

# Performance Investigation of Digital Lowpass IIR Filter Based on Different Platforms

Case Study

## Raaed Faleh Hassan

Middle Technical University, College of Electrical Engineering Technology,  
Department of control and automation eng. Techniques, Iraq  
dr-raaed\_alanbaki@mtu.edu.iq

**Abstract** – The work presented in this paper illuminates the design and simulation of a recursive or Infinite Impulse Response (IIR) filter. The proposed design algorithm employs the Genetic Algorithm to determine the filter coefficients to satisfy the required performance. The effectiveness of different platforms on filter design and performance has been studied in this paper. Three different platforms are considered to implement and verify the designed filter's work through simulation. The first platform is the MATLAB/SIMULINK software package used to implement the Biquad form filter. This technique is the basis for the software implementation of the designed IIR filter. The HDL – Cosimulation technique is considered the second one; it inspired to take advantage of the existing tools in SIMULINK to convert the designed filter algorithm to the Very high-speed integrated circuit Hardware Description Language (VHDL) format. The System Generator is employed as the third technique, in which the designed filter is implemented as a hardware structure based on basic unit blocks provided by Xilinx System Generator. This technique facilitates the implementation of the designed filter in the FPGA target device. Simulation results show that the performance of the designed filter is remarkably reliable even with severe noise levels.

---

**Keywords** – Low pass IIR Filter, Genetic Algorithm, Biquad form, HDL-co simulation, XSG

---

## 1. INTRODUCTION

Digital filters have become essential for removing unwanted signals in recently outspread communication and control systems [1]. Digital filters are categorized into two classes according to their impulse response [2]. The first category is the Finite Impulse Response (FIR), while the Infinite Impulse Response (IIR) is the second one [2]. As its name, the FIR filter possesses a finite impulse response due to the absence of the feedback loop, and it is also known as a non – recursive filter. On the other hand, the IIR filter has a feedback loop; therefore, it works recursively. The digital filter design aims to achieve high attenuation at the stopband with a steep roll-off. An IIR filter satisfies these requirements and significantly reduces the computational process compared with the FIR filter [3]. In Very Large Scale Integrated (VLSI) realization, it is preferable to adopt an IIR filter rather than an FIR filter. The IIR filter's drawback is the weakness of instability problems that can overcome by using advanced techniques [4] – [7]. A word length arises as a significant situation that influences the filter behavior in implementing the digital filters. Therefore, a finite – word – size becomes a crucial parameter in the digital filter design issue. The principal idea beyond this approach is to engage a proper filter structure for each application [8]. The IIR filter design can be achieved based on two direc-

tions: the transformation and optimization methods [9]. The transformation method performed by transforming the designed analog filter to the digital filter at some given specifications is the bilinear process which is the most widely used transformation method. This method's developed digital filter mostly suffers from poor behavior due to a multimodal error surface and instability [10]. These problems can be avoided when the design parameters are bounded in some criteria allocated using optimization methods [10] – [19]. The most prevalent optimization method is the Genetic Algorithm (GA) which can search versatile spaces and optimizes challenging functions that are very complicated to investigate [1], [8]. Employing GA for the digital filter design is preferable because it can construct the filter in any form with the lowest order [8]. Recently, the most interesting parallel processing hardware devices for implementing digital filter algorithms are the field-programmable gate array (FPGA) [2], [3], [19] – [22]. FPGA is a VLSI logic chip that can configure for realizing widespread logic functions [23]. No matter how the logic function is involved, the FPGA structure's nature facilitates the function's implementation in parallel, leading to very high-speed computation. However, the hardware implementation of digital systems using FPGA is somewhat tricky and needs the training to deal with Hardware Description Language (HDL). Therefore, a virtual environment for de-

signing, implementing, and testing the digital systems using FPGA is needed [24].

Recently, Matlab/Simulink provides this virtual environment utilizing co-simulation with ModelSim and implementing a digital system using the Xilinx system generator. Employing co-simulation with Modelsim permits the verification of an HDL module of the system against its Simulink model. In this case, the Simulink environment will act as a test bench that stimulates an HDL module [25]. The most realistic visualization of digital system performance can be achieved using Xilinx System Generator (XSG). Adopting this platform enables designing a digital system in the Simulink environment using Xilinx-specific blocks in the Simulink library. XSG design can easily be placed on the FPGA target device as a bit file with the possibility of obtaining the HDL code for further optimization and integration with other projects within the Xilinx environment. This paper explores these platforms' features to implement and analyze the performance of the digital low pass IIR filter.

## 2. DIGITAL IIR FILTER:

According to its name, this type of filter has an infinite impulse response, as indicated in the following difference equation:

$$y_n = \sum_{k=0}^M b_k x_{n-k} + \sum_{k=1}^N a_k y_{n-k} \quad N \geq M \quad (1)$$

Where  $y_n$  is the current sample of the filter output,  $y_{n-k}$  are the  $k^{th}$  previous samples of the output,  $x_{n-k}$  are the  $k^{th}$  previous samples, including the current sample of the input signal,  $a_k$  and  $b_k$  are the filter coefficients,  $M$  is the numerator's order, and  $N$  is the denominator's order [1], [2]. From eq. (1), it can be seen that the filter output  $y_n$  is determined recursively; in other words, it is a function of the current and previous samples of the input signal and the output signal's previous samples. Taking the  $z$  - transform for eq. (1) leads to the following system function:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2)$$

### 2.1 IIR FILTER REALIZATION:

Several practical realization architectures that implement the digital IIR filter were outlined in the literature; these are:

**DFI:** The digital IIR filter can be realized in the direct form I according to eq. (1) which comprises a non - recursive part and recursive part. The implementation of the digital IIR filter in this form is shown in Figure (1); for each output sample, it requires  $(N + M)$  delays,  $(N+M+1)$  multiplications, and  $(N+M)$  additions.

**DFII:** In the direct form II structure, shared delay elements between the recursive and non - recursive parts

of the digital IIR filter are employed, as shown in Figure (2). The determination process of each output sample requires  $\max(N \text{ or } M)$  delays,  $(N+M)$  additions, and  $(N+M+1)$  multiplications [2].

The benefit of DFII is its more economical utilization of the delay units. However, both separate the poles sections from the zero sections, DFII can share the delay units between them. Therefore, a significant reduction in the number of delay units can be achieved when adopting DFII. The drawback of the DFII is that the pole units precede the zero units and imposes an unfeasible dynamic range on the delay units' intersection at some frequencies. As a result, DFII suffers from overflow, while DFI has immune from this effect [2]. The transposed structure of the DFII, shown in Figure (3), precedes the zero sections and shares the delay units between zeros and poles. Therefore, this structure has the advantage of DFII with more robust behavior [23].

### Cascade Structure:

For the higher-order IIR filters (higher than 2) with transfer function given by eq. (2) with  $N \geq M$ , the transfer function can be factorized into a cascaded second-order subsystem as expressed below:

$$H(z) = \prod_{k=1}^{\frac{N+1}{2}} H_k(z) \quad (3)$$

Where 
$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}} \quad (4)$$

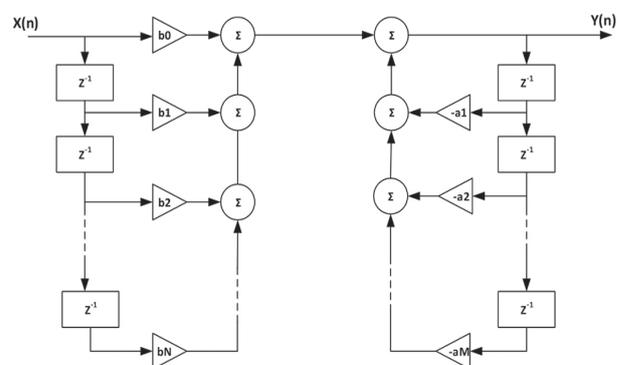


Fig. 1. DFI structure

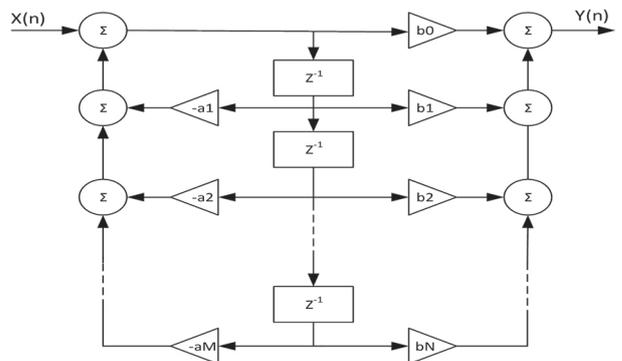
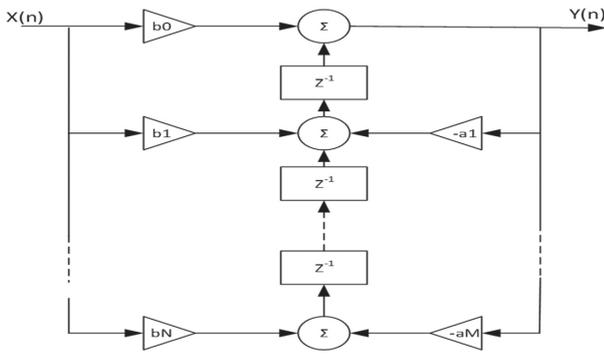
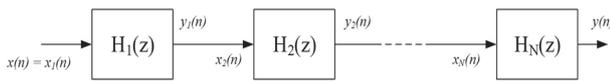


Fig. 2. DFII structure



**Fig. 3.** Transposed DFII Structure

The coefficients  $\{a_{ki}\}$  and  $\{b_{ki}\}$  in eq. (4) are real, which implies that any pair of real or complex conjugate poles or zeros are grouped in second-order or quadratic form. This structure reduces the effects of using finite word length representation of the filter coefficients [20]. The cascade structure can be implemented by a series-connected of first-order and/or second-order structures, as shown in Figure (4). Each of the 2<sup>nd</sup> order subsystems in eq. (3) can be implemented as DFI or DFII or transposed DFII.



**Fig. 4.** Cascade Structure

A parallel structure of an IIR filter can be synthesized based on a partial fraction expansion performed on  $H(z)$  in eq. (2). When  $N \geq M$  in eq. (2). Moreover, the poles are distinct, the partial – fraction expansion of  $H(z)$  produces the following results:

$$H(z) = C + \sum_{k=1}^N \frac{A_k}{1 - P_k z^{-1}} \quad (5)$$

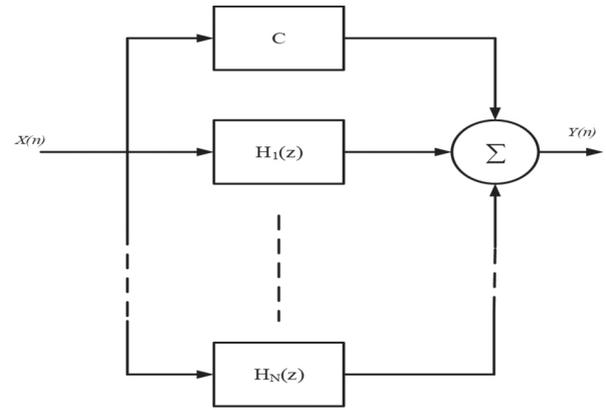
Where:  $P_k$  are the poles,  $A_k$  are coefficients, and  $C = b_N/a_N$ .

Figure (5) shows the block diagram representation of eq. (5). Generally,  $H(z)$  may contain some complex-valued poles. In this case, the produced coefficients  $A_k$  are also complex-valued. The avoidance of multiplication by complex numbers can be achieved by combining pairs of complex conjugate poles to form a 2<sup>nd</sup> order subsystem. Each of these subsystems has the form:

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}} \quad (6)$$

Where  $\{a_{ki}\}$  and  $\{b_{ki}\}$  coefficients are real-valued system parameters. The modified transfer function can now be expressed as

$$H(z) = C + \sum_{k=1}^{\frac{N+1}{2}} H_k(z) \quad (7)$$



**Fig. 5.** Parallel Structure

### Proposed Structure:

The design process's main task is to determine the filter coefficients ( $a_k$  &  $b_k$ ) to perform the desired performance while ensuring stability [8]. The most popular realization structure of the digital IIR filter is to stack some 1<sup>st</sup> order and (or) second-order in cascade. Therefore, Eq. (3) can be modified into the stacked of the cascaded second-order form (assuming  $M = N$ ):

$$H(z) = \prod_{k=1}^{N/2} G_k \frac{1 + c_{1k}z^{-1} + c_{2k}z^{-2}}{1 + d_{1k}z^{-1} + d_{2k}z^{-2}} \quad (8)$$

Let  $z = e^{j\Omega}$ , Eq. 8 becomes:

$$H(e^{j\Omega}) = \prod_{k=1}^{N/2} G_k \frac{1 + c_{1k}e^{-j\Omega} + c_{2k}e^{-2j\Omega}}{1 + d_{1k}e^{-j\Omega} + d_{2k}e^{-2j\Omega}} \quad (9)$$

The magnitude frequency response of the filter is determined as follows:

$$|H(e^{j\Omega})| = \left| \prod_{k=1}^{N/2} G_k \frac{1 + c_{1k}e^{-j\Omega} + c_{2k}e^{-2j\Omega}}{1 + d_{1k}e^{-j\Omega} + d_{2k}e^{-2j\Omega}} \right| \quad (10)$$

Furthermore, the phase-frequency response is determined as follows:

$$\angle H(e^{j\Omega}) = \tan^{-1} \left( \frac{\text{Im}(H(j\Omega))}{\text{Re}(H(j\Omega))} \right) \quad (11)$$

### 3. GENETIC ALGORITHM BASED IIR FILTER DESIGN:

Genetic Algorithm (GA) can be considered as the most popular and robust search algorithm in recent years [8] - [14]. It is established according to the theories of social evolution or natural selection. In these theories, natural selection can be regarded as an exploration procedure to evaluate the optimum DNA that maximizes a species' persist possibility to procreate, thereby spread the species. The optimum DNA was obtained during the DNA crossover through sexual reproduction with unqualified offspring's depletion due to low accommodation with the surroundings [22]. GA utilizes natural selection principles to accomplish the

task of evaluating the optimum solution of the prescribed problem. Initially, the population parameters are selected randomly to confirm that the exploration space is widely and evenly sampled. The fitness function is used to determine and choose the members of a population with reasonable solutions sufficient to produce the next generation. Similar to natural selection, the crossover process is adopted to perform the swapping between the sections of the randomly designated pairs. GA may be stuck in the suboptimal region, and the mutation is considered to permit the GA to skip outside this region. From many types of mutation, a swap mutation is employed in this work, in which two genes are selected randomly and interchange their positions. With predefined filter order, the objective of the IIR filter design is to determine its coefficients. When the GA is considered the filter design tool, the filter coefficients act as chromosomes that constitute the population. The fitness function can be derived from comparing the magnitude response given in Eq. (10) with the desired magnitude response.

For the low pass filter design issue, the chosen magnitude response is:

$$|H_d(e^{j\Omega})| = \begin{cases} 1 - \delta_p & \Omega \leq \Omega_p \\ \delta_s & \Omega \geq \Omega_s \end{cases} \quad (12)$$

Where:

$H_d(e^{j\Omega})$ : The desired magnitude response.

$\Omega_p, \Omega_s$ : The passband and stopband frequencies, respectively.

$\delta_p, \delta_s$ : The passband and stopband allowed ripples, respectively.

The error of the magnitude response is defined as follows:

$$E_p(\Omega) = |H_d(e^{j\Omega})| - |H(e^{j\Omega})| \quad (13)$$

The fitness function to be minimized is:

$$J_{min} = \sum_{\Omega} |E_p(\Omega)|^2 \quad (14)$$

The digital IIR filter considered in this paper will be a fourth-order low – pass filter, which has the following system function in cascade form:

$$H(z) = \prod_{k=1}^2 G_k \frac{1+c_{1k}z^{-1}+c_{2k}z^{-2}}{1+d_{1k}z^{-1}+d_{2k}z^{-2}} \quad (15)$$

Therefore, the population contains ten individuals; these are:  $[G_1 \ c_{11} \ c_{21} \ d_{11} \ d_{21} \ G_2 \ c_{12} \ c_{22} \ d_{12} \ d_{22}]$ . It must now be clear that to achieve IIR filter design, the GA is utilized to minimize the fitness function. The minimization process results in evaluating the best values of the individuals. Figure 6 shows the flow chart of the GA-based fourth-order low pass IIR filter design.

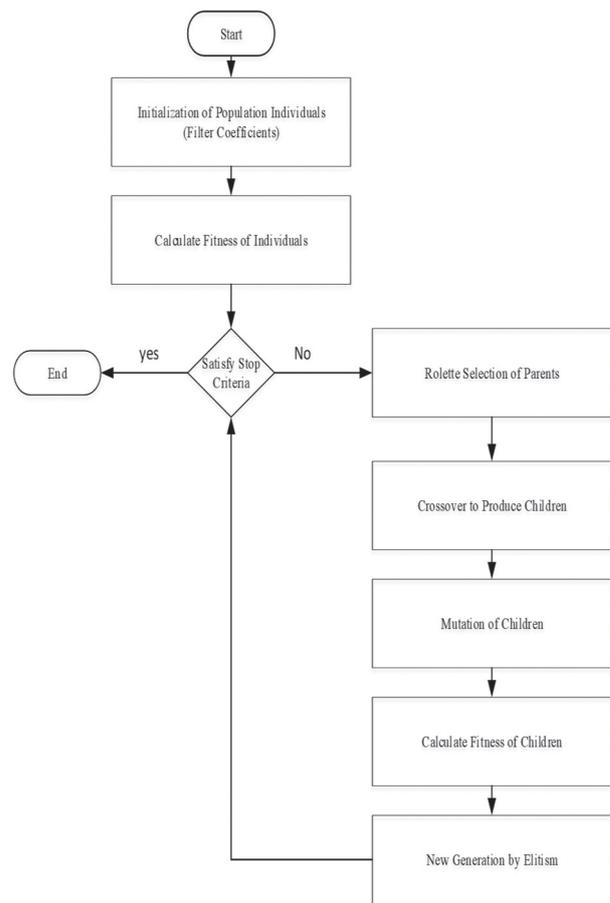


Fig. 6. GA flow chart

The GA optimization issue was performed using (optimtool) in the MATLAB software package. Table 1 shows the GA parameters setting

Table 1. GA parameters

Genetic Algorithm Parameters	
No. of variables	10
Maximum $\delta_p$ and $\delta_s$	0.1
Generation	2000
Population size	200
Crossover Function	Two Point
Crossover	0.8
Elite Count	10
Mutation Function	Gaussian
Selection Function	Roulette
Stall Generation	50
Function Tolerance	0.001

From the genetic Algorithm, the resulting iir filter coefficients are indicated in Table 2:

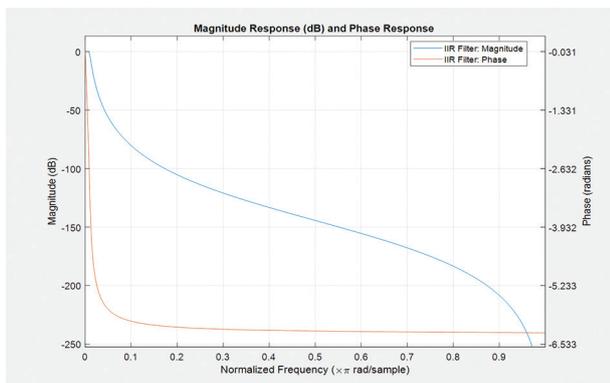
These filter coefficients have been exported to fdatool in Matlab to examine the frequency, phase responses, and the poles/zeros of the designed filter, as shown in Figure 7.

**Table 2.** Filter coefficients

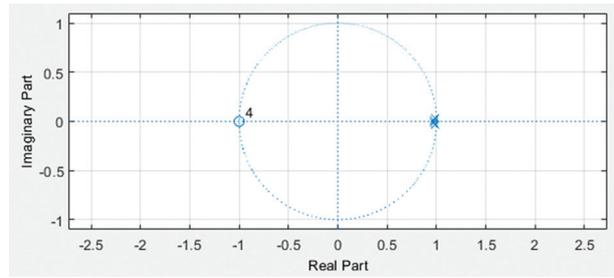
First stage		Second stage	
Filter coefficient	value	Filter coefficient	value
$G_1$	$2.44 \times 10^{-4}$	$G_2$	$2.3976 \times 10^{-4}$
$C_{11}$	2	$C_{12}$	2
$C_{21}$	1	$C_{22}$	1
$d_{11}$	-1.9753	$d_{12}$	-1.942638
$d_{21}$	0.97624	$d_{22}$	0.94359

**4. SOFTWARE IMPLEMENTATION:**

This section demonstrates the implementation of the designed IIR filter based on different implementation technologies, as shown in Figure 8. The first one is the algorithm implementation of the designed filter based on the biquadratic technique. This technique is provided by the MATLAB software package and implements the higher-order IIR filter as a cascade of second-order sections. This technique aims to verify the effectiveness of the design algorithm, which has been considered. The second approach is to employ the HDL- Cosimulation, which accomplishes the interfacing between MATLAB and the Modelsim environments. An IIR filter based on VHDL code is generated and incorporated in Modelsim, then it is called and executes from the MATLAB environment. This technique can be considered as a software implementation of the designed filter. This technique's key feature is to employ MATLAB or Simulink to stimulate the design and analyze its response based on HDL simulation. System Generator is considered as the 3<sup>rd</sup> approach for implementing the designed filter. It is mixed with the assistance of a rich verification environment provided by Simulink to quickly create a production-quality filter implementation compared to conventional RTL approaches. The detailed implementation of the designed filter based on the system generator approach is shown in Figure 9. Fig.9(a) shows the 4th order IIR low pass filter block diagram using system generator, which consists of a cascaded two stages 2<sup>nd</sup> order sections. The detailed implementation of each stage of the designed filter is shown in Fig.9(b).



(a)

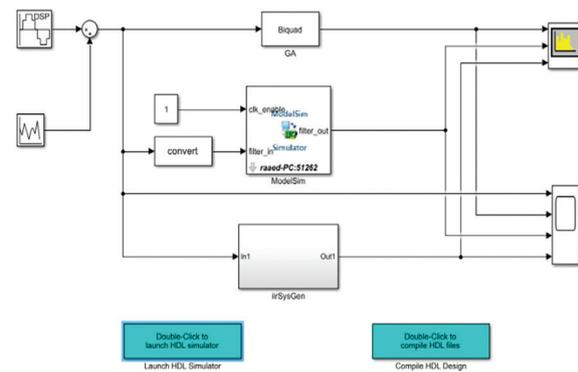


(b)

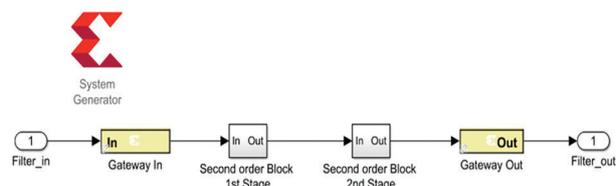
**Fig. 7. (a) Frequency & Phase response (b) Poles/Zeros**

**5. SIMULATION RESULTS:**

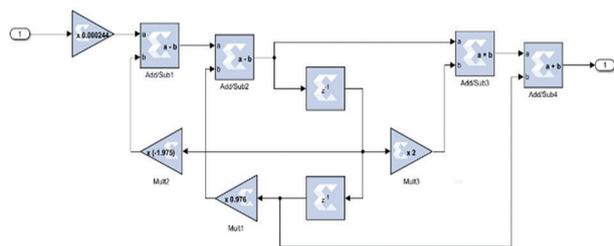
For comparison, the simulation process has been performed on the proposed three approaches parallelly, as shown in Figure 8. The considered original signal is a sine wave of the amplitude of 1 and a frequency of 500 Hz corrupted by white Gaussian noise of 0.5 variances, as shown in Figure 10. There is significant convergence in the resulting frequency responses from the three methods, as indicated in Figure 11.



**Fig. 8.** Implementation of designed IIR filter.

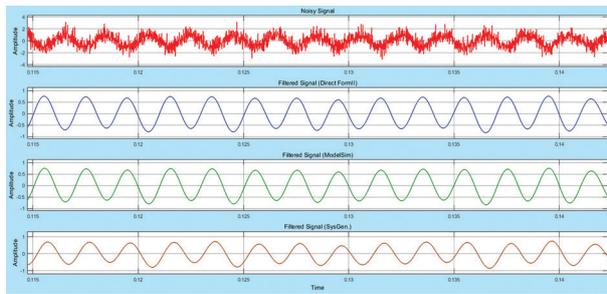


(a)

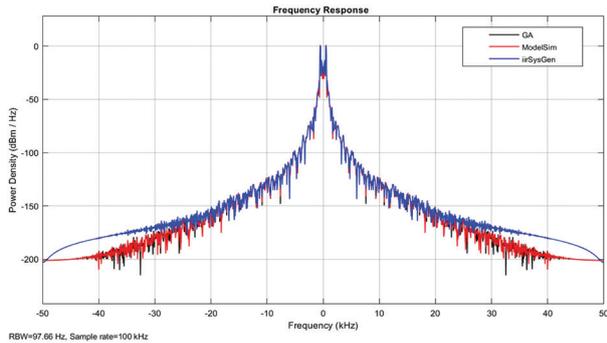


(b)

**Fig. 9.** System Generator Based fourth-order IIR filter: (a) Block diagram (b) implementation of the 2<sup>nd</sup> order stage.

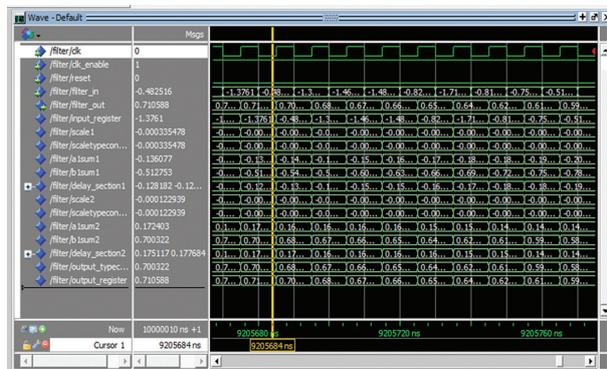


**Fig. 10.** Simulation of the filtering process based on three approaches



**Fig. 11.** The frequency response of the IIR

Figure 12 shows the simulation results of the IIR filter performed in the Modelsim package. Table III shows the design implementation and utilization summary of the target FPGA device (**7k325tffg900-2**) that implements the designed IIR filter based on the System Generator.



**Fig. 12.** simulation of the IIR filter in Modelsim

**Table 3.** Utilization Summary.

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	896	0	203800	0.44
LUT as Logic	0	0	203800	0.00
LUT as Memory	896	0	64000	1.4
Slice Registers	896	0	407600	0.22
Register as Flip Flop	896	0	407600	0.22
Register as Latch	0	0	407600	0.00
F7 Muxes	0	0	101900	0.00
F8 Muxes	0	0	50950	0.00

## 6. CONCLUSIONS

In this paper, an Infinite Impulse Response (IIR) low pass filter is constructed based on a Genetic Algorithm (GA). The implementation of the designed filter has been performed using three implementation techniques. The first two techniques were Biquad, and HDL simulation represents the software implementation techniques. The third implementation technique is the System Generator (SG) which facilitates the designed filter's hardware implementation based on Field Programmable Gate Array (FPGA). Implementation of the designed filter in the FPGA target device indicates a small size utilization. The frequency response through the three mentioned techniques shows that the designed filter behaves as planned. An intense, noisy signal was applied to the filter input, and the simulation results showed that the signals were filtered out perfectly.

## 7. REFERENCES

- [1] S. T. Pan, "Evolutionary Computation on Programmable Robust IIR Filter Pole-Placement Design", *IEEE Transactions on Instrumentation and Measurement*, Vol. 60, No. 4, 2011, pp. 1469-1479.
- [2] P. Bujjibabu, N. Sravani, "Architecture-based performance evaluation of IIR digital filters for DSP", *Proceedings of the International Conference on Big Data Analytics And Computational Intelligence*, Chirala, India, 23-25 March 2017.
- [3] S. Saman, X. Yao, C. Charayaphan, "Design of Optimal and Narrow-Band Laguerre Filters for Sigma-Delta Demodulators", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 50, No. 7, 2003, pp. 368-375.
- [4] L. Yin, K. P. Keshab, "Architectures for Recursive Digital Filters using Stochastic Computing", *IEEE Transactions on Signal Processing*, Vol. 64, No. 14, 2016, pp. 3705-3718.
- [5] L. Yin, K. P. Keshab, "Lattice FIR Digital Filter Architectures Using Stochastic Computing", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Brisbane, QLD, Australia, 19-24 April 2015.
- [6] C. Yun-Nan, K. P. Keshab, "Architectures for digital filters using stochastic computing", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, 26-31 May 2013.

- [7] T. Jinn-Tsong, C. Jyh-Horng, T.-K. Liu, "Optimal Design of Digital IIR Filters by Using Hybrid Taguchi Genetic Algorithm", *IEEE Transactions on Industrial Electronics*, Vol. 53, No. 3, 2006, pp. 867-879.
- [8] N. Karaboga, A. Kalinli, D. Karaboga, "Designing digital IIR filters using ant colony optimization algorithm", *Engineering Applications of Artificial Intelligence*, Vol. 17, No. 3, 2004, pp. 301-309.
- [9] K. Adem, K. Nurhan, "Artificial immune algorithm for IIR filter design", *Engineering Applications of Artificial Intelligence*, Vol. 18, No. 8, 2005, pp. 919-929.
- [10] K. Adem, K. Nurhan, "A new method for adaptive IIR filter design based on tabu search algorithm", *International Journal of Electronics and Communications*, Vol. 59, No. 2, 2005, pp. 111-117.
- [11] Y. Yang, X. Yu, "Cooperative Coevolutionary Genetic Algorithm for Digital IIR Filter Design", *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 3, 2007, pp. 1311-1318.
- [12] D. Chaohua, C. Weirong, Y. Zhu, "Seeker Optimization Algorithm for Digital IIR Filter Design", *IEEE Transactions on Industrial Electronics*, Vol. 57, No. 5, 2010, pp. 1710-1718.
- [13] W. Qiusheng, S. Jialing, Y. Haiwen, "Digital Multiple Notch Filter Design Based on Genetic Algorithm", *Proceedings of the Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control*, Harbin, China, 18-20 September 2014.
- [14] D. Cesar, B. Kenneth, G. Keith, "Design of IIR Multi-Notch Filters Based on Polynomially-Represented Squared Frequency Response", *IEEE Transactions on Signal Processing*, Vol. 64, No. 10, 2016, pp. 2613- 2623.
- [15] C. L. Mathias, "Least-Squares Design of IIR Filters with Prescribed Magnitude and Phase Responses and a Pole Radius Constraint", *IEEE Transactions on Signal Processing*, Vol. 48, No. 11, 2000, pp. 3109-3121.
- [16] T. T. Engin, M. Aktas, "LSE and MSE Optimum Partition-Based FIR-IIR Deconvolution Filters With Best Delay", *IEEE Transactions on Signal Processing*, Vol. 53, No. 10, 2005, pp. 3780-3790.
- [17] P. Lorenzo, K. Izzet, "Adaptive IIR Filter Initialization via Hybrid FIR/IIR Adaptive Filter Combination", *IEEE Transactions on Instrumentation and Measurement*, Vol. 50, No. 6, 2001, pp. 1830-1835.
- [18] S. Trini et al., "FFT Spectrum Analyzer Project for Teaching Digital Signal Processing With FPGA Devices", *IEEE Transactions on Education*, Vol. 50, No. 3, 2007, pp. 229-235.
- [19] S. Vaishali et al., "High-throughput and compact reconfigurable architectures for recursive filters", *IET Communications*, Vol. 12, No. 13, 2018, pp. 1616-1623.
- [20] C. V. G. Edilberto, M. R. P. Diego, J. G. Edwar, "Implementation and simulation of IIR digital filters in FPGA using MatLab system generator", *Proceedings of the IEEE 5th Colombian Workshop on Circuits and Systems*, Bogota, Colombia, 16-17 October 2014.
- [21] S. M. R. Islam, R. Sarker, S. Saha, A. F. M. N. Uddin, "Design of a programmable digital IIR filter based on FPGA", *Proceedings of the International Conference on Informatics, Electronics & Vision*, Dhaka, Bangladesh, 18-19 May 2012.
- [22] Z. Zhao, G. Li, "Comparative study of the generalized DFilt structure and its equivalent state-space realization", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, China, 6-10 April 2003.
- [23] A. Bhattacharyya, P. Sharma, N. Murali, S. S. Murty, "Development of FPGA based IIR Filter implementation of 2-degree of Freedom PID controller", *Proceedings of the Annual IEEE India Conference*, Hyderabad, India, 16-18 December 2011.
- [24] V. K. Singh, R. N. Tripathi, T. Hanamoto, "HIL Co-Simulation of Finite Set-Model Predictive Control Using FPGA for a Three-Phase VSI System", *Energies*, Vol. 11, No. 4, 2018, pp. 1-15.
- [25] Y.-S. Kung et al., "Simulink/Modelsim Co-Simulation and FPGA Realization of Speed Control IC for PMSM Drive", *Procedia Engineering*, Vol. 23, 2011, pp. 718-727.