Journal of Scientific & Industrial Research
Vol. 82, November 2023, pp. 1177-1183
DOI: 10.56042/jsir.v82i11.3268

An Insight into the Performance of Chaotic Sequences using Cascaded Mismatched Filters with Adaptive Performance of Radar Sequences using Adaptive Mismatched Filter

K Renu[1]* & P Rajesh Kumar[2]

[1]GST, GITAM (Deemed to be University), Visakhapatnam, A. P. 530 045, Andhra Pradesh, India

[2]A. U. College of Engineering (A), Andhra University, Visakhapatnam 530 003, Andhra Pradesh, India

Received 03 September 2021; revised 19 June 2023; accepted 07 October 2023

In modern Radar applications, optimal sequences have been used in many areas, such as communication systems, radar, and sonar, because of their minimal peak sidelobe level, which causes an increase in the signal-to-noise ratio with a good range resolution at the output. The literature survey shows various pulse compression techniques that are widely used to achieve superior range resolution and range detection performance. Several studies have been conducted on chaotic communication involving chaotic maps in recent years, producing promising results. These maps are used to generate different phase-coded sequences. The properties of the chaotic map sequences are almost random. The performance of these sequences has been studied with various optimization techniques in literature by employing a matched filter and a mismatched filter and is measured in terms of peak sidelobe ratio. But the performance has not improved significantly. This paper focused on improving performance using a new hybrid technique to design mismatched filters. This improvement is achieved by designing the coefficients of the mismatched filters using a combination of metaheuristic methods and an evolutionary algorithm for specializing in intensification and diversification. A significant improvement in the peak sidelobe ratio and range resolution is obtained when the mismatched filter is combined with adaptive filters at the output.

Keywords: Chaotic sequences, GHO, Peak sidelobe ratio, Pulse compression, Range resolution

Introduction

The practical problems associated with increasing the radar range with the desired range resolution and accuracy are addressed by pulse compression, which deals with the code that modulates the carrier during transmission. In radar, when the received echo signal from the target passes through a Matched Filter (MF), it generates the main lobe and many sidelobes at the output. The sidelobes may cause false alarms, or they may mask the peaks of the weak target echo signals. Therefore, having a minimum value of sidelobe peaks is better, resulting in a reduced peak sidelobe ratio. The minimum value of PSR indicates suppression of unwanted clutter and range sidelobes. Sidelobe reduction is where lots of research has been reported with considerable interest.

The primary purpose of matched filter (MF) and Mismatched Filter (MMF) is to suppress the sidelobes. Ackroyd and Ghani proposed a least-squares inverse filter approach to design a mismatched filter.[1] Baden and Cohen introduced the iteratively reweighted least squares method to improve the PSR value.[2] From the literature, it has been studied that the MMF coefficients can be optimized for obtaining good results with various input sequences. Based on this idea, Nunn[3] proposed an optimization concept for the signals or sequences with the least value of peak sidelobe in its ACF. Levanon[4] extended the optimization concept by recommending an optimized filter to minimize PSR. The coefficients of the optimized filter are designed so that it results in a minimum cost function when combined with input. This research aims to reduce the cost function in terms of PSR using a hybrid optimization technique. Here the input sequences are generated using a chaotic map in a chaos system that is highly influenced by initial conditions. The chaos theory has been discussed using different algorithms such as firefly algorithm[5], differential evolution algorithm[6], and genetic algorithm[7] etc. In this paper, the performance of this chaotic sequence is evaluated

*Author for Correspondence
E-mail: rkarra@gitam.edu

by cross correlating this sequence with the optimized coefficients of the filter.

Methodology

The research work presented in literature[8] compared the objective function of ternary sequences that is obtained from chaotic maps. These sequences are processed through the matched filter to get the auto-correlation pattern. The PSR achieved with MF was improved with the help of MMF. This paper optimizes MMF coefficients using the Grasshopper Optimization (GHO) technique and cascaded MMF designed with DE and GHO methods. The performance is examined by cross-correlating the chaotic phase-coded sequences with the filter coefficients.

The Differential Evolution algorithm is a direct search stochastic optimization technique suggested by Storn and Price in 1995.[(9,10)] The popularity of this method is due to its easy implementation of solving problems in different fields like synchronization and control of chaotic systems[11], image enhancement problems and parameter identification.[12] The optimization strategy followed for this algorithm is DE/rand/1/bin meaning thereby, random target vectors are chosen for the mutation. Here the bin acronym stands for the binomial decision rule, which is used for the recombination process.

Grasshopper Optimization (GHO) Algorithm

Over the last three decades, meta-heuristics optimization algorithms have attained exciting research areas. The GHO algorithm was suggested by Saremi *et al*.[13] This algorithm has attracted a lot of research interest in solving optimization problems. Much of the research work has been done and published using GHO. Some of the recent swarm intelligence algorithms depend on grasshopper's natural foraging and swarming behaviour. These insects are hazardous pests that damage crop production and agriculture as they move slowly in their infancy but have a wide range of activities in adulthood.[14] This characteristic of the swarm makes two modes in their search process: exploration and target search. Balancing intensification and diversification distinguish this method from other optimization techniques. One of the essential characteristics of the grasshopper is that it can effectively solve practical problems in the search space. GHO has many advantages over other evolutionary algorithms. It increases the average survival rate of grasshoppers and improves the random initial population. The abrupt changes at the initial stage of the optimization help the search on a global scale. These grasshoppers are then moving locally at the final step of optimization. The steps in GHO are presented in Fig. 1.

The swarming behaviour of grasshoppers is mathematically expressed in Eq. (1).

$$P_i = S_i + G_i + A_i \quad \dots (1)$$

where, P_i is the position of i[th] grasshopper, S_i the social interaction between grasshoppers and G_i indicate gravitational force on the i[th] grasshopper. A_i represents the wind direction. The expansion of Eq. (1) can be written in Eq. (2) by substituting S_i, G_i and A_i. The factor A_i is always tends toward the best solution $\widehat{T_d}$.

$$X_i^d = c \sum_{\substack{j=1 \\ j \neq i}}^{N} c \frac{ub_d - lb_d}{2} s\left(\left|x_j^d - x_i^d\right|\right) \frac{x_j - x_i}{d_{ij}} + \widehat{T_d} \quad \dots (2)$$

where, N is the number of grasshoppers. X_i^d is the current location of i[th] grasshoppers in D-dimensional solution space. ub_d and lb_d are the upper and lower limits of solution space. The function s(r) in Eq. (3) is a force function that describes the social interactions and can be defined as

$$s(r) = f e^{(-r/l)} - e^{-r} \quad \dots (3)$$

The factor c in Eq. (4) is used to decrease the declination of coefficients which is defined as

$$c = c_{max} - \frac{l(c_{max} - c_{min}}{L} \quad \dots (4)$$

where, c_{max} and c_{min} are the maximum and minimum values of c, 't' is the present iteration, and L represents the total iteration count. $\left|x_j^d - x_i^d\right|$ is the distance between i[th] and j[th] grasshopper. $\frac{x_j - x_i}{d_{ij}}$ represents its unit vector, f is the intensity of

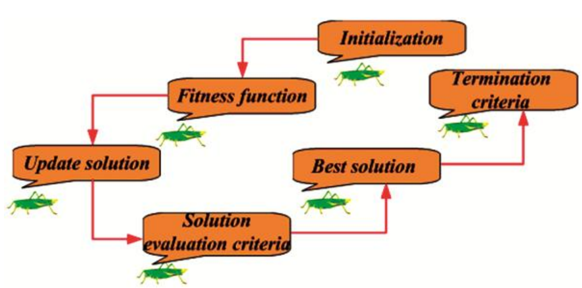

Fig. 1 — Steps in grasshopper optimization algorithms

attraction, and l is the attraction length in the s(r) function. $\widehat{T_d}$ is the value of the Dth dimension in the target, i.e., the best solution. Hence, from Eq. (2), it is evident that the next location of the grasshoppers is updated based on its current location, target position (global best position), and the location of other grasshoppers in the swarm. The first part of the equation indicates the location of the grasshopper relative to others in the swarm and the other part replicates the movement for food sources.

GHO provides the most promising target in the search space. The grasshoppers continuously migrate towards the promising target during the iterations to obtain the actual global optimum in the search space. Like other evolutionary algorithms, this method also considers the fitness function to control the search process and to get the optimal location in the search space to achieve the required objective function. The vector corresponding to this fitness value is considered a global optimal position vector transmitted to other grasshoppers in the swarm. Accordingly, the other grasshoppers adjust their positions until they achieve the target food. The accuracy of the target is improved with the approximation of global optima in proportion to iterations. It is to be noted that the exploitation characteristics of GHO satisfy the single test function, whereas the exploratory nature encourages the multimodal test function to fulfill.

However, the challenging problems involving complex test functions are easily solved with the proper balance between GHO's exploitation and exploration property. Therefore, it may conclude that grasshopper optimization is significantly superior to other existing optimization algorithms. The flow chart in Fig. 2 clearly explains the procedure to follow in GHO using the pseudo-code.

Pseudo code of Grasshopper Optimization Algorithm:

Initialize the population of grasshoppers Pi (i=1,2,….,n)
Initialize c_{max}, c_{min} and maximum value of iteration L
Compute the fitness f(Pi) of each grasshopper Pi
T= best solution
While (t< L) do
Update c using equation (4)
For i=1 to N (considering N number of grasshoppers in the population)
Do
Normalizing the distance between grasshoppers.
Update the current position of the grasshopper using Eq. (2)
Bring the current grasshopper back if it goes outside of the boundaries.
end for
Update T if a better solution found
t = t+1
end while
Return the best solution T

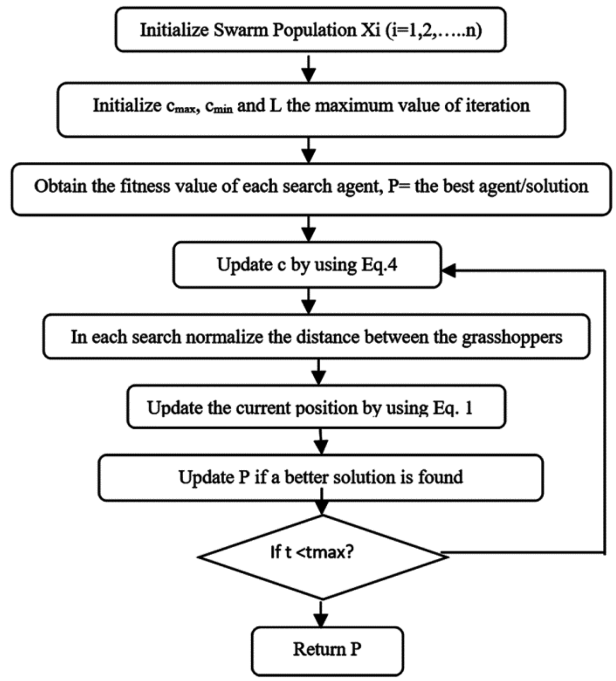

Fig. 2 — Flow chart diagram of GHO method

The main aim of the optimization is to compute the minimum value of the fitness or cost function of the given problem for the proper estimation of computation time, convergence rate, and accuracy. The block diagram in Fig. 3 shows the methodology adopted in the present work. This section describes the use of cascaded MMF, where two MMFs are connected in the cascaded form.[15,16] The coefficients of the first MMF are designed using differential evolution, whereas the second MMF is employed with the Grasshopper Optimization method.

The output of the cascaded filter is processed by using adaptive techniques such as the Least Mean Square (LMS) and Binary Step Size Least Mean Square (BSSLMS) algorithm, as suggested earlier.[17] These adaptive algorithms are used to update their coefficients to improve the performance of the objective function, which is considered PSR. If the filter coefficients of the MMF are defined as

$$H = \{h_0, h_1, \dots\dots h_{M-1}\} \quad \dots (5)$$

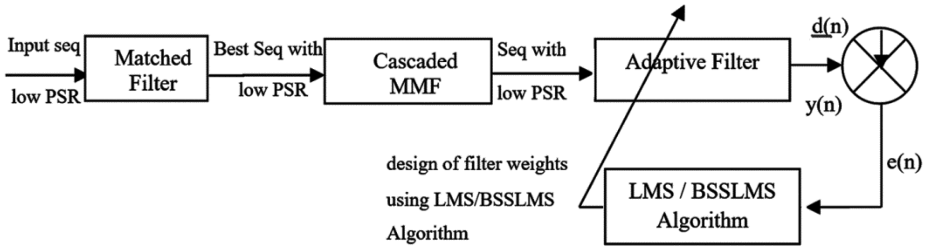

Fig. 3 — Block diagram representation of cascaded MMF followed by Adaptive filter

Table 1 — PSR analysis of Binary Logistic Sequence with MF, MMF and Adaptive Filters

Length of the Seq	PSR of MF (dB)	ASP of MF	PSR of Random POP (dB)	PSR of MMF(GHO) (dB)	ASP of MMF (GHO)	PSR of MMF (DE) (dB)	ASP of MMF (DE)	PSR of MMF (DE-GHO) (dB)	ASP of MMF (DE-GHO)	PSR of Adaptive LMS filter (dB)	ASP with LMS Filter	PSR of Adaptive BSSLMS filter (dB)	ASP with BSSLMS Filter ASP
20	−16.4782	0.15	−15.4934	−18.5910	0.1176	−20.2457	0.0972	−23.4677	0.0671	−32.4575	0.0238	−37.4665	0.0134
25	−18.4164	0.12	−15.6402	−20.4175	0.0953	−22.2164	0.0775	−26.5881	0.0468	−41.0119	0.0089	−41.1336	0.0088
30	−17.5012	0.1333	−17.2552	−20.9063	0.0901	−22.7878	0.0725	−25.8908	0.0508	−31.6292	0.0262	−31.3431	0.0271
35	−16.9020	0.1429	−16.8850	−19.6061	0.1046	−21.9802	0.0796	−25.6863	0.0520	−29.0737	0.0352	−29.8559	0.0322
40	−18.0618	0.1250	−17.0003	−20.6343	0.0930	−21.0055	0.0891	−23.5383	0.0665	−26.9241	0.0451	−26.9694	0.0448
45	−17.5012	0.1333	−159606	−18.5273	0.1185	−19.9384	0.1007	−21.5002	0.0841	−24.4970	0.0596	−23.2980	0.0684
50	−18.4164	0.1200	−17.5446	−19.7870	0.1025	−21.1079	0.0884	−23.6402	0.0658	−25.3826	0.0538	−25.6687	0.0521
60	−17.5012	0.1333	−16.0468	−18.4654	0.1193	−20.2564	0.0971	−22.2313	0.0773	−25.4373	0.0535	−26.0767	0.0497
70	−17.8171	0.1286	−17.4253	−19.7577	0.1028	−20.0439	0.0995	−22.2523	0.0772	−26.0849	0.0496	−26.1349	0.0493
80	−18.9769	0.1125	−17.3692	−20.3885	0.0956	−20.1087	0.0988	−22.1209	0.0783	−24.5199	0.0594	−25.0460	0.0559
90	−18.2570	0.1222	−17.9310	−19.5412	0.1054	−20.3289	0.0963	−23.0182	0.0706	−24.8114	0.0575	−25.1982	0.0550
100	−19.1721	0.1100	−17.7017	−19.8464	0.1018	−20.4521	0.0949	−22.8739	0.0718	−25.6413	0.0522	−26.0161	0.0500
150	−19.4394	0.1067	−18.4894	−20.5291	0.0941	−20.4272	0.0952	−22.6244	0.0739	−26.0373	0.0499	−26.0098	0.0501
200	−20	0.1000	−19.4613	−21.5451	0.0837	−21.2824	0.0863	−23.5918	0.0661	−26.5863	0.0468	−27.0246	0.0445
250	−20	0.1000	−19.4770	−21.1366	0.0877	−21.0270	0.0888	−22.8539	0.0720	−25.6982	0.0519	−26.6937	0.0463
300	−20.5993	0.0933	−19.9365	−21.6519	0.0827	−21.7719	0.0815	−23.7541	0.0649	−26.3410	0.0482	−27.7150	0.0411

The output of the MMF is given in Eq. (6)

$$G_k = \sum_{i=0}^{M-1} h_i \, s_{i-k} \quad for - (N-1) \leq k \leq (M-1) \qquad \dots (6)$$

where, M is the filter length and must have M ≥ N, s_i = 0 for i< 0 and i> N-1.

The Peak sidelobe ratio of the MMF is measured using Eq. (7).

$$PSR = 20 \log \frac{max\,|G_{k \neq M-N/2}}{G_{k=M-N/2}} \qquad \dots (7)$$

Each variable vector/solution in the randomly generated population is selected as the MMF impulse response coefficients of length M. The crossover rate and the mutation factor are assumed as 0.5 and 0.2, respectively. The population size and the number of iterations is initialized. As per the literature, the matched filter coefficients are the input sequence. But the MMF coefficients are in the form of [Z S Z] which have a slight deviation from the zero-padded matched filter output response. Here Z refers to a zero-padded sequence having length (M-N)/2. The filter length is thrice of the input sequence length.

Results & Discussion

The research work in this paper shows the improvement in the performance of chaotic phase-coded sequences using the proposed hybrid technique. The random population is considered as the MMF coefficients that are designed with different methods like differential evolution, Grass-Hopper optimization, and hybrid DE-GHO. The iteration count and population size are chosen as 200 each. The objective function PSR is measured at each step in the block diagram of Fig. 3 for the generated chaotic sequences. The comparison of binary logistic and improved logistic sequences in terms of PSR is given in Tables 1 & 2. The output of the cascaded MMF using DE-GHO for the binary and ternary logistic code of length 20 is shown in Fig. 4.

From Table 1, It is found that the 20-length binary logistic sequence has a PSR of −16.4782 dB with MF, −18.5910 dB with GHO-MMF, −20.2457 dB with DE-MMF, −23.4677 dB with cascaded MMF using DE with GHO. This value is then improved to −32.4575 dB by connecting the adaptive filter using LMS and −37.4665 dB using the BSSLMS algorithm.

Table 2 — PSR analysis of Binary Improved Logistic Sequence with MF, MMF and Adaptive Filters

Length of the Seq	PSR of MF (dB)	ASP of MF	PSR of Random POP (dB)	PSR of MMF(GHO) (dB)	ASP of MMF (GHO)	PSR of MMF (DE) (dB)	ASP of MMF (DE)	PSR of MMF (DE-GHO) (dB)	ASP of MMF (DE-GHO)	PSR of Adaptive LMS filter (dB)	ASP with LMS Filter	PSR of Adaptive BSSLMS filter (dB)	ASP with BSSLMS Filter ASP
20	−16.4782	0.1500	−15.1115	−19.2591	0.1089	−21.5558	0.0836	−24.3885	0.0603	−41.0718	0.0088	−39.6272	0.0104
25	−18.4164	0.1200	−15.6402	−20.4175	0.0953	−22.2154	0.0775	−26.5881	0.0468	−41.0119	0.0089	−41.1336	0.0088
30	−17.5012	0.1333	−17.6270	−22.4372	0.0755	−23.1621	0.0695	−27.7922	0.0408	−463232	0.0048	−38.3342	0.0121
35	−16.9020	0.1429	−16.3843	−19.0920	0.1110	−20.2859	0.0968	−22.8603	0.0719	−26.4631	0.0475	−27.5640	0.0419
40	−18.0618	0.1250	−17.0858	−20.1250	0.0986	−21.1656	0.0874	−24.0280	0.0629	−28.5388	0.0374	−27.6980	0.0412
45	−17.5012	0.1333	−16.7572	−19.3392	0.1079	−20.6555	0.0927	−22.6766	0.0735	−25.8685	0.0509	−25.7506	0.0516
50	−18.4164	0.1200	−17.2077	−20.2353	0.0973	−20.4115	0.0954	−23.6775	0.0655	−27.1717	0.0438	−27.7364	0.0410
60	−18.6611	0.1167	−16.8587	−19.1339	0.1105	−19.8600	0.1016	−22.6008	0.0741	−25.1611	0.0552	−25.5005	0.0531
70	−18.8402	0.1143	−17.6831	−20.1162	0.0987	−20.3498	0.0961	−22.2808	0.0778	−25.3858	0.0538	−26.0485	0.0498
80	−18.0618	0.1250	−16.4554	−18.7405	0.1156	−19.5989	0.1047	−21.5798	0.0834	−24.9394	0.0566	−24.3397	0.0607
90	−19.0849	0.1111	−17.5187	−19.5816	0.1049	−19.9161	0.1010	−22.0857	0.0787	−24.0032	0.0631	−24.9502	0.0566
100	−18.4164	0.1200	−17.7904	−19.8282	0.1020	−20.2950	0.0967	−22.7015	0.0733	−25.9002	0.0507	−26.6932	0.0463
150	−19.4394	0.1067	−18.8203	−20.7535	0.0917	−20.6359	0.0929	−22.8279	0.0722	−25.9365	0.0505	−26.3461	0.0482
200	−19.5762	0.1050	−19.2477	−20.9546	0.0896	−21.1712	0.0874	−23.2653	0.0687	−25.6201	0.0524	−25.992	0.0505
250	−20.3546	0.0960	−19.8895	−22.2633	0.0771	−21.7468	0.0818	−23.7132	0.0652	−26.4359	0.0477	−26.8180	0.0456
300	−20.9151	0.0900	−19.9763	−21.8871	0.0805	−21.6498	0.0827	−23.4159	0.0675	−26.6098	0.0467	−27.7443	0.0410

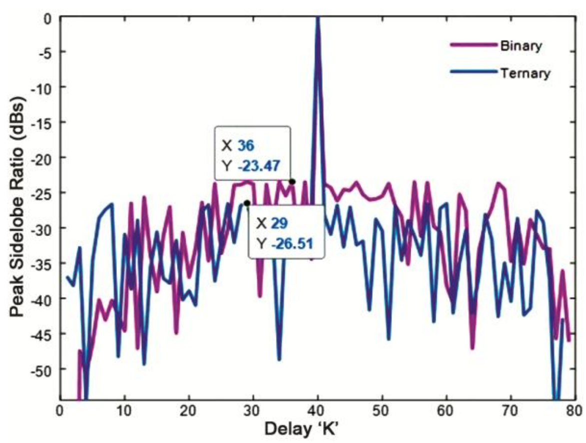

Fig. 4 — Cascaded MMF output of binary and ternary code

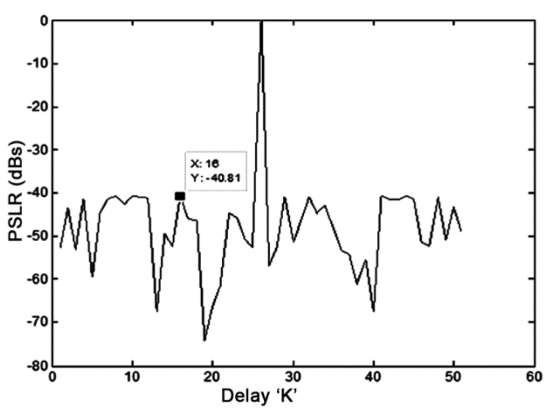

Fig. 5 — Cascaded MMF output of Barker code of length 13

Similarly, Table 2 shows the performance analysis of improved logistic sequences of different lengths. The cross-correlated output of MMF for 13 length Barker sequence is shown in Fig. 5 in which it is observed that the output is not symmetric.

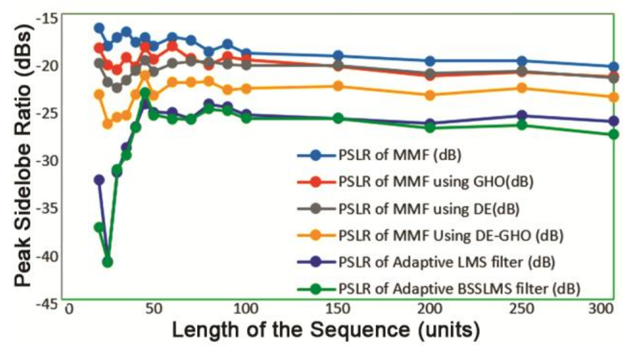

Fig. 6 — Performance comparison of binary logistic code for different lengths

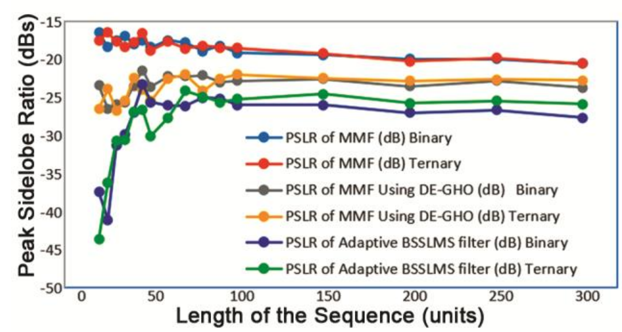

Fig. 7 — Performance comparison between binary and ternary codes

The graphical representation in Fig. 6 shows the comparison results obtained for the binary logistic sequence as per the block diagram with different optimization techniques. In almost all the sequence lengths, the value of PSR is more using adaptive filters with BSSLMS than compared to LMS. The deviation of the performance is represented in Fig. 7

Table 3 — PSR analysis of Ternary Logistic Sequence with MF, MMF and Adaptive Filters

Length of the Seq	PSR of MF (dB)	ASP of MF	PSR of Random POP (dB)	PSR of MMF(GHO) (dB)	ASP of MMF (GHO)	PSR of MMF (DE) (dB)	ASP of MMF (DE)	PSR of MMF (DE-GHO) (dB)	ASP of MMF (DE-GHO)	PSR of Adaptive LMS filter (dB)	ASP with LMS Filter	PSR of Adaptive BSSLMS filter (dB)	ASP with BSSLMS Filter ASP
20	−17.5012	0.1333	−15.0248	−19.4624	0.1064	−23.3002	0.0684	−26.5134	0.0472	−49.5110	0.0033	−43.6092	0.0066
25	−16.4782	0.1500	−15.2342	−18.4325	0.1198	−20.8654	0.0905	−23.9174	0.0637	−28.7052	0.0367	−36.2132	0.0155
30	−17.6921	0.1304	−15.7040	−20.0109	0.0999	−22.0000	0.0794	−26.7712	0.0459	−30.6975	0.0292	−30.7607	0.0290
35	−18.4164	0.1200	−15.8534	−20.1260	0.0986	−21.6917	0.0823	−25.4675	0.0533	−32.4800	0.0238	−30.6295	0.0294
40	−17.786	0.129	−15.4349	−18.4252	0.1199	−19.7365	0.1031	−22.4725	0.0752	−25.8782	0.0508	−26.9057	0.0452
45	−16.6502	0.1471	−15.9752	−19.1704	0.1100	−20.3659	0.0959	−23.9845	0.0632	−27.8874	0.0403	−26.6740	0.0464
50	−18.8402	0.1143	−16.8751	−20.7895	0.0913	−21.5706	0.0835	−25.4806	0.0532	−29.3431	0.0341	−30.0807	0.0313
60	−17.6921	0.1304	−16.7378	−20.7520	0.0917	−20.2218	0.0975	−22.5436	0.0746	−27.5124	0.0421	−27.6764	0.0413
70	−18.5884	0.1176	−16.3861	−19.2927	0.1085	−19.4418	0.1066	−22.0284	0.0792	−23.7586	0.0649	−24.1882	0.0617
80	−18.2155	0.1228	−16.3143	−18.9969	0.1122	−19.5769	0.1050	−24.1442	0.0781	−24.5848	0.0590	−24.9342	0.0567
90	−18.4597	0.1194	−17.7184	−19.5642	0.1051	−19.7995	0.1023	−22.5531	0.0745	−25.5702	0.0527	−25.6803	0.0520
100	−18.5314	0.1184	−16.8518	−19.2871	0.1086	−19.6523	0.1041	−21.9934	0.0795	−25.0646	0.0558	−25.2448	0.0547
150	−19.2442	0.1091	−17.6430	−20.2006	0.0977	−20.2627	0.0970	−22.4629	0.0753	−24.1357	0.0621	−24.5855	0.0590
200	−20.2848	0.0968	−19.2517	−21.3321	0.0858	−21.0905	0.0882	−22.8753	0.0718	−25.5536	0.0528	−25.7896	0.0513
250	−19.8618	0.1016	−19.1704	−21.0567	0.0885	−21.1177	0.0879	−22.6574	0.0736	−25.1042	0.0556	−25.4619	0.0533
300	−20.5993	0.0933	−19.6937	−21.0901	0.0882	−21.0065	0.0891	−22.8101	0.0724	−24.8396	0.0573	−25.8852	0.0508

Table 4 — PSR analysis of ternary improved logistic sequence with MF, MMF and adaptive filters

Length of the Seq	PSR of MF (dB)	ASP of MF	PSR of Random POP (dB)	PSR of MMF(GHO) (dB)	ASP of MMF (GHO)	PSR of MMF (DE) (dB)	ASP of MMF (DE)	PSR of MMF (DE-GHO) (dB)	ASP of MMF (DE-GHO)	PSR of Adaptive LMS filter (dB)	ASP with LMS Filter	PSR of Adaptive BSSLMS filter (dB)	ASP withBSSLMS Filter ASP
20	−20.8279	0.0909	−14.8504	−20.1309	0.0985	−24.0491	0.0627	−28.2971	0.0385	−67.2278	4.35*10^−4	−44.1552	0.0062
25	−19.0849	0.1111	−14.5911	−20.1791	0.0980	−22.1052	0.0785	−26.9982	0.0447	−46.3103	0.0048	−42.0557	0.0079
30	−19.0849	0.1111	−15.6993	−19.9760	0.1003	−20.6842	0.0924	−23.5140	0.0667	−30.2589	0.0307	−35.6193	0.0166
35	−18.5884	0.1176	−14.8790	−19.2322	0.1092	−20.2691	0.0969	−24.4892	0.0596	−27.7986	0.0407	−29.5084	0.0335
40	−18.5884	0.1176	−15.4193	−20.0221	0.0997	−20.1224	0.0986	−23.5035	00668	−27.2611	0.0433	−27.3968	0.0427
45	−18.4164	0.1200	−15.0988	−18.8990	0.1135	−19.1970	0.1097	−22.8168	0.0723	−26.9743	0.0448	−26.7416	0.0460
50	−18.4164	0.1200	−16.2884	−18.5614	0.1180	−19.1211	0.1106	−22.4155	0.0757	−24.6556	0.0585	−25.3526	0.0540
60	−18.7570	0.1154	−15.5359	−18.5299	0.1184	−18.9968	0.1122	−22.6518	0.0737	−25.5000	0.0531	−25.9931	0.0502
70	−18.5884	0.1176	−15.9355	−18.6981	0.1162	−19.3715	0.1075	−22.3446	0.0763	−24.3456	0.0606	−24.5348	0.0593
80	−19.0849	0.1111	−16.7969	−20.2535	0.0971	−19.8908	0.1013	−22.4326	0.0756	−25.7312	0.0517	−25.7165	0.0518
90	−18.6900	0.1163	−16.4654	−19.7348	0.1031	−19.1865	0.1098	−22.0894	0.0786	−23.8671	0.0641	−23.9772	0.0633
100	−19.0849	0.1111	−17.4694	−19.2057	0.1096	−19.5377	0.1055	−22.1766	0.0778	−24.9620	0.0565	−26.1371	0.0493
150	−19.6680	0.1039	−17.6942	−20.8245	0.0909	−20.2335	0.0973	−22.9604	0.0711	−24.0001	0.0631	−24.8094	0.0575
200	−20.7485	0.0917	−18.4780	−21.3907	0.0852	−20.5203	0.0942	−23.5510	0.0664	−26.0949	0.0496	−25.9238	0.0506
250	−20.7618	0.0916	−19.0024	−21.2837	0.0863	−20.8261	0.0909	−22.7746	0.0727	−25.2117	0.0549	−25.1840	0.0551
300	−20.9399	0.0897	−19.7297	−22.0026	0.0794	−20.8628	0.0905	−23.7233	0.0651	−25.5237	0.0529	−26.2356	0.0488

for binary and ternary sequences with MF, DE-GHO MMF and adaptive filter using BSSLMS.

The tabular analysis of the performance of ternary logistic and improved logistic codes are presented in Table 3 and 4 respectively.

Conclusions

This paper describes a cascaded matched filter whose coefficients are updated using different optimization techniques such as Differential Evolution, the Grass-hopper algorithm, and adaptive algorithms. Initially, the results are compared with that of MF. The proposed method improves the performance of chaotic binary and ternary sequences. In this approach, the PSR is first measured from the auto-correlation pattern using MF. Then the analysis is carried out, which involves implementing MMF and cascading MMF. The evolution of the direction of the grasshopper depends on the attraction intensity and attraction length towards food sources. A reduced PSR of −40.81 dB is obtained with a Barker code of length 13 using DE-GHO cascaded MMF. It is also observed that in the case of binary and ternary, the improved logistic sequence provides an improved PSR of −41.0718 dB and −67.2278 dB, respectively, for length 20. Therefore, the performance achieved with adaptive techniques provides better results for large-length sequences. The future scope of this research work can be extended to obtain better results by using the other polyphase sequences with different hybrid optimization techniques.

References

1 Ackroyd M H & Ghani F, Optimum mismatched filters for sidelobe suppression, *IEEE Trans Aerosp Electron Syst* (IEEE), **9** (1973) 214–218, https://doi.org/10.1109/TAES.1973.309769.

2 Baden J M & Cohen M N, Optimum peak sidelobe filters for bi-phase pulse compression, *Proc IEEE Int Radar Conf*, (1990) 252–259.

3 Nunn C, Constrained optimization applied to pulse compression codes and filters, *IEEE Int Radar Conf*, (2005) 190–194.

4 Levanon N, Cross-Correlation of long binary signals with longer mismatched filters, *IEEE Proc.-Radar Sonar Navig*, **152(6)** (2005) 377–382, https://doi.org/10.1049/ip-rsn:20050024.

5 Gandomi A H, Yang X S, Talatahari S & Alavi A H, Firefly algorithm with chaos, *Commun Nonlinear Sci Numer Simul*, **18(1)** (2013), 89–98, https://doi.org/10.1016/j.cnsns.2012.06.009.

6 Guo Z, Chen B, Min Y & Binggang C, Self-adaptive chaos differential evolution, edited by L Jiao, L Wang, X-B Gao, J Liu, and F Wu, Advances in Natural Computation: *Second Int Conf ICNC*, China, proc, Part I, **4221** of *Lecture Notes in Computer Science*, *Springer*, Berlin, Heidelberg, (2006) 972–975, https://doi.org/10.1007/11881070_128.

7 Ebrahim zadeh R & Jampour M, Chaotic genetic algorithm based on Lorenz chaotic system for optimization problems, *Int J Intell Syst Appl*, **5(5)** (2013) 19–24, https://doi.org/10.5815/ijisa.2013.05.03.

8 Renu K & Rajesh Kumar P, Performance of ternary sequence using adaptive filters, *Proc Int Conf Microelectronics, Electromagnetics Telecommunications*, *LNEE Series, Springer*, **471** (2018) 63–73, https://doi.org/10.1007/978-981-10-7329-8_7.

9 Seventline J B, Sharma G V K, Sridevi K, Elizabath Rani D & Raja Rajeswari K, Development and performance evolution of mismatched filter using differential evolution, *Int J Comput Appl*, **46(8)** (2012) 56–60.

10 Storn R & Price K, Differential evolution: a simple and Efficient Adaptive scheme for global optimization over continuous spaces, *Technical Report*, ***TR-95 - 12***, *International Computer Science Institute*, Berkeley, USA, (1995) 952.

11 Storn R & Price K, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *J Glob Optim*, **11** (1997) 341–359, https://doi.org/10.1023/A:1008202821328.

12 Chang W D, Parameter identification of Rossler's chaotic system by an evolutionary algorithm, *Chaos, Solitons & Fractals, Elsevier*, **29(5)** (2006) 1047–1053, https://doi.org/10.1016/j.chaos.2005.08.121.

13 Saremi S, Mirjalili S & Lewis A, Grasshopper optimization algorithm: *Theory and application*, *Adv Eng Softw*, **105** (2017) 30–47, https://doi.org/10.1016/j.advengsoft.2017.01.004.

14 Ewees A A, Mohamed A E, Alameer Z, Haiwang Y & Zhang J, Improving multilayer perceptron neural network using a chaotic grasshopper optimization algorithm to forecast iron ore price volatility, *Resour Policy*, *Elsevier*, **65(C)** (2020) 101555, https://doi.org/10.1016/j.resourpol.2019.101555.