FPGA based Synthesize of PSO Algorithm and its Area-Performance Analysis

Bharat Lal Harijan, Farrukh Shaikh, Burhan Aslam Arain
Institute of Information and Communication Technologies
Mehran University of Engineering and Technology,
Jamshoro, Sindh, Pakistan

Tayab Din Memon
Associate Professor
Mehran University of Engineering and Technology,
Jamshoro, Sindh, Pakistan

Imtiaz Hussain Kalwar
Associate Professor
DHA Suffa University, Karachi, Sindh,
Pakistan

***Abstract*—Digital filters are the most significant part of signal processing that are used in enormous applications such as speech recognition, acoustic, adaptive equalization, and noise and interference reduction. It would be of great benefit to implement adaptive FIR filter because of self-optimization property, linearity and frequency stability. Designing FIR filter involves multi-modal optimization problems whereas conservative gradient optimization technique is not useful to design the filter. Hence, Particle Swarm Optimization (PSO) algorithm is more flexible and optimization technique based on population of particles in search space and alternative approach for linear phase FIR filter design. PSO improves the solution characteristic by giving a novel method for updating swarm's position and velocity vector. Set of optimized filter coefficients will be generated by PSO algorithm. In this paper, PSO based FIR Low pass filter is efficiently designed in MATLAB and further Xilinx System Generator tool is used to efficiently design, synthesize and implement FIR filter in FPGA using SPARTEN 3E kit. For an example specifications, output of PSO algorithm is obtained that is set of optimized coefficients whose response is approximating to the ideal response. Hence, functional verification of the proposed algorithm has been performed and the error between obtained filter and ideal filter is minimized successfully. This work demonstrates the effectiveness of the PSO algorithms in parallel processing environment as compared to the Remez Exchange algorithm.**

Keywords—Particle swarm optimization (PSO); Remez Exchange Algorithm; FPGA implementation; FIR filter

I. Introduction

Digital filters enable us to pass some frequencies unaltered, while totally blocking others. Generally digital filters consist of two types; finite Impulse response (FIR) and infinite Impulse Response (IIR). An exactly linear phase response can be generated by FIR Filter and no any phase distortion or noise present in the output signal which is required in wide verity of telecommunication applications i.e. echo cancellation, noise and interface reduction, speech or image encoding. Different techniques are accessible for design the FIR filter. Window method is most frequently used tool but this method is not much capable to efficiently control the of frequency response in several bands of frequency [1]. Remez Exchange Algorithm or Parks–McClellan (PM) algorithm is ordinary method for designing FIR filter but this method has some limitation of high pass band ripples and computational complexity [2], [3]. It is good to design filter using optimization algorithm because of less mean squire error between desired response and actual response [3]. Optimization is not new techniques while numerous efforts have been already made for optimum design. Like Genetic Algorithm [3], Particle Swam Optimization algorithm [4], Differential Evolution [5], Artificial Bee Colony [6] are implemented for filer design. These methods showed themselves fairly effective by providing better control of performance constraints in addition to high stopband attenuation. Genetic Algorithms gives the effective result for local optimum but not successful in fining global optimum, PSO technique is able to solve problem [7]. Software based PSO algorithm increases the run time because of iterative process, additional processing time and storage is needed for FIR filter implementation [8]. PSO gives the better solutions over GA, processing time of one iteration of PSO algorithm gives higher process speed for optimization problems rather than genetic algorithm [8]. Implementation of digital filers based on FPGA which is flexible, low power, low cast and area sufficient provide better performance and superior to traditional approach [9].

Designing FIR low pass filter using traditional methods require more coefficients if sharp cutoff or no phase distortion is required and actual response $H(\Omega)$ is not more approximating to desired frequency response $Hd(\Omega)$ within a given specification in magnitude and phase [11], [12]. In recent past, one of the alternatives to this approach reported is short word length DSP systems [13], [14] in which sigma-delta modulation is a key element. However, in this research paper we have attempted to present the PSO based FIR filter designed in MATLAB; output of PSO is set of optimized coefficients whose response is approximating to the ideal response. Main objective is to efficient design, synthesize and functional verification of the optimized and original FIR low pass filter using Xilinx System Generator and implement in FPGA through hardware co-simulation, and to perform comparative analysis between both. This work will culminate with development of single-bit ternary PSO algorithm.

II. FIR Filter Design in Matlab using PSO Algorithm

In this section, PSO based FIR Low pass filer design and its implementation in MATLAB is discussed.

A. FIR Low Pass Filter Design

FIR filters are non-recursive filters and only depends upon past input information never on past output information [10]. Designed filter frequency response is given as:

$$H_d(e^{jw}) = \sum_{n=0}^{N} h[n]\, e^{-jwn} \quad (1)$$

where h[n] shows filter's impulse response, N is order is filter with N+1 length. Ideal response of Low Pass filter is defined as;

$$H_i(e^{jw}) = \begin{cases} 1, & for\ 0 \le w \le w_c \\ x, & otherwise \end{cases} \quad (2)$$

where w_c shows cutoff frequency of LP filter. Here obtained filter is designed by Remez Exchange Algorithm, this algorithm provides so many ripples in stop band, for sharp cutoff more coefficients are required. PSO algorithm is used to overcome this problem by minimizing the error between Remez and Ideal filter. The error equation is;

$$E(w) = [H_i(e^{jw}) - H_d(e^{jw})] \quad (3)$$

$H_i(e^{jw})$ is the ideal frequency response and $H_d(e^{jw})$ is frequency response of the approximate filter.

B. Particle Swarm Optimization (PSO) Algorithm

PSO is the optimization technique used to determine the search space for specified problem to find the setting or constraint that essential to maximize the specific object [15]. This global optimization technique was, first introduced by J. Kennedy and R. C. Eberhart in 1995, based on common behavior of fish schooling or bird flocking [16]. PSO algorithm can solve optimization-based problems, in this research PSO is used to optimize the FIR filter coefficients to minimize the error. PSO algorithm is iterative process and initialized with population consist of N particles and every particle initialized to random position. For each iteration the error fitness function is used to measure the fitness value of each particle *i* in the search space. Then velocity vector is calculated which influenced by the particles individual experience as well as the experience of its neighbors. Velocity vector is further used to update the particles position which defines the filter coefficients. The velocity update equation for particle is given as:

$$v_i^{t+1} = w.v_i^t + c_1.r_1(p_i^t - x_i^t) + c_2.r_2(p_g^t - x_i^t) \quad (4)$$

And position updating equation is:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5)$$

Superscripts t and t+1 represent the index of preceding and subsequent iterations, w is inertia coefficient, r_1 and r_2 are considered as uniformly distributed random numbers, c_1 and c_2 is cognitive acceleration term and social acceleration term and p_i and p_g are particle best position and swarm best position.

C. Designing Steps

Step I. In the very first step, specifies parameters that are required for designing FIR LP filter; Frequency of Sampling = 1kHz, W_{asps}=0.25, W_{stop}=0.3, Passband ripples= 0.1 and Stopband ripples = 0.01, filter order = 10 (Total no. of coefficients = 11).

Step II. Initialize Swarm size (Particles) = 250, w = 0.65, $c_1 = c_2$ = 2.05, Dimensions (No. of coefficients) D = 11, and maximum iteration *itmax* =100.

Step III. Create the initial particle vectors by utilizing above parameters and calculate initial value of error fitness function for the entire population by using (3).

Step VI. Error fitness vector is being used to calculate the minimum error value and calculate *pbest* (individual best) and *gbest (*group best) from entire swarm.

Step V. Update velocity and the position (filter coefficients) according to (4) & (5), which is to be considered as particle initial vector, error fitness is calculated form updated parameters also *pbest* and *gbest* is calculated accordingly.

Step VI. If values of vector *pbest* and *gbest* considered in Step V are improved than those calculated in Step IV, replaced the vector and no change otherwise.

Step VII. Repeat continuously from Step IV to Step VI till convergence conditions is meet (error fitness value equals minimum error fitness or reaching ***itmax***).

In Fig. 1, frequency response of PSO, Ideal and Remez algorithm-based FIR LP filters, for swam size N = 250 and *itmx*=100 is shown. By increasing swarm size, ripples in stop band are reduced at great extent and with increasing the iteration PSO algorithm gives the sharp cutoff at the cost of more chip area and performance degradation.

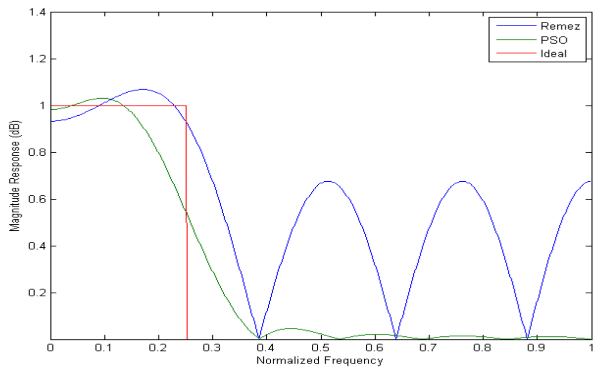

Fig. 1. Frequency response of PSO, ideal and Remez algorithm-based FIR low pass filter.

TABLE I. ORIGINAL COEFFICIENTS AND PSO BASED OPTIMIZED COEFFICIENTS

H(n)	**Original Coefficients**	**PSO Coefficients**
H(1)	0.0539051351555210	-0.0299519908221535
H(2)	-0.267487017557134	-0.0573261235259919
H(3)	0.100504853383030	-0.0536428075595944
H(4)	0.198866656248658	-0.00336950466085290
H(5)	0.247575005806795	0.0847531392346484
H(6)	0.265721847550549	0.179660912255896
H(7)	0.247575005806795	0.243609681976491
H(8)	0.198866656248658	0.251158067844883
H(9)	0.100504853383030	0.201996549974828
H(10)	-0.267487017557134	0.121085694673855
H(11)	0.0539051351555210	0.0444379358512675

III. FIR LP FILTER DESIGN UISNG XILINX SYSTEM GENERATOR

Xilinx System Generator is a programing tool used to develop efficient DSP algorithm and implement on FPGA. Due to reprogrammable capability of FPGA, implemented filter coefficients can be changed easily as per requirement [17]. System generator block set is available in MATLAB Simulink and it is high level programing tool for developing high performance DSP systems in FPGA [18-19]. System Generator enables the user to integrate with Simulink and it can easily generate synthesizable VHDL and Verilog code.

In this work, initially FIR Low pass filter is designed using FDA tool for the specification given outlined in designing steps. Further, PSO algorithm is applied on obtained coefficients and output of PSO is optimized coefficients whose response is approximate to ideal response. Xilinx system generator 14.7 is used for efficient direct form-I FIR low pass filter design and implemented in Spartan 3E FPGA kit through co-simulation.

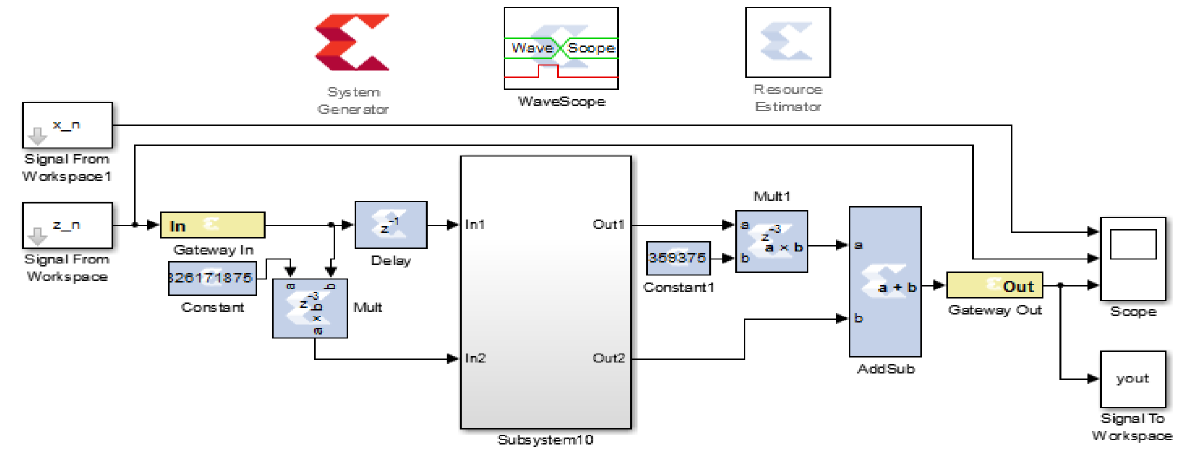

Fig. 2. Simulation model of direct form I FIR low pass filter.

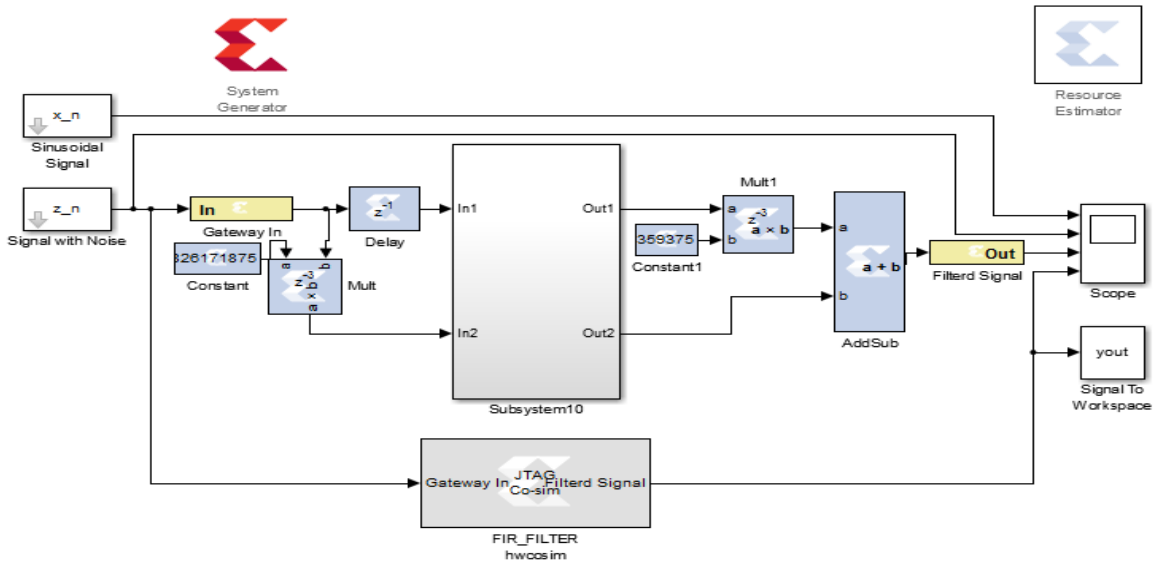

Fig. 3. Hardware co-simulation model of direct form I FIR low pass filter.

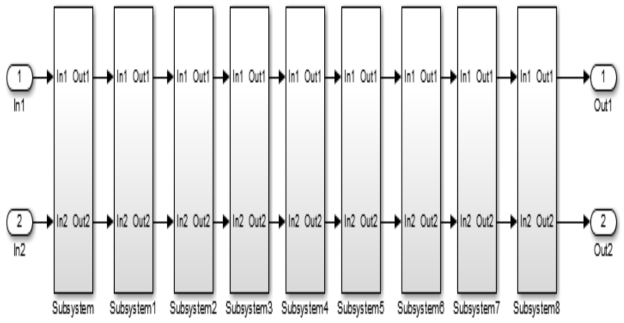

Fig. 4. Internal structure of direct Form I FIR filter model.

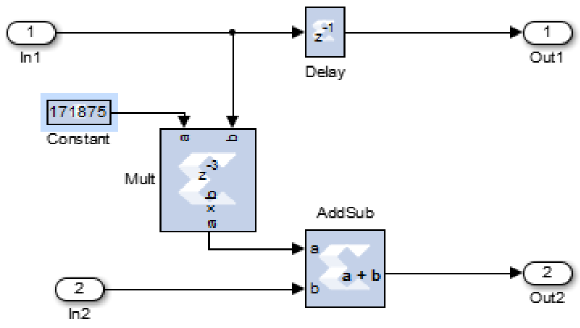

Fig. 5. Subsystem internal structure.

Simulation and Hardware Co-simulation model is shown in Fig. 2 and 3. JTAG cable shown in Fig. 2 is used for communication between Xilinx System Generator and Spartan 3E FPGA kit. System generator block set generate the of JTAG block of compatible signal for Spartan 3E kit. Resource Estimator is used to calculate the resources used by the device. It is used only when hardware is connected. Fig. 4 and 5 shows subsystem and internal structure of FIR Filter.

IV. Simulaiton Results and Discussion

Sinusoidal test signal of 125Hz frequency is generated in MATALB workspace as shown in Fig. 6 and White Gaussian Noise is added to original signal with Signal to Noise Ratio (SNR=1). Noisy signal is used as input of FIR filter. In first place, we used FIR filter model as shown in Fig. 2 with original filter coefficients, output signal of original filter is shown in Fig. 7 which contains more noise present in input signal. In Fig. 8, better output response is obtained, while we have used same model as shown in Fig. 2 but PSO optimized coefficients are employed as shown in Table I. This output signal is also taken to workspace in order to draw the spectrum as shown in Fig. 9, 10 and 11. The mean square error is computed using (3), and obtained Error = 0.3447933 when filter designed by Remez Exchange Algorithm and Error = 0.06442020 while filter designed by PSO algorithm. Area utilization is also observed using Spartan 3E kit, Table II shows area utilization of FIR filter using Spartan 3E FPGA kit.

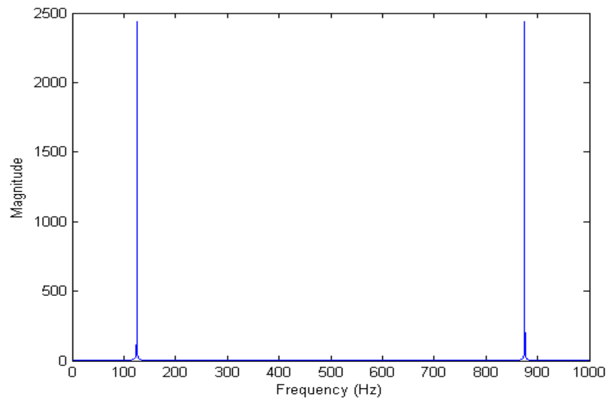

Fig. 6. Frequency spectrum of input signal.

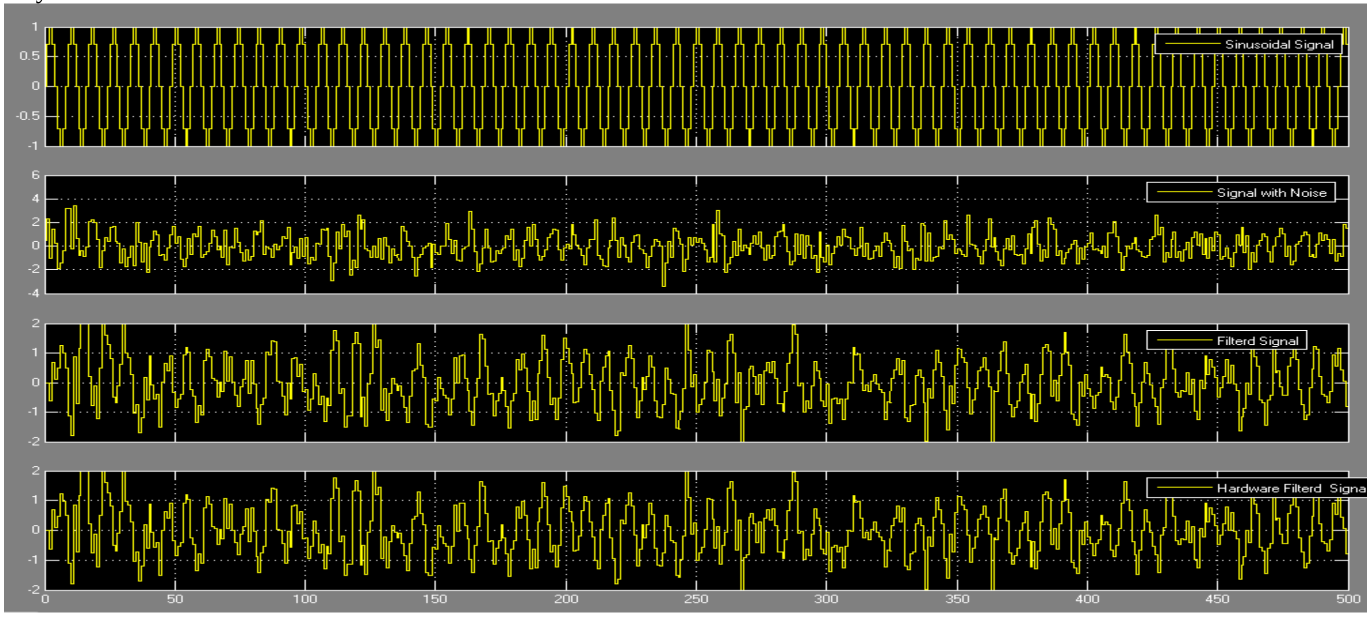

Fig. 7. Software simulation and hardware filtered output signal with original filte.

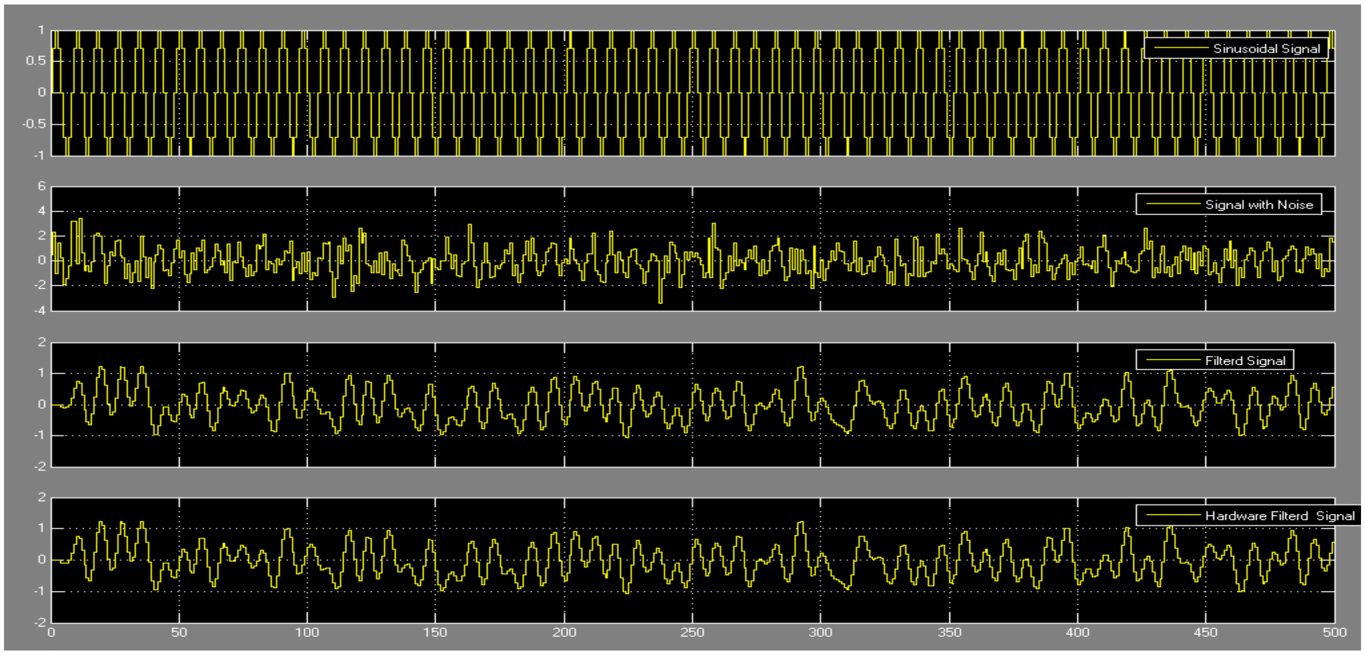

Fig. 8. Software simulation and hardware filtered output signal with optimized filter.

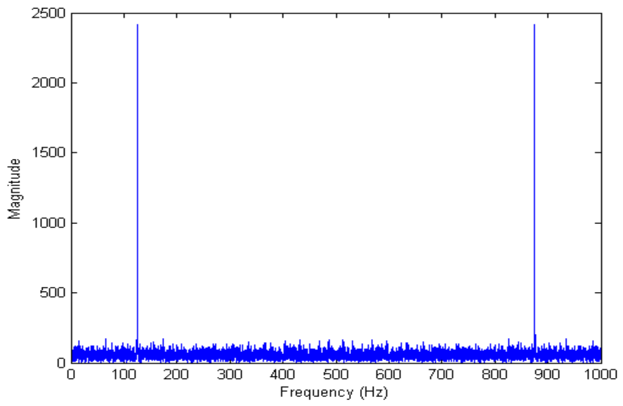

Fig. 9. Frequency spectrum of noisy signal.

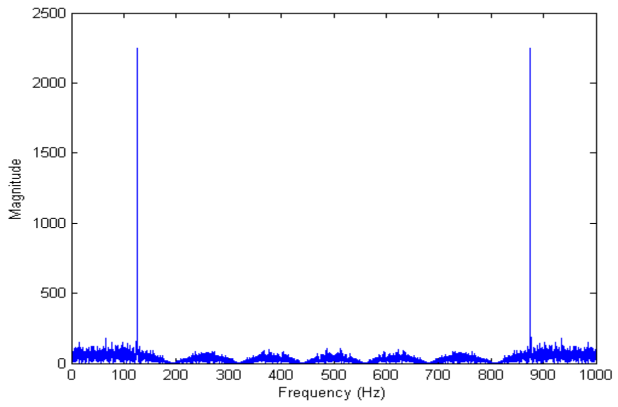

Fig. 10. Frequency spectrum of output signal of original filter.

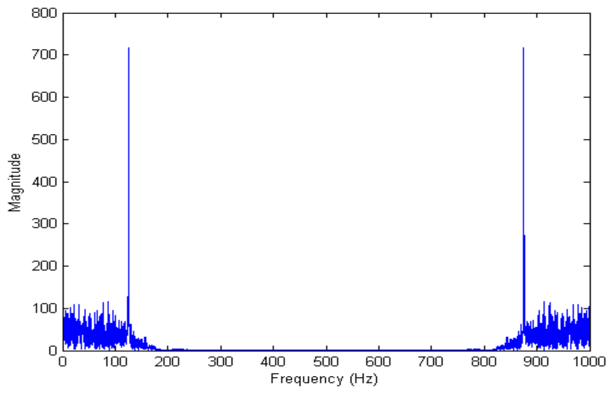

Fig. 11. Frequency spectrum of output signal of optimized filter.

In Fig. 9, frequency spectrum of noisy signal contains 125Hz original signal frequency and SNR=1 is shown. Whereas, Fig. 10 shows the output of original filter which shows the noise is present in the filtered signal and Fig. 11 shows the output of PSO based filter which contains less noise as compared to original filter. The area utilization by the PSO algorithm in FPGA given in Table II is quite small amount as compared to the available resources of the device.

TABLE II. AREA UTILIZATION OF FIR FILTER USING SPARTAN 3E KIT

Logic Unitization	Used	Available	Utilization
Number of Slices	82	4656	1%
Number of Slice Flip Flops	161	9312	1%
Number of IOBs	60	232	25%
Number of GCLKs	1	24	4%

V. Conclusion

In this paper, we have designed PSO based FIR filter in MATLAB that is further efficiently designed and synthesized using Xilinx System Generator in FPGA. Functional verification of the Remez Exchange Algorithm and PSO Algorithm based FIR low pass filter is performed through hardware co-simulation in Spartan 3E FPGA device. It is demonstrated that error in the PSO algorithm is successfully minimized. Area utilization of the PSO algorithm is also reported that is well below the available resources that shows much more room is available for improvement in the algorithm by increasing order of the filter.

Point to the future work is to compare this algorithm with other recursive algorithm and finally develop single-bit ternary FIR-like filter by employing these techniques.

References

[1] B. Luitel, G. K. Venayagamoorthy, "Differential Evolution Particle Swarm Optimization for Digital Filter Design," in IEEE Cong. on Evolution Comput., pp. 3954-3961, 2008.

[2] F. Shaikh, T.D. Memon and I. H. Kalwar, "Design and Analysis of Linear Phase FIR Filter in Fpga using PSO Algorithm," in 6th Mediterranean Conf. on Embd. Comput., Montenegro, 2017.

[3] K. Pardeep and S. Kaur, "Optimization of FIR Filters Design using Genetic Algorithm," Int. J. of Emerg. Trends and Techno. in Comput. Sci., vol. 1, no. 3, 2012.

[4] Neha and A. P. Singh, "Design of Linear Phase low pass FIR Filter using Particle Swarm Optimization Algorithm," Int. J. of Comput. Appl., vol. 98, no.3, pp. 0975–8887, 2014.

[5] Wei Zhong, "Linear phase FIR Digital Filter Design using Differential Evolution Algorithms," M.S. thesis, Dept. of Elect. & Comput. Eng., University of Windsor, Ontario, Canada, 2016.

[6] A. K. Dwivedi, S. Ghosh and N. D. Londhe, "Modified Artificial Bee Colony Optimization-Based FIR Filter Design with Experimental Validation using fpga," Inst. of Elect. and Techno. Signal Processing J., 2017: Available doi: 10.1049/iet-spr.2015.0214.

[7] A. Praneeth and P. K. Shah, "Design of FIR Filter using Particle Swarm Optimization," Int. Adv. Research. J. in Sci, Eng. and Techno., vol. 3, no. 5, 2016.

[8] B. A. Mohamed sadek and SAKLY Anis, "FPGA Implementation of Parallel Particle Swarm Optimization Algorithm and Compared with Genetic Algorithm," Int. J. of Adv. Comput. Sci. and Appl., vol. 7, no. 8, 2016.

[9] R. Thakur and K. Khare, "High speed FPGA Implementation of FIR filter for DSP Applications," Int. J. of Mod. and Opt., vol. 3, no. 1, 2013.

[10] L.Tan and J. Jiang, "Finite impulse response filter design," Digital Signal Processing Fundamentals and Applications, 2nd ed. pp 217-290, ELSVIER.

[11] P. M. Palangpour, "Fpga Implementation of PSO Algorithm and Neural Networks," M.S. thesis, Missouri University of Science and Technology, 2010.

[12] P. Fodisch, A. Bryksa, B. Lange, W. Enghardt and P. Kaever, "Implementing high order FIR filters in FPGAs," 2016: Available arXiv:1610.03360v2.

[13] A. Chang, T. D. Memon, Z. M. Hussain, I. H. Kalwar, and B. S. Chowdhry, "Design and Analysis of Single-Bit Ternary Matched Filter," Wireless Personal Communications, pp. 1-15, 2018.

[14] Tayab D Memon, P. Beckett, and A. Z. Sadik, "Power-Area-Performance Characteristics of FPGA based sigma-delta modulated FIR Filters," Journal of Signal Processing Systems (JSPS) vol. 70, pp. 275-288, 2013.

[15] J. Blondin, "Particle Swarm Optimization," Tutorial, 2009.

[16] J. Kennedy and R. Eberhart, "Particle Swarm Optimization" in Proceedings of the IEEE Int. Conf. on Neural Networks, vol. 4, pp 1942–1948, 1995.

[17] S. Roy, L. Srivani and D. T. Murthy, "Digital Filter Design Using FPGA," Int. J. of Eng. and Innovat. Techno., vol. 5, no. 4, 2015.

[18] K. Sahu and R. Sinha, "FIR filter Designing using Matlab Simulink and Xilinx System Generator," Int. Res. J. of Eng. and Techno., vol. 2 no. 8, 2015.

[19] System Generator for DSP user guide, Xilinx UG640 v. 14.3, 2012.