynomials for MIMO radar waveform generation,” IET Radar, Sonar Navigat., vol. 11, no. 2, pp. 330–340, Feb. 2017.

[23] S. Hong, F. Zhou, Y. Dong, Z. Zhao, Y. Wang, and M. Yan, “Chaotic phase-coded waveforms with space-time complementary coding for mimo radar applications,” IEEE Access, vol. 6, pp. 42086–42083, 2018.

[24] W. Fan, J. Liang, H. C. So, and G. Lu, “Min-max metric for spectrally compatible waveform design via log-exponential smoothing,” IEEE Trans. Signal Process., vol. 68, pp. 1075–1090, 2020.

[25] A. Haimovich, R. Blum, and L. Cimini, “MIMO radar with widely separated antennas,” IEEE Signal Process. Mag., vol. 25, no. 1, pp. 116–129, 2008.

[26] A. M. Haimovich, “MIMO radar: A multi-sensor spatially diverse radar architecture,” New Jersey Inst. Technol., Newark, NJ, USA, Tech. Rep. AFRL-SR-AR-TR-09-0046, 2008.

[27] A. G. Stove, A. L. Humre, and C. J. Baker, “Low probability of intercept radar strategies,” IEEE Proc. Radar. Sonar Navigat., vol. 151, no. 5, pp. 249–260, Oct. 2004.

[28] F. Ju, Y. Sun, J. Gao, Y. Hu, and B. Yin, “Image outlier detection and feature extraction via L1-Norm-Based 2D probabilistic PCA,” IEEE Trans. Image Process., vol. 24, no. 12, pp. 4834–4846, Dec. 2015.
HAMID ESMAEILI NAJAFABADI received the B.S. degree in electronics from the Isfahan University of Technology, in 2005, the M.S. degree in communication from the Amirkabir University of Technology, in 2008, and the Ph.D. degree in electrical engineering from the University of Isfahan, in 2017. He has collaborated with some industrial groups, including the Cheetah Group and ICTI. He is currently a Postdoctoral Associate with the University of Calgary. His current research interests include data science, deep learning, radar and optimization theories, generative adversarial networks, and pipeline defect detection.

HENRY LEUNG (Fellow, IEEE) is currently a Professor with the Department of Electrical and Computer Engineering, University of Calgary. Before joining the University of Calgary, he was with the Department of National Defence (DND), Canada, as a Defence Scientist. His current research interests include information fusion, machine learning, the IoT, nonlinear dynamics, robotics, signal, and image processing. He is a fellow of the SPIE. He is an Associate Editor of IEEE Circuits and Systems Magazine and Journal of Sensors. He is a Topic Editor on “robotic sensor” of the International Journal of Advanced Robotic Systems. He is also an Editor of the Springer book series on “Information Fusion and Data Science.”

PETER W. MOO (Senior Member, IEEE) received the B.Sc. degree in mathematics and engineering from Queen’s University, Kingston, in 1993, and the M.S.E. and Ph.D. degrees in electrical engineering: systems from the University of Michigan, in 1995 and 1998, respectively. Since 1999, he has been with the Defence Research and Development Canada (DRDC), where he is currently an Expert Level Defence Scientist and the Group Leader of the Radar Sensing and Exploitation Section. He is a DRDC’s Lead Scientist for naval phased array radar and high frequency surface wave radar. He is coauthor of the book Adaptive Radar Resource Management (Academic Press, 2015). He has coauthored more than 50 journal articles, conference papers, and scientific reports. His research interests include radar resource management, multiple-input multiple-output radar, and space-time adaptive processing. He serves on the Editorial Board for IET Radar, Sonar & Navigation. He is the Chair of TTCP ISTAR Technical Panel 2 on RF Sensing, and is the Chair of NATO Research Task Group SET-223 on Adaptive Radar Resource Management.

---

K. Skretting and K. Engan, “Recursive least squares dictionary learning algorithm,” IEEE Trans. Signal Process., vol. 58, no. 4, pp. 2121–2130, Apr. 2010.

G. Martinsson, A. Gillman, E. Liberty, N. Halko, V. Rokhlin, S. Hao, Y. Shkolnisky, P. Young, J. Tropp, M. Tygert, and F. Woolfe, “Randomized methods for computing the Singular Value Decomposition (SVD) of very large matrices,” Work. Alg. Mod. Data Sets, Palo Alto, CA, USA, Jun. 2010.