EURASIP Journal on Advances
in Signal Processing

RESEARCH Open Access

Fast algorithm for designing periodic/aperiodic sequences with good correlation and stopband properties

Liang Tang*, Yongfeng Zhu and Qiang Fu

Abstract

Periodic/aperiodic sequences with low autocorrelation sidelobes are widely used in many fields, such as communication and radar systems. Besides the correlation property, the frequency stopband property is often considered in the sequence design when the systems work in a crowded electromagnetic environment. In this paper, we aim at designing periodic/aperiodic sequences with low autocorrelation sidelobes and arbitrary frequency stopbands, and propose an efficient algorithm named FFT (fast Fourier transform)-based conjugate gradient algorithm. To calculate the step size efficiently, a method based on Taylor series expansion is developed. By changing the number of FFT points, the proposed algorithm can be easily used to generate periodic/aperiodic sequences. Since the gradient and step size can be implemented by FFT operations and Hadamard product, the whole algorithm is computationally efficient and can be used to design very long sequences. Numerical experiments show that the proposed algorithm has better performance than the state-of-the-art algorithms in terms of the running time.

Keywords: Autocorrelation, Integrated sidelobe level, Stopband property, Gradient, Periodic/aperiodic sequence design

1 Introduction

Sequences with low periodic or aperiodic autocorrelation sidelobes have been studied for a long time since the 1950s. The applications of such sequences cover both military field and civil field, such as the pulse compression radar and wireless communication [1, 2]. In pulse compression radar, the output of the pulse compression can be regarded as the convolution of each scattering point and the waveform's autocorrelation. As the traditional waveforms (e.g., linear frequency modulation (LFM) signal) suffer high range sidelobes, the detection performance and imaging quality of radar systems are limited. Thus, sequences with low autocorrelation sidelobes are applied to suppress the sidelobe jamming from the adjacent strong scattering points and thus improve the detection performance of weak target [3]. Moreover, sequences with low autocorrelation sidelobes have many advantages in the code division multiple access (CDMA) systems [4], for instance, reducing the influence of the multipath effect and improving the synchronization precision.

Owing to the extensive application and great practical value of the sequences with low autocorrelation sidelobes, many scholars have shown a great interest in generating such sequences. The related studies can be classified into two categories. The first category is the design of binary sequences by using the search methodologies, such as exhaustive search method [5] and evolutionary algorithm [6]. And the second category is the design of polyphase sequences [7–13]. In recent years, with the development of the optimization theory, the problem of designing polyphase sequences with low autocorrelation sidelobes has been a research hotspot. Stochastic optimization algorithms [7, 8] were applied to design such sequences at the early stage. However, because of the high computational burden, these algorithms are impractical and unable to design sequences with large length. In order to fix this problem more efficiently, algorithms named CAN (cyclic algorithm new) and Accelerated-MISL (accelerated monotonic minimizer for integrated sidelobe level) which were both based on fast Fourier transform (FFT)

*Correspondence: tangliang12@nudt.edu.cn
[1] National University of Defense Technology, Deya Road, Changsha, China

were proposed in [9, 10]. In [9], the CAN algorithm was proposed to design unimodular sequences of large length by minimizing a criterion which is "almost equivalent" to the integrated sidelobe level (ISL) metric. Different from the CAN algorithm, the Accelerated-MISL algorithm [10] was deduced via majorization-minimization (MM) method. Since the Accelerated-MISL algorithm is based on an acceleration scheme which can greatly reduce the iteration number, it outperforms the CAN algorithm on both the merit factor (MF) and the computational efficiency. Besides, there are some literatures on designing sequences with zero correlation zone [11–13] or low periodic autocorrelation [14].

Apart from the correlation property, the stopband property (which means suppressing several frequency bands) is usually considered in sequence design for suppressing narrowband interferences or avoiding the certain frequency bands that reserved for some applications, such as navigation and military communications [15–19]. Since the discontinuous frequency of the transmit waveform leads to high autocorrelation sidelobes, the correlation and stopband properties are both considered. At the early stage, the cyclic algorithms named SCAN (stopband CAN) and WeSCAN (weighted SCAN) were proposed in [20] to design sequences with good correlation and stopband properties. But since the WeSCAN requires N FFT operations and the eigenvalue decomposition at each iteration, its convergence speed is slow. Afterwards, the improved cyclic algorithms named SDCA (steepest descent-based cyclic algorithm) [21] and MMSE-WISL [22] are proposed for sparse frequency waveform design on the basis of the CAN and WeCAN (weighted CAN) algorithms [9]. However, like the WeSCAN algorithm, these two improved algorithms are also time-consuming and not suitable for long waveform design [22]. In recent years, some efficient algorithms, such as the majorization-minimization (MM) method [10], gradient-based algorithms [23, 24], Lagrange programming neural network (LPNN) [25], alternating projection [26], and alternating direction method of multipliers (ADMM) [27], have also been applied to unimodular sequence design with stopband property, where LPNN and ADMM can be applied to generate both the periodic and aperiodic waveforms. It is worth noting that the sidelobe suppression means flatting the sequence spectrum. Thus, the suppression of frequency stopbands seems more difficult than the sidelobe suppression, which means that suppressing the stopbands increases more complexity.

In this paper, we consider the problem of designing periodic/aperiodic sequences with low autocorrelation sidelobes and arbitrary frequency stopbands. By using the relationship between the autocorrelation function and power spectrum density (PSD), the problem is formulated as an unconstrained minimization problem with respect to the sequence phase in frequency domain. To solve this problem, an efficient algorithm named FFT-based conjugate gradient algorithm (which we call FCGA) is proposed. Unlike the traditional gradient algorithm, the searching step size of the FCGA which is hard to calculate directly is deduced via the Taylor series expansion. As the gradient and step size of the FCGA can be implemented by FFT operations and Hadamard product, the algorithm is efficient and can design sequences with large length. Moreover, by selecting the number of the FFT points, the proposed algorithm can be easily applied to design periodic or aperiodic sequences.

The remaining sections of the paper are organized as follows. In Section 2, the sequence design problem that incorporates both the correlation and stopband properties is formulated. In Section 3, the phase gradient and step size are derived, and then, a modified descent direction is given to guarantee the monotonicity of the FCGA. To show the effectiveness of the proposed algorithm, several numerical experiments are presented in Section 4. Finally, Section 5 gives the conclusion.

Notation: Boldface upper case letters denote matrices while boldface lower case letters denote column vectors. $(\cdot)^*$, $(\cdot)^T$, and $(\cdot)^H$ denote the complex conjugate, transpose, and conjugate transpose, respectively. $\|\cdot\|$ denotes the Euclidean norm of vectors. $\circ$ denotes the Hadamard product. $\mathbf{X}\,(m,n)$ denotes the (mth,nth) element of matrix $\mathbf{X}$. $\mathrm{R}e(\cdot)$ and $\mathrm{I}m(\cdot)$ denote the real and imaginary part, respectively. Diag ($\mathbf{x}$) is a diagonal matrix formed with $\mathbf{x}$. $\mathcal{F}_N(\cdot), \mathcal{F}_N^{-1}(\cdot)$ denote the $N-$point FFT and inverse FFT (IFFT) operations, respectively.

2 Problem formulation

As mentioned in Section 1, this paper focuses on the problem of designing sequences with good correlation and frequency stopband properties. That is to say, the sequences should satisfy two conditions: one is low autocorrelation sidelobes (here we minimize the power of the sidelobes, i.e., ISL metric); the other one is low stopband power. In the following, two criterions are established to measure the autocorrelation and the stopband power, and then, we formulate the design problem as an unconstrained minimization problem.

2.1 Correlation property

Let $\{x_n\}_{n=0}^{N-1}$ denote the complex sequence to be designed. The vector form of the sequence can be expressed as

$$\mathbf{x} = [x_0, ..., x_{N-1}]^T. \tag{1}$$

The periodic and aperiodic autocorrelations of the sequence are respectively defined as

$$\tilde{r}_k = \sum_{n=0}^{N-1} x_n x^*_{(n-k) \bmod N} = \tilde{r}^*_{-k},$$
$$r_k = \sum_{n=k}^{N-1} x_n x^*_{n-k} = r^*_{-k}, k = 0, ..., N-1. \tag{2}$$

Then, the corresponding integrated sidelobe levels (ISLs) which express the goodness of the periodic and aperiodic correlation properties are given by

$$I_{per} = \sum_{k=1}^{N-1} |\tilde{r}_k|^2, \quad I_{aper} = \sum_{k=1}^{N-1} |r_k|^2. \tag{3}$$

Let

$$\mathbf{r}_1 = \left[\tilde{r}_0, \tilde{r}_1, ..., \tilde{r}_{N-1}\right]^T,$$
$$\mathbf{r}_2 = \left[r_0, r_1, ..., r_{N-1}, 0, r^*_{N-1}, ..., r^*_1\right]^T. \tag{4}$$

denote the periodic and aperiodic autocorrelation vectors, respectively. Then, I_{per} and I_{aper} can be written more compact as

$$I_{per} = \sum_{k=0}^{N-1} |\tilde{r}_k|^2 - |\tilde{r}_0|^2 = \mathbf{r}_1^H \mathbf{r}_1 - N^2,$$
$$I_{aper} = \frac{1}{2}\left(\sum_{k=1-N}^{N-1} |r_k|^2 - |r_0|^2\right) = \frac{1}{2}\left(\mathbf{r}_2^H \mathbf{r}_2 - N^2\right), \tag{5}$$

where we assume $\tilde{r}_0 = r_0 = \sum_{n=0}^{N-1} x_n x^*_n = N$. Let $\mathbf{F}_{\tilde{N}}$ be the $\tilde{N} \times \tilde{N}$ discrete Fourier transform (DFT) matrix with the following expression

$$\mathbf{F}_{\tilde{N}}(m, n) = e^{-j\frac{2mn\pi}{\tilde{N}}}, 0 \le m, n < \tilde{N}. \tag{6}$$

Then, the $\tilde{N} \times \tilde{N}$ inverse discrete Fourier transform (IDFT) matrix is $\mathbf{F}^H_{\tilde{N}}/\tilde{N}$. It is well known that the autocorrelation function is the inverse Fourier transform of the power spectrum density (PSD); thus, we have the following relationship [13, 14]:

$$\mathbf{r}_1 = \mathcal{F}_N^{-1}(\mathbf{p}_1) = \frac{1}{N}\mathbf{F}_N^H \mathbf{p}_1,$$
$$\mathbf{r}_2 = \mathcal{F}_{2N}^{-1}(\mathbf{p}_2) = \frac{1}{2N}\mathbf{F}_{2N}^H \mathbf{p}_2. \tag{7}$$

where $\mathbf{p}_1, \mathbf{p}_2$ are the PSD of lengths N and $2N$, respectively. By substituting (7) into (5), the frequency domain expressions of I_{per} and I_{aper} are given by

$$I_{per} = \frac{1}{N^2}\mathbf{p}_1^H \mathbf{F}_N \mathbf{F}_N^H \mathbf{p}_1 - N^2 = \frac{1}{N}\left(\mathbf{p}_1^H \mathbf{p}_1 - N^3\right),$$
$$I_{aper} = \frac{1}{2}\left(\frac{1}{4N^2}\mathbf{p}_2^H \mathbf{F}_{2N} \mathbf{F}_{2N}^H \mathbf{p}_2 - N^2\right) = \frac{1}{4N}\left(\mathbf{p}_2^H \mathbf{p}_2 - 2N^3\right), \tag{8}$$

where $\mathbf{F}_{\tilde{N}} \mathbf{F}^H_{\tilde{N}} = \tilde{N}\mathbf{I}_{\tilde{N}}, \tilde{N} = N, 2N$, and $\mathbf{I}_{\tilde{N}}$ denotes the $\tilde{N} \times \tilde{N}$ identity matrix.

Generally, the problem of designing periodic/aperiodic sequences with low autocorrelation sidelobes can be regarded as a minimization problem of the periodic/aperiodic ISL metrics. From (8), it is easy to see that the metrics of the periodic and aperiodic autocorrelations have the same form. Thus, by ignoring the constant terms in (8), the periodic/aperiodic ISL metrics can be equivalent to the following unified criterion:

$$J_{ACF} = \frac{1}{2\tilde{N}}\mathbf{p}^H \mathbf{p}, \tag{9}$$

where $\mathbf{p} = [p_0, ..., p_{\tilde{N}-1}]^T$ and $\tilde{N}$ is the number of FFT points. When $\tilde{N} = N$, (9) is the criterion for the periodic autocorrelation. And when $\tilde{N} = 2N$, (9) becomes the criterion for the aperiodic autocorrelation.

2.2 Stopband property

In crowded electromagnetic environment, the signal spectrum is often polluted by the interference or the signals from other users. Thus, designing sequences with frequency stopbands is sometimes quite necessary. Without loss of generality, we consider the normalized frequencies here.

Assume the set of frequency stopbands $\Omega \subset [0, 1]$ can be written as

$$\Omega = \bigcup_{k=1}^{N_s} \left(f_{k1}, f_{k2}\right), \tag{10}$$

where $\left(f_{k1}, f_{k2}\right)$ denotes one stopband and N_s denotes the number of the stopbands. Considering the $\tilde{N}-$point FFT operations, the corresponding point set of the stopband set can be expressed as

$$\Omega_1 = \bigcup_{k=1}^{N_s} \left[\lfloor \tilde{N} f_{k1} \rfloor, \lceil \tilde{N} f_{k2} \rceil\right] \subset \left[0, \tilde{N}\right], \tag{11}$$

where $\lfloor \cdot \rfloor, \lceil \cdot \rceil$ are the floor and ceiling operations. Therefore, the suppression of the stopband set Ω is equivalent to minimizing the PSD in Ω_1, i.e., $\mathbf{p}(k), k \in \Omega_1$. Let us define the frequency weight vector as

$$\mathbf{w}_f = \left[\bar{w}_0, ..., \bar{w}_{\tilde{N}-1}\right]^T,$$
$$\bar{w}_p = \begin{cases} 1, & p \in \Omega_1 \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

As the PSD is nonnegative, the criterion for the stopband property can be constructed as the square of the Euclidean norm of the weighted PSD:

$$J_{PSD} = \left\|\mathbf{w}_f \circ \mathbf{p}\right\|^2 = \mathbf{p}^H \text{Diag}\left(\mathbf{w}_f\right) \mathbf{p}. \tag{13}$$

Actually, the criterion J_{PSD} is the total power in frequency stopbands. Thus, by minimizing (13), the spectral power in Ω can be suppressed.

2.3 The minimization problem

In order to suppress the autocorrelation sidelobes and the stopband power, both J_{ACF} and J_{PSD} should be minimized. Thus, the design problem can be regarded as a bi-objective optimization (or Pareto optimization) problem. However, since the frequency stopband leads to high sidelobes, it is impossible to find a solution that minimizes both J_{ACF} and J_{PSD}. To make a compromise between correlation sidelobes and frequency stopbands, here we apply the scalarization procedure, i.e., turning the bi-objective optimization problem into a single-objective optimization problem. After scalarizing the problem, the single objective function that incorporates (9) and (13) can be expressed as

$$\begin{aligned} J_T &= \lambda J_{PSD} + (1-\lambda) J_{ACF} \\ &= \lambda \mathbf{p}^H \mathrm{Diag}\left(\mathbf{w}_f\right)\mathbf{p} + \frac{1-\lambda}{2\tilde{N}}\mathbf{p}^H\mathbf{p} \\ &= \mathbf{p}^H \mathrm{Diag}\left(\mathbf{w}\right)\mathbf{p}. \end{aligned} \tag{14}$$

where $\lambda \in [0,1]$ is a weight coefficient by which we can balance the relative weight between J_{ACF} and J_{PSD}, and $\mathbf{w} = \lambda \mathbf{w}_f + \frac{1-\lambda}{2\tilde{N}}$.

Defining $\mathbf{f} = \left[f_0, ..., f_{\tilde{N}-1}\right]^T$ as the $\tilde{N}-$point DFT of $\mathbf{x}$, and

$$\mathbf{a}_p = \left[1, e^{j\omega_p}, ..., e^{j\omega_p(N-1)}\right]^T, \omega_p = \frac{2\pi}{\tilde{N}}p, p = 0, ..., \tilde{N}-1, \tag{15}$$

then one can get

$$\begin{aligned} &\mathbf{f} = \left[\mathbf{a}_0^H\mathbf{x}, ..., \mathbf{a}_{\tilde{N}-1}^H\mathbf{x}\right]^T. \\ &\mathbf{p} = \mathbf{f}^* \circ \mathbf{f} = \left[\mathbf{x}^H\mathbf{a}_0\mathbf{a}_0^H\mathbf{x}, ..., \mathbf{x}^H\mathbf{a}_{\tilde{N}-1}\mathbf{a}_{\tilde{N}-1}^H\mathbf{x}\right]^T. \end{aligned} \tag{16}$$

By substituting (16) into (14), the objective function J_T can be reformulated as

$$J_T = \sum_{p=0}^{\tilde{N}-1} w_p \left(\mathbf{x}^H\mathbf{a}_p\mathbf{a}_p^H\mathbf{x}\right)^2, \tag{17}$$

where $w_p = \lambda \bar{w}_p + \frac{1-\lambda}{2\tilde{N}}$ is the pth element of the weight vector $\mathbf{w}$.

In general, the transmitter components such as power amplifier have a maximum amplitude clip [2]. To maximize the transmitter power, it is desired for sequence amplitude to reach this amplitude clip, which means that the sequence should be unimodular. Therefore, considering the unimodular constraint, the problem of interest can be written as

$$\begin{aligned} &\min\ J_T = \sum_{p=0}^{\tilde{N}-1} w_p \left(\mathbf{x}^H\mathbf{a}_p\mathbf{a}_p^H\mathbf{x}\right)^2 \\ &\text{s.t.}\quad |x_n| = 1, \quad n = 0, ..., N-1. \end{aligned} \tag{18}$$

From (18), we can see that the unimodular constraint is the constraint related to the sequence amplitude. Thus, the minimization problem can be solved by optimizing the sequence phase. Let

$$\boldsymbol{\phi} = \left[\phi_0, ..., \phi_{N-1}\right]^T \tag{19}$$

denote the phase vector of the sequence $\mathbf{x}$, then we have

$$\mathbf{x} = \left[e^{j\phi_0}, ..., e^{j\phi_{N-1}}\right]^T. \tag{20}$$

Thus, the problem (18) can be seen as an unconstrained minimization problem on $\boldsymbol{\phi}$:

$$\min\quad J_T\left(\boldsymbol{\phi}\right) = \sum_{p=0}^{\tilde{N}-1} w_p \left(\mathbf{x}^H\mathbf{a}_p\mathbf{a}_p^H\mathbf{x}\right)^2. \tag{21}$$

3 The FCGA algorithm

A good property of the Accelerated-MISL algorithm in [10] is the monotone decreasing property. But this algorithm minimizes the majorization function rather than the original function. To minimize the objective function (21) directly and guarantee the monotonically decreasing property of the algorithm, the conjugate gradient algorithm (CGA) is applied here. In this section, we first deduce the gradient and the step size, which are two important parts of the CGA, and then summarize the FCGA algorithm.

3.1 Phase gradient

To deduce the phase gradient $\nabla_{\boldsymbol{\phi}} J_T$, we first work out the derivative of the function $J_T\left(\boldsymbol{\phi}\right)$ with respect to the phase $\phi_i, i = 0, ..., N-1$:

$$\frac{\partial J_T}{\partial \phi_i} = 2\sum_{p=0}^{\tilde{N}-1} w_p \left(\mathbf{x}^H\mathbf{a}_p\mathbf{a}_p^H\mathbf{x}\right) \frac{\partial \mathbf{x}^H\mathbf{a}_p\mathbf{a}_p^H\mathbf{x}}{\partial \phi_i}. \tag{22}$$

Let

$$\mathbf{A}_p = \mathbf{a}_p\mathbf{a}_p^H, \quad \mathbf{A}_p\left(m+1, n+1\right) = d_{mn}^p, \quad 0 \le m, n \le N-1. \tag{23}$$

Then, the gradient $\nabla_{\boldsymbol{\phi}} J_T$ can be written as

$$\nabla_{\boldsymbol{\phi}} J_T = 2\sum_{p=0}^{\tilde{N}-1} w_p \left(\mathbf{x}^H\mathbf{A}_p\mathbf{x}\right) \nabla_{\boldsymbol{\phi}} \left(\mathbf{x}^H\mathbf{A}_p\mathbf{x}\right). \tag{24}$$

In the following, we derive the explicit expression of $\nabla_{\boldsymbol{\phi}}\left(\mathbf{x}^H\mathbf{A}_p\mathbf{x}\right)$. According to (20) and (23), the expansion of $\mathbf{x}^H\mathbf{A}_p\mathbf{x}$ is given by

$$\mathbf{x}^H\mathbf{A}_p\mathbf{x}=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}a_{mn}^p e^{j(\phi_n-\phi_m)}. \tag{25}$$

Then, the derivative of $\mathbf{x}^H\mathbf{A}_p\mathbf{x}$ with respect to ϕ_i can be denoted as

$$\frac{\partial \mathbf{x}^H\mathbf{A}_p\mathbf{x}}{\partial\phi_i}=jx_i\sum_{m=0}^{N-1}a_{mi}^p x_m^* - jx_i^*\sum_{n=0}^{N-1}a_{in}^p x_n. \tag{26}$$

Since $\mathbf{A}_p = \mathbf{A}_p^H$, we have $a_{mi}^p = \left(a_{im}^p\right)^*$. Thus, the derivative (26) can be rewritten as

$$\frac{\partial \mathbf{x}^H\mathbf{A}_p\mathbf{x}}{\partial\phi_i}=2\mathrm{Re}\left(-jx_i^*\sum_{n=0}^{N-1}a_{in}^p x_n\right). \tag{27}$$

By stacking (27) in a vector, the gradient $\nabla_{\boldsymbol{\phi}}\left(\mathbf{x}^H\mathbf{A}_p\mathbf{x}\right)$ is given by

$$\nabla_{\boldsymbol{\phi}}\left(\mathbf{x}^H\mathbf{A}_p\mathbf{x}\right)=2\mathrm{Re}\left(-j\mathbf{x}^*\circ\left(\mathbf{A}_p\mathbf{x}\right)\right). \tag{28}$$

According to (24) and (28), the explicit expression of phase gradient is given by (see the Appendix 1)

$$\nabla_{\boldsymbol{\phi}}J_T=4\tilde{N}\mathrm{Re}\left(-j\mathbf{x}^*\circ\mathbf{y}_{1:N}\right), \tag{29}$$

where $\mathbf{y}_{1:N}$ denotes the first N elements of $\mathbf{y}$, and

$$\mathbf{y}=\mathcal{F}_{\tilde{N}}^{-1}\left(\mathbf{w}\circ\left|\mathcal{F}_{\tilde{N}}\left(\mathbf{x}\right)\right|^2\circ\mathcal{F}_{\tilde{N}}\left(\mathbf{x}\right)\right). \tag{30}$$

3.2 Step size calculation via Taylor series expansion

The traditional methods for obtaining the step size are the linear search methods. As the linear search methods cost much iteration and computation, the best way to obtain the step size is direct calculation. In this subsection, we adopt the Taylor series expansion to deduce the step size.

Assume the present iteration point is $\mathbf{x}^k$, and the corresponding iteration direction is $\mathbf{d}^k=\left[d_0^k,...,d_{N-1}^k\right]^T$. Then, the next iteration point can be expressed as

$$\boldsymbol{\phi}^{k+1}=\boldsymbol{\phi}^k+\mu\mathbf{d}^k, \mathbf{x}^{k+1}=\mathbf{x}^k\circ\left[e^{j\mu d_0^k},...,e^{j\mu d_{N-1}^k}\right]^T, \tag{31}$$

where μ is the step size. Essentially, the linear search problem is a minimization problem with respect to μ which can be expressed as follows:

$$\min_{\mu>0}\ J_T\left(\mu\right)=\sum_{p=0}^{\tilde{N}-1}w_p\left(\left(\mathbf{x}^{k+1}\right)^H\mathbf{A}_p\mathbf{x}^{k+1}\right)^2. \tag{32}$$

Let

$$\mathbf{z}^{k+1}=\left[\left(\mathbf{x}^{k+1}\right)^H\mathbf{A}_0\mathbf{x}^{k+1},...,\left(\mathbf{x}^{k+1}\right)^H\mathbf{A}_{\tilde{N}-1}\mathbf{x}^{k+1}\right]^T, \tag{33}$$

then the derivative of $J_T\left(\mu\right)$ with respect to μ can be written as

$$\begin{aligned}\frac{\partial J_T\left(\mu\right)}{\partial\mu}&=\frac{\partial\left(\mathbf{w}^T\left(\mathbf{z}^{k+1}\circ\mathbf{z}^{k+1}\right)\right)}{\partial\mu}\\&=2\mathbf{w}^T\left(\mathbf{z}^{k+1}\circ\frac{\partial\mathbf{z}^{k+1}}{\partial\mu}\right).\end{aligned} \tag{34}$$

To simplify (34), we first expand $\left(\mathbf{x}^{k+1}\right)^H\mathbf{A}_p\mathbf{x}^{k+1}$ as follows:

$$\left(\mathbf{x}^{k+1}\right)^H\mathbf{A}_p\mathbf{x}^{k+1}=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}a_{mn}^p\left(x_m^k\right)^*x_n^k e^{j\mu\left(d_n^k-d_m^k\right)}, \tag{35}$$

where x_n^k, d_n^k are respectively the $(n+1)$th element of $\mathbf{x}^k$ and $\mathbf{d}^k$. By using the Taylor series expansion and then keeping the first three terms, the exponential term $e^{j\mu\left(d_n^k-d_m^k\right)}$ in (35) can be approximated as

$$e^{j\mu\left(d_n^k-d_m^k\right)}\approx 1+j\mu\left(d_n^k-d_m^k\right)-\frac{\mu^2}{2}\left(d_n^k-d_m^k\right)^2. \tag{36}$$

By substituting (36) into (35), $\left(\mathbf{x}^{k+1}\right)^H\mathbf{A}_p\mathbf{x}^{k+1}$ can be rewritten as

$$\begin{aligned}&\left(\mathbf{x}^{k+1}\right)^H\mathbf{A}_p\mathbf{x}^{k+1}\\&=\left(\mathbf{x}^k\right)^H\mathbf{A}_p\mathbf{x}^k-2\mu\mathrm{I}m\left(\left(\mathbf{x}^k\right)^H\mathbf{A}_p\mathbf{x}_1^k\right)\\&\quad-\mu^2\left(\mathrm{Re}\left(\left(\mathbf{x}^k\right)^H\mathbf{A}_p\mathbf{x}_2^k\right)-\left(\mathbf{x}_1^k\right)^H\mathbf{A}_p\mathbf{x}_1^k\right),\end{aligned} \tag{37}$$

where $\mathbf{x}_1^k=\mathbf{x}^k\circ\mathbf{d}^k$, $\mathbf{x}_2^k=\mathbf{x}^k\circ\mathbf{d}^k\circ\mathbf{d}^k$. Let us define

$$\begin{aligned}\mathbf{f}^k&=\mathcal{F}_{\tilde{N}}\left(\mathbf{x}^k\right)=\left[\mathbf{a}_0^H\mathbf{x}^k,...,\mathbf{a}_{\tilde{N}-1}^H\mathbf{x}^k\right]^T,\\\mathbf{f}_1^k&=\mathcal{F}_{\tilde{N}}\left(\mathbf{x}_1^k\right)=\left[\mathbf{a}_0^H\mathbf{x}_1^k,...,\mathbf{a}_{\tilde{N}-1}^H\mathbf{x}_1^k\right]^T,\\\mathbf{f}_2^k&=\mathcal{F}_{\tilde{N}}\left(\mathbf{x}_2^k\right)=\left[\mathbf{a}_0^H\mathbf{x}_2^k,...,\mathbf{a}_{\tilde{N}-1}^H\mathbf{x}_2^k\right]^T.\end{aligned} \tag{38}$$

Then, according to (33), (37), and (38), we have

$$\mathbf{z}^{k+1}=\left|\mathbf{f}^k\right|^2-2\mu\mathbf{t}_2-\mu^2\mathbf{t}_1, \tag{39}$$

where

$$\begin{aligned}\mathbf{t}_1&=\mathrm{Re}\left(\left(\mathbf{f}^k\right)^*\circ\mathbf{f}_2^k\right)-\left(\mathbf{f}_1^k\right)^*\circ\mathbf{f}_1^k,\\\mathbf{t}_2&=\mathrm{I}m\left(\left(\mathbf{f}^k\right)^*\circ\mathbf{f}_1^k\right).\end{aligned} \tag{40}$$

By substituting (39) into (34), the derivative in (34) can be written more compactly as

$$
\begin{aligned}
&\frac{\partial J_T(\mu)}{\partial \mu} \\
&= -4\mathbf{w}^T\left(\left(\left|\mathbf{f}^k\right|^2 - 2\mu\mathbf{t}_2 - \mu^2\mathbf{t}_1\right)\circ(\mathbf{t}_2+\mu\mathbf{t}_1)\right) \\
&= -4\left(a\mu^3 + b\mu^2 + c\mu + d\right),
\end{aligned}
\tag{41}
$$

where

$$
\begin{aligned}
&a = -\mathbf{w}^T(\mathbf{t}_1\circ\mathbf{t}_1), \\
&b = -3\mathbf{w}^T(\mathbf{t}_1\circ\mathbf{t}_2), \\
&c = -2\mathbf{w}^T(\mathbf{t}_2\circ\mathbf{t}_2) + \mathbf{w}^T\left(\left|\mathbf{f}^k\right|^2\circ\mathbf{t}_1\right), \\
&d = \mathbf{w}^T\left(\left|\mathbf{f}^k\right|^2\circ\mathbf{t}_2\right).
\end{aligned}
\tag{42}
$$

Let $\partial J_T(\mu)/\partial\mu = 0$, then we have

$$
a\mu^3 + b\mu^2 + c\mu + d = 0. \tag{43}
$$

It is well known that there is at least a real root in the roots of a cubic equation with real coefficients. Thus, the approximate step size can be calculated directly by solving (43). Since the searching direction is descendant, problem (32) has at least one minimum point greater than 0. Then, we can choose the positive root of (43) which is closest to zero as the step size.

3.3 Algorithm summary

After deducing the phase gradient and the step size, it is easy to solve the unconstrained problem (21) by using the conjugate gradient algorithm (CGA). In order to calculate the search direction effectively, here we apply the classical Polak-Ribiere-Polyak-CGA (PRP-CGA) whose direction can be calculated according to the gradient. The searching direction of the PRP-CGA can be expressed as

$$
\mathbf{d}^{k+1} = -\mathbf{g}^{k+1} + \frac{\left(\mathbf{g}^{k+1}-\mathbf{g}^k\right)^T\mathbf{g}^{k+1}}{\left\|\mathbf{g}^k\right\|^2}\mathbf{d}^k, \tag{44}
$$

where $\mathbf{g}^k$ and $\mathbf{d}^k$ are the gradient and searching direction of the kth iteration, respectively. Since the step size is an approximate value, $\left(\mathbf{g}^{k+1}\right)^T\mathbf{d}^k$ may not be zero. Thus, $\mathbf{d}^{k+1}$ may not be descendant, i.e.,

$$
\begin{aligned}
&\left(\mathbf{g}^{k+1}\right)^T\mathbf{d}^{k+1} \\
&= -\left\|\mathbf{g}^{k+1}\right\|^2 + \frac{\left(\mathbf{g}^{k+1}-\mathbf{g}^k\right)^T\mathbf{g}^{k+1}}{\left\|\mathbf{g}^k\right\|^2}\left(\mathbf{g}^{k+1}\right)^T\mathbf{d}^k < 0
\end{aligned}
\tag{45}
$$

is not always satisfied. For guaranteeing that the searching direction is descendant, we apply the following modified direction:

$$
\begin{aligned}
\tilde{\mathbf{d}}^{k+1} &= -\mathbf{g}^{k+1} + \frac{\left(\mathbf{g}^{k+1}-\mathbf{g}^k\right)^T\mathbf{g}^{k+1}}{\left\|\mathbf{g}^k\right\|^2}\mathbf{d}^k, \\
\mathbf{d}^{k+1} &= \begin{cases} \tilde{\mathbf{d}}^{k+1}, & \left(\mathbf{g}^{k+1}\right)^T\tilde{\mathbf{d}}^{k+1} < 0 \\ -\mathbf{g}^{k+1}, & \left(\mathbf{g}^{k+1}\right)^T\tilde{\mathbf{d}}^{k+1} \geq 0 \end{cases}.
\end{aligned}
\tag{46}
$$

From (46), we can see that when the approximate step size can guarantee the direction $\tilde{\mathbf{d}}^{k+1}$ of the PRP-CGA is descendant (i.e., $\left(\mathbf{g}^{k+1}\right)^T\tilde{\mathbf{d}}^{k+1} < 0$), the $\tilde{\mathbf{d}}^{k+1}$ is selected as the searching direction. But when the step size makes $\tilde{\mathbf{d}}^{k+1}$ no longer a descent direction (i.e., $\left(\mathbf{g}^{k+1}\right)^T\tilde{\mathbf{d}}^{k+1} \geq 0$), we choose the negative gradient direction as the searching direction. By doing so, the descent direction is always guaranteed. On the basis of the above derivation, the FCGA algorithm can be summarized in Algorithm 1.

```
Algorithm 1 FCGA
Initialization: $k = 0, \lambda, \mathbf{w}_f, N, \tilde{N}, \mathbf{x}^0, \mathbf{f}^0 = \mathcal{F}_{\tilde{N}}\left(\mathbf{x}^0\right)$,
$\mathbf{g}^0 = \nabla_{\phi}J_T\left(\mathbf{x}^0\right), \mathbf{d}^0 = -\mathbf{g}^0$.
Repeat
1: $\mathbf{f}_1^k = \mathcal{F}_{\tilde{N}}\left(\mathbf{x}^k\circ\mathbf{d}^k\right), \mathbf{f}_2^k = \mathcal{F}_{\tilde{N}}\left(\mathbf{x}^k\circ\mathbf{d}^k\circ\mathbf{d}^k\right)$.
2: $\mathbf{t}_1 = \mathrm{Re}\left(\left(\mathbf{f}^k\right)^*\circ\mathbf{f}_2^k\right) - \left(\mathbf{f}_1^k\right)^*\circ\mathbf{f}_1^k, \mathbf{t}_2 = \mathrm{Im}\left(\left(\mathbf{f}^k\right)^*\circ\mathbf{f}_1^k\right)$.
3: Compute the coefficients $a, b, c, d$ according to (42).
4: Solving the cubic equation (43), and then choose the positive root which is closest to zero as the step size $\mu^k$.
5: $\mathbf{x}^{k+1} = \mathbf{x}^k\circ e^{j\mu^k\mathbf{d}^k}, \mathbf{f}^{k+1} = \mathcal{F}_{\tilde{N}}\left(\mathbf{x}^{k+1}\right)$.
6: $\mathbf{y} = \mathcal{F}_{\tilde{N}}^{-1}\left(\mathbf{w}\circ\left|\mathbf{f}^{k+1}\right|^2\circ\mathbf{f}^{k+1}\right)$.
7: $\mathbf{g}^{k+1} = 4\tilde{N}\mathrm{Re}\left(-j\left(\mathbf{x}^{k+1}\right)^*\circ\mathbf{y}_{1:N}\right)$.
8: Compute the searching direction $\mathbf{d}^{k+1}$ according to (46).
9: $k = k + 1$.
Until convergence
```

From Algorithm 1, we can see that the proposed algorithm can be easily implemented by the FFT operations and the Hadamard product. In order to analyze the per iteration computation of the algorithm, we calculate the numbers of additions and multiplications for each step in Algorithm 1, and the result is shown in Table 1. As shown in this table, the total number of the additions and multiplications is $21\tilde{N} + 12N + 8\tilde{N}\log\tilde{N} \leq 33\tilde{N} + 8\tilde{N}\log\tilde{N}$. With the increase of $\tilde{N}$, the total number approaches $8\tilde{N}\log\tilde{N}$, i.e., the computation of four FFT operations. Thus, the computation complexity of each iteration is about $\mathcal{O}(\tilde{N}\log\tilde{N})$. Like the other FFT-based algorithms

Table 1 The numbers of additions and multiplications for each step in Algorithm 1

Step	Addition	Multiplication
1	$3N + 2\tilde{N}\log\tilde{N}$	$2\tilde{N}\log\tilde{N}$
2	$3\tilde{N}$	$\tilde{N}$
3	$10\tilde{N}$	$5\tilde{N}$
4	0	0
5	$N + \tilde{N}\log\tilde{N}$	$\tilde{N}\log\tilde{N}$
6	$2\tilde{N} + \tilde{N}\log\tilde{N}$	$\tilde{N}\log\tilde{N}$
7	N	0
8	$4N$	$3N$
Total	$15\tilde{N} + 9N + 4\tilde{N}\log\tilde{N}$	$6\tilde{N} + 3N + 4\tilde{N}\log\tilde{N}$

(such as the CAN algorithm and the Accelerated-MISL algorithm), the FCGA is also computationally efficient.

In fact, the per iteration computation is not the only factor that affects the computational efficiency of the algorithm. Another factor is the iteration number. For example, the per iteration computation of the CAN (needs two FFT operations at each iteration) and Accelerated-MISL (needs four FFT operations at each iteration) algorithms are $15\tilde{N} + 8\tilde{N}\log\tilde{N}$ and $4\tilde{N}\log\tilde{N}$ [9, 10], respectively. Although the per iteration computation of the Accelerated-MISL is larger than that of the CAN, the Accelerated-MISL is more efficient than the CAN [10]. Compared to these two algorithms, the per iteration computation of the FCGA is a little larger. But it does not mean that the FCGA is slower than the CAN and Accelerated-MISL algorithms. In the next section, we will provide several experiments to validate the effectiveness of the proposed algorithm. Actually, in our test, the FCGA is more efficient due to fewer iterations.

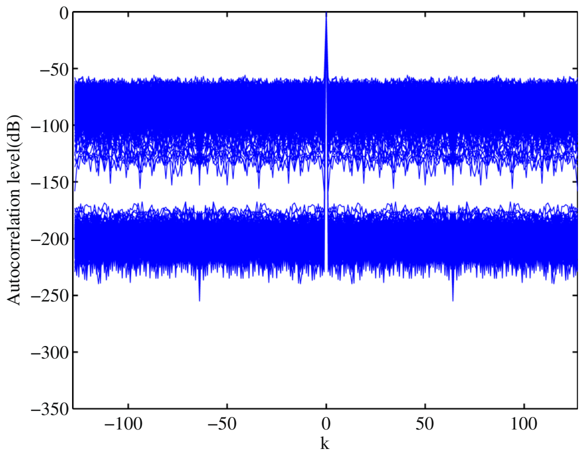

Fig. 1 Autocorrelation level of the sequences generated by PeCAN

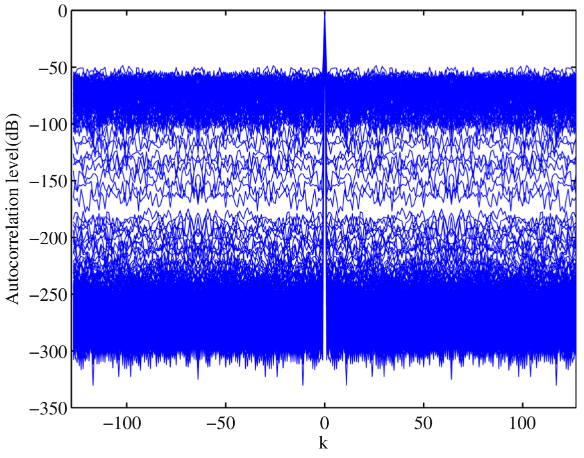

Fig. 2 Autocorrelation level of the sequences generated by ADMM

4 Simulation results and discussion

To illustrate the effectiveness of the proposed FCGA and compare the performance with the existing ones, several numerical experiments are presented in this section. In the first two experiments, we carry out the design of periodic and aperiodic sequences with low autocorrelation sidelobes, respectively. Then, we consider the problem of designing sequences with stopband property in the third experiment. All the experiments are performed on a PC with a 3.60-GHz i7-4790 CPU and 8-GB RAM. The software environment is Matlab 2012b. In the following experiments, if there is no special declaration, all the algorithms are initialized by the unimodular sequence $\mathbf{x}^0 = \left\{e^{j2\pi\phi_n}\right\}_{n=0}^{N-1}$, where $\{\phi_n\}_{n=0}^{N-1}$ are independent random variables uniformly distributed in $[0, 1]$.

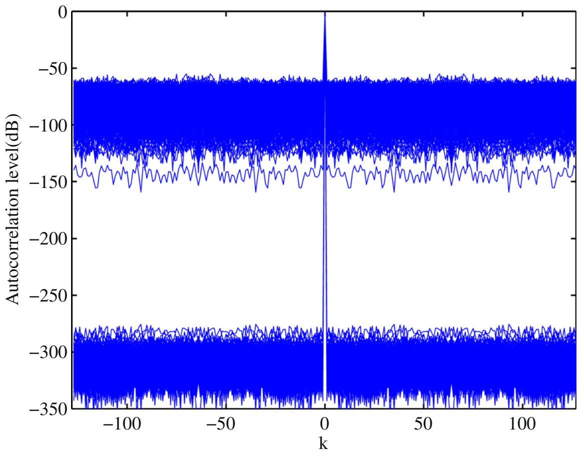

Fig. 3 Autocorrelation level of the sequences generated by FCGA

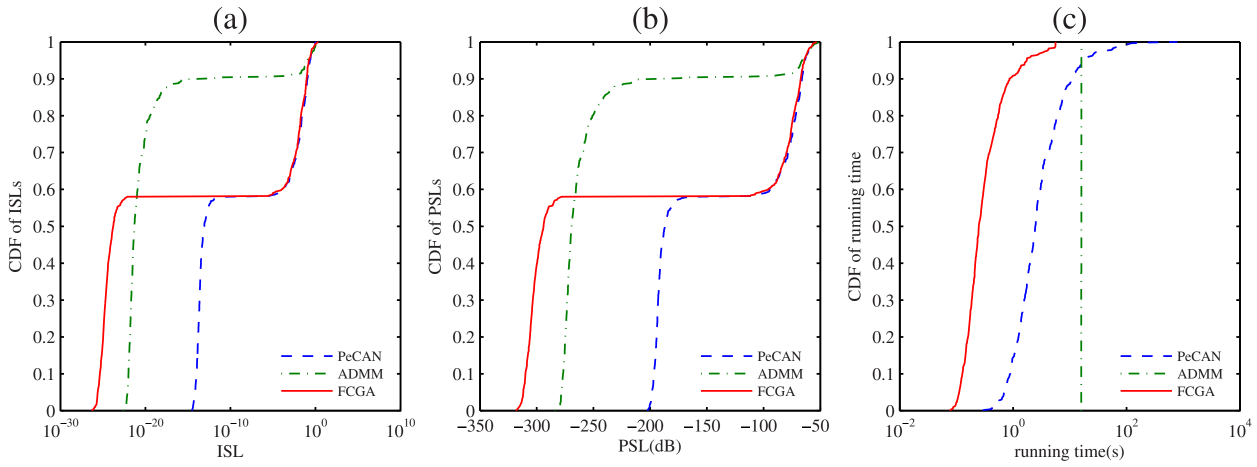

Fig. 4 The cumulative distribution functions of the **a** ISLs, **b** PSLs, and **c** running time

4.1 Periodic sequence design with low autocorrelation sidelobes

First, we apply the proposed algorithm to generate periodic sequences of length $N = 128$ with impulse-like autocorrelation function (i.e., the case of $\lambda = 0, \tilde{N} = N$) and compare the performance with the classical PeCAN [14] and the state-of-the-art ADMM [27] algorithms. Both algorithms are efficient at generating periodic sequence with impulse-like autocorrelation. The Matlab code of the PeCAN was downloaded from the website [28] of the book [2]. The stopping criterions of PeCAN and ADMM algorithms are the same as the ones applied in [2] and [27], and the FCGA is stopped when $\left\|\mathbf{x}^{k+1} - \mathbf{x}^{k}\right\| \leq 10^{-14}$. In this experiment, we initialize the three algorithms by 1000 random unimodular sequences and then generate 1000 sequences for each algorithm. The normalized autocorrelation level (NAL) of the sequences designed by the PeCAN, ADMM, and FCGA are shown in Figs. 1, 2, and 3, respectively, where the NAL is defined as

$$r_{norm}(k) = 20\log_{10}\left|\frac{r_k}{r_0}\right|,\ k = 1-N, ..., N-1, k \neq 0. \tag{47}$$

It can be observed that in all these figures, there exist two distinct ranges for the sidelobe distribution. This is because that for all algorithms, the optimal sequence is a local optimal solution. When the algorithms are initialized by random sequences, the generated sequence may reach global optimal (sequence with sidelobes that are less than − 300 dB) or not. Comparing Figs. 1, 2, and 3, we can see that the FCGA can generate the best sequence whose peak sidelobe level (PSL) is about − 318 dB.

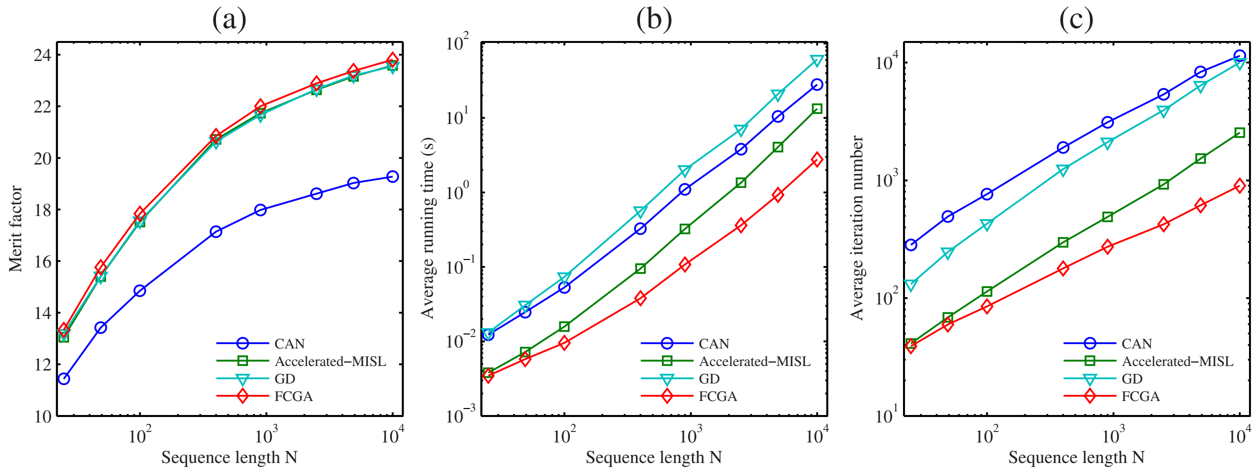

Fig. 5 a–**c** The performance metrics of different algorithms initialized by random sequences

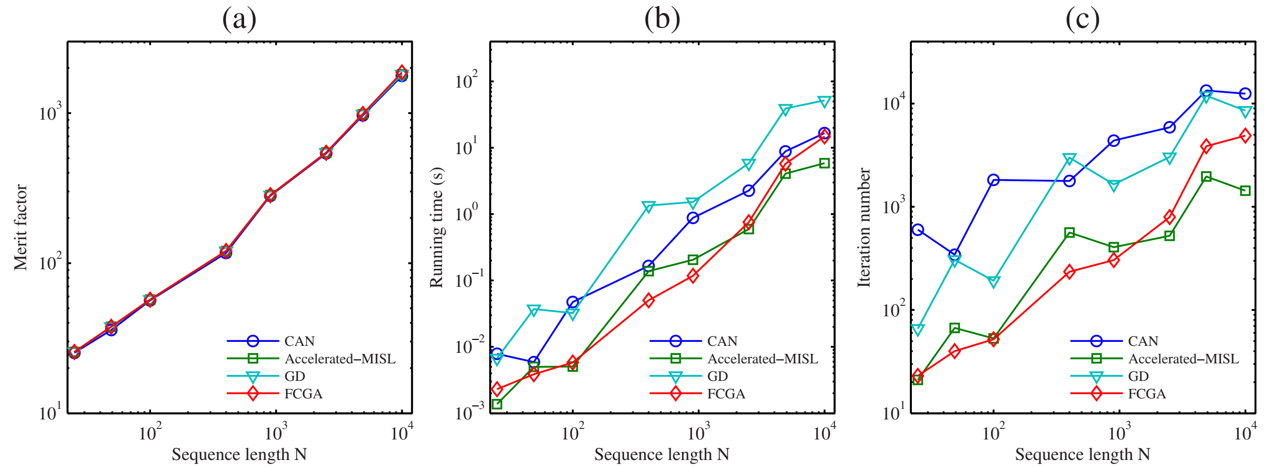

Fig. 6 a–c The performance metrics of different algorithms initialized by Frank sequence

To fully compare the sidelobe performance, we calculate the ISLs and PSLs of all 1000 sequences for each algorithm and record the running time of each trial. The cumulative distribution functions (CDF) of the ISLs, PSLs, and running time are presented in Fig. 4. Since both ISL and PSL are the criterions related to the autocorrelation, the CDF of the ISLs and PSLs are similar as can be seen from Fig. 4a, b. In Fig. 4b, we can see that about 55.7% of the PSL of the FCGA sequences is less than − 285 dB, while the PeCAN and ADMM algorithms are difficult to produce sequences with such PSL. Compared with the PeCAN, the overall sidelobe performance of the FCGA is better. The only drawback of the proposed algorithm is that about 41.8% of the PSLs of the FCGA sequences are located in [−100, −50] dB, while the ADMM has the proportion of only 10.3%. However, as shown in Fig. 4c, the running time of the FCGA is about an order of magnitude smaller than that of the PeCAN and ADMM algorithms, which indicates the former is more computationally efficient. It is worth noting that since the ADMM is stopped after 2×10^5 iterations, the running time of ADMM is basically unchanged.

4.2 Aperiodic sequences design with low autocorrelation sidelobes

In this subsection, we investigate the performance of the proposed algorithm by designing aperiodic sequences (i.e., the case of $\lambda = 0, \tilde{N} = 2N$). To show the advantage of the proposed algorithm, we compare the performance with several algorithms, including the CAN [9], the Accelerated-MISL [10], and the gradient descent (GD) algorithm in [23]. Among the three contrast algorithms,

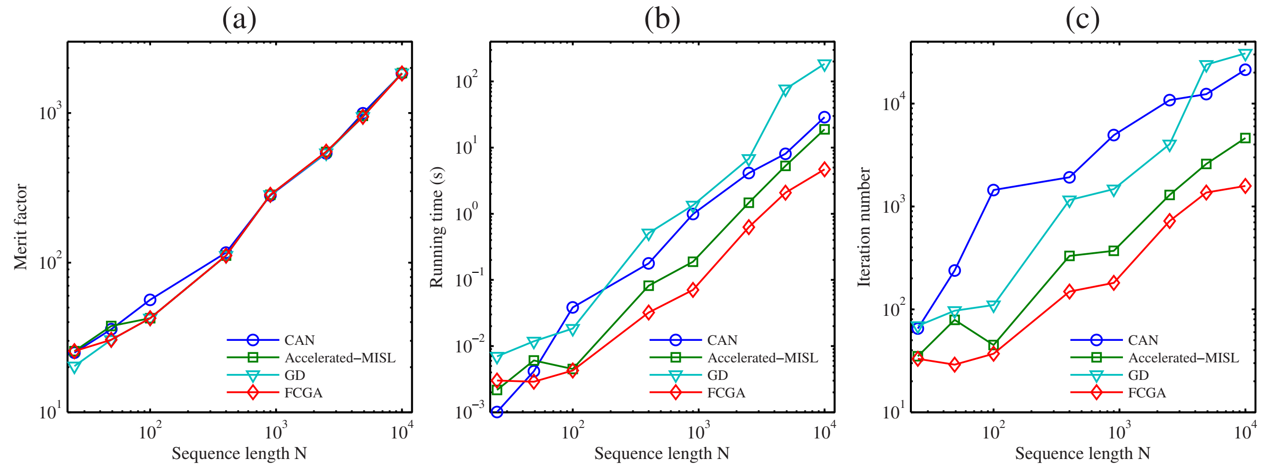

Fig. 7 a–c The performance metrics of different algorithms initialized by Golomb sequence

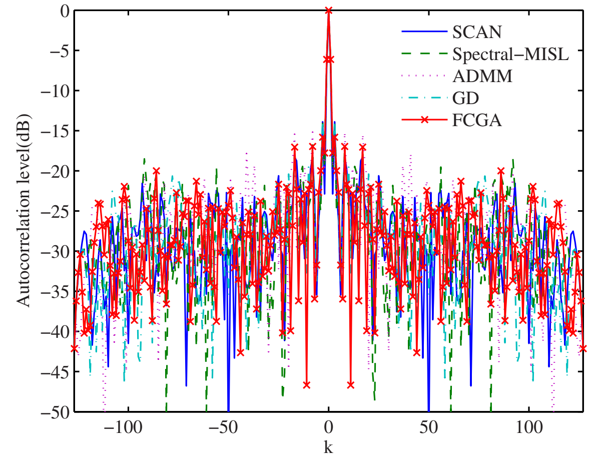

Fig. 8 Autocorrelation level of the sequences generated by five different algorithms

the CAN is the classical algorithm for this design problem, and the other two algorithms are proposed recently. All these algorithms are good at designing aperiodic sequence with low sidelobes and also have high computational efficiency. Since the ADMM is inferior to the CAN and Accelerated-MISL algorithms in terms of the aperiodic autocorrelation sidelobes [27], here we do not take the ADMM into account. The stopping criterion is set to be $\left\|\mathbf{x}^{k+1}-\mathbf{x}^{k}\right\| \leq 10^{-3}$ for all algorithms.

In this experiment, we compare the quality of the designed sequences and the running times of these algorithms. Generally, the sequence quality can be measured by the merit factor (MF) (the large the better) which is defined as

$$\mathrm{MF}=\frac{|r_0|^2}{2\sum_{k=1}^{N-1}|r_k|^2}=\frac{N^2}{2\mathrm{ISL}}. \tag{48}$$

First of all, we use the random sequences to initialize the algorithms. To obtain the average values of the MF and running time, we repeat the four algorithms 300 times for each of the following lengths: $N = 2^5, 2^6, ..., 2^{13}$. Figure 5 shows the performance metrics (average merit factor, average running time, and average iteration number) of different algorithms. As we can see in Fig. 5a, the average merit factors of the sequences designed by the FCGA, GD, and Accelerated-MISL are basically the same. This is because that these three algorithms are all gradient-based algorithms which are different from the CAN. Actually, the MF of the sequences designed by the FCGA is a little larger than that of the GD and Accelerated-MISL. From Fig. 5b, it can be easily observed that the FCGA is faster than the other three algorithms, especially when the sequence length is very large. Although the FCGA costs a little more computations than the CAN and Accelerated-MISL algorithms at each iteration, the superiority of the FCGA comes from the reduction of the iteration number. Figure 5c shows the average iteration numbers of the four algorithms. From this figure, we can see that the average iteration number of the FCGA is the least. This may be because that (i) the FCGA minimizes the objective function (ISL) directly, rather than a more loose objective function which the Accelerated-MISL applied, and (ii) in our test, the step size of the FCGA seems more accurate than that of the GD in [23], which means that the FCGA has faster convergence speed. Taking the merit factor and the running time into account, the proposed FCGA is better than the other three algorithms.

Next, we use the Frank sequence and Golomb sequence to initialize the four algorithms, and the results are shown in Figs. 6 and 7, respectively. It is well known that both the Frank sequence and the Golomb sequence have good autocorrelation property. Thus, after initializing by these two kinds of sequences, the algorithms can produce sequence with better sidelobe performance. In Figs. 6a and 7a, the MF of different algorithms are almost the same and far greater than the result in Fig. 5a. This indicates that the Frank sequence and Golomb sequence can lead to higher MF than the random sequence. Moreover, from Figs. 6b and 7b, we can see that the FCGA is still an efficient and competitive algorithm when initialized by the Frank sequence or the Golomb sequence.

4.3 Sequence design with stopband property

To further verify the effectiveness of the FCGA, here we consider the problem of designing aperiodic sequence with low autocorrelation sidelobes and stopband property.

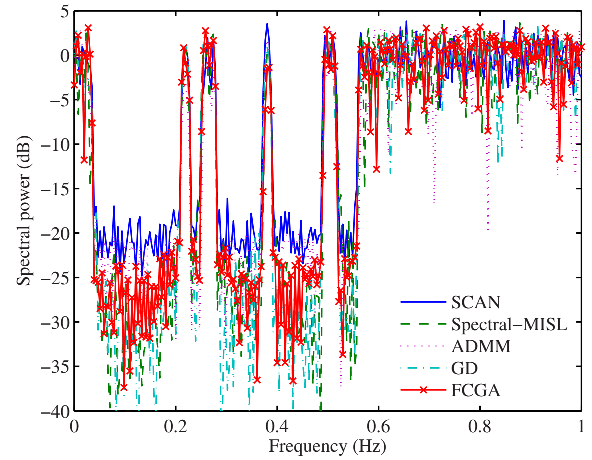

Fig. 9 Spectral power of sequences generated by five different algorithms

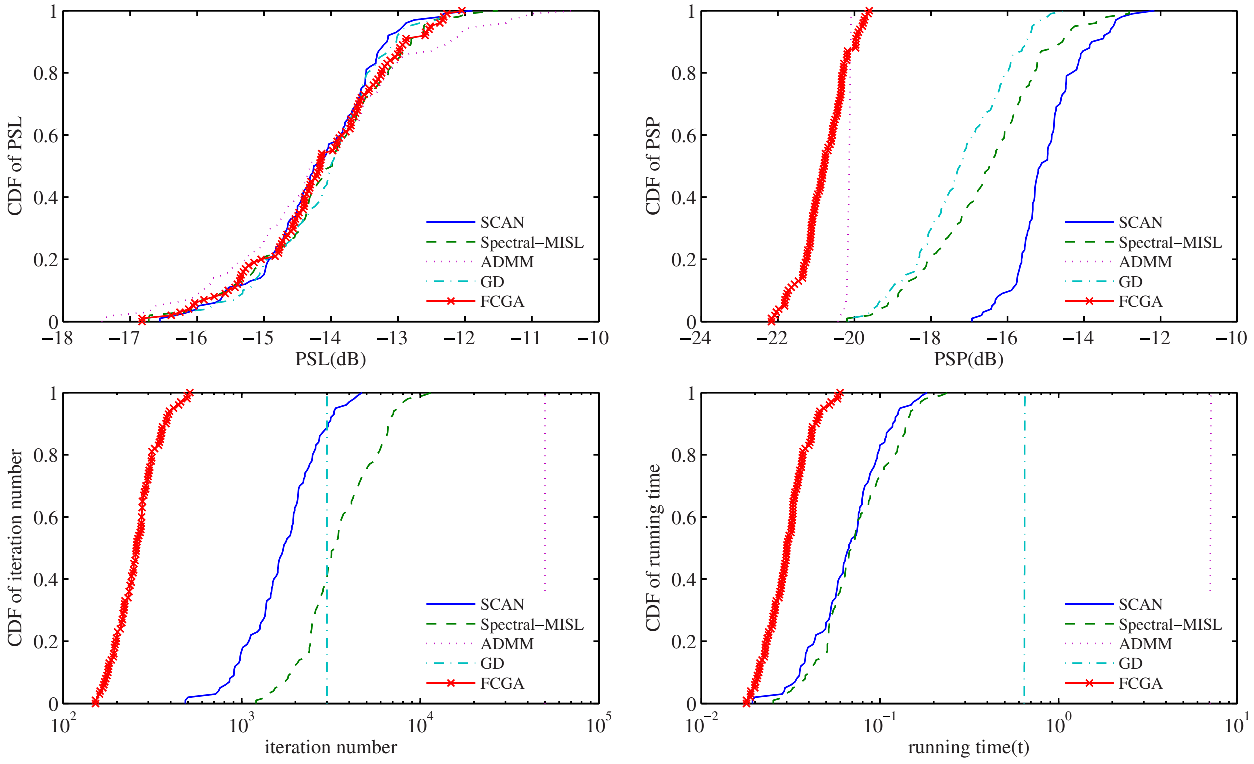

Fig. 10 The cumulative distribution functions of the PSL, PSP, iteration number, and running time

The contrast algorithms are the SCAN [20], Spectral-MISL [10], ADMM, and GD [23] respectively, where the ADMM is the modified version for the synthesis of aperiodic unimodular sequences. Assume that the sequence length is $N = 128$, and the normalized frequency stopband considered here is $[0.04, 0.21] \cup [0.23, 0.25] \cup [0.28, 0.37] \cup [0.39, 0.49] \cup [0.52, 0.56]$. In this experiment, we choose $\left\|\mathbf{x}^{k+1} - \mathbf{x}^{k}\right\| \leq 10^{-3}$ as the stopping criterion of the SCAN, Spectral-MISL, and FCGA algorithms. And the weight coefficient λ in these three algorithms are set to be 0.9, 10^4, 0.9, respectively. The parameters of the ADMM are the same as those in [27]. For the GD algorithm, we set the maximum iteration number $T = 3000$, $\alpha_{acf} = 0.1$, and $\alpha_{spec} = 0.9$.

Figures 8 and 9 give an example of the autocorrelation level and spectrum power of the sequences designed by the five algorithms. The PSL of the sequences designed by the SCAN, Spectral-MISL, ADMM, GD, and FCGA algorithms are − 14.01 dB, − 14.09 dB, − 15.64 dB, − 13.80 dB, and − 15.81 dB, respectively. And the corresponding peak stopband power (PSP) are − 16.12 dB, − 15.71 dB, − 20.13 dB, − 17.77 dB, and − 20.73 dB. Like the other four algorithms, the proposed algorithm can suppress the autocorrelation sidelobes and the spectral power in frequency stopbands simultaneously.

Next, to eliminate the randomness, we repeat the algorithms 100 times and then record the PSL, PSP, iteration number N_I, and running time t. Figure 10 shows the cumulative distribution functions of the four performance parameters $(\text{PSL}, \text{PSP}, N_I, t)$. And the average values of the parameters are provided in Table 2. From Fig. 10 and Table 2, we can see that the PSL of these algorithms are basically the same, while the PSP of the ADMM and proposed algorithm (which have comparable performance) are lower than that of the other three algorithms. Moreover, the running time in Fig. 10 indicates that the proposed algorithm is more computationally efficient than the other four algorithms owing to the fewer iterations.

Table 2 Comparison of the performance parameters between the five algorithms

	PSL (dB)	PSP (dB)	N_I	$t(s)$
SCAN	− 14.20	− 14.96	1887	0.071
Spectral-MISL	− 14.07	− 16.55	3953	0.084
ADMM	− 14.14	− 20.14	50000	7.097
GD	− 14.09	− 17.21	3000	0.646
FCGA	− 14.13	− 20.78	269	0.028

5 Conclusions

In this paper, we have proposed an efficient algorithm named FCGA for designing periodic/aperiodic sequences with low autocorrelation sidelobes and low frequency stopband power. The FCGA algorithm is derived based on the conjugate gradient algorithm which can guarantee the monotonicity of the objective function. By changing the number of FFT points, the design of periodic and aperiodic sequences can be easily achieved. Since the main step of the FCGA can be implemented by FFT operations and Hadamard product, the proposed algorithm is computationally efficient and can be used to design sequences of large length. Several numerical experiments have been provided to demonstrate the superiority of the proposed algorithm.

Appendix 1: Derivation of phase gradient

By substituting (23) and (28) into (24), the gradient $\nabla_{\phi} J_T$ can be rewritten as

$$\begin{aligned}\nabla_{\phi} J_T &= 2\sum_{p=0}^{\tilde{N}-1} w_p \left(\mathbf{x}^H \mathbf{A}_p \mathbf{x}\right) 2\mathrm{R}e\left(-j\mathbf{x}^* \circ \left(\mathbf{A}_p \mathbf{x}\right)\right) \\ &= 4\mathrm{Re}\left(-j\mathbf{x}^* \circ \left(\sum_{p=0}^{\tilde{N}-1} \mathbf{a}_p w_p \left(\mathbf{x}^H \mathbf{a}_p \mathbf{a}_p^H \mathbf{x}\right) \mathbf{a}_p^H \mathbf{x}\right)\right) \\ &= 4\mathrm{Re}\left(-j\mathbf{x}^* \circ \left(\left[\mathbf{a}_0, ..., \mathbf{a}_{\tilde{N}-1}\right] \begin{bmatrix} w_0 \left|\mathbf{a}_0^H \mathbf{x}\right|^2 \mathbf{a}_0^H \mathbf{x} \\ \vdots \\ w_{\tilde{N}-1} \left|\mathbf{a}_{\tilde{N}-1}^H \mathbf{x}\right|^2 \mathbf{a}_{\tilde{N}-1}^H \mathbf{x} \end{bmatrix}\right)\right).\end{aligned} \tag{49}$$

Defining $\mathbf{A} = \left[\mathbf{a}_0, ..., \mathbf{a}_{\tilde{N}-1}\right]$, then we have

$$\begin{aligned}\begin{bmatrix} w_0 \left|\mathbf{a}_0^H \mathbf{x}\right|^2 \mathbf{a}_0^H \mathbf{x} \\ \vdots \\ w_{\tilde{N}-1} \left|\mathbf{a}_{\tilde{N}-1}^H \mathbf{x}\right|^2 \mathbf{a}_{\tilde{N}-1}^H \mathbf{x} \end{bmatrix} &= \begin{bmatrix} w_0 \\ \vdots \\ w_{\tilde{N}-1} \end{bmatrix} \circ \begin{bmatrix} \left|\mathbf{a}_0^H \mathbf{x}\right|^2 \\ \vdots \\ \left|\mathbf{a}_{\tilde{N}-1}^H \mathbf{x}\right|^2 \end{bmatrix} \circ \begin{bmatrix} \mathbf{a}_0^H \mathbf{x} \\ \vdots \\ \mathbf{a}_{\tilde{N}-1}^H \mathbf{x} \end{bmatrix} \\ &= \mathbf{w} \circ \left|\mathbf{A}^H \mathbf{x}\right|^2 \circ \left(\mathbf{A}^H \mathbf{x}\right),\end{aligned} \tag{50}$$

where function $|\cdot|^2$ is applied element-wise to the vector. According to (49) and (50), $\nabla_{\phi} J_T$ can be expressed as

$$\nabla_{\phi} J_T = 4\mathrm{R}e\left(-j\mathbf{x}^* \circ \left(\mathbf{A}\left(\mathbf{w} \circ \left|\mathbf{A}^H \mathbf{x}\right|^2 \circ \left(\mathbf{A}^H \mathbf{x}\right)\right)\right)\right). \tag{51}$$

From the expression of $\mathbf{A}$, we can see that the $\tilde{N} \times N$ matrix $\mathbf{A}^H$ is composed of the first N columns of $\tilde{N} \times \tilde{N}$ DFT matrix $\mathbf{F}_{\tilde{N}}$, i.e., $\mathbf{A}^H = \mathbf{F}_{:,1:N}$ (to facilitate the derivation, here we ignore the subscript of $\mathbf{F}_{\tilde{N}}$). By substituting $\mathbf{F}$ for $\mathbf{A}$, (51) can be rewritten as

$$\nabla_{\phi} J_T = 4\tilde{N}\mathrm{R}e\left(-j\mathbf{x}^* \circ \left(\frac{1}{\tilde{N}} \mathbf{F}_{:,1:N}^H \left(\mathbf{w} \circ \left|\mathbf{F}_{:,1:N} \mathbf{x}\right|^2 \circ \left(\mathbf{F}_{:,1:N} \mathbf{x}\right)\right)\right)\right). \tag{52}$$

Therefore, the phase gradient (52) can be implemented by FFT operations. Let

$$\mathbf{y} = \mathcal{F}_{\tilde{N}}^{-1}\left(\mathbf{w} \circ \left|\mathcal{F}_{\tilde{N}}\left(\mathbf{x}\right)\right|^2 \circ \mathcal{F}_{\tilde{N}}\left(\mathbf{x}\right)\right), \tag{53}$$

then the phase gradient can be further simplified as

$$\nabla_{\phi} J_T = 4\tilde{N}\mathrm{R}e\left(-j\mathbf{x}^* \circ \mathbf{y}_{1:N}\right), \tag{54}$$

where $\mathbf{y}_{1:N}$ denotes the first N elements of $\mathbf{y}$.

Abbreviations

ADMM: Alternating direction method of multipliers; CAN: Cyclic algorithm new; CDF: Cumulative distribution function; CDMA: Code division multiple access; FCGA: FFT-based conjugate gradient algorithm; (I)DFT: (Inverse) discrete fourier transform; (I)FFT: (Inverse) fast fourier transform; ISL: Integrated sidelobe level; LFM: Linear frequency modulation; LPNN: Lagrange programming neural network; MF: Merit factor; MISL: Monotonic minimizer for integrated sidelobe level; MM: Majorization-minimization; MMSE: Minimum mean square error; NAL: Normalized autocorrelation level; PeCAN: Periodic-correlation cyclic algorithm new; PRP-CGA: Polak-Ribiere-Polyak-CGA; PSD: Power spectrum density; PSL: Peak sidelobe level; PSP: Peak stopband power; SCAN: Stopband cyclic algorithm new; SDCA: Steepest descent-based cyclic algorithm; WeCAN: Weighted cyclic algorithm new; WeSCAN: Weighted stopband cyclic algorithm new

Acknowledgements

The authors want to thank XiaoLi Sun for her great help in Latex editing and the anonymous reviewers for their valuable comments.

Funding

This work is supported by the Key Projects in the National Science & Technology Pillar Program during the Twelfth Five-year Plan Period (No. 2014BAK12B00) and the National Natural Science Foundation of China (No. 61101186 and No. 61101179).

Availability of data and materials

Please contact the author for data (Matlab codes) requests.

Authors' contributions

LT conceived and devised the idea. LT and YZ designed and performed the experiments. YZ and QF analyzed the experiment results. All authors contributed to writing and editing the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 30 March 2017 Accepted: 22 August 2018
Published online: 17 September 2018

References

1. S. W. Golomb, G. Gong, *Signal design for good correlation: for wireless communication, cryptography, and radar*. (Cambridge University Press, Cambridge, 2005)
2. H. He, J. Li, P. Stoica, *Waveform design for active sensing systems: a computational approach*. (Cambridge University Press, Cambridge, 2012)
3. N. Levanon, E. Mozeson, *Radar signals*. (Wiley, New York, 2004)
4. D. Tse, P. Viswanath, *Fundamentals of wireless communication*. (Cambridge University Press, New York, 2005)
5. S. Mertens, Exhaustive search for low-autocorrelation binary sequences. J. Phys. A. Math. General. **29**(18), 473–481 (1996)

6. W. H. Mow, K. L. Du, W. H. Wu, New evolutionary search for long low autocorrelation binary sequences. IEEE Trans. Aerosp. Electron. Syst. **51**(1), 290–303 (2015)
7. C. J. Nunn, G. E. Coxson, Polyphase pulse compression codes with optimal peak and integrated sidelobes. IEEE Trans. Aerosp. Electron. Syst. **45**(2), 775–781 (2009)
8. P. Borwein, R. Ferguson, Polyphase sequences with low autocorrelation. IEEE Trans. Inf. Theory. **51**(4), 1564–1567 (2005)
9. P. Stoica, H. He, J. Li, New algorithms for designing unimodular sequences with good correlation properties. IEEE Trans. Signal Process. **57**(4), 1415–1425 (2009)
10. J. X. Song, P. Babu, D. P. Palomar, Optimization methods for designing sequences with low autocorrelation sidelobes. IEEE Trans. Signal Process. **63**(15), 3998–4009 (2015)
11. F. C. Li, Y. N. Zhao, X. L. Qiao, A waveform design method for suppressing range sidelobes in desired intervals. Signal Process. **96**, 203–211 (2014)
12. J. X. Song, P. Babu, D. Palomar, Sequence design to minimize the weighted integrated and peak sidelobe levels. IEEE Trans. Signal Process. **64**(8), 2051–2064 (2016)
13. H. Wu, Z. Y. Song, H. Q. Fan, Q. Fu, Designing sequence with low sidelobe levels at specified intervals based on psd fitting. Electron. Lett. **51**(1), 99–100 (2015)
14. P. Stoica, H. He, J. Li, On designing sequences with impulse-like periodic correlation. IEEE Signal Process. Lett. **16**(8), 703–706 (2009)
15. A. Aubry, A. De Maio, Y. Huang, M. Piezzo, A. Farina, A new radar waveform design algorithm with improved feasibility for spectral coexistence. IEEE Trans. Aerosp. Electron. Syst. **51**(2), 1029–1038 (2015)
16. W. Rowe, P. Stoica, J. Li, Spectrally constrained waveform design. IEEE Signal Process. Mag. **31**(3), 157–162 (2014)
17. M. J. Lindenfeld, Sparse frequency transmit and receive waveform design. IEEE Trans. Aerosp. Electron. Syst. **40**(3), 851–861 (2004)
18. Z. T. Luo, K. Lu, X. Y. Chen, Z. S. He, Wideband signal design for over-the-horizon radar in cochannel interference. Eurasip J. Adv. Signal Process. **2014**(1), 159 (2014)
19. P. Ge, G. L. Cui, L. J. Kong, J. Y. Yang, Unimodular sequence design under frequency hopping communication compatibility requirements. Eurasip J. Adv. Signal Process. **2016**(1), 137 (2016)
20. H. He, P. Stoica, J. Li, in *2nd International Workshop on Cognitive Information Processing (CIP) 2010*. Waveform design with stopband and correlation constraints for cognitive radar (IEEE, Elba, 2010), pp. 344–349
21. G. Wang, Y. Lu, Designing single/multiple sparse frequency waveforms with sidelobe constraint. IET Radar Sonar Navig. **5**(1), 32–38 (2011)
22. H. Wu, Z. Y. Song, H. Q. Fan, Y. X. Li, Q. Fu, A new algorithm for sparse frequency waveform design with range sidelobes constraint. Chinese J. Electron. **24**(3), 604–610 (2015)
23. B. O'Donnell, J. M. Baden, in *2016 IEEE Radar Conference*. Fast gradient descent for multi-objective waveform design (IEEE, Philadelphia, 2016), pp. 1–5
24. D. Zhao, Y. Wei, Y. Liu, Spectrum optimization via FFT-based conjugate gradient method for unimodular sequence design. Signal Process. **142**, 354–365 (2018)
25. J. L. Liang, H. C. So, C. S. Leung, J. Li, A. Farina, Waveform design with unit modulus and spectral shape constraints via Lagrange programming neural network. IEEE J. Sel. Topics Signal Process. **9**(8), 1377–1386 (2015)
26. X. Feng, Y. C. Song, Z. Q. Zhou, Y. N. Zhao, Designing unimodular waveform with low range sidelobes and stopband for cognitive radar via relaxed alternating projection. Int J Antennas Propag. **2016**, 1–9 (2016)
27. J. L. Liang, H. C. So, J. Li, A. Farina, Unimodular sequence design based on alternating direction method of multipliers. IEEE Trans. Signal Process. **64**(20), 5367–5381 (2016)
28. The Matlab code of the PeCAN. http://www.sal.ufl.edu/book/.