Optimization Methods for Solving the Low Autocorrelation Sidelobes Problem

U. Tan*†, O. Rabaste‡, C. Adnet*, F. Arlery*, J.-P. Ovarlez†‡
*Thales Air Systems, 91470 Limours, France
†SONDRA – CentraleSupélec, 91192 Gif-Sur-Yvette cedex, France
‡ONERA, The French Aerospace Lab, 91123 Palaiseau cedex, France

***Abstract*—In this paper, a discussion is made on the optimization methods that can solve the low autocorrelation sidelobes problem for polyphase sequences. This paper starts with a description and a comparison of two algorithms that are commonly used in the literature: a stochastic method and a deterministic one (a gradient descent). Then, an alternative method based on the Random Walk Metropolis-Hastings algorithm is proposed, that takes the gradient as a search direction. It provides better results than a steepest descent alone. Finally, this autocorrelation question is handled differently, considering a mismatched filter. We will see that a mismatched filter performs impressively well on optimized sequences.**

***Index Terms*—Aperiodic autocorrelation – Optimization algorithms – Gradient descent – Mismatched filter**

I. Introduction

Since 1950, sets of sequences with some properties (autocorrelation, cross-correlation) have been sought. In radar for instance, they are often of interest, because a signal with a "good" autocorrelation property may be distinguished from a time-shifted version of itself, while the cross-correlation property enables a signal to be set apart from another signal.

In terms of radar detection, the autocorrelation function is usually referred to a "matched filter". If the matched filter presents high sidelobes, it may be problematic in the presence of multiple targets. In that case, weak targets can be hidden by a stronger one. Hence, one should find sequences with low autocorrelation sidelobes. This is the objective of this paper.

Searching for those sequences can be seen as an optimization problem, the minimization of a function called "energy". This energy quantifies the autocorrelation sidelobe level of a sequence, like the Merit Factor introduced by Golay [1]. There exists a lot of methods that can provide suboptimal solutions to this optimization problem. Stochastic methods are usually helpful, as no hypothesis on the cost function is required ([2], [3], [4], [5]); their convergence to a global minimum is *almost certain* theoretically. A gradient descent, also known as a steepest descent, is a fast algorithm for finding a local minimum of a function [6]. In this paper, we propose another stochastic method, based on the Metropolis-Hastings algorithm. It generates random variables, but in the descent direction defined by the gradient.

Furthermore, this autocorrelation problem can be handled differently using mismatched filters. A mismatched filter may highly reduce the sidelobe level, at a cost of some SNR (*Signal-to-Noise Ratio*) loss. As it is possible to obtain numerically the optimal mismatched filter [7], we will try to make some connections between the autocorrelation level of a sequence and the gain that can be obtained with the optimal mismatched filter.

This article only deals with polyphase sequences. But it is possible to extend the following procedure for binary sequences (usually referred as "LABS" for *Low Autocorrelation Binary Sequence* in the literature), and for sets of sequences with "good" autocorrelation and cross-correlation ([8], [9]).

This paper is organized as follows. Section II describes and compares two algorithms, a random search and a steepest descent, in order to solve the low autocorrelation sequence problem. Section III suggests another algorithm, based on the Metropolis-Hastings algorithm, that combines the two above-mentioned ones. The last section studies the efficiency of a mismatched filter on an optimized-or-not sequence.

II. Comparison of two optimization methods in the low autocorrelation sidelobes problem

In order to search for sequences with low autocorrelation sidelobes, the procedure employed in this article is inspired from a recent article (Baden *et al.* [6]). It is based on a real function, called "energy", that quantifies the energy present in the autocorrelation sidelobes of a given sequence. Hence, this search is equivalent to a minimization problem.

Let us consider a polyphase sequence $\mathbf{a}$ of length N and of constant modulus, *i.e.* $\mathbf{a} = [a_1, ..., a_N]^T = \left(\mathrm{e}^{j2\pi\alpha_i}\right)_{i\in[\![1,N]\!]}$. Assume that $a_i = 0$ for $i < 1$ and $i > N$.

The discrete aperiodic autocorrelation, denoted by $\mathbf{c} = \mathbf{a} * \mathbf{a}$, is the sequence:

$$c_m = \sum_{i=1}^{N} a_i a_{i+m}^* \quad (-N < m < N)\,, \tag{1}$$

where we note $(.)^*$ the complex conjugate operator. By including some weighting $\mathbf{w}$ to allow shaping of the sidelobes, and an exponent $p \in \mathbb{N}$ to control the peak sidelobe, the autocorrelation sidelobe energy E_a is defined by:

$$E_a(a) = \sum_{m=-N+1}^{N-1} w_m \left(c_m c_m^*\right)^p\,. \tag{2}$$

Finding the minimum of E_a, the cost function throughout this paper, is a difficult task. Indeed, it is a non convex problem

(because of the constant modulus hypothesis) that must be optimized over a large set of variables (of size equal to the length N of the sequence). Suboptimal approaches are usually considered in the literature: stochastic and deterministic methods.

Hence, in this section, we recall quickly two well-known optimization methods, one from each approach: a random search and a gradient descent [6]. We compare then their performances on the optimization problem mentioned earlier. The next section will propose an original alternative that mixes some aspects of the two previous-cited methods.

A. Stochastic Methods

Stochastic methods ([2] to [5]) are characterized by the presence of randomness in the optimization process. The iterative process can be divided into two steps, specific to each method. A generation step creates candidates by mutation or exploration of the domain. A selection step evaluates these candidates through the cost function. One can notice that the mutation operator is quite sneaky, as the linear combination of two sequences with low autocorrelation sidelobes does not necessarily provide another one with better sidelobes... Common stochastic methods for designing polyphase sequence includes genetic algorithms [4], evolutionary strategies [5], tabu search [3], etc.

The principle of a stochastic method is described in the table 1, with the most basic algorithm, the random search.

Algorithm 1 Random Search

```
random initialization a
repeat
    generation of a candidate b
    if E_a(b) < E_a(a) then
        a = b
    end if
until a stop criterion is met
```

Like most methods, stochastic methods have some advantages and some drawbacks. A short list is given below.

Advantages

- Stochastic methods can be used to solve every optimization problem, as they don't need any hypothesis on the cost function (smoothness, convexity...).
- Their global convergence is *almost certain* theoretically...

Drawbacks

- ... but it cannot be established in practice.
- The longer the sequence is, the slower the algorithms are.

B. Gradient Descent

The gradient descent is a fast algorithm for finding a local minimum of a function. It starts with an initial guess of the solution, and as many times as needed, moves it towards the opposite direction of the gradient at that point. Table 2 gives its procedure. Pros and cons of a gradient descent are the following:

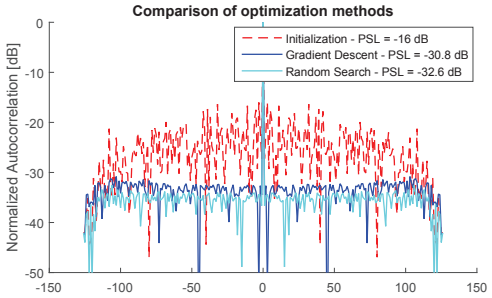


Figure 1. Optimization of the autocorrelation of a polyphase sequence of length $N = 127$

Advantages

- It is pretty fast.
- The more information we have on the cost function, the more accurate the gradient descent can be (*cf.* Quasi-Newton methods for instance)

Drawbacks

- It needs some hypothesis on the cost function (existence of partial derivatives at least)
- The algorithm is only guaranteed to converge to a local minimum.

Algorithm 2 Gradient Descent

```
parameter initialization: step γ
random initialization: sequence a
repeat
    while f(a − γ∇_E(a)) > f(a) do
        reduce γ
    end while
    a = a − γ∇_E(a)
until a stop criterion is met
```

Partial derivatives of the energy function E_a, with respect to the phase angle of $\mathbf{a}$, denoted α exist; their calculation is detailed in [6], and the result (3) is given below, with $\boldsymbol{\eta}$ a sequence that depends on $\mathbf{c}$. So it makes sense to use a gradient descent.

$$\frac{\partial E_a}{\partial \alpha_j} = -2p\,\Im\left[a_j((\boldsymbol{\eta}\circ\mathbf{c}^*)*\mathbf{a})_j\right] - 2p\,\Im\left[a_j((\boldsymbol{\eta}\circ\mathbf{c})*\mathbf{a}^r)_{N+1-j}\right], \quad (3)$$

There are other deterministic algorithms, like the CAN algorithm, introduced by Stoica *et al.*, and described in [10]. This algorithm expresses the problem in the frequency domain, and solves it with a minimization-majoration technique. However, it suffers from a high computational complexity, due to the use of singular value decompositions (SVD).

C. First results: Comparison of the two methods

In this part, a random search and a steepest descent are applied in order to find sequences with low autocorrelation sidelobes. Figure 1 shows the improvements we can get from a random sequence of length $N = 127$.

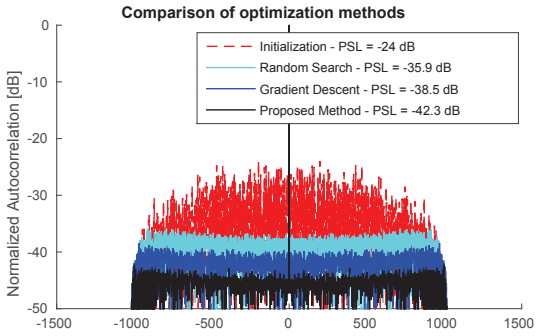
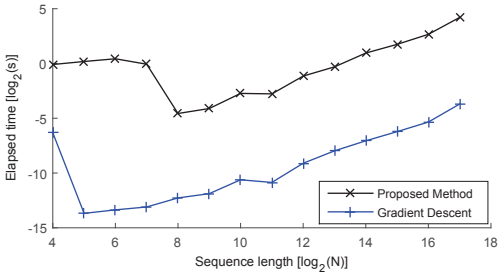


Figure 2. Optimization of the autocorrelation of a polyphase sequence, of length $N = 1024$



Sequence length N	2^4	2^5	2^6	2^7	2^8	2^9
PSL-objective (dB)	-20	-27	-29	-33	-35	-37
Sequence length N	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	
PSL-objective (dB)	-39	-41	-42	-43	-44	

Figure 3. Comparison of the convergence rate, according to the sequence length and its associated PSL-objective (logarithmic scale)

In this example, one can observe that both methods perform well: around 15 dB are gained from a random initialization, with a little more for the random search. That gives us a PSL (*Peak Sidelobe Level*) at around 36 dB. Notice that this result is consistent, compared to other stochastic methods [11] described in the literature. Although the computational time is not really a reliable indicator (as it depends on a lot of parameters), the difference between the two methods is quite impressive: the gradient descent has converged in a few seconds, while it has been about minutes for the random search.

The procedure is repeated in figure 2, with a sequence of length $N = 1024$. This time, the trend is reversed in favour of the gradient descent: a difference of 3.5 dB can be observed between the two methods. As a random search explores "naturally" the whole domain, the increasing number of degrees of freedom implies a lengthier computation time. However, for practical reasons, the iteration process has to be stopped after a finite reasonable amount of time, thus implying a degradation of performance for high length sequences.

III. An alternative method based on the gradient

A steepest descent is a deterministic algorithm. Since this algorithm will converge to the first local optimum on the descent path, its performance strongly depends on the initialization. Hence, it may be interesting to study the performance of a multi-start gradient, or to use a gradient descent during the generation process of a stochastic method, as suggested in [11]. On the other side, stochastic methods are not trapped by local minima, and may thus explore the whole search domain, at the cost of extra computational time.

In this section, we propose an alternative method for minimizing the energy E_a. It combines two aspects of the methods described earlier: the exploration of the heuristics, and the fast convergence of a steepest descent. Our method is based on the Random Walk Metropolis-Hastings algorithm, originally proposed by Metropolis *et al.* in 1953. It allows the evaluation of the cost function in different areas of the search domain. The generation process in the solution we propose is not done on the neighbourhood, but on the line defined by the gradient. The step parameter (just like the one in a descent) is determined pseudo-randomly, avoiding to fall into a local minimum. Table 3 describes this algorithm, where $\mathrm{rand}(x, y)$ denotes a random number between x and y. Its main feature is on the choice of the generation step. The proposed iterative process faces three possible cases: if the cost function at the new proposed candidate is improved, this candidate is accepted, and we randomly increase the descent step in order to possibly go further in the descent direction (case (a)). If the new candidate does not provide an improvement, then it may still be accepted with some probability $P(\delta)$ (b). Finally, if it is not accepted (c), we propose to decrease the step in order to increase the probability to accept a new candidate at the next iteration.

Algorithm 3 Proposed Method

input: selection law P, initial step γ_0
random sequence initialization $\mathbf{a}$
step initialization, $\gamma = \gamma_0$
repeat
  if $\delta := E(a - \gamma \nabla_E(a)) - E(a) < 0$ **then**
    $a = a - \gamma \nabla_E(a)$
    $\gamma = \gamma_0 + \mathrm{rand}(0, \gamma_0)$ ▷ (a)
  else if $P(\delta) < \mathrm{rand}(0, 1)$ **then**
    $a = a - \gamma \nabla_E(a)$ ▷ (b)
  else
    $\gamma = \gamma_0 - \mathrm{rand}(0, \gamma_0)$ ▷ (c)
  end if
until a stop criterion is met

Figure 2 provides an overview of the efficiency of the proposed algorithm. A gain of 3.5 dB – in terms of PSL – is obtained, compared to a standard gradient descent. Figure 3 compares the convergence rate of the two methods by setting a PSL-objective to reach. As the gradient descent can be computed in $O(N\log(N))$ operations [6], one can see that it is also the case for the proposed algorithm, up to a multiplicative factor.

Thus, the proposed method offers better results, at the cost of a reasonable computational cost. So it is appropriate to use it for designing long polyphase sequences.

IV. Mismatched filtering

From here, this article has only been focused on the autocorrelation function of a sequence, that is to say on the matched filter. Such a procedure is thus limited to exactly N degrees of freedom (the length of the sequence) to obtain the best possible PSL. However it is possible to replace the matched filter by a different filter, that will not be optimal in terms of Signal-to-Noise Ratio (SNR) but may provide a better PSL. Using such a mismatched filter provides additional degrees of freedom in order to optimize the PSL, thus allowing to obtain better PSL performance at the cost of a controllable loss in processing gain. The number of additional degrees of freedom will be equal to the length of the mismatched filter, that can be chosen to be longer than the considered sequence.

Let $\mathbf{f}$ be a mismatched filter of length N_f. The output of the mismatched filter is the sequence $\mathbf{d} = \mathbf{a} * \mathbf{f}$, where:

$$d_m = \sum_{i=1}^{N_f} a_i f_{i+m}^* \quad (-N_f < m < N_f)\,. \tag{4}$$

Apart from the Peak-to-Sidelobe Ratio (PSLR), the efficiency of a mismatched filter can be measured by the Signal-to-Noise Ratio (SNR) and the loss in processing gain (LPG). These quantities are defined by:

$$\mathrm{SNR} = 10\log_{10}(|\mathbf{a}^H\mathbf{f}|)\,, \tag{5}$$

$$\mathrm{LPG} = 10\log_{10}\left(\frac{(|\mathbf{a}^H\mathbf{f}|)^2}{(\mathbf{a}^H\mathbf{a})(\mathbf{f}^H\mathbf{f})}\right) \tag{6}$$

A recent article [7] explains how to generate the optimal mismatched filter. The procedure is based on the resolution of an optimization problem, the minimization of the PSLR. The latter can be cast as a Quadratically Constrained Quadratic Program (QCQP). This proposed QCQP is shown to be convex, so it is possible – for instance by using an interior point method – to find the global minimum, and thus *i.e.* the optimal mismatched filter. In this procedure, the LPG can also be constrained to a given value.

In this section, we suggest to inspect the efficiency of the optimal mismatched filter, according to the initialization. So, we consider two sequences of length $N = 127$: a random one, and the same sequence, optimized with the proposed method in section III. Their optimal mismatched filter, of length $N_f = 3N - 1$ is computed and applied; results are shown figure 4, on the top and the bottom respectively.

We first notice that, interestingly, the matched filter of the optimized sequence presents a better PSLR than the mismatched filter of the initial random sequence, with a difference of approximately 6 dB. Of course, the PSLR provided by the mismatched filter for the optimal sequence is much better. Note also that surprisingly the gain provided by the mismatched filter compared to the matched filter is better for the optimized sequence than for the initial sequence.

Hence, in order to obtain a sequence with low sidelobes in autocorrelation, it seems interesting to start with an optimization of the sequence itself, with methods like the one suggested in the previous section, and then employ an optimal mismatched filter on it. Following this procedure, the figure 4 shows that it is possible, for a sequence of length $N = 127$, to obtain sidelobes with a really flat profile at -52 dB, and with a loss of only 0.3 dB. Such a level is very interesting, especially for sequences with such length!

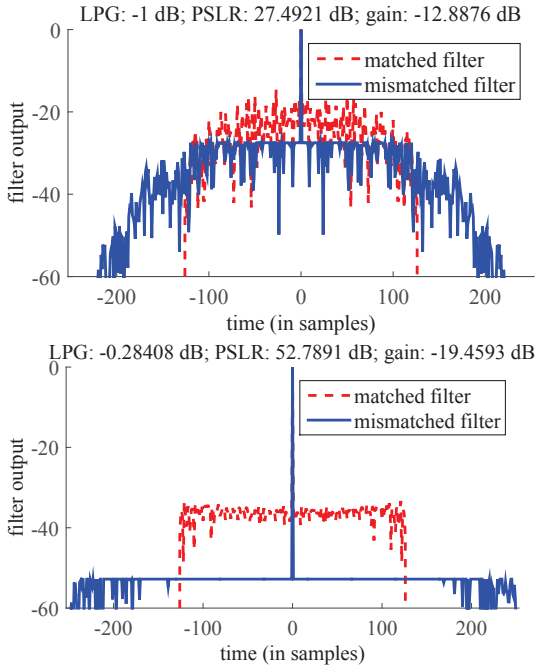


Figure 4. Matched filter and mismatched filter for a initial sequence of length $N = 127$. On top: non-optimized sequence. On the bottom: optimized sequence.

V. Conclusion

In this paper, a method for finding a sequence with low sidelobes in autocorrelation has been presented. It explores a lot of candidates, because of the random feature of the Metropolis-Hastings algorithm, but *good* ones, as they are chosen in the direction of the gradient. Obtained results are better than what we can obtain with a steepest descent, at the cost of some additional but reasonable computational time. This method can be extended to obtain sets of sequences with low sidelobes in both autocorrelation and cross-correlation.

By definition, a mismatched filter provides a gain on the Peak to Sidelobe Ratio, at a cost of a loss in processing gain. We have observed that this gain is higher on sequences that are optimized than those that are not. The procedure that consists in an optimization of a sequence, and an application of the optimal mismatched filter on it seems promising.

Ongoing work will be focused on:

- Studying the connection between the length of the sequence, its optimal mismatched filter and the PSL (*Peak Sidelobe Level*)
- Jointly optimizing the code and its mismatched filter
- Applying these methods on the coherent MIMO ambiguity function case

ACKNOWLEDGMENT

This work is partly supported by a DGA-MRIS scholarship.

REFERENCES

[1] M. Golay, "Sieves for low autocorrelation binary sequences," *IEEE Transactions on Information Theory*, vol. 23, no. 1, pp. 43–51, Jan 1977.

[2] M. Malekzadeh *et. al*, *Intelligent Computing Technology: 8th International Conference, ICIC 2012, Huangshan, China, July 25-29, 2012. Proceedings*. Springer Berlin Heidelberg, 2012, ch. Optimization of Orthogonal Poly Phase Coding Waveform Based on Bees Algorithm and Artificial Bee Colony for MIMO Radar.

[3] D. Liu, Y. Liu, and H. Cai, "Orthogonal polyphase code sets design for mimo radar using tabu search," in *Intelligent Control, Automatic Detection and High-End Equipment (ICADE), 2012 IEEE International Conference on*, July 2012, pp. 123–127.

[4] H. H. Dam, H. J. Zepernick, and H. Luders, "Polyphase sequence design using a genetic algorithm," in *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, vol. 3, May 2004, pp. 1471–1474 Vol.3.

[5] H. Deng, "Polyphase code design for orthogonal netted radar systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 11, pp. 3126–3135, Nov 2004.

[6] J. M. Baden, M. S. Davis, and L. Schmieder, "Efficient energy gradient calculations for binary and polyphase sequences," in *Radar Conference (RadarCon), 2015 IEEE*, May 2015, pp. 0304–0309.

[7] O. Rabaste and L. Savy, "Mismatched filter optimization for radar applications using quadratically constrained quadratic programs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 4, pp. 3107–3122, 2015.

[8] F. Arlery, R. Kassab, U. Tan, and F. Lehmann, "Efficient gradient method for locally optimizing the periodic/aperiodic ambiguity function," in *2016 IEEE Radar Conference, Philadelphia, USA, "in press"*, 2016.

[9] U. Tan, C. Adnet, O. Rabaste, F. Arlery, J.-P. Ovarlez, and J.-P. Guyvarch, "Phase code optimization for coherent mimo radar via a gradient descent," in *2016 IEEE Radar Conference, Philadelphia, USA, "in press"*, 2016.

[10] J. Song, P. Babu, and D. P. Palomar, "Optimization methods for designing sequences with low autocorrelation sidelobes," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3998–4009, Aug 2015.

[11] A. Milewski and D. Szczegielniak, "A hybrid optimization method for designing polyphase sequences," *Hydroacoustics*, vol. 9, pp. 125–132, 2006.