



18 - 20 Dec, 2009

Generation of Pulse Compression Codes Using NSGA-II

Ajit Kumar Sahoo #1, Ganapati Panda #2, Pyari Mohan Pradhan #3

*#Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela, Orissa, India*

1ajitsahool@gmail.com

**School of Electrical Sciences*

Indian Institute of Technology, Bhubaneswar, Orissa, India

2ganapati.panda@gmail.com

3pyarimohan.pradhan@gmail.com

Archived in Dspace@NITR

<http://dspace.nitrkl.ac.in/dspace>

Generation of Pulse Compression Codes Using NSGA-II

Ajit Kumar Sahoo ^{#1}, Ganapati Panda ^{*2}, Pyari Mohan Pradhan ^{*3}

*#Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela, Orissa, India
¹ajitsahoo1@gmail.com*

**School of Electrical Sciences
Indian Institute of Technology, Bhubaneswar, Orissa, India
²ganapati.panda@gmail.com
³pyarimohan.pradhan@gmail.com*

Abstract—Pulse compression technique avoids the transmission of a signal having small pulse width and high peak power for better range resolution by transmitting phase or frequency modulated large pulse width signal having comparatively low peak power signal. This paper demonstrates an application of non dominated sorting genetic algorithm-II (NSGA-II), a multiobjective algorithm, to generate biphasic pulse compression codes from length 49 to 59. But the aperiodic autocorrelation function (ACF) of the phase coded signals contains range sidelobes which act as self clutter in radar target detection. The efficiency of these codes depends upon the energy content in the range sidelobes of their autocorrelation functions. Peak sidelobe level (PSL) and integrated side lobe level (ISL) are the two performance measures for pulse compression codes. A code is chosen for an application if the ISL and PSL value is within tolerable limits. In this work PSL and ISL are chosen as the objective functions for generating biphasic code.

keywords—NSGA-II, pulse compression, PSL, ISL, biphasic code.

I. INTRODUCTION

Pulse compression is used in peak power limited radars to transmit long waveforms with sufficient energy to detect targets while simultaneously achieving range resolution. The range resolution is determined by the bandwidth of the transmitted signal. Phase or frequency modulation is used for obtaining a high bandwidth, while maintaining constant amplitude. The performance of range resolution would be optimal, if the coded waveform has an impulsive autocorrelation. Biphasic pulse compression is mostly used in radar system in which phase of the transmitted pulse is restricted to 0 degree or 180 degree change of the carrier. Phase coded (PC) waveforms support better range-resolution as compared to linear frequency modulated (LFM) pulses because the windowing functions used, with LFM pulses to lower time sidelobes, cause a broadening in the mainlobe response and deterioration in range resolution. But the autocorrelation of phase coded waveforms contain range sidelobes, which have a negative influence on the detection performance of radar. These sidelobes can mask useful information and may give undesired gain to the received signal from the unwanted target. A desirable property of the compressed pulse is that it should

have low sidelobes in order to prevent a weak target from being masked in the time sidelobes of a nearby stronger target. The lower the sidelobes, relative to the mainlobe peak, the better the main peak can be distinguished and, therefore the better is the corresponding code.

Sequences having low off-peak signal value at the output of the matched filter have found extensive applications in radar and communication systems. Binary pulse compression codes such as the Barker code [1] or maximal-length sequences [2] are extensively used in many radar systems. The digital matched filter for the binary code is much less complex than the polyphase code, and the clutter rejection performance of the binary code is fair while that of the chirp method is poor. Barker sequences are the only known binary sequences with the lowest PSL of unity. The longest available barker sequence is of length 13. Many practical applications require longer codes to achieve higher signal to noise ratio. By exhaustive computer search program, Lindner [3] has found all binary sequences up to length 40 with minimum PSL. Cohen *et al.* [4] have extended those results up to length 48. Using a neural network approach, Hu *et al.* [5] have obtained useful binary sequences for lengths up to 100. Kerdock *et al.* [6] have found the maximum length of a code for a given PSL. Rao and Reddy [7] have obtained larger length binary sequences with minimum sidelobes in their aperiodic autocorrelation pattern by cyclically shifting the Legendre sequences. Levanon [8] has used mismatch processing technique to optimize integrated or peak sidelobe levels of long binary signals. In the literature the pulse compression codes are generated by minimizing either PSL or ISL. In this paper we have used NSGA-II algorithm to minimize both ISL and PSL simultaneously. The pareto front of NSGA-II algorithm for a particular code length contains more than one solution having different PSL and ISL values. The selection of a pulse compression codes depends upon the application and the environmental conditions. If the application is radar designed for a scenario dominated by distributed clutter then ISL is more important. On the other hand if the application requires detection of targets in the presence of large discrete clutter then the PSL is more

important.

The organization of the paper is as follows. In section II, the basis of generating new aperiodic codes is outlined as a multiobjective optimization problem in which the PSL and ISL are simultaneously optimized. The multiobjective tool i.e NSGA-II used as to optimize these two objective functions is discussed in section III. The new sequences generated by the proposed method through simulation study using MATLAB and are listed in section IV. Finally in section V the conclusion of the investigation is highlighted.

II. MULTIOBJECTIVE FORMULATION

Let L be the length binary sequence and is given by

$$S = \{s_1, s_2, s_3, \dots, s_L\} \quad (1)$$

where each element of S has value is either +1 or -1.

The output of the matched filter that is the aperiodic autocorrelation function for positive delays is given as

$$C_k(S) = \sum_{i=1}^{L-k} s_i s_{i+k} \quad (2)$$

where $k = 0, 1, 2, \dots, L - 1$

A significant problem inherent in biphasic pulse compression is that the autocorrelation function does not yield a perfect impulse, that means it does not produce $C_k(S) = 0$ for $k \neq 0$. Any non zero value of $C_k(S)$ for $k \neq 0$ is referred to as sidelobe where as the zero-offset correlation value $C_0(S)$ is called the mainlobe. The difference between a pulse compression waveform and a simple pulse waveform is in the existence and magnitude of these sidelobes. The amplitude of the main peak of an autocorrelation is a measure of signal-to-noise ratio improvement for a given code. The sidelobes limit the usefulness of a code regardless of the strength of the mainlobe. This is because sidelobes signify "self noise" which may be much greater than the system noise. Codes are chosen for a given application based on their length and sidelobe levels.

There are two frequently used sidelobe measures. The first one is the peak sidelobe level(PSL) which is the largest sidelobe in the autocorrelation function of the code and is defined as

$$PSL = \text{Max} |(C_k(S))|, k \neq 0 \quad (3)$$

The second is the integrated sidelobe level (ISL) defined as the total energy contained in all the sidelobes. As the signal is real valued the autocorrelation is real and symmetric about the zero delay. The ISL is represented as

$$ISL = 2 \sum_{k=1}^{L-1} C_k^2(S) \quad (4)$$

The PSL and ISL which are represented in (3) and (4) are simultaneously minimized using NSGA-II algorithm to achieve desired aperiodic codes.

III. NONDOMINATED SORTING GENETIC ALGORITHM-II

In 1967, Rosenberg suggested a genetic search to the simulation of genetics and the chemistry of a population of single-celled organisms with multiple properties or objectives [10]. Genetic algorithm (GA) was originally proposed by J. Holland [11] which imitates Nature's robust way of evolving successful organisms. The first practical algorithm, called vector evaluated genetic algorithm (VEGA), developed by Schaffer [12]. One of the constraints of VEGA is its bias towards some Pareto-optimal solutions. Subsequently Deb and Srinivas [13] proposed the nondominated sorting genetic algorithm (NSGA) which is a popular nondomination based genetic algorithm for multi-objective optimization problem. It is a very effective algorithm but has been generally not preferred for its heavy computational complexity, lack of elitism and for choosing the optimal parameter value for sharing parameter. A modified version, NSGA-II was developed by Deb *et al.* [14] as a better sorting algorithm which incorporates elitism and without using a sharing parameter. A brief outline of NSGA-II algorithm is given below.

- 1) **Population Initialization:** The population contains a set of chromosomes. Each chromosome is initialised randomly with binary bits having length same as the code length.
- 2) **Non-Dominated sort:** The initialized population is sorted based on non-domination
 - for each individual p in main population P perform the following
 - Initialize the set of individuals dominated by p $S_p = \phi$.
 - Initialize the number of individuals that dominate p i.e. $n_p = 0$.
 - for each individual q in P
 - * if p dominates q then
 - $S_p = S_p \cup \{q\}$
 - * else if q dominates p then
 - $n_p = n_p + 1$
 - if $n_p = 0$ then p belongs to the first front and rank of individual p i.e. $p_{rank} = 1$. Update the first front F_1 by adding p to front one i.e. $F_1 = F_1 \cup \{p\}$
 - This is carried out for all the individuals in main population P .
 - Initialize the front counter $i = 1$
 - perform the following if i^{th} front is nonempty i.e. $F_i \neq \phi$
 - $Q = \phi$. The set Q is used to store the members of the next front.
 - for each individual p in front F_i
 - * for each individual q in S_p
 - $n_q = n_q - 1$, decrement the domination count for individual q
 - if $n_q = 0$, set $q_{rank} = i + 1$. Update Q i.e. $Q = Q \cup q$.

TABLE I
SEQUENCES

Seq Len	PSL	ISL	Seq
49	5	472	00111111001111110110000001001011001101001010101
	6	464	00111111001111110110100001001111001101001010101
50	5	658	0001001001110011110010010100011011101010111111010
	6	642	1001000001110011110010010100011011101010111111010
51	5	594	110000010000000110100110011100010001011110010101101
	6	506	100010010000000110100110111100010001011110010101001
52	5	604	000011001100110000111100000001010001011010101101101
	6	564	010100111000111010010111011011010100000001101100010011
53	5	708	1100100110100100000001000100111010111100001010110001
	6	700	11111011011101101000011100111001000010111010101100100
54	5	750	011110101101001000100000110000111010100010011101110100
	6	734	110110101101001000100000110000111010100010011101110000
55	5	830	1000111100010001011111001000110101011111110101101001001
	6	822	1001101100010001111110001000011101011111110101111001011
56	5	920	00111000100011011100011010001000000010011011010000111101
	6	776	00111010100011011100011010001000000010111011010000100101
57	5	1040	110011010111110101110110101110000100010010000111111011000
	6	784	110110010111110101110110101110001100010010000111111011000
58	5	890	100101000111011100010101111100001110100111011001000000010
	6	698	100101000111001100010101101100001110100111011001000000000
59	5	994	0000010010110000111111110001010100011110100011001001110101
	6	818	000001011011010011111101100010101110011110100011011001110100

– $i = i + 1$.

– set Q is the next front and hence $F_i = Q$.

3) **Crowding Distance Assignment:** Once the non dominated sorting is complete the crowding distance is assigned. Since the individuals are selected based on rank and crowding distance, all the individuals in the population are assigned a crowding distance value front wise. The crowding distance is calculated as

- For each front F_i , n is the number of individuals.
 - Initialize the distance for all individuals to be zero i.e. $F_i(d_j) = 0$, where j corresponds to an individual in front F_i .
 - for each objective function m
 - * Sort the individuals in front F_i based on objective function m i.e $I = \text{sort}(F_i, m)$.
 - * Assign infinite distance to boundry values for each individual in F_i i.e $I(d_1) = \infty$ and $I(d_n) = \infty$
 - * for $k = 2$ to $(n - 1)$
 - $I(d_k) = I(d_k) + \frac{I(k+1) \cdot m - I(k-1) \cdot m}{f_m^{\max} - f_m^{\min}}$
 - where $I(k) \cdot m$ is the value of the m^{th} objective function of k^{th} individual in I

4) **Selection:** The individuals are selected using a binary tournament selection. The selection is carried out as follows.

- a) An individual is selected if its non domination rank is smaller than the other.

- b) If both the individuals belong to the same front i.e both have same non domination rank then the individual having higher crowding distance is selected

5) **Genetic Operators:** Single point crossover and mutation operations as used in GA [15] are also employed in NSGA-II.

6) **Recombination and Selection:** The offspring population is combined with the current generation population and the total population is sorted based on nondomination. The new generation is filled by chromosomes from each front subsequently until the population size exceeds the current population size N . If by adding all the individuals in front F_j the population exceeds N then individuals in front F_j are selected based on their crowding distance sorted in the descending order until the population size is N .

IV. SIMULATION RESULTS

The NSGA-II algorithm dealt in section III is used to obtain the desired non periodic binary strings yielding lowest possible PSL and ISL. PSL and ISL are the two cost functions which are minimized using NSGA-II algorithm to generate binary strings varying the lengths from 49 to 59. The population size N is chosen as 150 for each code generation. The PSL and ISL values for each chromosome are determined according to (3) and (4) respectively. The population is sorted based on non-domination. Each individual in first front are given a rank value

of 1. Similarly each individual in second front is assigned a rank value of 2 and so on. Parents are selected from the population on the basis of rank and crowding distance by using binary tournament selection in which an individual is selected if its rank is less than the other. If the rank of both individual are same then crowding distances are compared. The selected population generate offspring by using crossover and mutation operators as discussed in section III. The value of crossover and mutation probability used are 0.8 and 0.1 respectively. The population including the current population and current off springs are sorted based on non-domination and only the best N individuals are selected. Solutions belonging to the best nondominated set, pareto front having rank 1, are best solution in the combined population. If the size of this pareto front is smaller than N then all the members are chosen for the new population. The remaining members of the new population are chosen from the subsequent nondominated fronts in the order of their ranking. This procedure is continued until no more sets can be accommodated. If by adding a front the new population size crosses N then the members from that front are chosen based on crowding distance. To choose exactly N population members, we sort the solutions of the front in descending order of the crowding distance and choose the best solutions, those are having higher crowding distance, to fill all population slots. The number of generations is chosen as 200. In multi objective algorithms a pareto front contains more than one solution for each sequence length. Two codes for each sequence length is given in Table 1. 0's are used in place of -1's to conserve space.

V. CONCLUSION

By using NSGA-II algorithm as given in section III a list of sequences of length 49-59 are obtained, and tabulated in Table I, which also contains PSL and ISL values of the sequences. The use of a sequence depends upon the environmental conditions. For a fixed sequence if the environment is dominated by distributed clutter then the sequence having low ISL value can be used. On the other hand if the environment is dominated by large discrete clutter then the code having low PSL value can be chosen.

REFERENCES

- [1] A.W. Rihaczek, and R.M. Golden, *Range sidelobe suppression for Barker codes*, IEEE Trans. Aerosp. Electron. Syst., Vol.7, No.6, pp. 1087-1092, Nov.1971.
- [2] M. I. Skolnik, *Radar Handbook (2nd ed.)*. New York: McGraw-Hill, 1990.
- [3] J. Lindner, *Binary sequences up to length 40 with best possible autocorrelation function*, Electron. Lett., vol. 11, no. 21, p. 507, Oct.1975.
- [4] M. N. Cohen, M. R. Fox, and J. M. Baden, *Minimum peak sidelobe pulse compression codes*, in Proc. IEEE Int. Radar Conf., pp.633-639, 1990.
- [5] F. Hu, P. Z. Fan, M. Darnell, and F. Jin, *Binary sequences with good aperiodic autocorrelation functions obtained by neural network search*, Electron. Lett., vol. 33, no. 8, pp. 688-689, Apr. 1997.
- [6] A.M. Kerdock, R.Mayer and D. Bass *Longest binary pulse compression codes with given peak sidelobe levels*, Proc. of IEEE, Vol.74, No.2, p. 366, Feb. 1986.
- [7] K.V. Rao, and V.U. Reddy, *Biphase sequences with low sidelobe autocorrelation function*, IEEE Trans. Aerosp. Electron. Syst., Vol.22, No.2 pp. 128-133, Mar.1986.

- [8] N. Levanon, *Cross-correlation of long binary signals with longer mismatched filters* IEE Proc. Radar Sonar Navig., Vol. 152, No. 6, pp.377-382, Dec.2005.
- [9] J.M. Baden, and M.N. Cohen, *Optimal peak sidelobe filters for biphase pulse compression* Proc. IEEE Int. Radar Conf., pp. 249-252, 1990.
- [10] R. Rosenberg, *Simulation of Genetic Populations with Biochemical Properties*, PhD thesis, 1967.
- [11] J. H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor (1975).
- [12] J. D. Schaffer, *Some experiments in machine learning using vector evaluated genetic algorithms*, PhD Thesis, Nashville, TN: Vanderbilt University.
- [13] N. Srinivas and K. Deb, *Multiobjective function optimization using nondominated sorting genetic algorithms*, Evolutionary Computation , vol. 2, no. 3, pp. 221-248, 1995
- [14] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Tran. on Evolutionary Computation, Vol.6, No.2, pp.182-197, Apr. 2002.
- [15] G.Panda, BabitaMajhi, D.Mohanty, A.K.Sahoo, *A GA-Based pruning strategy and weight update algorithm for efficient nonlinear system identification*, Second International Conference Pattern Recognition and Machine Intelligence (PREMI), ISI Kolkata, pp.244-251, India, 2007.