ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 10(36), DOI: 10.17485/ijst/2017/v10i36/119177, September 2017* ISSN (Online) : 0974-5645

Design of IIR Filter Using PSO Algorithm and its Implementation in FPGA

Umema Tasleem Shaikh[1], Imtiaz Hussain Kalwar[2], Tayab Din Memon[1] and Faiza Shaikh[1]

[1]Mehran University of Engineering and Technology, Indus Hwy, Jamshoro 76062, Pakistan; umematasleem@hotmail.com, tayabdin82@gmail.com, shaikhfaiza30@gmail.com
[2]DHA Suffa University, Imtiaz.hussain@faculty.muet.edu.pk

Abstract

In this paper, Particle Swarm Optimization (PSO) algorithm based Adaptive Infinite Impulse Response (IIR) filter is designed. PSO algorithm is inspired by bird or fish swarm movement. PSO as global optimization technique provides advantages of simple implementation, quickly convergence providing better results and robustness for avoiding local minima problem. The designed PSO based Adaptive filter is then implemented to work as a noise canceller. Simulation results shows that the designed filter is effective with good noise suppression. The PSO based filter is also synthesized in FPGA and run fittings in small commercial devices Stratix-II and cyclone-II

Keywords: Adaptive Infinite Impulse Response (IIR) Filter, FPGA implementation, Particle Swarm Optimization, VHDL

1. Introduction

Filter is usually used to retrieve the desired information and avoid all the undesired information such as noise, which is created due to inevitable conditions of the environment, from the input signal. Digital filters are popular to reduce the undesired noise as it provides reliability and high accuracy, reduced sensitivity and small physical size to component tolerances as compared to analog filters. Digital filters are now favorable for applications like wideband image processing systems and digital communication systems having high data rate. Digital filters are generally classified into two main types namely; Finite Impulse Response (FIR) filter and Infinite Impulse Response (IIR) filter. IIR digital filters offer computational efficiency, reduced system delay and improved selectivity as compared to FIR digital filters with comparable approximated accuracy. However, the digital IIR filters lacks in efficiency, practical applicability and global optimization robustness. To overcome these in-capabilities PSO offers viable digital IIR filter designing tool because of its simple implementation and may requires few parameters to control its convergence.

Kennedy and Eberhart developed the evolutionary computation technique of PSO algorithm in 1995[1]. PSO as global optimization technique offers advantages of robustness, fast convergence rate and simplicity in implementation. The PSO algorithm is a social-psychological based adaptive algorithm; an individual from population adjusts by returning random value towards formally successful regions. Velocity and position updates are the two primary operators of Particle Swarm. It is basically an iterative process and during iteration, particle velocity is updated from current to new value, global best position and previous best position and the next position will be assigned from the new assigned velocity. This process will be iterated until the error will be minimized.

Field Programmable Gate Arrays (FPGAs) are becoming popular for digital circuits implementation as dedicated hardwired realization of digital filters is essential for high-speed and small area digital filtering applications and FPGA has aided them to practically realize the adaptive filter algorithm. The PSO algorithm

*Author for correspondence

mapping on FPGA will be straight-forward as it will be composed of following three modules: PSO module, adaptive filter module and unknown IIR filter module.

2. IIR Filter Design

Following are the five steps for designing a digital IIR filter using PSO Algorithm:

2.1 Filter Specification

In this step, for designing low pass IIR filter following specifications are used: Sampling Frequency Fs=1000Hz, cutoff frequency=0.3, stop and ripple=0.01 and pass band ripple=0.1.

2.2 Coefficient Calculation

At this step, the reference coefficients will be determined which will act as global best coefficient for the PSO algorithm implementation. For this, Butterworth filter coefficient can be used which follows the above specification

2.3 Fitness Function Generation

PSO algorithm basically minimized the error by using fitness function. In fitness function, error between ideal and a designed filter is taken. As mentioned earlier, for IIR Butterworth filter is used. Fitness function is defined as:

$$Fitness = \frac{1}{N}\sum_{k+1}^{N}(ideal(k) - actual(k)) \qquad (1)$$

Whereas, the ideal response is defined as:

$$ideal(k) = \qquad (2)$$

Where, ideal (K) and actual (K) are the magnitude response of the ideal and the actual filter, where N is the number of samples[2].

2.4 PSO Algorithm Implementation

Following are the steps for implementing PSO algorithm:

2.4.1 Initial Particles

Randomly generate initial population matrix of order (i x j).

2.4.2 Evaluate Particles

The fitness value for each set of particles in the population is need to be calculated using Equation1.

2.4.3 Update Global Best

Select the global best particles which have the minimum fitness value.

2.4.4 Update Personal Best

By comparing each particle's newly calculated fitness value, with the previous value, select particle's personal best.

2.4.5 Update Velocity of the Particles

Using the following equation, update each particle's velocity:

$$V_{ij}^{k+1} = w\,(V_{ij}^{k} + c_1 r_1^{k}\left(pbest_{ij}^{k} - x_i^{k}\right) + c_2 r_2^{k}(gbest_{ij}^{k} - x_i^{k})) \qquad (3)$$

2.4.6 Update Position of the Particle

Using the following equation, update the each particle's position:

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \qquad (4)$$

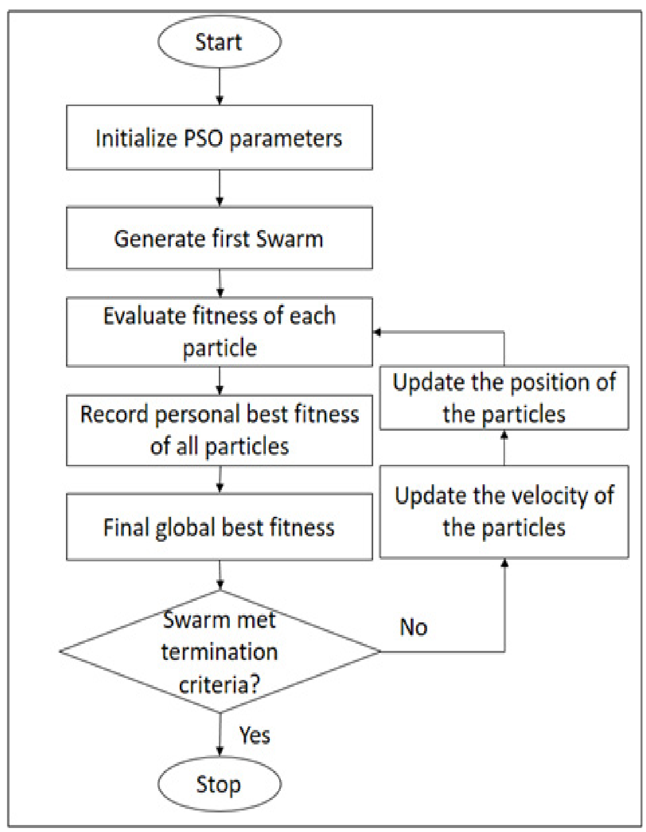

Figure 1. PSO algorithm flow chart.

In the above equations, x_{ij}^{k} and v_{ij}^{k} represents the position and the speed of the particle "i" at its k^{th} times having the j^{th} dimension; $pbest_{ij}^{k}$ represents the particle's optimist position having the j^{th} dimension quantity of individual "i" at its k^{th} times. $gbest_{ij}^{k}$ is swarm's optimist position. Speed limits $-V_{jmax}$, and V_{jmax} are used to keepparticle within the desired searching space. c_1 and c_2 are the cognitive acceleration and social acceleration coefficients, which aid in deciding the length flying of the particle to select optimal position of the particle. r_1 and r_2 represents random number having value between 0-1[3].

2.4.7 Termination

Unless the maximum number of iteration is reached repeat the whole process from second step.

PSO algorithm flow chart is shown below in Figure 1.

3. Simulation and Discussion

IIR filter is implemented using following specification. For initial global best filter coefficients, Butterworth filter coefficients is used having Sampling Frequency Fs= 1000Hz, cutoff frequency= 0.3, stop b and ripple= 0.01, pass b and ripple= 0.1. Initial global best particle=Butterworth filter coefficients (for reference), swarm size =50, cognitive constant = 2.05, Society constant= 2.05, inertia weight = 0.4, max number of iterations = 125, number of cascaded IIR filter coefficients= 14 (for order 6).

3.1 MATLAB Simulation

The PSO algorithm based 6^{th} order low pass IIR filter is simulated using MATLAB R2014b and obtained following coefficients (Table 1).

The plot of PSO based IIR filter using above specification obtained in Matlab as (Figure 2):

Passband: $0 \le \omega \le 0.27\pi$

Stopband: $0.3 \le \omega \le \pi$

Pass-band Ripple Magnitude:

$0.9945 \le |H(e^{j\omega})| \le 1.013$

(0.0185)

Stop-band Ripple magnitude:

$|H(e^{j\omega})| \le 0.0957$

(0.0957)

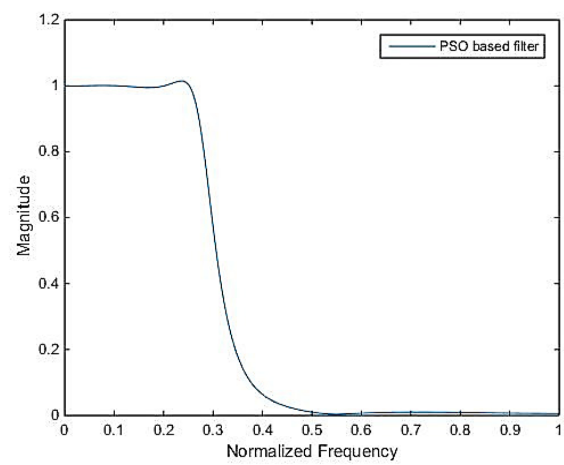

Figure 2. 6th order IIR Filter frequency response.

Table 1. PSO Optimized Coefficients: (order: 6)

h(N) Nume-rator Co-efficients	PSO	Butter-worth	h(N) Denomi-nator Co-efficient	PSO	Butter-worth
hn(0)	0.03201	0.00258	hn(0)	1.05193	1
hn(1)	0.02841	0.01551	hn(1)	-2.43369	-2.37972
hn(2)	0.04848	0.038775	hn(2)	3.02459	2.910406
hn(3)	0.05506	0.051701	hn(3)	-2.06931	-2.055131
hn(4)	0.04854	0.038775	hn(4)	0.88002	0.877923
hn(5)	0.02835	0.01551	hn(5)	-0.21204	-0.20986
hn(6)	0.0322	0.002585	hn(6)	0.03284	0.021831

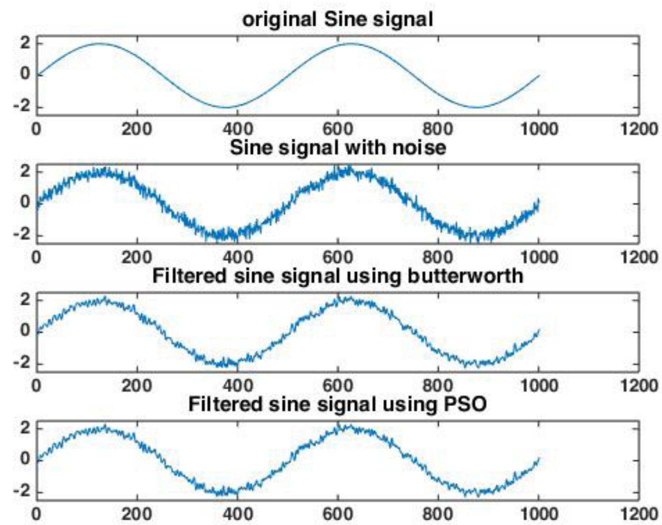

Figure 3. IIR filter implementation on a sinusoidal signal.

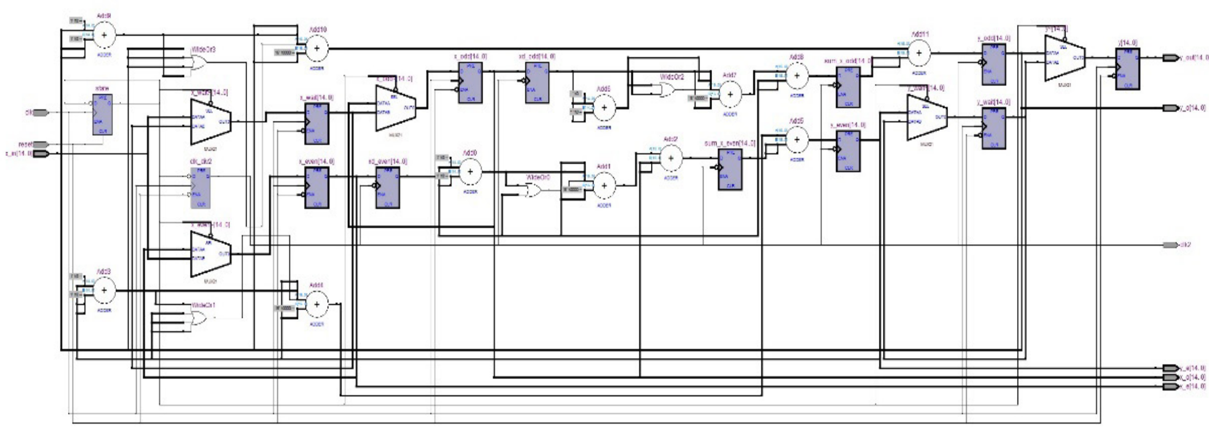

Figure 4. RTL view of IIR filter in FPGA.

Figure 3 shows the implementation of designed filter on a sinusoidal signal.

3.2 FPGA Implementation

The PSO based Low pass IIR filter is simulated to VHDL and realized on various FPGAs using Quartus-II 9.1. The purposed filter's RTL view is shown in Figure 4. A test bench is also generated. The results show that the output of filter was available after 185ns delay while the initially 1000ns time limit was selected for simulation. Hardware utilization of different devices like, Stratix-II and Cyclone-II, has also been compared for the algorithm. The completed analyses were based on 6 input LUTs utilization. The logic utilization for PSO based IIR Filter is shown in Table 2.

Table 2. Results of area-performance of two devices in FPGA

Device	N	Logic Utilization	LUTs	Clock
Cyclone II	6	3722	7631	111MHz
STRATIX II	6	4138	8113	139MHz

4. Conclusion and Future Work

Swarm intelligence based PSO is a new exploratory optimization method, which can be implemented on vast variety of applications. In comparison with other, PSO algorithm is very simple and only requires few parameters to make fully functionally developed application. Here PSO algorithm is used for designing IIR filter in MATLAB and synthesized in FPGA's small commercial devices cyclone-II and Stratix-II. The results show that, Stratix-II shows better performance while Cyclone-II provides better chip area perspective. So as per application, this tradeoff may be used. As far as future work is concern, we can compare the PSO based filter results with GA based filter and some other optimization technique and implement this designed filter on real time applications.

5. References

1. Krusienski DJ, Jenkins WK. Particle swarm optimization for adaptive IIR filter structures, IEEE Congress on Evolutionary Computation, CEC 2004. 2004; 1. Crossref.
2. Ranjit Kaur, Damanpreet Singh. Particle swarm optimization algorithm for designing optimal IIR digital filter, International Journal of Emerging Technologies in Computational and Applied Sciences –IJETCAS; 2014. p. 225–30.
3. Qinghai Bai. Analysis of particle swarm optimization algorithm, Computer and Information Science. 2010; 3(1). DOI: Crossref.
4. Sharbari Banerjee, Amitabha Sinha. Performance analysis of different DSP algorithms on advanced microcontroller and FPGA, ACTEA 2009, IEEE International Conference; July 15 2009. DOI: 10.1109/ACTEA.2009.5227848
5. Sheng Chen, Bing L. Luk. Digital IIR filter design using particle swarm optimization, Int. J. Modelling, Identification and Control. 2010; 9(4). Crossref.
6. Neha Gupta, Mohammad Junaid Khan. Analysis of the IIR filter design using particle swarm optimization method, International Journal of Scientific Research in Recent Sciences. IJSRRS, 2015; 1(1):34–38.
7. Tim Hendtlass. Fitness estimation and the particle swarm optimisation algorithm, IEEE Congress on Evolutionary Computation; 2007. https://doi.org/10.1109/CEC.2007.4425028.
8. Dean J Krusienski, Kenneth Jenkins. The application of particle swarm optimization to adaptive IIR phase equalization. Conference Paper in Acoustics, Speech, and Signal Processing; 1988. DOI: Crossref ICASSP-88., 1988 International Conference on June; 2004.
9. Lipika Gupta, Rajesh Mehra. Modified PSO based adaptive IIR filter design for system identification on FPGA, International Journal of Computer Applications (0975 – 8887). May 2011; 22(5).
10. Ranjit Singh, Sandeep K Arya. Application of particle swarm optimization for the design of IIR digital filters, Int. J. on Recent Trends in Engineering and Technology. Mar 2012; 7(2).