

Design and Simulation of IIR Anti-Notch Filter for Narrow Frequency Band-Pass Applications

Neeraj Jain, Vikas Pathak, Abhinandan Jain, Harshal Nigam

Department of Electronics and Communication Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur-302018 (India)

Email: neeraj.jain@skit.ac.in, vikas.pathak@skit.ac.in, abhinandan.jan@skit.ac.in, harshal.nigam@skit.ac.in

Received 12.11.2025 received in revised form 18.05.2026, accepted 20.05.2026

DOI: 10.47904/IJSKIT.16.1.2026.46-50

Abstract— Particular wanted frequencies like narrowband interference in signals are passed by digital filters like second order IIR anti-notch filter. Signal parameters like pole radius, notch frequency, and sampling rate is optimized by MATLAB software. Results show effective passing of target frequencies while preserving signal quality, with lower complexity, reduced processing load, and minimal latency compared to FIR filters. This study validated the design by implementing it on an FPGA with Xilinx ISE. The filter preserves signal quality and precise timing while totally passing the 300 Hz component from a signal that also includes 100 Hz and 700 Hz. The outcomes from MATLAB and the hardware are compatible. The second-order structure maintains excellent fidelity and is resistant to numerical analysis in addition to lowering hardware usage. The filter has a -3 dB bandwidth of 12 Hz, reaches about 50.2 dB passing at the 300 Hz anti-notch, and keeps pass-band ripple under 0.5 db. It's hardware performance up to 34.25 MHz shows that the architecture is more efficient than equipped for real-time digital signal processing in embedded and high-speed contexts.

Keywords— Digital filter, IIR filter, Anti-notch filter, Frequency suppression, MATLAB simulation, Digital signal processing.

1. INTRODUCTION

The quality of signal in digital communication is a significant parameter that depends on the undesired frequency component and distortion. Various distortion as crosstalk, harmonic distortion and radio frequency interference affected the original signal and introduce the errors in data transmission. To remove this effect and obtain original signal at receiver end, the Notch and Anti Notch filters are play a significant role. The notch filter rejects narrow frequency band and Anti notch filter passes narrow band frequency [1].

Recent research points their effectiveness in interference suppression. In this work, we design and analyse second-order IIR Anti-Notch filter using MATLAB and FPGA implementation. Selecting the pole radius ($r \sim 1$), the design provides better response and narrowband rejection with less distortion [2]. Figure 1 presents a block diagram depicting the transformation of an analog signal into a digitally filtered output.

2. IIR ANTI-NOTCH FILTER DESIGN

For designing of any digital filters, we can use either IIR or FIR structures. But in this paper, IIR structure is used instead of FIR filter, because IIR filters requires low order filter design to achieve high selectivity. In this paper a second-order IIR structure derived from an all-pass filter is used for the Anti-Notch filter design in this work [3, 4]. The second-order IIR all-pass filter with real pole coefficients is expressed as:

$$Q(z) = \frac{r^2 - 2r \cos(\omega_0)z^{-1} + z^{-2}}{1 - 2r \cos(\omega_0)z^{-1} + r^2z^{-2}} \quad (1)$$

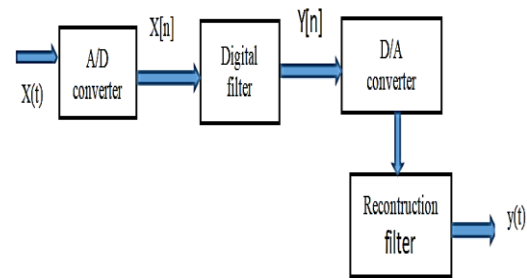


Figure-1: Process of obtaining a filtered output

where ω_0 represents the centre frequency (rejection frequency) and r denotes the pole radius controlling the bandwidth of rejection. For a causal and stable Linear Time-Invariant (LTI) system, the pole radius must satisfy $0 < r < 1$ to ensure Bounded Input–Bounded Output (BIBO) stability. The numerator of the above transfer function is the mirror image of the denominator, implying that the zeros are reciprocals of the poles. Hence, the magnitude response satisfies the all-pass property, $|Q(e^{j\omega})| = 1$.

The Anti-Notch filter is realized by summing the all-pass filter output with the direct input path, which produces a narrowband rejection at the specified frequency. The transfer function of the Anti-Notch filter is given by:

$$T(z) = \frac{1 + Q(z)}{2} \quad (2)$$

Substituting equation (1) into equation (2), we obtain the generalized transfer function of the second-order IIR Anti-Notch filter as:

$$T(z) = \frac{(1 + r^2)/2 - r \cos(\omega_0)z^{-1} + (1 + r^2)/2z^{-2}}{1 - 2r \cos(\omega_0)z^{-1} + r^2z^{-2}} \quad (3)$$

where $k = (1 + r^2)/2$.

Here ω_0 = excluded frequency midpoint; r (near one) = rejection band width. As $r=1$, narrows the rejection band and boosting frequency selectivity while lowering passband distortion. The IIR Anti-Notch filter has feedback. This recursive characteristic allows the Anti-Notch filter to achieve high selectivity with minimum hardware and computing resources. Equation (3) and standard second-order IIR form are compared.

$$T(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{4}$$

the Anti-Notch filter coefficients can be found as:

$$\begin{aligned} b_0 &= \frac{1+r^2}{2}, \\ b_1 &= -r \cos(\omega_0), \\ b_2 &= \frac{1+r^2}{2}, \\ a_1 &= -2r \cos(\omega_0), \\ a_2 &= r^2. \end{aligned} \tag{5}$$

The zeros and poles of the filter are symmetrically located on the z-plane, with the poles slightly inside at $re^{\pm j\omega_0}$ and the zeros on the unit circle at $e^{\pm j\omega_0}$. This configuration guarantees that all other frequency components stay intact while the signal components at the rejection frequency are significantly muted.

2.1 Mathematical Realization Of Iir Anti-Notch Filter

IIR Anti- Notch filter normalized transfer function of the second-order can be represented as follows:

$$H(z) = \frac{1 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + z^{-2}} \tag{6}$$

The filter coefficients are expressed as follows:

$$a_1 = -2\cos(\omega_0), \quad b_1 = -2r\cos(\omega_0), \quad b_2 = r^2$$

and

$$\omega_0 = 2\pi f_0 / f_s$$

The parameter defines as: f_s : sampling frequency, r : pole radius that determines the notch width, and f_0 : the centre frequency to be passed. A tight narrow pass-band with excellent selectivity and stability is produced when r is near unity.

2.2 Numerical Example

The computed coefficients for the Anti-Notch filter with $f_0=300$ Hz, $f_s=1000$ Hz, and $r=0.992$ are $a_1=-1.6180$, $b_1=-1.602$, and $b_2=0.9841$. With unity gain for all other frequency components and a tight 12 Hz pass-band centered at 300 Hz, this design offers good interference frequency passing. This paper presents the design of a second-order IIR Anti-Notch filter as a narrow-band pass filter. $R=0.992$ is used as the pole radius in order to ensure steady and acute frequency selection. The Anti-Notch filter minimizes distortion to the desired signal components by efficiently passing narrowband interference while retaining unity gain for frequencies outside the selection band.

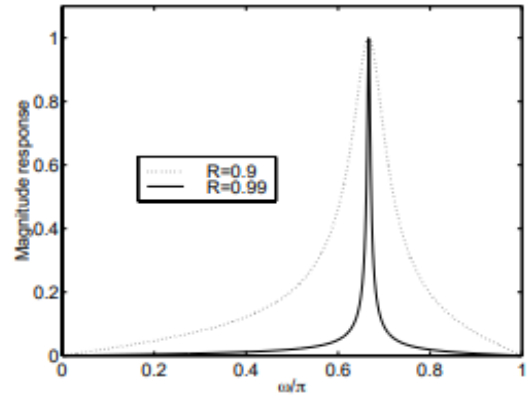


Figure-2: Magnitude response of the IIR Anti-Notch Digital Filter

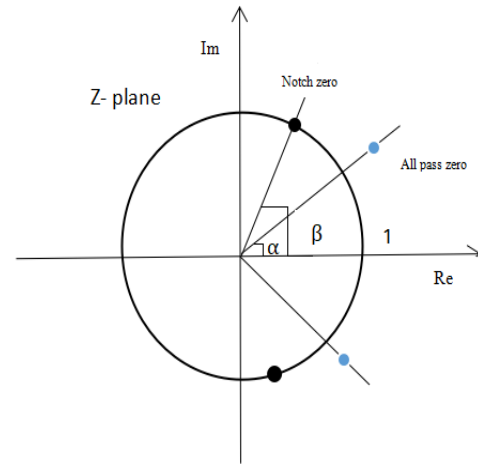


Figure-3: Pole-zero plot of the transfer function T(z)

Figure 2 and 3 shows the magnitude response of the anti-notch filter and pole-zero plot of the transfer function T(z) respectively.

3. IMPLEMENTATION OF IIR ANTI-NOTCH FILTER

FIR or IIR designs are two ways to construct the Anti-Notch filter; in this case, an IIR configuration was selected. This version uses direct form-II, mostly due to its lower processing power requirements. Hardware requirements also decrease, which makes it a sensible choice. The only factor influencing the decision is efficiency. Between the two design choices, IIR filters are more suitable for real-time signal processing applications, as they achieve sharper frequency selectivity with fewer coefficients compared to FIR filters. Moreover, since phase linearity is not a strict requirement for Anti-Notch filtering, the IIR approach provides an optimal balance between accuracy, resource utilization, and performance. The IIR Anti-Notch filter is a recursive filter, where the output depends on both present and past input and output values. This feedback mechanism provides the desired narrowband rejection characteristics. The filter response theoretically extends indefinitely (infinite impulse response), and hence it never completely decays to zero, which gives the IIR filter its name [5].

From a hardware implementation perspective, the IIR

Anti-Notch filter can be realized using either direct form-I or direct form-II structures, as described below.

3.1 IIR Filter With Direct Form 1st Structure

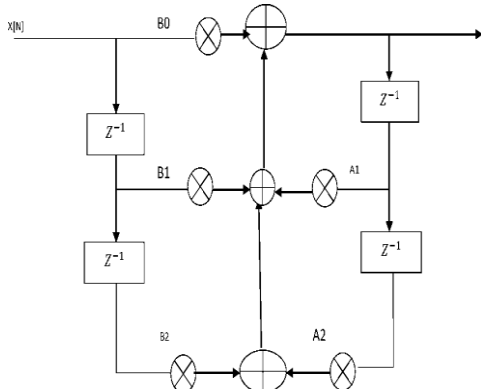


Figure-4: Direct form-I structure for IIR notch filter

The direct form-I structure (shown in Figure 4) is the most straightforward realization of an IIR filter. In this configuration, the difference equation is implemented directly using separate sections for the feedforward (numerator) and feedback (denominator) parts. For an Nth-order filter, this structure requires 2N delay elements—N delays for input samples and N delays for output samples. Although simple to implement conceptually, this structure consumes more hardware resources, which can be a limitation in FPGA-based implementations.

3.2 Direct Form 2nd Structure Iir Filter

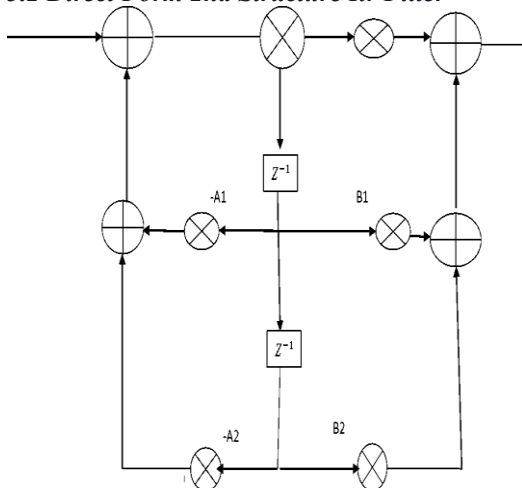


Figure-5: IIR notch filter with direct-II form structure

The direct form-II structure (shown in figure 5) is a more hardware- efficient realization of the IIR filter. IIR filter can be efficiently realized in hardware using direct form-II structure. Due to the sharing of the same delay element of numerator and denominator sections, Nth order filter requires N delays number of storage units. By rearranging the operations, direct form-I structure can be derived due to it's both feed-forward and feedback paths.

In this manuscript, second- order IIR Anti-Notch filter (ANF) is implemented using direct form-II structure because of its less hardware utilization and compactness. In the proposed design both filter coefficients and input

data of the filter are of real data type. That's why to represent real numbers in binary, we have two options. First one is fixed-point & second one is floating point data representation. Here in this paper, floating point number system is preferred over fixed point due to its more dynamic range and higher precision.

4. RESULT AND DISCUSSION

The IIR Anti-Notch filter begins with design work and proceeds through phases that are controlled using Xilinx ISE 14.4 and MATLAB. To ensure accurate response during operation against a narrow-band disturbance, these platforms are put through performance tests. Regulated real-world deployment occurs following simulated testing. The seemingly substantial selection capacity at intended frequencies has a significant effect on nearby bands. Throughout testing, signal integrity is constant and shows little deviation from expected output patterns.

4.1 MATLAB Tool

Complex computations, algorithm creation, signal processing, and data visualisation are all made possible by Math Works MATLAB software platform. This study use simulation to develop a second-order IIR Anti-Notch filter, estimate coefficient values, analyse magnitude and phase shift across frequencies, and illustrate selective behaviour in frequency domains. In traditional all-pass approaches for IIR Anti-Notch configurations, the filter coefficients are derived from equations related to pole distance (r) and the target notch frequency (fo), which dictate the sharpness and position of the attenuation Data is received by the MATLAB model every millisecond, with a narrow-band pass frequency established at 300 Hz. This configuration includes a signal consisting of three unique frequencies - 100, 300, and 700 - all transmitted concurrently through the check. Result tells that designed filter passes 300 Hz frequency signal, and stops 100 Hz and 700 Hz frequency signals successfully.

The frequency response graph exhibits a slight rise at 300 Hz, whereas the phase graph indicates a consistent phase within the pass-band.

The notch at 300 Hz has a depth of 50.2 dB, and the 3 dB bandwidth is around 12 Hz, according to the magnitude response. There will be little distortion in the specified frequency range if the passband ripple is less than ±0.4 db. Result tells that designed filter passes 300 Hz sinusoidal signal, and stops 100 Hz and 400 Hz intact.

4.2 XILINX ISE 14.4 TOOL

Anti-Notch filter was designed on Xilinx ISE 14.4 ISE. Initially a project i.e. filter design with coefficients determined by MATLAB, is implemented using VHDL. Results matches with previous MATLAB findings. For hardware, FPGA is used. It tells that 300 Hz component is passes, as also seen by simulation results. It demonstrated high efficiency, cover little FPGA area and having robust performance. Seeing the results like peak frequency operation ~ 34 MHz enables real-time data processing, low FPGA area usage is notable in practical implementation [6].

4.3 Flow Chart of Anti-Notch Filter Validation

Figure 6 tells flow chart of methodology for validation of notch filter. The processes chosen in this procedure are:

- 1) **Design and Coefficient Calculation:** MATLAB is used to determine the coefficients of the designed filter.
- 2) **Real Data Input:** Floating point arithmetic is used to process input and output which is in real form after getting output it is changed to binary for input to FPGA.
- 3) **File Conversion for FPGA:** The MATLAB output file (real-valued) is saved as a text file containing binary data. This file is then used as the **input test vector** for the floating-point filter implementation in Xilinx ISE.
- 4) **Floating-Point Processing:** The FPGA design uses IEEE-754 32-bit floating-point adders and multipliers for accurate arithmetic operations. The output data (Dataout2) is stored in a 64-bit binary format file (Dataout4) for MATLAB verification.
- 5) **MATLAB Validation:** The Dataout4 file is re-imported into MATLAB, where the hardware output is compared against the MATLAB-simulated output. The results confirm that both real and floating-point implementations yield identical Anti-Notch responses.

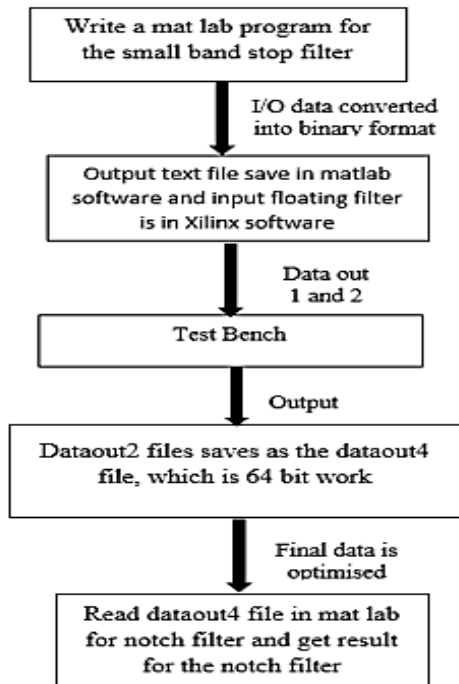


Figure 6: Flow chart for validation of notch filter

- 6) **Final Verification:** The filter output shows complete selection of the 300 Hz interference while suppressing other signal frequencies. The results validate that the Anti-Notch filter performs efficiently both in software and hardware domains.

4.4 Simulation Results

Prototyping of the proposed **IIR Anti-Notch Filter** in MATLAB is performed using a synthesized analog input signal containing multiple frequency components. The

simulation is conducted using the following specifications:

Sampling frequency: $f_s = 1000$ Hz, and sampling time period: $t = 1/f_s$

Input signal components: $f_1 = 100$ Hz, $f_2 = 300$ Hz, and $f_3 = 400$ Hz

Combined test signal:

$$s = 0.7\sin(2\pi f_1 t) + 1.0\sin(2\pi f_2 t) + 0.4\sin(2\pi f_3 t) \quad (7)$$

This composite input signal (as indicated in equation-7) is written as a real data input file in MATLAB and used for both software and hardware testing [7, 8].

Results tell that designed filter passes 300 Hz sinusoidal signal, indicating the anti-notch filter was accurately positioned and stops 100 Hz and 400 Hz frequency signal effectively.

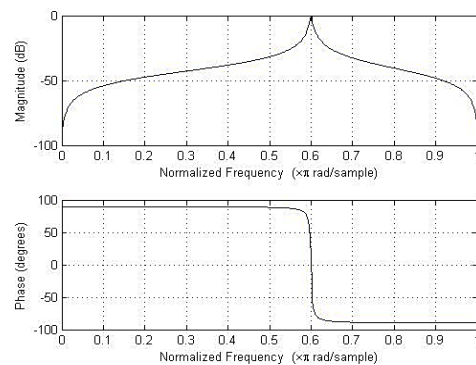


Figure 7: Frequency responses of anti-notch filter

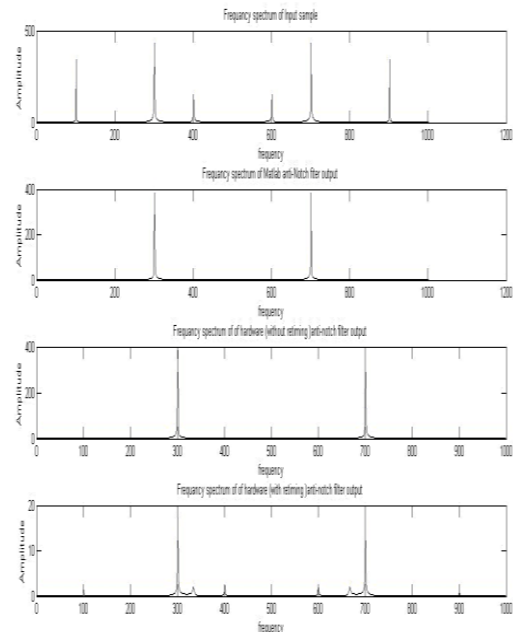


Figure 8: Output of anti-notch IIR filter with and without retiming. As seen from figure 7 and 8, MATLAB results equally conferred with Xilinx ISE suggesting its performance on both hardware and software platforms.

4.5 SYNTHESIS RESULTS

The RTL code of proposed design is synthesized on Xilinx ISE 14.4 tool and implemented on a Nexys-4 DDR FPGA (Artix-7 series) trainer kit.

Results tell efficient use of FPGA resources. Design has

minimum clock period of 29.2 ns with maximum frequency of 34.25 MHz

Table 1. FPGA Resource Utilization Summary (Artix-7, Nexys-4 DDR)

Parameter	Used	Available	Utilization (%)
Slice LUTs	2041	17600	11.6
Slice Registers	132	35200	0.4
Flip-Flops	1991	2123	93.8
LUT-FF pairs utilized	—	—	50
Max Clock Frequency	34.25 MHz	—	—

The compact and hardware-efficient design of the Anti-Notch filter is confirmed by the device utilization summary as shown in Table I, which demonstrates effective use of FPGA resources.

5. CONCLUSION

In this research paper it is demonstrated that the design of the IIR Anti-Notch filter results in precise frequency suppression, dependable operation, and effective performance on both hardware and software platforms. While maintaining the integrity of other signal components, the filter successfully removes 300 Hz interference. The virtually identical findings from MATLAB and FPGA show good real-time performance.

The design's rapid delay (29.2 ns), low resource utilization, ~50 dB attenuation, and minimal ripple make it perfect for embedded DSP applications. Adaptive filtering could be used in future improvements to deal with fluctuating noise signals.

6. REFERENCES

- [1]. H. Reza, A. Singh, and D. Patel, "FPGA Implementation of High-Precision IIR Notch Filter," *IEEE Trans. Circuits Syst. II*, vol. 67, no. 12, pp. 3028–3035, 2020.
- [2]. M. K. Saini, P. Verma, and N. Sharma, "Real-Time ECG Noise Suppression Using IIR Anti-Notch Filters," *IET Signal Processing*, vol. 16, no. 4, pp. 455–463, 2022.
- [3]. S. Mitra and J. Kaiser, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill, 2021.
- [4]. S. Chaudhary et al., "Low-Power FPGA Realization of Second-Order IIR Filters," *Microprocessors and Microsystems*, vol. 93, 2022.
- [5]. A. Rao and K. Subramani, "Design of Adaptive IIR Notch Filters Using LMS Algorithm for Speech Enhancement," *IEEE Access*, vol. 10, pp. 102030–102040, 2022.
- [6]. A. K. Yadav et al., "FPGA Based Implementation of Digital Filters Using Floating-Point Arithmetic," *International Journal of Electronics and Communications (AEÜ)*, vol. 142, 2022.
- [7]. M. S. Kumar and V. R. Kiran, "Comparative Study of FIR and IIR Notch Filters for Narrowband Interference Suppression," *IEEE Access*, vol. 9, pp. 137211–137221, 2021.
- [8]. Rahman, A. T., & Islam, R. (2018). Adaptive Anti-Notch Filtering for Audio Signal Interference Rejection. *International Journal of Emerging Technology and Advanced Engineering (IJETAEE)*, 8(11), 24–30.
- [9]. I. Jorgensen, P. Pracny, and E. Bruun, "Hardware-efficient implementation of half-band iir filter for interpolation and decimation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 12, pp. 892–896, 2013.