Scientific Research and Essays Vol. 5 (20), pp. 3049-3058, 18 October, 2010
Available online at http://www.academicjournals.org/SRE
ISSN 1992-2248 ©2010 Academic Journals

Review

A novel UWB pulse design method using particle swarm optimization algorithm

Seyed Noorodin Keshavarz[1*], Mostafa Attaran Kakhki[1], Mahdi Gholami Omali[2] and Mehdi Hamidi[2]

[1]Department of Electrical Engineering Ferdowsi University, Mashhad, Iran,
[2]Department of Electrical Engineering, Sajjad University, Mashhad, Iran

Accepted 3 June, 2010

In this paper a new combined Gaussian UWB pulse waveform, optimized using particle swarm optimization (PSO) algorithm is presented. In fact since a Gaussian-derived pulse itself can not satisfy FCC spectral constraints for the entire spectrum of a UWB signal, we are persuaded into properly weighted combinations of Gaussian derivatives creating a spectrum maximally close to FCC standards for a UWB system. In this report we have investigated three methods: Random combination, combination based upon least square error using PSO algorithm, and combination based on minimizing the area between FCC standard emission mask and PSD of the designed signal using PSO algorithm. Finally a pulse generated by first, forth, and fifth-order derivatives of Gaussian function is presented which is better compared with its previously designed counterparts when it comes to spectral performance even for frequencies less than 1 GHz.

Key words: Gaussian derivative, PSO algorithm, pulse design, UWB pulse, spectral shaping.

INTRODUCTION

In recent years, the ultra-wideband (UWB) radio has attracted increasing interests for its potential applications in short-range high-data-rate wireless communications (Win and Scholtz, 1998; Win and Scholtz, 2000). With its enormous bandwidth, UWB signaling provides fine temporal resolution and offers the potential for ample multi-path diversity. It is well known that the UWB system has two essential requirements (Shan et al., 2005). The first is to alleviate the possible interference with other existing systems (Hämäläinen et al., 2006; Lei et al., 2006). Because UWB radio systems operate with extremely large bandwidths and have to coexist with many existing systems, the equivalent isotropically radiated power (EIRP) must comply with the regulation by some organizations, for example, the Federal Communications Commission (FCC) in the United States, the FCC has regulated the main frequency band of the UWB systems to be between 3.1 and 10.6 GHz (Federal Communication Commission, 2002). The second requirement is the optimal receiving characteristics. The transmission reliability of a UWB system is determined by the received signal-to-noise ratio (SNR). Given the stringent limitations of transmission power, maximizing of the received SNR requires efficient utilization of the bandwidth and power allowed by the FCC mask. It is well-known that the spectrum of the transmitted signal is effectively determined by that of the underlying UWB pulse and the need to fully utilize the allocated bandwidth. On the other hand, the orthogonality of received pulses has notable impact on the performance of the correlation receiver (Parr et al., 2003; Michael et al., 2002). Therefore, the design of the pulse shape is important in UWB systems.

The Gaussian monocycle pulse is commonly used in UWB impulse radio, however, it exhibits a poor fit to the FCC spectral mask and thus is not desirable for practical usage. There have been challenges of designing new pulses, such as the modified Hermite pulses (MHP) and a pulse design algorithm utilizing ideas of prolate spheroidal wave function (Michael et al., 2002). However, frequency shifting and bandpass filters are required for the MHP of order 0 or 1 for higher order MHP, respecttively, to satisfy the FCC spectral mask. Paper (Parr et al., 2003) designed the pulses based on the dominant eigenvectors of a channel matrix constructed by sampling the spectral mask. Pulses generated from different

*Corresponding author. E-mail: snkt20@yahoo.com.

eigenvectors are mutually orthogonal, and conform to the FCC spectral mask. However, they cannot achieve the optimal spectral utilization, and require a high sampling rate (64 GHz) that is difficult to implement in practice.

Digital FIR filter solutions based on the Parks–McClellan (PM) algorithm (Mc Clellan and Parks, 1973) have also been exploited for shaping UWB pulses under mask-fitting requirements (Luo et al., 2003). The PM design facilitates good approximations of the FCC spectral mask in a minimax sense but does not directly optimize the spectral utilization of the pulse. Moreover, trial-and-error is required to find suitable values for the parameters implicit in a PM design, such as the edges tolerances of the pass and stop bands, and the frequency weighting of the approximation error.

The raise Cosine impulse has better spectrum characteristic and satisfy the bandwidth constraints of FCC, but it cannot be realized using a simple circuit. Another fundamental method is to modulate the sinusoid wave form using the window function such as Gaussian window and squared raised Cosine window. Recently several methods that satisfy the FCC mask have been proposed (Parr et al., 2003; Corral et al., 2002), which are more practical for implementation (Matsuo et al., 2005). Besides this, other UWB waveform design methods were proposed in some literatures. For example, paper (Ramfrez- Mireles, 2002) formulated the signal design for binary UWB communications taking into consideration the particular characteristics of UWB propagation in a dense multi-path channel. Papers (Han and Nguyen, 2002; Chen et al., 2004) investigated the design for the source pulses and antennas, and the optimization of the UWB signals with some considerations about antenna systems. In paper (Liang et al., 2003), the authors focused on the UWB radar waveform design without constraints. Paper (Shan et al., 2005) presented and studied three frequency-domain models based on the differential evolution for optimizing source pulses and detection templates in UWB radio systems. This paper (Nakache and Molisch, 2006) studies the design of signaling waveforms for time-hopping impulse UWB radio (TH-IR) with limits on the power spectral density.

The results in paper (Jayaweera, 2005) showed that in additive white Gaussian noise (AWGN) with coherent reception, the optimal (in the sense of minimum probability of error) performance of waveform design is achieved by choosing the modulation time shift parameter. Design of the family of orthogonal and spectrally efficient UWB waveforms was proposed (Igor and Ryuji, 2007; Michael et al., 2001).

Here, we research pulse design method based on Gaussian derivatives that were produced easily and used broadly, proposed a simple pulse design method making use of a linear multi-pulse combination of a Gaussian derivation and through analyzing the effect of the shape factor α_i and the weight factor w_i of linear combination on spectrum distribution we proposed to optimize its weight vector and shaping factor by PSO algorithm and we will do experiments to prove this method efficient. The rest of this paper is organized as follows; description of Gaussian derivatives; presentation of PSO algorithm; the design of Gaussian mixed UWB pulse waveforms via PSO algorithm is addressed and some design examples are presented, and lastly concluding remarks are provided.

GAUSSIAN DERIVATIVES

The basic Gaussian pulse in time domain is shown as:

$$f(t)=-\frac{\sqrt{2}}{\alpha}\exp\left(\frac{-2\pi t^2}{\alpha^2}\right) \quad (1)$$

where, $\alpha^2=4\pi\sigma^2$ is pulse shaping factor and σ^2 is variance.

The k^{th} derivative of $f(t)$, is expressed as $f^{(k)}(t)$ and its frequency domain expression through the Fourier transform is:

$$F_k(f)=\sqrt{\alpha}\left(2\pi f\right)^k\exp\left(-\pi f^2\alpha^2/2\right) \quad (2)$$

The power spectral density (PSD) of $F_k(f)$ is expressed as:

$$P_k(f)=\left|F_k(f)\right|^2=\alpha\left(2\pi f\right)^{2k}\exp\left(-\pi^2 f^4\alpha^4/4\right) \quad (3)$$

Pulse shaping factor

As pulse shaping factor α changes, Gaussian pulse duration will also change. Generated waveforms of Gaussian pulse with α changing in the time domain and frequency domain were illustrated in Figure 1 and 2. From Figure 1 and 2, it is clear that the pulse width decreases as α increases.

Derivative order

If α is constant and k increases, then the spectrum energy will distribute in the high frequency band. To explain the influence, the frequency domain waveform of Gaussian derivatives was illustrated in Figure 3. Where, α equaled to 0.625 ns. It is clear that the PSD moves to higher frequency with the order augmentation of Gaussian derivative.

PARTICLE SWARM OPTIMIZATION ALGORITHM

The particle swarm paradigm, that was only a few years ago a curiosity, has now attracted the interest of researchers around the globe. This gives an overview of

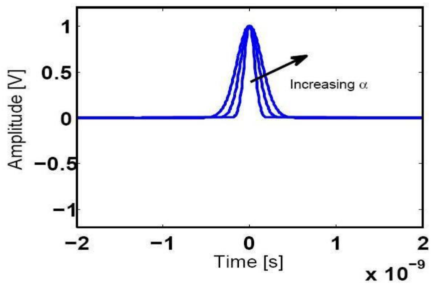

Figure 1. The Gaussian pulse changes with α (Time domain).

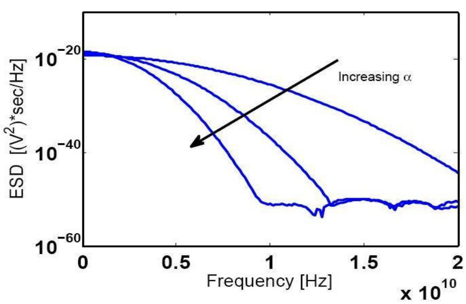

Figure 2. The Gaussian pulse changes with α (Frequency domain).

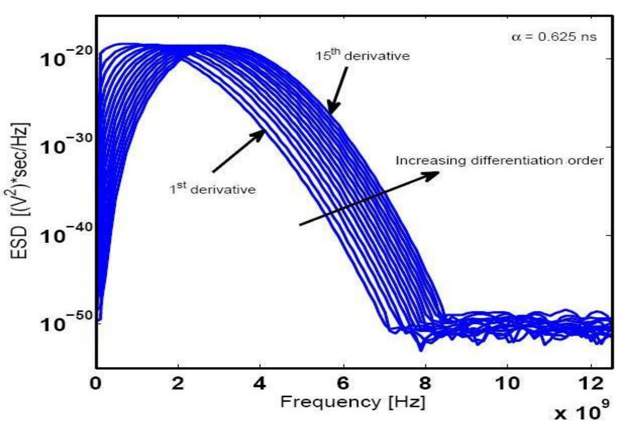

Figure 3. The Gaussian pulse changing with the order augmentation of Gaussian derivative (frequency domain).

important work that gave direction and impetus to research in particle swarms as well as some interesting new directions and applications (James and Russel,1995; James and Russel, 2001). Things change fast in this field as investigators discover new ways to do things, and new things to do with particle swarms. It is impossible to cover all aspects of this area within the strict page limits of this paper.

The initial ideas on particle swarms of Kennedy and Eberhart were essentially aimed at producing computational intelligence by exploiting simple analogues of social interaction, rather than purely individual cognitive abilities. The first simulations (Kennedy and Eberhart, 1995) were influenced by Heppner and Gren-ander's work (Heppner and Grenander, 1990) and involved analogues of bird flocks searching for corn. These developed (Kennedy and Eberhart, 1995) into a powerful optimization method—Particle Swarm Optimization (PSO).

In PSO, a number of simple entities—the particles—are placed in the search space of some problem or function, and each evaluates the objective function at its current location. Each particle then determines its movement through the search space by combining some aspect of the history of its own current and best (best-fitness) locations with those of one or more members of the swarm, with some random perturbations. The next iteration takes place after all particles have been moved. Eventually the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function.

Each individual in the particle swarm is composed of three D-dimensional vectors, where D is the dimensionality of the search space. These are the current position $\vec{x}_i$, the previous best position $\vec{p}_i$, and the velocity $\vec{v}_i$.

The current position $\vec{x}_i$ can be considered as a set of coordinates describing a point in space. On each iteration of the algorithm, the current position is evaluated as a problem solution. If that position is better than any that has been found so far, then the coordinates are stored in the second vector, $\vec{p}_i$. The value of the best function result so far is stored in a variable that can be called $pbest_i$ (for "previous best"), for comparison on later iterations. The objective, of course, is to keep finding better positions and updating $\vec{p}_i$ and $pbest_i$. New points are chosen by adding $\vec{v}_i$ coordinates to $\vec{x}_i$, and the algorithm operates by adjusting $\vec{v}_i$, which can effectively be seen as a step size.

The particle swarm is more than just a collection of particles. A particle by itself has almost no power to solve any problem; progress occurs only when the particles interact. Problem solving is a population-wide phenomenon, emerging from the individual behaviors of the particles through their interactions. In any case, populations are organized according to some sort of communication structure or topology, often thought of as a social network. The topology typically consists of bidirectional edges connecting pairs of particles, so that if j is in i's neighborhood, i is also in j's. Each particle communicates with some other particles and is affected by the best point found by any member of its topological

Table 1. Key PSO vocabulary.

Some Key terms used to describe PSO	
Particle/ Agent	One single individual in the swarm.
Location/Position	An agent's N dimensional coordinates which represents a solution to the problem.
Swarm	The entire collection of agents.
Fitness/Cost/ Evaluation Function	A single number representing the goodness of a given solution.
Personal best (pbest)	The location in parameter space of the best fitness retuned for a specific agent.
Global best (gbest)	The location in parameter space of the best fitness retuned for entire swarm.
V_{max}	The maximum allowed velocity in a given direction.

neighborhood. This is just the vector $\vec{p}_i$ for that best neighbor, which will be denoted with $\vec{p}_g$.

In the particle swarm optimization process, the velocity of each particle is iteratively adjusted so that the particle stochastically oscillates around $\vec{p}_i$ and $\vec{p}_g$ locations. In fact PSO is an evolutionary algorithm based on the intelligence and cooperation of group of birds or fish schooling. It maintains a swarm of particles where each particle represents a potential solution. The PSO algorithm particles are flown through a multidimensional search space, where the position of each particle is adjusted according to its own experience and that of its neighbors. Table 1 shows some key terms used to describe PSO (Robinson and Sammi, 2004). The (original) process for implementing PSO is shown in Algorithm 1.

Algorithm 1: Original PSO

1. Initialize a population array of particles with random positions and velocities on D dimensions in the search space.
2. Loop
3. For each particle, evaluate the desired optimization fitness function in D variables.
4. Compare particle's fitness evaluation with its pbest_i. If current value is better than pbest_i, then set pbest_i equal to the current value and $\vec{p}_i$ equal to the current location $\vec{x}_i$ in D-dimensional space.
5. Identify the particle in the neighborhood with the best success so far, and assign its index to the variable g.
6. Change the velocity and position of the particle according to the following equation:

$$\begin{aligned} &\vec{v}_i \leftarrow \vec{v}_i + \vec{U}(0,\phi_1)\otimes(\vec{p}_i - \vec{x}_i) + \vec{U}(0,\phi_2)\otimes(\vec{p}_g - \vec{x}_i) \\ &\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i \end{aligned} \quad (4)$$

7. If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), exit loop.
8. End loop

Note that, $\vec{U}(0,\phi_i)$ represents a vector of random numbers uniformly distributed in [0, ϕ_i] which is randomly generated at each iteration for each particle. $\otimes$ is component-wise multiplication. In the original version of PSO, each component of $\vec{v}_i$ is kept within the range $[-V_{max}, +V_{max}]$.

Parameters

The basic PSO described above has a small number of parameters that need to be fixed. One parameter is the size of the population. This is often set empirically on the basis of the dimensionality and perceived difficulty of a problem. Values in the range 10 - 50 are quite common. The parameters ϕ_1 and ϕ_2 in (4) determine the magnitude of the random forces in the direction of personal best $\vec{p}_i$ and neighborhood best $\vec{p}_g$. These are often called acceleration coefficients. The behavior of a PSO changes radically with the value of ϕ_1 and ϕ_2. Interestingly, we can interpret the components $\vec{U}(0,\phi_1)\otimes(\vec{p}_i - \vec{x}_i)$ and $\vec{U}(0,\phi_2)\otimes(\vec{p}_g - \vec{x}_i)$ in (4) as attractive forces produced by springs of random stiffness, and we can approximately interpret the motion of a particle as the integration of Newton's second law. In this interpretation, $\phi_1/2$ and $\phi_2/2$ represent the mean stiffness of the springs pulling a particle. It is not surprising then that by changing ϕ_1 and ϕ_2 one can make the PSO more or less "responsive" and possibly even unstable, with particle speeds increasing without control. The value ϕ_1 = ϕ_2 = 2.0, almost ubiquitously adopted in early PSO research, did just that. However, this is often harmful to the search and needs to be controlled. The technique originally proposed to do this was to bound velocities so that each component of $\vec{v}_i$ is kept within the range $[-V_{max}, +V_{max}]$. The choice of the parameter V_{max} required some care since it appeared to influence the balance between exploration and exploitation. The use of hard bounds on velocity, however present some problems. The optimal value of V_{max} is problem-specific, but no reasonable rule of thumb is known. Further, when V_{max} was implemented, the particle's trajectory failed to converge. Where one would hope to shift from the large-scale steps that typify exploratory search to the finer, focused search of exploitation, V_{max} simply chopped off the particle's oscillations, so that some hopefully satisfactory compromise will be seen throughout the run.

Inertia weight

Motivated by the desire to better control the scope of the search, reduce the importance of V_{max}, and perhaps eliminate it altogether, the following modification of the PSO's update equations was proposed (Shi and Eberhart, 1998b):

$$\begin{aligned} \vec{v}_i &\leftarrow \omega \vec{v}_i + \vec{U}(0,\phi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0,\phi_2) \otimes (\vec{p}_g - \vec{x}_i) \\ \vec{x}_i &\leftarrow \vec{x}_i + \vec{v}_i \end{aligned} \quad (5)$$

where ω was termed the "inertia weight." Note that if we interpret $\vec{U}(0,\phi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0,\phi_2) \otimes (\vec{p}_g - \vec{x}_i)$ as the external force, $\vec{f}_i$, acting on a particle, then the change in a particle's velocity (that is, the particle's acceleration) can be written as $\Delta \vec{v}_i = \vec{f}_i - (1-\omega)\vec{v}_i$. That is, the constant $1-\omega$ acts effectively as a friction coefficient, and so ω can be interpreted as the fluidity of the medium in which a particle moves. This perhaps explains why researchers have found that the best performance could be obtained by initially setting ω to some relatively high value (e.g., 0.9), which corresponds to a system where particles move in a low viscosity medium and perform extensive exploration, and gradually reducing ω to a much lower value (e.g., 0.4), where the system would be more dissipative and exploitative and would be better at homing into local optima. It is even possible to start from values of $\omega > 1$, which would make the swarm unstable, provided that the value is reduced sufficiently to bring the swarm in a stable region (the precise value of ω that guarantees stability depends on the values of the acceleration coefficients).

Naturally, other strategies can be adopted to adjust the inertia weight. For example, in (Eberhart and Shi, 2000) the adaptation of ω using a fuzzy system was reported to significantly improve PSO performance. Another effective strategy is to use an inertia weight with a random component, rather than time-decreasing. For example, (Eberhart and Shi, 2001) successfully used $\omega = U(0.5, 1)$. There are also studies (Zheng et al ., 2003), in which an increasing inertia weight was used to obtain good results. With (2) and an appropriate choice of ω and of the acceleration coefficients, ϕ_1 and ϕ_2, the PSO can be made much more stable, so much that one can either do without V_{max} or can set V_{max} to a much higher value, such as the value of the dynamic range of each variable (on each dimension). In this case, V_{max} may improve performance, though with use of inertia or constriction techniques, it is no longer necessary for damping the swarm's dynamics.

Constriction coefficients

Though the earliest researchers recognized that some form of damping of the dynamics of a particle was necessary, the reason for this was not understood. But when the particle swarm algorithm is run without restraining velocities in some way, these rapidly increase to unacceptable levels within a few iterations. Kennedy (1998) noted that the trajectories of nonstochastic one-dimensional particles contained interesting regularities when $\phi_1 + \phi_2$ was between 0.0 and 4.0. Clerc's analysis of the iterative system led him to propose a strategy for the placement of "constriction coefficients" on the terms of the formulas; these coefficients controlled the convergence of the particle and allowed an elegant and well-explained method for preventing explosion, ensuring convergence, and eliminating the arbitrary V_{max} parameter. The analysis also takes the guesswork out of setting the values of ϕ_1 and ϕ_2.

Clerc and Kennedy (2002) noted that there can be many ways to implement the constriction coefficient. One of the simplest methods of incorporating it is the following:

$$\begin{aligned} \vec{v}_i &\leftarrow \chi(\vec{v}_i + \vec{U}(0,\phi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0,\phi_2) \otimes (\vec{p}_g - \vec{x}_i)) \\ \vec{x}_i &\leftarrow \vec{x}_i + \vec{v}_i, \end{aligned} \quad (6)$$

where $\phi = \phi_1 + \phi_2 > 4$ and

$$\chi = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (7)$$

When Clerc's constriction method is used, ϕ is commonly set to 4.1, $\phi_1 = \phi_2$, and the constant multiplier χ is approximately 0.7298. This results in the previous velocity being multiplied by 0.7298 and each of the two $(\vec{p} - \vec{x})$ terms being multiplied by a random number limited by 0.7298 × 2.05 ≈ 1.49618.

GAUSSIAN MIXED UWB PULSE DESIGN

In this paper, the possibility of tuning the PSD of a generated pulse by combining a few single reference Gaussian

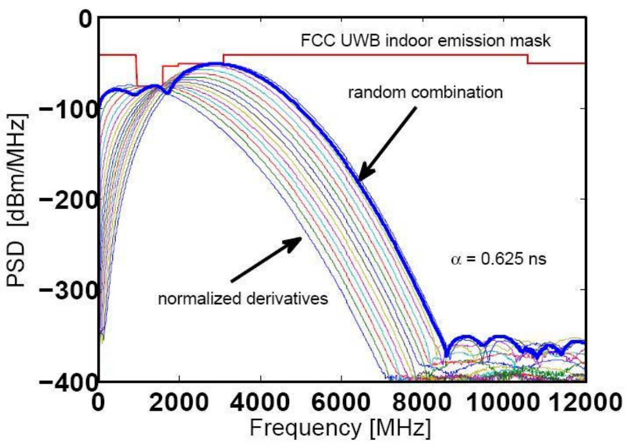

Figure 4. Random combination (case 1).

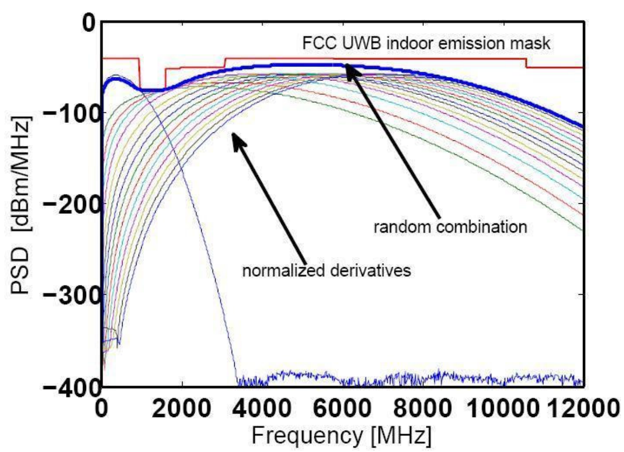

Figure 5. Random combination (case 2).

pulse waveform and its derivatives to adjust the PSD to the mask was analyze. A possible approach is to use linear combinations of N Gaussian derivatives, each being characterized by a given α value and combined them linearly with different weight factors into a pulse. Note that the combination of N derivatives and the possibility of choosing different α values for different derivatives provide a high degree of flexibility in the generation of pulse waveforms. The combinational pulse is:

$$p(t)=\sum_{i=1}^{N} a_i f^{(i)}(t,\alpha_i) \quad (8)$$

where a_i is weight factor and α_i is pulse shaping factor. We want to design and compare combinational UWB pulses from two aspects: Random combination, and error minimization such as the Least Square Error (LSE) or the area between standard UWB emission mask and the PSD of the linear combination signal via PSO algorithm.

Random combination

Now, we will analyze the problem of UWB emission mask approximation with combination of Gaussian derivatives and perform the approximation through two cases: In the first case, all derivatives have the same shape factor α, while in the second case, different derivatives take different α values.

We generate a set of coefficient for the first 15 derivatives in a random way and check if the PSD of the linear com-bination of the functions obtained with these coefficients satisfies the emission limits. In order to gain better weight coefficients, we repeated choosing another sets of coefficients until the distance between the mask and PSD of the generated waveform falls below a fixed threshold.

Figures 4 and 5 showed the PSD of the waveforms obtained by random combination of the first 15 deriva-tives plotted against the FCC emission mask. The weight coefficients and shape factors of the combined UWB pulses for case 1 and 2 are shown in Table 2

Figure 4 shows that the combination of several Gaussian deri-vatives leads to a good approximation of the emission mask, particular in the band 0.96 GHz- 3.6 GHz. Outside this band, power is less efficiently used. But the spectral performance of case 2 is better than case 1 and single Gaussian derivation pulse.

UWB pulse design using PSO algorithm

In this section, we describe the PSO algorithm for UWB pulse design. First of all, we define the number of individuals as N and the dimension of the search space of an optimization as *d.* The position of an individual at k^{th} generation (iteration) is denoted by $X_k[i]\in R^d$, that is, its velocity and evaluation value are denoted by $V_k[i]\in R^d$ and $J_k[i]\in R$, respectively. The index i= 1, 2, ... , N is the number of the individual. An individual's best position and its evaluation value in past are denoted by $p_k[i]\in R^d$ and $J_{pk}[i]\in R$. The best position and evaluation value of whole swarm including past generation are denoted by $g_k\in R^d$ and $J_{gk}\in R$. You can see the behavior of an individual in 2-dimensional search space in Figure 6.

This algorithm is described as follows:

[PSO algorithm steps]

[step 1]

Generate the individuals $X_0[i], \forall i\in[1,2,...,N]$ of initial gen-eration (k = 0) randomly. [step 2]

Compute the evaluation value and update $p_k[i]$ for all individuals.

$$p_k[i]=X_k[i], \text{ if } J_k[i]>J_{pk}[i], \forall i. \quad (9)$$

Table 2. The weight coefficients and shape factors of the combined UWB pulses for case 1 and case 2.

Case 2		Case 1		
w_i	α_i	w_i	α_i	i
-0.082	1.525 ns	-0.038	0.625 ns	1
0.355	0.325 ns	0.003	0.625 ns	2
0.076	0.325 ns	-0.013	0.625 ns	3
0.831	0.325 ns	0.045	0.625 ns	4
-0.860	0.325 ns	-0.044	0.625 ns	5
-0.973	0.325 ns	0.091	0.625 ns	6
0.903	0.325 ns	-0.029	0.625 ns	7
0.030	0.325 ns	0.236	0.625 ns	8
0.500	0.325 ns	0.505	0.625 ns	9
0.217	0.325 ns	0.062	0.625 ns	10
0.697	0.325 ns	0.615	0.625 ns	11
0.494	0.325 ns	-0.778	0.625 ns	12
-0.144	0.325 ns	0.078	0.625 ns	13
0.181	0.325 ns	-0.550	0.625 ns	14
0.790	0.325 ns	0.817	0.625 ns	15

[step 3]
Update g_k by the following equation.

$$g_k[i]=X_k[i],\ \text{if}\ J_k[i]>J_{gk}[i],\ \forall i. \tag{10}$$

[step 4]

Generation changes with the following equation.

$$X_{k+1}[i]=X_k[i]+V_k[i],\qquad \forall i$$
$$V_k[i]=\kappa\times\{V_{k-1}[i]+c_1\times\Gamma(g_k-X_k[i])+c_2\times\Lambda(p_k[i]-X_k[i])\},\ \forall i, \tag{11}$$

where, $c_1, c_2, \kappa \in R$ and $\Gamma, \Lambda \in R^{d\times d}$ are denoted by

$$\Gamma=\text{diag}[\gamma_1, \gamma_2, ..., \gamma_d]$$
$$\Lambda=\text{diag}[\lambda_1, \lambda_2, ..., \lambda_d] \tag{12}$$

where, $\gamma_i \in [0,1]$, $\lambda_i \in [0,1]$ ($i=1,2,...,d$) are uniform pseudorandom numbers.
[step 5]
$k=k+1$, then repeat [step 2] through [step 5].

Random selection is not the only possible strategy for determining the set of coefficients in the linear combination. In fact if the weight vector of linear combination was selected randomly, the attained combination pulse may not be the optimum, so we propose to optimize the weight vector using PSO. In PSO algorithm, the evaluation function definition is very important. The goal is to optimize weight vector so as to make combination pulse close to FCC emission mask to the greatest extent, that is, to minimize error between them. So we defined two evaluation functions below:

1. Least square error (LSE) evaluation function:

A more systematic way of selecting such coefficients is our procedure for error minimization, in which the following error function must be minimized:

$$e_{s1}(t)=\int_{-\infty}^{+\infty}|e_1(t)|^2dt=\int_{-\infty}^{+\infty}\left|f(t)-\sum_{k=1}^{N}a_kf_k(t)\right|^2dt \tag{13}$$

In equation (13), f(t) is the target function or FCC emission mask. Note that since requirements are specified in terms of meeting a PSD, the error equation (13) could be rewritten as follows:

$$e_1=\int_{-\infty}^{+\infty}|P_M(f)-F(f)|^2df \tag{14}$$

where $P_M(f)$ is FCC emission mask and $F(f)$ is PSD of linear combination pulse.

2. The error function e_2 which is the area between standard UWB emission mask and the PSD of the linear combination pulse error function.
We define this error function as:

$$e_2=\left|\int_0^{10.6}(P_M(f)-F(f))df\right| \tag{15}$$

e_2 is the area between FCC emission mask and the PSD of combined Gaussian derivatives pulse. e_1 and e_2 are error functions and we want to find the weight vector that minimizes the error functions e_1 and e_2.

Coding rules

In PSO, individuals of the k^{th} generation have information of position $X_k[i]$, velocity $V_k[i]$ and its evaluation value $J_k[i]$. In addition, these individuals remember a past best position $p_k[i]$ of itself. PSO is an optimization technique by the generational change, in which each particle searches for the position with the highest evaluation value. The generational change is implementted based on $X_k[i]$, $V_k[i]$, $p_k[i]$, and past best position of the swarm g_k. Here, we would use PSO to optimize the vector $[\alpha_j \quad w_j]$ of linear combination of Gaussian derivatives. We denote this vector by $X_k^{d_m}[i]$, where i (i=1,2,...,N) is the i^{th} particle at k^{th} generation, m is the length of each particle or the dimension of the search space of PSO algorithm. Combination pulses are composed of the 15 single Gaussian derivatives pulses, so the

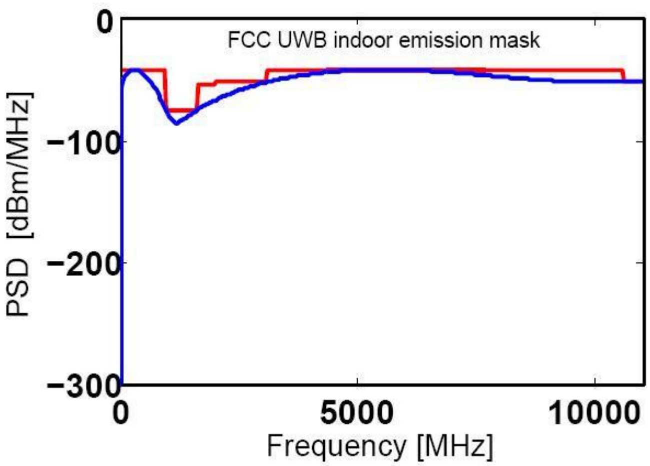

Figure 7. The PSD of PSO linear combined UWB pulse using e_1 .

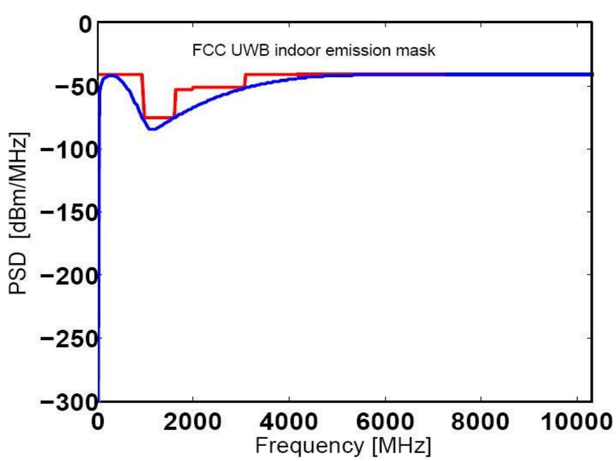

Figure 8. The PSD of PSO linear combined UWB pulse using e_2

the length of vector is 30. When optimizing problems, PSO involves parameters such as population size N, that is, the number of individuals in population, dimension of the search space of optimization as d, and the maximum iteration S. Here N is supposed to be 10, the maximum iteration is chosen to be 20 and d equals to 30.

At the first step we generate the initial population randomly, and then in the following, step 2 through step 5 are applied to all individuals. For example for i=1, we have:

$$\begin{array}{ll} X_0^{d_m}[1]=\left[X_0^{d_1}[1] \ \cdots \ X_0^{d_{30}}[1]\right] & 1^{th} \text{ particle at } 1^{th} \text{ generation} \\ \vdots \qquad \vdots \qquad \vdots & \vdots \\ X_0^{d_m}[10]=\left[X_0^{d_1}[10] \ \cdots \ X_0^{d_{30}}[10]\right] & 10^{th} \text{ particle at } 1^{th} \text{ generation} \end{array} \quad (16)$$

Compute the evaluation value according to the equation (14) or (15) for minimizing error function and then update $p_k[i]$ for all individuals based on (9). Then update g_k by equation (10). After that we update generation of the swarm from k to k+1 by equation (11) and then repeat the algorithm. Now we perform re-initialization of the swarm by the following formulation returns to Gaussian derivatives.

$$X_k[i]=g_k-\delta(R_{max}-R_{min})\times(2\gamma-1) \quad (17)$$

Individuals which go out of the search area are returned to following position:

$$X_k[i]=R_{max}-2\delta\cdot\gamma(R_{max}-R_{min}) \quad \text{if} \quad X_k[i]>R_{max}$$

$$X_k[i]=R_{min}+2\delta\cdot\gamma(R_{max}-R_{min}) \quad \text{if} \quad X_k[i]<R_{min} \quad (18)$$

where R_{min} and R_{max} are minimum and maximum value of the search area respectively, δ is the range of a controllable power factor, and $\gamma\in[0,1]$ denotes a uniform pseudorandom number.

SIMULATION RESULTS

In Figure 7 and 8 PSO optimized the vector $[\alpha_j \quad w_j]$ of linear combined of Gaussian derivatives. As we applied PSO algorithm using equations (14) and (15) for j=1, 4, 5 based on linear combination of Gaussian derivatives, we obtained the optimum value of vecto-r $[\alpha_1 \quad \alpha_4 \quad \alpha_5 \quad w_1 \quad w_4 \quad w_5]$ equaled to $[1.96\,\text{ns} \quad 0.0758\,\text{ns} \quad 0.231\,\text{ns} \quad -0.0079 \quad -0.0323 \quad 0.0391]$ and $[2.044\,\text{ns} \quad 0.100\,\text{ns} \quad 0.211\,\text{ns} \quad -0.0180 \quad -0.0313 \quad 0.2872]$,respectively. From Figure 7 and 8, it could be understood that the optimized pulse by PSO algorithm met the FCC emission mask in whole frequency band; it was better than single Gaussian derivative and had much larger PSD than random combination pulse under satisfying condition.

Because duration of UWB pulse is very short, so it often differentiates different users by TH-code. This share of the common spectrum is THMA in IR-UWB system. As to PSO combined pulses in Figure 7 and 8, we use the standard Gaussian approximation (SGA) algorithm to evaluate the BER performance (Zhang and Aaron, 2005; Scholtz, 1993). It is supposed that the modulation method is 2PPM and the controllable power of transmission of Pr_b is shown as:

$$Pr_b=\frac{1}{2}erfc\left(\sqrt{\left(\left(E_b^{(1)}/N_0\right)^{-1}+\left(\gamma_R\Big/2R_b(N_u-1)\int_{-T_M}^{T_M}R_0^2(\tau)d\tau\right)^{-1}\right)^{-1}\Big/2}\right) \quad (19)$$

where E_b/N_0 is the signal-to-noise ratio, R_b is data

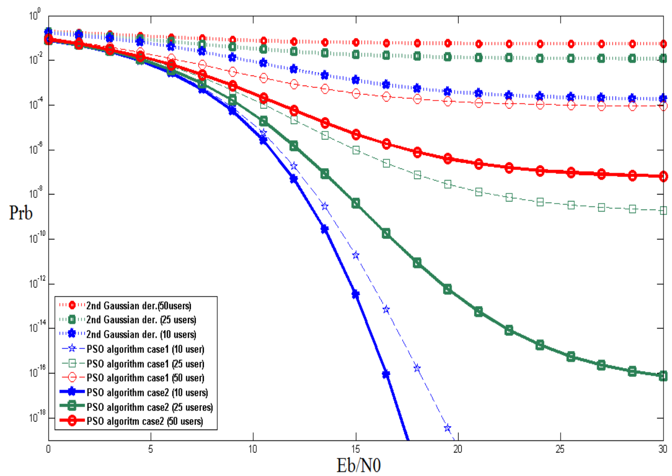

Figure 9. The $\Pr_b$ of PSO linear combined UWB pulse using e_1 .

transmission speed, N_u is the user number and $N_u - 1$ is the MUI number, $R_0(t)$ is a proportional coefficient and $\gamma_R \leq 1$. The simulation parameters are set as following: $R_b = 35$ Mbps, $\gamma_R = 1$, N_u = 10, 25 and 50, T_M = 0.5e-9 and E_b/N_0 is changeable in range [0,30] dB. The simulation results compared the PSO combined UWB pulse (cases 1 and 2) and Gaussian doublet pulses are shown in Figure 9.

CONCLUSIONS

IR-UWB signals are characterized by an interesting feature; their spectral properties may be appropriately tuned by playing with a variety of parameters. In this paper, we analyzed the issue of tuning spectral properties of to-be-radiated IR-UWB signals to reference spectral patterns by pulse shaping. We showed that the impulsive nature of the carrier of IR allows translating the problem of matching a spectrum, into the problem of finding a best waveform match. We proposed a method for obtaining the approximating function based on constructing differential Gaussian pulse that made use of a linear combination of only three pulses of Gaussian derivatives. To find the best weight coefficient, we proposed to optimize its weight vector and shaping factor by particle swarm optimization technique. Results obtained by application of the proposed algorithm were presented for three case studies: Random combination, error minimizetion such as the Least Square Error (LSE) and the area between standard UWB emission mask and the PSD of the linear combination signal via particle swarm optimization algorithm. In each three cases, the set of base functions was formed by the first 15 derivatives of the Gaussian pulse.

The first case study refers to approximating FCC indoor UWB emission mask by using a pure trial-and-error method. In order to gain better weight coefficients, we optimized the weight vector and shaping factor by particle swarm optimization algorithm and a similar analysis was carried out in second and third case studies. A good spectrum matching was obtained, especially in the cases two and three.

This paper combined Gaussian derivatives linearly to implement UWB pulse designed at the same time proposed to optimize pulse shape factor and weight vector using PSO. We adopted first, fourth and fifth order Gaussian derivatives as basic functions, and constructed the PSO model pulse for optimizing pulse shaping and weight vector. The simulation results showed that, these two combination pulses meet FCC emission mask better in comparison with the random combination pulse or single derivative pulse which proved the UWB pulse design method based on PSO to be the best method.

Frequency shifting is required for pulses generated using Hermit polynomials for the designed pulses to meet the FCC spectral mask, and also, the spectrum of the pulses generated using Hermit polynomials of order 2 or higher contain multiple lobes of approximately equal amplitude requires the use of a bandpass filter. This filtering will result in a loss of usable signal energy which is another disadvantage of the Hermit polynomial method as compared to our pulse design algorithm and leads to a direct transmitter implementation since no multiplier is needed to frequently shift the frequency in order to move the pulse spectrum into the desired frequency mask. Experimental results show that obtained waveform can satisfies the constraints well, and it also shows the superiority of this method to other approaches. Therefore it can be considered as a potential scheme for the practical UWB systems.

ACKNOWLEDGMENT

This work was supported, in part, by a grant from the Iran Telecommunications Research Center.

REFERENCES

Win MZ, Scholtz RA (1998). Impulse radio: How it works. IEEE Commun. Lett., 2: pp. 36–38.

Win MZ, Scholtz RA (2000). Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. IEEE Trans. Commun., 48: 679–689.

Shan DM, Chen ZN, Wu XH (2005). Signal optimization for UWB radio systems. IEEE Trans. Antenn. Propag., 53 (7): 2178–2184.

HämäläinenM, . Iinatti,J, Oppermann I, Latva-aho M, Saloranta J, . Isola A (2006). Co-existence measurements between UMTS and UWB systems., doi:10.1049/ip-com:20050319. IEE Proc. Commun., 153(1): 153–158.

Lei J, Jianxin G, Ren C,, Yongsheng W (2006). A new method for interference mitigation with coexisting spectrum users for UWB pulse radio. IET International Conference on Wireless Mobile and Multimedia Networks Proceedings (ICWMMN 2006) doi:10.1049/cp:20061303. (CP525), p.138.

Federal Communication Commission (2002). Revision of part 15 of the commission’s rules regarding Ultra-Wideband transmission system. First Report and Order, ET Docket, FCC 02-48, February 14:1-153.

Parr B, Cho B., Wallace K, Ding Z (2003). A novel ultra wideband pulse design algorithm. IEEE Commun. Lett., 7: 219–221.

Michael LB, Ghavami M, Kohno R (2002). Multiple pulse generator for ultra-Wideband communication using Hermite polynomial based orthogonal pulses. In: Proc. IEEE Conference on Ultra Wideband Systems and Technologies, pp. 47–51.

Mc Clellan JH, Parks TW (1973). A unified approach to the design of optimum FIR linear-phase digital filters. IEEE Trans. Circuit Theory, 20: 697–701.

Luo X, Yang L, Giannakis GB (2003). Designing optimal pulse-shapers for UWB radios. J. Commun. New. 5(4): 344–353.

Corral CA, Sibecas S, Emami S, Stratis G (2002). Pulse spectrum optimization for ultra-wideband communication. In: Proc. 2002 IEEE Conference on Ultra Wideband Systems and Technologies (UWBST 2002), Baltimore, pp. 21–23.

Matsuo M, Kamada M, Habuchi H (2005). Design of UWB pulses based on B-splines. In: Proceedings of the IEEE International Symposium on Circuits and Systems, C3L-M2 (CD-ROM).

Ramfrez-Mireles F (2002). Signal design for ultra-wide-band communications in dense multi-path. IEEE Trans. Veh. Tech., 51: 1517–1521.

Han J,. Nguyen C (2002). A new ultra-wideband, ultra-short monocycle pulse generator with reduced ringing. IEEE Microwave Wireless Comp. Lett., 12(6):. 206–208.

Chen ZN, Wu XH, Li N, Yang HF, Chia MYW (2004). Considerations for source pulses and antennas in UWB radio systems. IEEE Trans. Antenn. Propag., 52: 1739–1748.

Liang QJ, Shi XY, Wei YF (2003). The waveform design of transmitted signal in UWB radar. Modern Radar, pp. 44–50.

Shan DM, Chen ZN, Wu XH (2005). Signal optimization for UWB radio systems. IEEE Trans. Antenn. Propag., 53 (7): 2178–2184.

Nakache YP, Molisch AF (2006). Spectral shaping of UWB signals for time-hopping impulse radio. IEEEJ. Select. Areas Commun., 24(4): 738–744.

Jayaweera SK (2005). Signal design for non-coherent PPM modulation with applications to UWB communications. IEEE Commun. Lett., 9 (5): 411–413.

Igor D, Ryuji K (2007). Design of the family of orthogonal and spectrally efficient UWB waveforms. IEEE J. Select. Topics Signal Process., 1(1): pp. 21-30.

Michael LB Ghavami M Kohno R (2001). Hermit Function Based Orthogonal Pulses for Ultra Wideband Communication. In Proc. WPMC’ 01.

James K, Russel E (1995). Particle swarm optimization. Proceeding of IEEE International Conference on Neural Networks, pp.1942-1948.

James Kennedy and Russel Eberhart (2001). Swarm Inteligence. Morgan Kaufmann Pub.

Kennedy J, Eberhart RC (1995). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks IV Piscataway: IEEE. pp. 1942–1948.

Heppner H,Grenander U (1990). A stochastic non-linear model for coordinated bird flocks. In S. Krasner (Ed.), The ubiquity of chaos Washington: AAAS, pp. 233–238.

Robinson J Sammi YR (2004). Particle swarm optimization in electromagnetic. IEEE Trans. Antenn. Propag., 52 (2): 397–407.

Shi Y, Eberhart, RC (1998b). A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway: IEEE, pp. 69–73.

Eberhart RC, Shi Y (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In Proceedings of the IEEE congress on evolutionary computation (CEC), San Diego, CA. Piscataway: IEEE, pp. 84–88.

Eberhart RC, Shi Y (2001). Tracking and optimizing dynamic systems with particle swarms. In Proceedings of the IEEE congress on evolutionary computation (CEC), Seoul, Korea. Piscataway: IEEE. pp. 94–100.

Zheng, YL, Ma, LH, Zhang, LY, Qian, JX (2003). On the convergence analysis and parameter selection in particle swarm optimization. In Proceedings of the IEEE international conference on machine learning and cybernetics, Piscataway: IEEE. pp. 1802–1807.

Kennedy J (1998). The behavior of particles. In V.W. Porto, N. Saravanan, D.Waagen and A. E. Eiben (Eds.), Lecture notes in computer science. Evolutionary programming VII: Proceedings of the 7th annual conference on evolutionary programming, San Diego, CA. Berlin: Springer, pp. 581–589.

Clerc M, Kennedy J (2002). The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput., 6(1): 58–73.

Zhang H, Aaron TG (2005). Biorthogonal PPM-THMA-UWB Communications. IEEE Tranactions on Wireless Communications, 4, p.3.

Scholtz RA (1993). Multiple Access with Time-Hopping Impulse Modulation. IEEE Military Conf., 2: pp. 447-450.