Low Sidelobe Pseudo-Orthogonal Code Sets Through Particle Swarm Optimization

Zach Dunn$^{1,2}$, Mark Year$^{1,2}$, Faruk Uysal$^2$, Caleb Fulton$^{1,2}$

$^1$ School of Electrical and Computer Engineering, University of Oklahoma, Norman, Oklahoma
$^2$ Advanced Radar Research Center, University of Oklahoma, Norman, Oklahoma

Abstract—Future and emerging radar systems are experiencing growing reliance on MIMO functionality in order to support demand for finer resolution and greater data collection under increasingly stringent spectral requirements. In order to support MIMO functionality while preserving spectral usage requirements, pseudo-orthogonal waveform sets are desired. In order to balance the complicated and often contradicting parameter requirements involved in creating pseudo-orthogonal waveform sets, a particle swarm-based optimization technique is introduced. This technique is shown to be applicable to several types of pseudo-orthogonal code sets, including a polyphase coded waveform set and a non-linear frequency modulated waveform set formed utilizing a novel parameterization method.

Keywords—Pseudo-Orthogonal Code Sets, Particle Swarm Optimization

I. INTRODUCTION

In recent years, growing performance demands of next generation radar systems have become increasingly at odds with tightening spectral requirements. The desire for finer range resolution while measuring the full polarimetric scattering matrix over a given and tightly enforced frequency range has led to the need for implementation of pulse compression in conjunction with pseudo-orthogonal waveform sets. Pulse compression is a technique involving the design of a waveform that, when used in conjunction with its matched filter, results in the concentration of a majority of the waveform’s energy over a short duration while retaining the overall energy of a longer waveform [1]. A pseudo-orthogonal waveform set is defined as a collection of waveforms that each display desirable pulse compression characteristics when processed with their own matched filter, but exhibit extremely low cross-correlation characteristics when processed with any of the other codes in the set’s matched filters. While the concept of a pseudo-orthogonal waveform set is easy to comprehend, an actual pseudo-orthogonal waveform set, where all the waveforms are assumed to operate simultaneously over the same bandwidth, is rather difficult to assemble. This difficulty stems from the dependence of pulse compression on a cross-correlation and auto-correlation based process. As a result, a direct optimization solution for pulse compressed waveforms has not yet been found. Therefore, iterative optimization methods must be introduced. This paper proposes the use of particle swarm optimization for the purpose of finding pseudo-orthogonal waveform sets within a given desired set of system parameters. While particle swarm optimization has been previously utilized for antenna array optimization [2-4] and detection and identification of targets [5,6], its application to the optimization of a variety of pseudo-orthogonal code sets has been relatively unexplored.

II. PARTICLE SWARM OPTIMIZATION

A. Background

The particle swarm methodology for the optimization of non-linear functions was first proposed as a biologically inspired algorithm [7]. In these early developments, researchers realized that the overall algorithm, originally designed to model the evolution of movement throughout flocks of birds or schools of fish, could be adapted to optimizing multi-dimensional non-linear problems through a process of intelligent progression of test points within an appropriate state space. This is accomplished by creating an overall fitness function that is a function of all system variables, and that determines the “desirability” of the overall system output given the current variable values. The fitness function is then evaluated by a given number of particles, each at a different “coordinates” within the multi-variate space, at each iteration of the algorithm. After each iteration of the algorithm, the particles adjust their coordinates so that the overall swarm of particles approaches the coordinates giving the most desirable known fitness function result, all the while more closely evaluating as-yet untested coordinates close to the best known coordinates. This makes particle swarm optimization extremely effective at navigating large multi-variate state spaces and finding optimized solutions, whereas traditional techniques (e.g. Newton-Raphson) more easily suffer from limitations due to local minimization of an error function in non-convex problems.

At the initialization of the algorithm, a preset number of “particles”, or fitness function evaluation points, are generated at random within the multivariate space. In addition to randomly generated coordinates, each particle also has a randomly assigned “velocity”, or preset rate of change for each variable dimension. The particles’ future positions are determined by their calculated velocity at each iteration, effectively giving the particles an “inertia” so that the overall swarm is not too easily swayed throughout evaluation of the iterations. All of the particles’ current states’ are then evaluated by the fitness function, and the lowest fitness function personal best seen by each individual particle (and its associated coordinates) are saved. Additionally, each particle also saves the coordinates giving the local best solution, which is the position giving the best solution within the nearest subset of particles, where the
subset size is a predetermined fraction of the total particles in the simulation. Each particle then evaluates its velocity vector, which determines its coordinates for the next iteration. The velocity vector is determined by summing a modifier based on the distance between the particle’s current coordinates and the particle’s personal best coordinates, a modifier based on the distance between the particle’s current coordinates and the particle’s local best coordinates, and a weighted version of the previous iteration’s velocity vector. The modifiers based on distance between coordinates each rely on the distance multiplied by random samples from a uniform distribution between zero and one, where the random sample for each modifier is regenerated every iteration.

For the iterations following the initial iteration, the coordinates of all the points are updated with their individually calculated velocities from the previous iteration. If any of the new points fall outside the preset allowable range of parameters, the violating coordinates are adjusted to satisfy the allowed range. The fitness function is then evaluated for all particle coordinates. For each particle, if it’s current coordinates result in a more desirable result than the previously saved personal best, then the personal best is overwritten with the current coordinates. Each particle’s local best is also reassessed, and the velocity vector for each particle is recalculated for the next iteration. The algorithm halts iterations either when the relative change in fitness function over several iterations has decreased to nearly zero, or when a preset program time limit has been reached.

B. Determining a Fitness Function

The actual execution of the particle swarm optimization algorithm lends itself to application to a vast array of non-linear optimization problems. Therefore, the main algorithm component that makes particle swarm optimization relevant to a specific problem is the fitness function. Fitness functions are evaluated such that lower values equate to more desirable system solutions. While some non-linear problems are relatively straightforward to determine the fitness function and are easy to visualize, such as finding the minimum of a polynomial based two dimensional surface, others are much more abstract and difficult to determine. For the purpose of finding pseudo-orthogonal waveform sets, the types of waveforms desired and waveform characteristics to be optimized must first be chosen. Next, the parameters needed to assemble the desired code types that only affect the chosen characteristics must be chosen before the fitness function can be determined. Possible fitness function characteristics for determining pseudo-orthogonal waveform sets can include maximizing main lobe power, minimizing beamwidth, minimizing peak sidelobe (PSL) levels, and minimizing cross-correlation levels with other codes’ matched filters within the set.

It is not required for a fitness function to be convex, but a non-convex fitness function allows the possibility of the particles to coalesce around a coordinate representing a local minima within the available coordinate space that may not be the global minima. Therefore, it has been found to be beneficial to execute particle swarm optimization several times for a given optimization problem, saving the results for each overall optimization. The fitness function is then executed for each of the saved results, and the coordinates giving the best overall result are then saved as the “optimal solution.” This multiple-optimization approach seems to be especially useful for non-convex fitness functions where the number of a coordinate dimensions approaches a noticeable fraction of the number of particles.

III. OPTIMIZED PSEUDO-ORTHOGONAL CODE SETS

A. Polyphase Coded Waveform Set

The first attempt at creating a pseudo-orthogonal waveform set revolved around creating a pair of length $N$ constant modulus polyphase coded waveforms. Similar to a previous approach, each bit in the code was treated as an individual variable to be optimized [8,9]. Each of the $N$ bits was assigned a variable representing that particular bit’s phase, with the final assembled waveforms shown as

$$R_{poly, 1}[k] = e^{j \psi_1[k]}$$

and

$$R_{poly, 2}[k] = e^{j \psi_2[k]}$$

where $1 \leq k \leq N$ and $0 \leq \psi_1[k], \psi_2[k] < 2\pi$. The collection of all the phase variables compromised the coordinate space to be traversed by the particles. The fitness function $F_{poly}$ was created by calculating the autocorrelation of both waveforms, as well as the cross-correlation between the two waveforms. The magnitude of the peak sidelobe levels from both autocorrelations were summed with the maximum magnitude present in the cross-correlation to yield the fitness function, which is shown below

$$F_{poly} = \left| PSL \left(S_{11}(R_{poly, 1}, R_{poly, 1})\right) \right|$$

$$+ \left| PSL \left(S_{22}(R_{poly, 2}, R_{poly, 2})\right) \right|$$

$$+ max |S_{12}(R_{poly, 1}, R_{poly, 2})|$$

where $S_{xy}$ is defined as the discrete cross correlation, shown as

$$S_{xy} = \sum_{-\infty}^{+\infty} x[n] y^* [n+k], -\infty < k < +\infty.$$  

This simple fitness function encourages a desirable pseudo-orthogonal waveform set by equally penalizing high peak sidelobes from either autocorrelation as well as a high cross-correlation magnitude. This fitness function is not convex though, so the multiple-optimization approach should be used to identify as desirable a result as possible.

Using a multiple-optimization approach with 12 overall iterations, a pair of length 800 polyphase coded waveforms was generated. A length 800 code was chosen as this corresponds with a 20 µs pulse occupying a 40 MHz 4 dB bandwidth. It was seen that the resulting waveforms displayed PSLs of -26.8 dB and -25.9 dB, respectively, with a maximum cross-correlation response of -25.9 dB. These results are shown in Figure 1. While only a pair of polyphase coded waveforms were generated for this example, it should be noted that extending this approach to generate pseudo-orthogonal waveform sets containing more than two polyphase coded waveforms is relatively straightforward.
B. Non-Linear Frequency Modulated Waveform Set

A common waveform set treated as “pseudo-orthogonal” is the pairing of an upchirp and downchirp Linear Frequency Modulated (LFM) waveform that each span the same bandwidth. It has been shown that Non-Linear Frequency Modulated (NLFM) waveforms can result in very low PSLs [10,11]. It was hypothesized that an optimized pair of upchirp and downchirp NLFM waveforms could result in an improved pseudo-orthogonal set over the traditional LFM waveform pair. While the resulting waveform pair should also have improved pseudo-orthogonal characteristics over a similarly optimized polyphase coded waveform set, by nature the approach of pairing of an upchirp with a downchirp is not easily modified to generate NLFM pseudo-orthogonal waveform sets containing more than two waveforms. Similar to a previous Genetic Algorithm-based approach [12], it was decided to create NLFM waveforms by shaping the instantaneous frequency with a spline function of time by preventing multiple possible frequencies for a given point in time. This is enforced by the conditional statement

\[
\text{if } X_{bi}[i] < (X_{bi}[i - 1] + D_{buf})
\]

\[
(X_{bi}[i], Y_{bi}[i]) = (X_{bi}[i] - M[i], Y_{bi}[i] + M[i])
\]

where \(D_{sp}\) is the horizontal spacing between adjacent baseline coordinate points, and \(D_{buf}\) is a user-determined distance buffer in the normalized space to prevent vertical segments in the instantaneous frequency plots.

Once the modifying factors have been found, a spline function is used to smoothly interpolate between the spline coordinates. The resulting spline is then renormalized in time to span the pulse length \(T\), and the normalized frequencies are multiplied by one half the bandwidth, effectively creating the waveform’s true instantaneous frequency as a function of time \(f_{inst}[t]\). This instantaneous frequency is then cumulatively summed and used as the phase terms of the final upchirp and downchirp waveforms, shown below respectively as

\[
R_{NLFM,1}[t] = e^{j 2\pi \sum_{i=1}^{t} f_{inst}[i]} \quad \frac{-T}{2} \leq t \leq \frac{T}{2}
\]

and

\[
R_{NLFM,2}[t] = e^{-j 2\pi \sum_{i=1}^{t} f_{inst}[i]} \quad \frac{-T}{2} \leq t \leq \frac{T}{2}
\]

The fitness function is simply the maximum magnitude present in the cross-correlation summed with the PSL, where the PSL is the highest magnitude point outside the ideal main lobe. The ideal mainlobe width is determined by a Gaussian curve approximation as described in [13]. The fitness function for the NLFM particle swarm optimization is shown as

\[
F_{NLFM} = \left| PSL \left( S_{11}(R_{NLFM,1}, R_{NLFM,2}) \right) \right| + \max \left| S_{12}(R_{NLFM,1}, R_{NLFM,2}) \right|
\]

Using a multiple-optimization approach with 12 overall iterations and \(N = 20\), a pair of 5 μs 10 MHz NLFM waveforms was generated. It was seen that the resulting waveforms display PSLs of -30.7 dB with a maximum cross-correlation response of -17.5 dB. These results are shown compared to a traditional
LFM pairing in Figure 2, and it is seen that while the cross-correlation is only 0.05 dB lower than the traditional LFM case, the PSL is 17.2 dB lower in the presented NLFM waveform set. These results help to demonstrate the viability of particle swarm optimization for the purpose of generating pseudo-orthogonal waveform sets within a given bandwidth and set of parameters that is much more desirable than the traditional LFM based waveform set.

IV. CONCLUSION

Particle swarm optimization offers a promising approach for generating pseudo-orthogonal waveform sets under various constraints for next generation radar systems. This offers the possibility of creating waveform sets with optimal autocorrelation and cross-correlation characteristics while satisfying restrictive spectral requirements. This was demonstrated through a constant modulus polyphase coded example, as well as a constant modulus NLFM waveform created with a novel parametric methodology utilizing spline interpolation. Therefore, through the construction and assignment of a clever fitness function, pseudo-orthogonal waveform sets can be easily and efficiently assembled and utilized for MIMO, polarimetric, and other modern and emerging radar architectures.

REFERENCES