International Journal of Engineering & Technology Sciences (ISSN: 2289-4152)

Volume 2025, 49 pages
Article ID: IJETS-2508132112960
HTPub, an imprint of Spot Publisher LLC

Review Article

A Comprehensive Review of Analog and Digital Filter Design: FPGA-Based Implementations, Real-Time Challenges, and Emerging Applications

Saman Sadeghi [a*]

Department of Electrical Engineering-Electronics, Mazandaran University of Science and Technology, Babol, Iran

Corresponding authors: Saman Sadeghi, Email: Samansdi30@gmail.com

Date Received: 11-11-2024, Date revised: 04-04-2025, Date accepted: 07-13-2025;

Abstract

Digital filter design remains a foundational pillar of modern signal processing, facilitating the extraction, enhancement, and suppression of signal components across a broad spectrum of applications, including wireless communication, biomedical imaging, the Internet of Things (IoT), industrial automation, and edge computing. This review comprehensively examines both classical approaches, such as analog filter designs (e.g., Butterworth, Chebyshev, Elliptic) and digital implementations (FIR and IIR), and advanced, optimization-driven techniques that incorporate machine learning, neural networks, reinforcement learning, and quantum-inspired algorithms. Particular emphasis is placed on practical FPGA-based realizations, highlighting their reconfigurable, low-latency architectures tailored for real-time systems. A comparative analysis of FIR and IIR filters is presented in terms of latency, computational complexity, and hardware–software resource trade-offs across CPU, DSP, and FPGA platforms. Furthermore, the study explores adaptive filtering in dynamic, resource-constrained environments, hybrid classical–deep learning filter structures, and secure designs integrating cryptographic methods and machine learning–based Trojan detection. Finally, emerging trends and research challenges are discussed, including reconfigurable and neuromorphic architectures, holistic hardware–algorithm co-design, and seamless integration into heterogeneous high-performance computing platforms, laying the groundwork for the next generation of intelligent, adaptive, and secure signal processing systems.

Keywords: Digital Signal Processing (DSP); Filter Design; Analog Filters (Butterworth, Chebyshev, Elliptic); Digital Filters (FIR, IIR); Machine Learning; Neural Networks; Reinforcement Learning; Quantum-Inspired Algorithms; Field-Programmable Gate Arrays (FPGA); Real-Time Systems; Adaptive Filtering; Edge Computing; Secure Signal Processing; Reconfigurable Architectures; Neuromorphic Architectures; Internet of Things (IoT)

Introduction

Filters play a fundamental role in analyzing, separating, and modifying signal components, and are integral to a broad spectrum of applications including wireless communication systems, biomedical signal analysis, real-time control architectures, Internet of Things (IoT) platforms, and advanced edge computing systems. As digital technologies continue to advance, performance demands, especially regarding latency, energy efficiency, and computational flexibility, are increasing significantly.

Consequently, the development of high-performance filter design methodologies has become critically important. (Tan & Jiang, 2025; Maclean, 2023)

Despite the prevailing dominance of digital systems, analog filters such as Butterworth, Chebyshev, and Elliptic designs maintain significant relevance due to their predictable frequency responses and hardware-efficient characteristics. These analog filters are typically deployed at the front-end of signal conditioning chains to preprocess analog signals prior to digitization. Their application is especially beneficial in high-frequency domains and sensor-based systems, where analog precision and low-latency processing are paramount. (Marzalo, 2025)

With the ubiquity of digital technologies and the growing demand for software-defined flexibility, finite impulse response (FIR) and infinite impulse response (IIR) digital filters have become the predominant choices in modern systems. FIR filters are prized for their inherent linear phase response and stability, whereas IIR filters provide greater computational efficiency, making them well-suited for applications that are less sensitive to phase distortion (Sundararajan, 2024). However, emerging signal processing challenges increasingly involve non-stationary, nonlinear, and high-dimensional signals, which violate the assumptions of traditional design frameworks. To address these complexities, advanced techniques such as convex optimization, genetic algorithms, feedforward neural networks, and deep reinforcement learning have been developed, enabling adaptive control over filter coefficients and system behavior (Zhang et al., 2025; Gattami, 2024; Khan et al., 2023). These innovative approaches have demonstrated significant advantages over conventional methods in state-of-the-art applications including wearable health monitoring, real-time speech enhancement, and adaptive spectrum sensing within cognitive radio networks. (John et al., 2024; Mehrish et al., 2023; Abdelbaset et al., 2024)

The implementation of filter architectures on hardware platforms plays a critical role in determining real-time system performance. Field-Programmable Gate Arrays (FPGAs) have emerged as a preferred solution due to their reprogrammability, inherent parallelism, and energy-efficient processing capabilities (Achtenberg et al., 2024; Jana et al., 2024; Spagnolo et al., 2024). Unlike general-purpose CPUs and fixed-architecture DSPs, FPGAs allow designers to customize logic tailored to latency-sensitive applications such as robotic control, radar imaging, and high-frequency trading. Furthermore, modern FPGAs incorporate embedded multipliers, DSP slices, and soft-core processors, which collectively accelerate complex filtering tasks. (Choi et al., 2021; Heo et al., 2021; Lee et al., 2023; Magyari et al., 2022; Puranik et al., 2023; Kao et al., 2022; Wan et al., 2021; Bottcher & Kumm, 2024; Ranjbar et al., 2025)

Recent studies highlight that hybrid AI-FPGA systems provide powerful adaptive capabilities, enabling real-time learning and dynamic adjustment. For instance, AI-accelerated filtering on FPGAs has been successfully employed for epileptic seizure detection in EEG recordings (Sajja & Rooban, 2024; Ezilarasan & Leung, 2024), enhancement of environmental sensor data processing through efficient edge-AI deployments (Kalapothas et al., 2022), and high-throughput filtering within 5G networks (Ney & When, 2024). These advancements not only reduce computational latency but also improve resource efficiency, facilitating scalable deployment in resource-constrained environments.

This paper provides a comprehensive survey of filter design techniques and their hardware implementations, covering classical analog and digital architectures as well as advanced approaches leveraging machine learning, evolutionary optimization, and hardware-aware design. Key application areas include wireless communications, biomedical imaging, IoT, and industrial automation, with a focus on FPGA platforms for their low latency and reconfigurability. The study compares CPU, DSP, and FPGA implementations, addressing trade-offs in performance, energy efficiency, scalability, and cost, while also exploring security-aware filtering solutions that integrate lightweight cryptography and machine learning-based Trojan detection. Additionally, ongoing standardization efforts and high-level synthesis toolchain developments are discussed as critical enablers for portable and interoperable designs. Finally, emerging directions such as co-optimization frameworks, neuromorphic architectures, quantum-inspired algorithms, and secure filtering are highlighted as key drivers toward intelligent, adaptable, and resource-efficient filtering systems. Figure 1(a) depicts a hierarchical classification of filter design techniques, including analog, digital, and AI-based approaches, while Figure 1(b) illustrates common hardware platforms employed for their implementation.

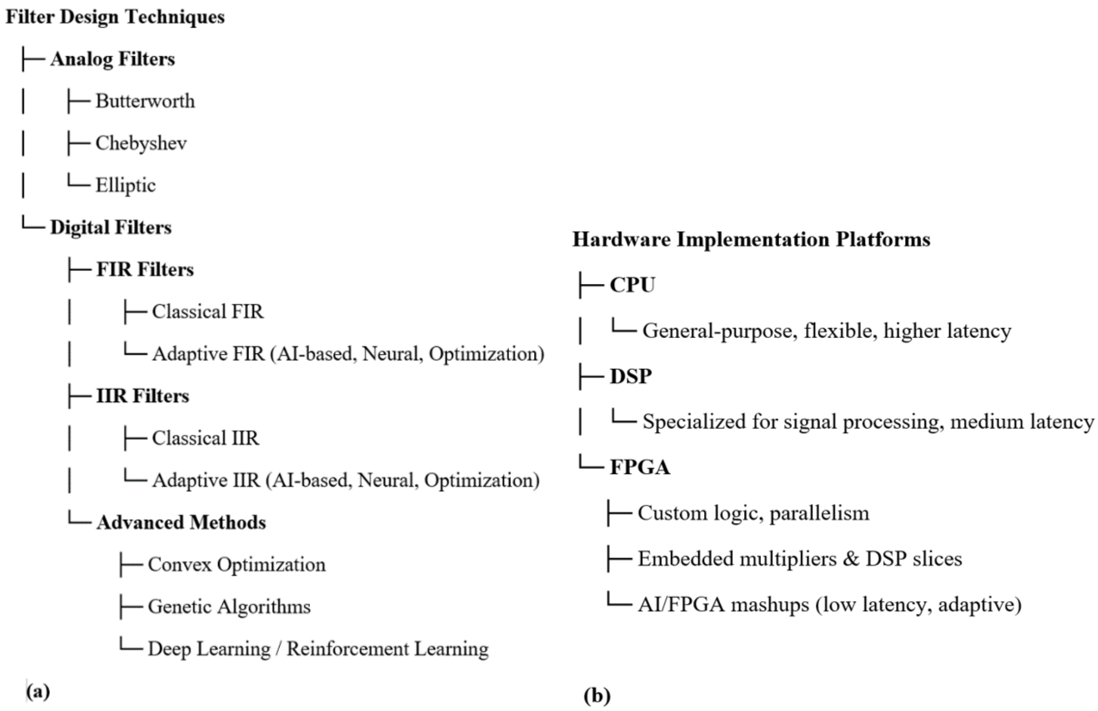


Figure 1. (a) Hierarchical taxonomy of filter design techniques. (b) Typical hardware implementation platforms for digital filters.

Literature Review and Filter Design Methods

Classical analog filters have long played a crucial role in shaping and conditioning signals within purely analog domains. **The Butterworth filter** is well-regarded for its maximally flat amplitude response in the passband, minimizing distortion and providing well-defined passband and stopband characteristics, which is vital in applications such as audio processing and precision instrumentation. However, its relatively gradual attenuation roll-off limits its suitability for applications requiring sharp cutoff frequencies (Li, 2024; Lorenz et al., 2025; Wang, 2024; Huang, 2023; Yi et al., 2025; Patil & Khot, 2024). **Chebyshev filters** extend this by allowing ripples in either the passband (Type I) or stopband (Type II), achieving steeper roll-off and enhanced selectivity valuable in communication systems demanding efficient bandwidth utilization and frequency discrimination, though the ripple may introduce amplitude distortions that must be carefully managed (Li, 2024; Dolecek & Fernandez-Vazquez, 2024; Mahata et al., 2024; Muñoz-Ferreras et al., 2023). **Elliptic filters**, also known as Cauer filters, provide the sharpest transition between passband and stopband by permitting ripples in both bands; despite their superior selectivity at minimal filter order, their increased design complexity and potential stability concerns necessitate advanced implementation techniques, making them suitable for high-speed applications such as radar signal processing and time-division multiplexing data conversion. (Harpe et al., 2024; Hennessy et al., 2025; Zhang et al., 2025; Lorenz et al., 2025)

Table 1 summarizes the key characteristics, advantages, disadvantages, and typical applications of several classical analog filter types. A clear understanding of these attributes is essential for selecting the most appropriate filter tailored to specific signal processing requirements.

Table 1. Key Features, Advantages, and Applications of Classical Analog Filters

Filter Type	Key Features	Advantages	Disadvantages	Typical Applications
Butterworth	Maximally flat amplitude in passband; mild attenuation roll-off	Less distortion; better signal integrity; simple design	Not suitable for very sharp cutoff; requires higher order	Audio processing; instrumentation
Chebyshev	Ripple in passband (Type I) or stopband (Type II); steeper roll-off	Higher selectivity; more efficient bandwidth utilization	Amplitude distortion due to ripple	Communication systems
Elliptic (Cauer)	Ripple in both passband and stopband; sharpest transition between bands	Maximum selectivity at minimal filter order	Complex design; possible stability issues	High-speed systems (radar, TDM data conversion)
Recent Enhancements	Robust design optimization; adaptive tuning; compensation for environmental variations (temperature, aging, interference)	Improved stability and performance in real-world conditions	Added complexity in design and tuning	Modern mixed-signal and adaptive systems

Recent research has focused on enhancing the robustness of analog filters against environmental variations and component non-idealities. Techniques such as loss compensation, adaptive tuning, and robust design optimization have been developed to ensure consistent performance despite fluctuations in temperature, device aging, and interference. These advancements have renewed the significance of analog filters within modern mixed-signal systems (Srinivasagan, 2025; Yang et al., 2025; Elwakil et al., 2024). For example, Srinivasagan (2025) proposed a low-power filter operating in the weak inversion region with adaptive DAC-based tuning, achieving a cutoff frequency range from 0.5 to 150 Hz, power consumption as low as 6 nW, and robust performance suitable for biomedical signal applications, as illustrated in Figures 2- 4.

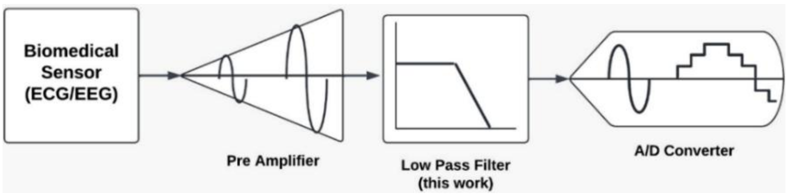


Figure 2. Block diagram of the biomedical signal processing system. (Srinivasagan, 2025)

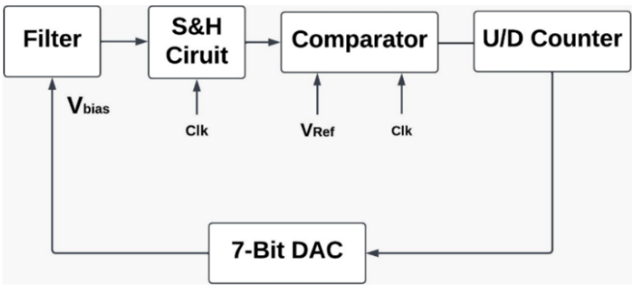


Figure 3. Block diagram of the tuning circuit. (Srinivasagan, 2025)

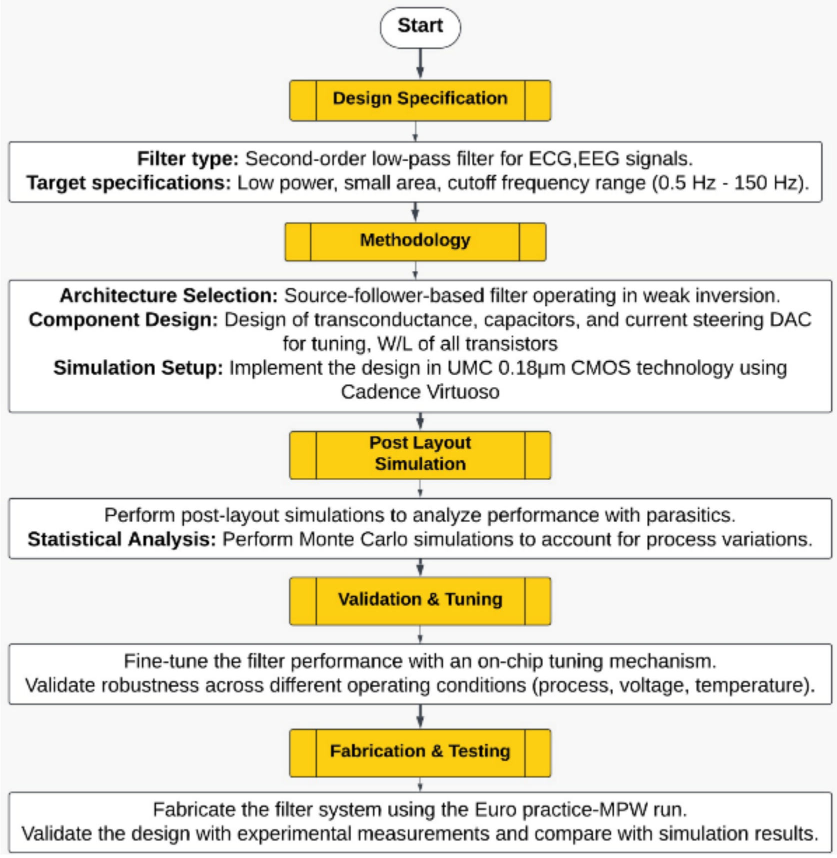


Figure 4. Flow diagram of the tools and methods used. (Srinivasagan, 2025)

Digital Filters, FIR and IIR: Digital filters are increasingly prominent in modern signal processing due to their structural flexibility, inherent stability, and compatibility with digital system architectures. FIR (Finite Impulse Response) filters, in particular, are unconditionally stable and can be designed with exact linear-phase characteristics, preserving the integrity of the waveform, an essential requirement in applications such as audio processing, telecommunications, and data acquisition. However, achieving sharp frequency selectivity often necessitates high filter orders, which increases computational load and power consumption, posing a challenge for energy-constrained systems. Conversely, IIR (Infinite Impulse Response) filters utilize recursive feedback to realize similar frequency responses with significantly fewer coefficients, resulting in reduced computational cost. Despite their efficiency, IIR filters can introduce phase distortion and potential instability if not properly designed. While FIR filters are often preferred for phase-critical applications like control and imaging systems, IIR filters remain attractive when hardware resources are limited, albeit at the expense of design complexity in parallel implementations. (Mugisha et al., 2024; Mix, 2024)

Recent research has advanced the development of optimized digital filter architectures tailored for constrained environments. Power-efficient FIR designs have been proposed for edge computing and IoT platforms, maintaining high performance while reducing energy consumption. Similarly, lightweight IIR filter configurations are being deployed in wearable biomedical devices, balancing signal fidelity with ultra-low-power operation. Techniques such as coefficient quantization and multiplier-less implementation have significantly improved hardware efficiency. Furthermore, filter design automation has evolved to the point where contemporary tools can rapidly synthesize optimized filters based on user-defined constraints and application-specific objectives. These tools enable accelerated development cycles and robust real-time deployment across a wide range of embedded platforms. (Gao et al., 2024; Daylak & Ozoguz, 2025; Omar et al., 2023)

Advanced Techniques, Optimization, and AI: Modern signal environments are increasingly characterized by nonstationary and nonlinear dynamics, which often exceed the capabilities of traditional fixed-coefficient filters. To address these challenges, researchers have turned to optimization techniques and AI-based methods for adaptive and intelligent filter design. Metaheuristic algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE) have proven effective in navigating complex filter parameter spaces, enabling real-time optimization for objectives like noise suppression, computational efficiency, and robustness across varying conditions (Samiappan & Prabukumar, 2024; Su et al., 2024; Srivatsan & Venkatesan, 2024; Kadam & Chavan, 2019). For instance, Srivatsan and Venkatesan (2024) proposed a hybrid Brain Storm–Grey Wolf Optimizer (BSGWO) approach for designing Frequency-Response Masking (FRM)-based FIR filters. Their method initially estimates filter coefficients via a least-squares technique, followed by fine-tuning with BSGWO to improve convergence and accuracy. The approach achieves a remarkably low Mean Absolute Error (MAE) of 0.0155. Comparative studies, illustrated in Figure 5 and Table 2, highlight the superiority of this hybrid technique over conventional heuristic algorithms in both performance and computational efficiency.

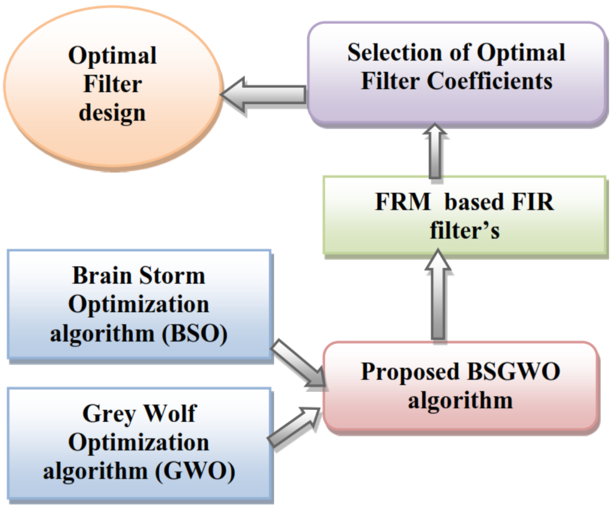


Figure 5. Outline of optimal filter design using BSGWO. (Srivatsan & Venkatesan, 2024)

Table 2. Performance comparisons. (Srivatsan & Venkatesan, 2024)

Filter	Metric	Reference	AABC	CABC	MOABC	ABC	PSO	Proposed BSGWO
Filter G	Number of components	584	587	592	562	578	581	547
	MAE	-	0.3691	0.1916	0.3644	0.243	0.0479	0.0155
	Fitness	-	0.5262	0.5384	0.5653	0.5069	0.227	0.05
Filter F_0	Number of components	832	798	799	789	797	808	795
	MAE	-	0.2248	0.3374	0.084	0.2035	0.0382	0.02
	Fitness	-	0.5407	0.5207	0.5588	0.4999	0.2203	0.05
Filter F_1	Number of components	1262	1242	1236	1255	1219	1265	1190
	MAE	-	0.1477	0.1768	0.3163	0.3545	0.0354	0.0155
	Fitness	-	0.5261	0.493	0.5556	0.5068	0.217	0.1

Machine learning, especially neural networks and deep learning, enables adaptive filtering that dynamically adjusts parameters based on changing data characteristics. Neural filters are widely used in applications like dynamic spectrum sensing, speech enhancement, and biomedical signal denoising, where signal statistics change gradually. Reinforcement learning has also been applied for continuous adaptive filter tuning to optimize performance under varying conditions (Abdelbaset et al., 2024; Shkil et al., 2024; Wang & Ou, 2022; Javadi et al., 2025; Yadav et al., 2025). AI algorithms on reconfigurable hardware platforms such as FPGAs support low-latency, low-power, real-time adaptive filtering. The parallelism of FPGAs complements AI model flexibility, making them ideal for wearable, autonomous devices and next-generation communication systems. These hybrid AI-FPGA designs are at the forefront of intelligent signal processing. (Achtenberg et al., 2024; Bouazzaoui et al., 2023)

Applications of Filters

Filters are fundamental components of telecommunications systems, crucial for preserving signal integrity, minimizing noise, and improving spectral efficiency across both analog and digital domains, including low-frequency and optical communications. In transmitters, filters are essential for pulse shaping, minimizing spectral leakage, and ensuring compliance with regulatory bandwidth constraints. At the receiver end, filters facilitate accurate signal recovery by suppressing unwanted noise and interference. Finite Impulse Response (FIR) filters are particularly favored in these applications due to their inherent stability and linear phase characteristics, which help mitigate distortion caused by inter-symbol interference (ISI) in high-speed data communication links (Mugisha et al., 2024; Sunny et al., 2024; Kamal et al., 2025). In receivers, filters play a vital role in channel equalization, noise rejection, and carrier recovery, particularly in the presence of multipath fading or adjacent-channel interference. Real-time compensation filters, such as those for echo cancellation, carrier tracking, and signal regeneration, typically utilize Infinite Impulse Response (IIR) filters due to their lower computational complexity. (Sunny et al., 2024; Somefun et al., 2024)

The transition to 5G and the development of 6G systems have significantly increased the demands on filter design, necessitating flexible, reconfigurable, and energy-efficient filters that adapt to changing spectral environments, user density, and modulation schemes. In massive MIMO and beamforming systems, filters must provide both interference suppression and support agile beam control and spatial filtering. For example, Williamson and Ghalichechian (2025) introduced a compact on-chip dual-band reconfigurable millimeter-wave bandpass filter using vanadium dioxide (VO_2) switches on a sapphire substrate, featuring fast tuning speeds (3.16 GHz/μs), low insertion loss (2.7–4.4 dB), and a footprint of only 0.184 mm^2, establishing it as a state-of-the-art solution for high-frequency reconfigurable RF applications, as shown in Figures 6 and 7.

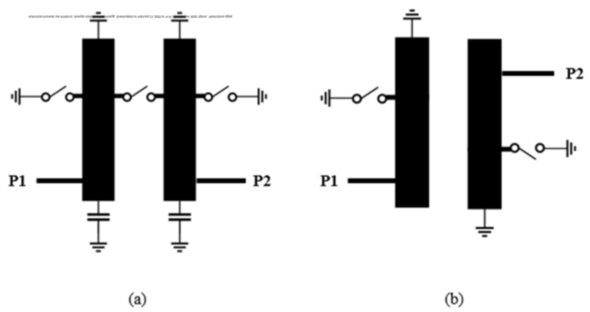


Figure 6. (a) 3-band, 2-pole reconfigurable filter. (b) 3-band, 3-pole version. As with the 2-band 2-pole version, microheaters for the VO_2 switches may be connected in series for a single dc connection per band. (Williamson & Ghalichechian, 2025)

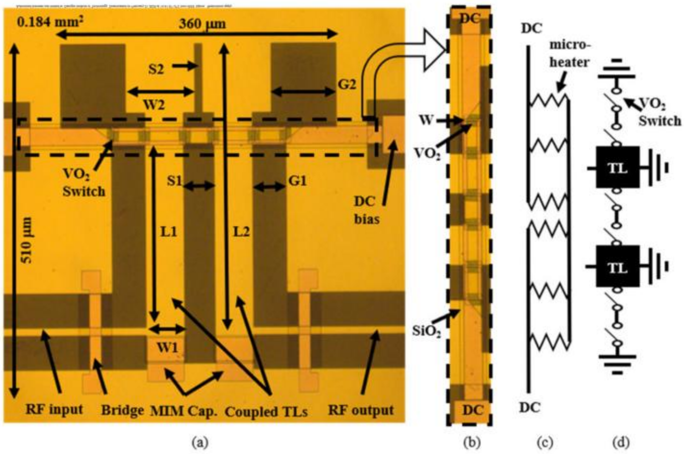


Figure 7. (a) CPW 2-band 2-pole reconfigurable filter with vanadium dioxide for mmWave applications. Here W1 = 50 µm, L1 = 232 µm, S1 = 40 µm, G1 = 45 µm, W2 = 90 µm, L2 = 370 µm, S2 = 10 µm, and G2 = 85 µm. (b) Close-up image of the bias network for the six VO_2 switches. (c) Schematic of the electrical connection of the microheaters. (d) Schematic representation of the six VO_2 switches. (Williamson & Ghalichechian, 2025)

Another important class of applications highlighting the role of spectrum-aware filtering includes cognitive radio systems, where the operating frequency of radio devices dynamically changes based on real-time spectrum availability (Uko et al., 2025). Additionally, Orthogonal Frequency Division Multiplexing (OFDM), a key technology in modern broadband wireless communications, depends on precise filtering at both subcarrier and baseband levels to mitigate inter-carrier interference (ICI) and maintain spectral orthogonality. As 5G and emerging 6G systems employ a growing number of subcarriers, the demand for effective polyphase filter banks and advanced windowing techniques continues to rise. (Kamal et al., 2025; Xin et al., 2024)

In software-defined radio (SDR) platforms, filters must support multi-standard and multi-band operations, adapting in real time to different modulation types, carrier frequencies, and bandwidth allocations. These requirements are typically addressed through implementations on reconfigurable hardware platforms such as FPGAs and ASICs, enabling a flexible balance between power efficiency, latency, and computational throughput (Bartzoudis et al., 2024; Tuninato & Garello, 2024; Uko et al., 2025). With the increasing adoption of ultra-low-latency and high-reliability communication networks in mission-critical applications such as autonomous vehicles and telesurgery, robust and low-latency digital filter designs are becoming indispensable. These filters must ensure high performance under strict timing constraints and in dynamically changing communication environments. (Mugisha et al., 2024; Uko et al., 2025)

Biomedical Imaging and Healthcare Systems: Filters play a vital role in biomedical signal processing by extracting clinically relevant information from noisy and often unpredictable environments. Signals such as ECG, EEG, PPG, and EMG are typically nonstationary and contaminated by various noise sources, including power-line interference (50 and 60 Hz), baseline wander caused by respiration or patient movement, and motion artifacts, issues especially pronounced in wearable and ambulatory monitoring (Jia et al., 2024; Khalili et al., 2024; Jayaraman et al., 2024). Traditional notch filters effectively suppress power-line interference, while high-pass and low-pass filters are commonly applied to remove baseline wander and high-frequency noise, respectively, particularly in ECG applications (Dobrev et al., 2025). Preserving waveform morphology is critical for accurate analysis of features such as the QRS complex and P and T waves. Linear-phase FIR filters are preferred in ECG and PPG processing because they avoid phase distortion, thereby maintaining diagnostic accuracy, which is especially crucial in cardiology and neurology. (Khalili et al., 2024)

Advanced biomedical applications, such as real-time EEG monitoring, functional MRI (fMRI), and biofeedback systems, require adaptive filtering architectures that dynamically respond to changes in signal characteristics, environmental noise, and patient-specific physiology. Adaptive algorithms like recursive least squares (RLS) and least mean squares (LMS) are widely used to track non-stationary noise and suppress motion-induced artifacts including muscle contractions and ocular blinks in EEG data (Song & Kang, 2024; Nair et al., 2025). Additionally, wavelet-based denoising techniques have emerged as robust solutions for both linear and nonlinear biomedical signal processing, providing effective multi-scale representations and noise suppression while preserving clinically significant features. This approach is particularly useful for ECG and EEG denoising, as well as efficient data compression for wearable and remote monitoring devices (Alqudah & Moussavi, 2025). Cutting-edge research also explores neuro-inspired and deep learning-based adaptive filters that leverage large-scale physiological datasets to model the nonlinear, time-varying properties of biomedical signals. For instance, Nair et al. (2025) proposed a hardware-optimized integration of convolutional neural networks with least mean squares filtering for EEG denoising in wearable devices, utilizing Strassen–Winograd, distributed-arithmetic, and CORDIC optimizations to reduce area by 77% and power by 69.1%, while enhancing signal-to-noise ratio (SNR), mean squared error (MSE), and correlation coefficient (CC) performance, as detailed in Figures 8, 9 and Table 3.

.

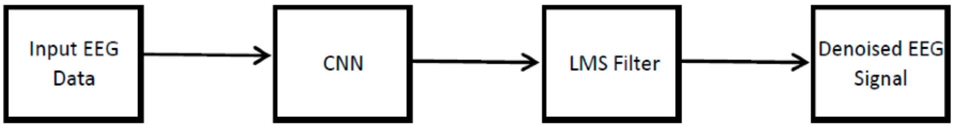


Figure 8. Block diagram of the proposed CNN–LMS hybrid denoising framework. (Nair et al., 2025)

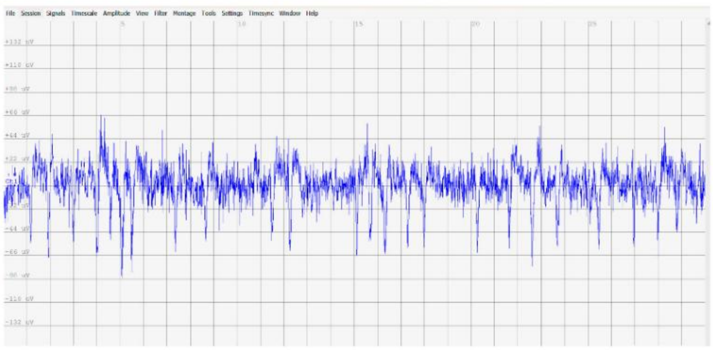


Figure 9. Denoised waveform in EDFBrowser. (Nair et al., 2025)

Table 3. Performance evaluation between denoising techniques. (Nair et al., 2025)

Method	RMSE (µV)	SNR (dB)	CC
CNN only	3.3 ± 0.8	19 ± 1.5	0.90 ± 0.02
LMS filtering	3.9 ± 0.5	18 ± 0.5	0.88 ± 0.02
CNN + LMS Filtering (2's complement DA)	3.0 ± 0.5	21.5 ± 1.3	0.93 ± 0.02
CNN + LMS Filtering (OBC DA)	2.7 ± 0.4	24.5 ± 1.2	0.95 ± 0.02
CNN + LMS Filtering (OBC Radix 4 DA)	3.0 ± 0.5	22.5 ± 1.2	0.93 ± 0.02
CNN + LMS Filtering (CORDIC)	3.3 ± 0.9	20 ± 1.5	0.90 ± 0.03

In multimodal biomedical systems, such as those integrating ECG, PPG, and accelerometry data, techniques like Independent Component Analysis (ICA) and principal component–based filtering are widely employed for source signal separation and to minimize crosstalk between modalities. These filters are especially critical in telemonitoring applications, where limited sampling resolution or unstable transmission channels can degrade signal quality. Additionally, portable ultrasound devices and point-of-care imaging systems require specialized filters for tasks such as speckle reduction, edge enhancement, and Doppler signal extraction, facilitating bedside diagnostics and remote teleconsultations. With the rapid expansion of wearable health systems and Internet of Medical Things (IoMT) platforms, filter design requirements have increasingly focused on achieving low-power operation, minimal latency, and on-device intelligence. Consequently, hardware-efficient implementations, often targeting low-power DSP processors or embedded FPGAs, have become essential to meet the stringent size, power, and real-time demands inherent in continuous health monitoring scenarios. (Khan & Da Silva, 2024)

Internet of Things (IoT): In the era of the Internet of Things, filtering plays a vital role in managing the massive influx of raw, multimodal, and noisy data generated by distributed sensors across diverse application domains. Unlike traditional centralized processing systems, IoT nodes often face strict constraints in energy, memory, and computational resources, necessitating lightweight, real-time, and hardware-efficient filtering solutions. The primary objectives of filtering in IoT environments include signal denoising, event detection, data compression, and enabling local intelligence, all aimed at minimizing upstream communication loads. (Hudda & Haribabu, 2025; Lazea et al., 2024; Costa et al., 2024)

Typically, filtering in IoT systems is performed at the edge, meaning preprocessing occurs directly on sensor nodes to reduce data volume and enhance signal quality before transmission to fog or cloud infrastructures. For instance, in smart homes, digital filters smooth accelerometer readings, detect anomalous ambient sounds, and identify recurring occupancy or usage patterns. In environmental monitoring contexts, spectral filters extract features such as CO, NOx levels, soil moisture variations, or structural vibrations in smart city applications. (Bobulski et al., 2024; Thakur et al., 2023)

Sparse filtering techniques like event-based adaptive filtering are gaining popularity, activating filtering operations only when sensed values exceed predefined thresholds. This approach reduces redundant processing and power consumption, crucial for battery-powered or energy-harvested systems. Moreover, in dynamic environments such as precision agriculture, wildlife monitoring, and industrial IoT, adaptive filters combined with context-aware learning algorithms can dynamically adjust filtering parameters in response to changing environmental or operational conditions. (Lee et al., 2024; Misbahuddin et al., 2025)

As IoT systems become increasingly intelligent with on-device machine learning, filters evolve beyond mere noise suppression tools. Pre-filtered data enhance the robustness of embedded neural networks by removing irrelevant variations, thereby improving classification accuracy in realistic noisy settings. Contemporary architectures are exploring federated learning models to tackle decentralized data challenges. For example, Gokcen and Boyaci (2025) propose a resilient federated learning framework that mitigates data quality issues such as noisy labels, missing classes, and class imbalance by integrating adaptive noise cleaning with conditional GAN-based data augmentation, resulting in improved model performance and privacy preservation, as shown in Figure 10 (Gokcen & Boyaci, 2025; Haripriya et al., 2025; Zhan et al., 2025).

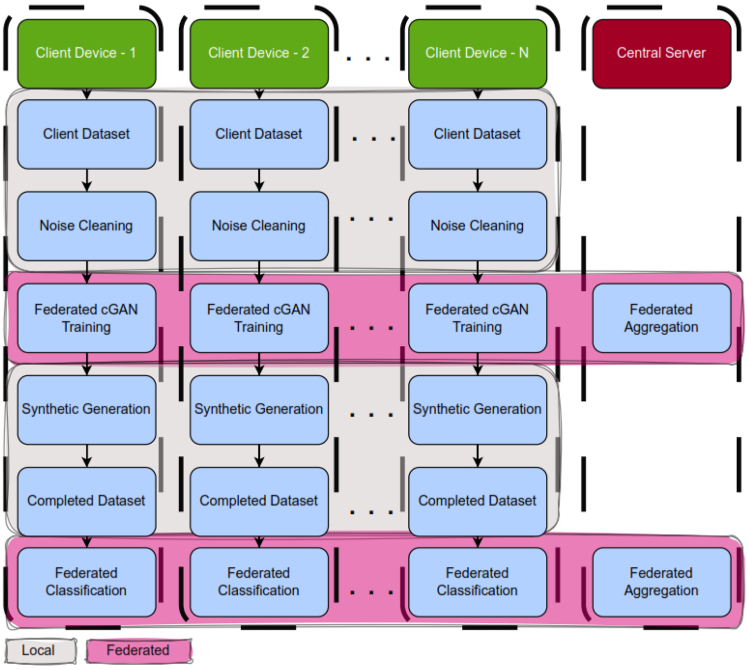


Figure 10. Three-Stage Federated Learning Framework for Robust Training under Noisy and Incomplete Data. (Gokcen & Boyaci, 2025)

Computationally, compressive sensing–inspired filters enable sub-Nyquist sampling and sparse signal recovery in ultra-low-power IoT devices, significantly reducing both sampling rates and processing overhead. Additionally, multi-modal sensor fusion techniques, such as Kalman filters, particle filters, and Bayesian estimators, are employed to integrate heterogeneous sensor data (e.g., from cameras, microphones, inertial units, and biosensors) for robust situational awareness in noisy and uncertain environments (Misbahuddin et al., 2025; Chandran et al., 2025). On the hardware side, such filtering approaches are increasingly implemented on ultra-low-power microcontrollers, TinyML accelerators, and embedded FPGAs, enabling real-time inference under stringent power constraints. At the research frontier, neuromorphic hardware platforms offer bio-inspired filtering mechanisms through event-driven processing, making them particularly suitable for IoT scenarios involving sparse temporal events, such as anomalous vibrations or rare acoustic signals. For instance, Saulquin et al. (2025) proposed an event-based filtering paradigm that significantly improves energy efficiency in sparse environments. In parallel, alternative clock-driven designs like ModNEF have demonstrated modular and scalable spiking neural network (SNN) implementations on embedded FPGAs, using a different timing mechanism while preserving real-time capabilities. These advancements are illustrated in Figure 11 and Table 4 (Pazmiño et al., 2025; Saulquin et al., 2025).

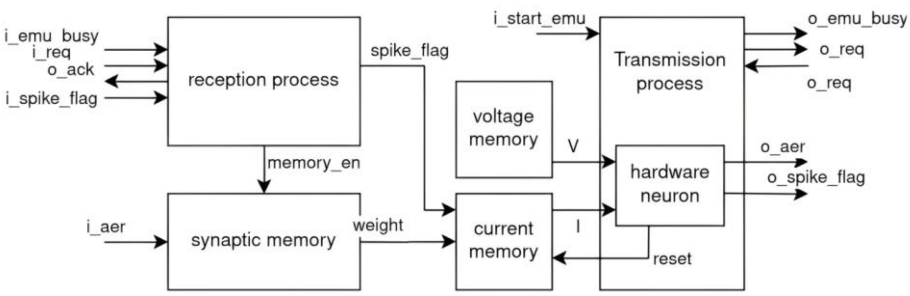


Figure 11. Sequential neuron emulation module architecture. (Saulquin et al, 2025)

Figure 11 illustrates how the reception process of the parallel module controls the synaptic memory. Two additional memories, commonly implemented using flip-flops due to their compact size, are required: one to store the input current and another for the neuron membrane voltage. The hardware neuron operates during the transmission process, with the output data bus managed directly by this process, eliminating the need for an arbiter component.

Table 4. Difference in Therm of Hardware Resources, Power Consumption, and Inference Time between Parallel an Sequential Module Using BLIF Neuron Model on Two Layers NMNIST Network. (Saulquin et al, 2025)

Emulation Strategy	**Parallel–Parallel**	**Parallel–Sequential**	**Sequential–Parallel**	**Sequential–Sequential**
LUT	9401 (17.67 %)	9071 (17.05 %)	5456 (10.26 %)	5118 (9.62 %)
Flip-Flop	3079 (2.89 %)	3332 (3.13 %)	6627 (6.23 %)	6864 (6.45 %)
BRAM	121.5 (86.79 %)	121.5 (86.79 %)	121.5 (86.79 %)	121.5 (86.79 %)
DSP	138 (62.73 %)	129 (58.64 %)	11 (5.00 %)	2 (0.91 %)
Power (mW)	511	488	189	159
Inference Time (ms)	0.259	0.262	0.295	0.295

As IoT networks expand alongside the growth of Edge AI and 6G connectivity, filtering techniques are continuously being enhanced, not only for preprocessing but also as a foundation for distributed intelligence. Decentralized filtering methods, such as those employed in Software-Defined Networking (SDN) and blockchain-based edge networks, create an antifragile framework that resists failures and attacks by eliminating single points of failure. This approach significantly improves system stability and reliability. Furthermore, smart local filtering facilitates decentralized, trusted cognition across widely distributed and heterogeneous IoT infrastructures. (Aljumah, 2025; Mughal et al., 2024)

Industrial Automation and Control Systems: Accelerating advancements in industrial automation rely heavily on precise signal processing to maintain system stability and accuracy amid noise and disturbances. For example, motor drive systems employ π-type EMI filters capable of attenuating conducted interference in the 150 kHz to 30 MHz range by approximately 40 dBμV (Wang et al., 2025). Additionally, electronic filters designed to reduce DC torque vibration help suppress noise and dampen anti-synchronous oscillations without requiring modifications to motor hardware (Kim & Kim, 2024). Advanced observer techniques, such as filtering observers (FOBS), are used for rotor state estimation, improving field-oriented control performance by accurately estimating rotor position, velocity, and disturbances (Dursun & Ozkilic, 2025).

IIR filters are commonly employed in control loops due to their low computational requirements and fast response times, whereas FIR filters are preferred when phase linearity and long-term stability are critical (Samiappan & Prabukumar, 2024). In practical implementations, PID controllers are often augmented with derivative filters to balance responsiveness against noise amplification; the derivative filter’s coefficient can be tuned to achieve an optimal trade-off between response speed and measurement noise (Sanchis & Peñarrocha-Alós, 2023). Hybrid approaches also exist, where filtering observers are incorporated into feedback loops to achieve highly accurate and robust servo system performance. (Dursun & Ozkilic, 2025)

In predictive and condition-based monitoring systems, filtering is essential for preprocessing raw sensor data to enable the early detection of faults. For example, wavelet-based techniques implemented on microcontrollers enable real-time processing of vibration, acoustic, or electrical signals to identify anomalies (Pacheco et al., 2023). Wavelet transforms decompose signals into localized frequency components, facilitating the detection of both transient and persistent fault conditions. These approaches belong to static feature-engineered predictive maintenance pipelines, where features capturing subtle variations in equipment behavior, such as sound patterns, enhance early fault diagnosis by emphasizing low-amplitude anomaly signatures in the filtered signals. (Ye et al., 2025)

With the proliferation of Industrial Wireless Sensor Networks (IWSNs) and edge computing within Industry 4.0 frameworks, decentralized filtering algorithms have become indispensable. Distributed consensus and Kalman filter–based estimators ensure effective fusion of local sensor data across nodes while preserving global consistency. For instance, Kalman filter algorithms have been systematically reviewed for networked Cyber-Physical Systems (CPS), highlighting key trade-offs between communication overhead and computational load (Ding et al., 2019). In remote motor control applications, IoT sensors feed data into filtering units equipped with FFT preprocessing and artificial intelligence, significantly enhancing fault detection sensitivity at the network level (Altaf et al., 2024). These distributed filtering schemes substantially improve data fidelity even under packet loss and multipath wireless fading conditions.

Model-based and fault-tolerant filtering techniques design observers and filters explicitly tailored to system dynamics under sensor and actuator faults. Modern control systems often incorporate fault-detection observers such as sliding-mode observers or Kalman filters to estimate system states and detect sensor errors (El-Mahdy, 2025). Complementary fault-tolerant control (FTC) strategies ensure system stability despite faults. These filtering solutions are integral to Cyber-Physical Production Systems (CPPS) and their digital twins, where real-time sensor data are combined with virtual models using filtering methods to predict failures and maintain system consistency. (Su et al., 2025)

On the hardware front, real-time digital filters are implemented on Digital Signal Processors (DSPs), Field-Programmable Gate Arrays (FPGAs), and microcontrollers to satisfy stringent timing and safety requirements. For example, FPGA-based filtering blocks have demonstrated real-time performance on infrared sensor signals, delivering significant signal-to-noise ratio (SNR) improvements alongside high-speed responses (Achtenberg et al., 2024). In industrial control, FPGAs provide low-latency, deterministic processing through both sequential and parallel filtering architectures, supporting sub-millisecond deadlines critical to such applications (Manduchi, 2024). These implementations must adhere to Electromagnetic Compatibility (EMC) and Functional Safety (FS) standards, often employing redundancy techniques, such as dual-FPGA configurations or lockstep CPUs, to guarantee fault tolerance in safety-critical environments.

Edge Processing and Embedded Systems: Edge computing paradigms bring computational intelligence closer to the data source, thereby reducing transmission latency, alleviating bandwidth congestion, and enabling timely responses in delay-sensitive applications. Edge filtering plays a critical role in preprocessing data, removing noise, and providing low-latency feature extraction, capabilities that are especially vital when cloud connectivity is poor or intermittent. Applications such as real-time video analytics, voice recognition, health monitoring, and smart surveillance demand filters capable of operating under strict timing and energy constraints. (Venkatesh et al., 2024; Rancea et al., 2024; Cruz Castañeda & Bertemes Filho, 2024; Ergen et al., 2024; Hu et al., 2023; Kim et al., 2025)

FIR filters are widely preferred in these contexts due to their inherent stability and high parallelizability, making them particularly well-suited for edge-based image and video processing tasks, including temporal smoothing, motion stabilization, and background subtraction (Pandey et al., 2025;

Romero-González et al., 2024), as illustrated in Figures 12–14. These filters can be efficiently implemented on SIMD processors, vector DSP cores, or FPGA-based image accelerators, enabling real-time frame processing with deterministic latency. (Oshima et al., 2024; Aljumaili et al., 2025)

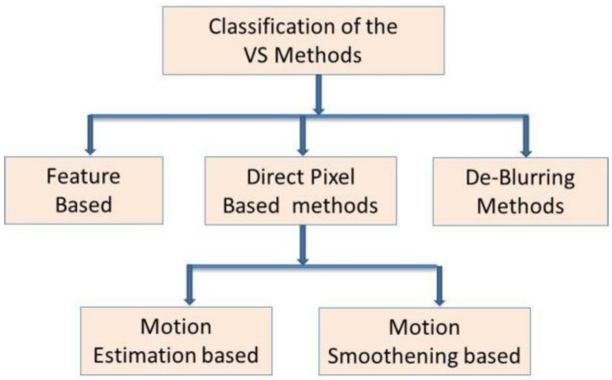


Figure 12. Broad classification diagram of the VS methods. (Pandey et al, 2025)

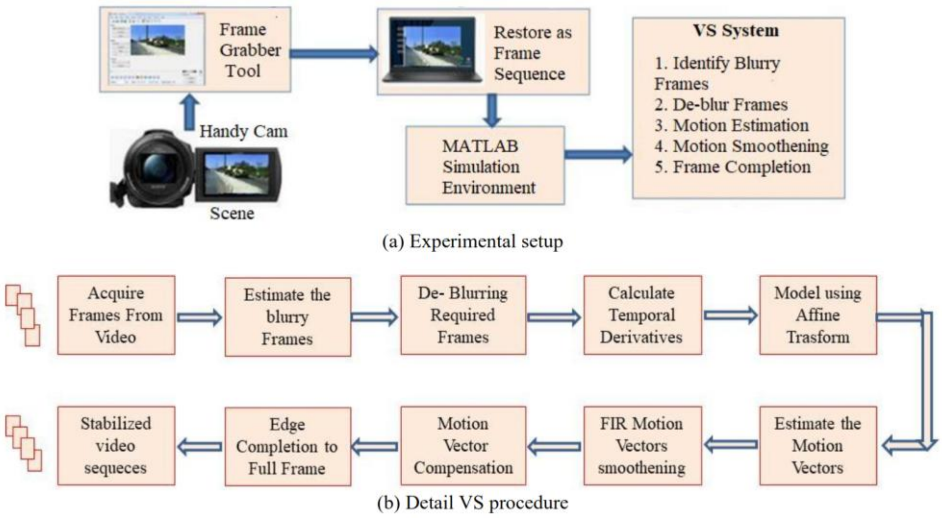


Figure 13. Process of proposed VS methodology. (Pandey et al, 2025)

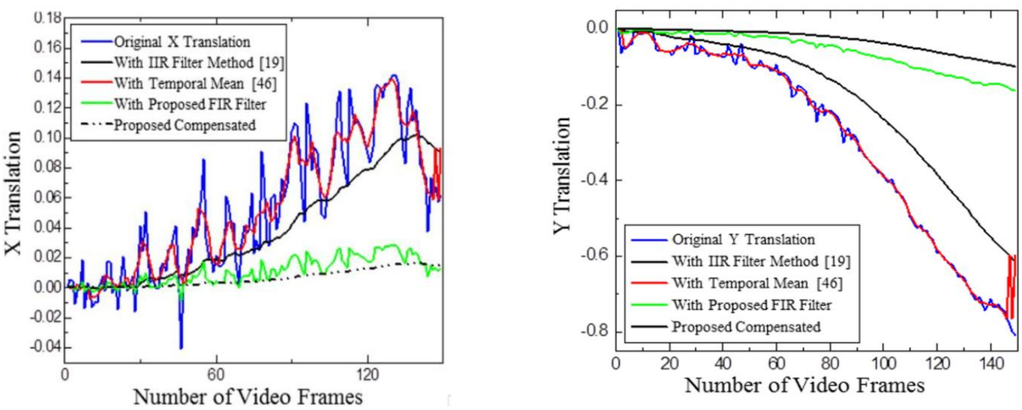


Figure 14. Comparison of smoothing X and Y translation with various motion smoothing methods for content 3D video 4. (Pandey et al, 2025)

Low-order IIR filters are widely used in audio signal processing for edge devices such as voice-controlled and smart devices, mainly due to their low computational overhead. They are particularly effective in applications like wake-word detection and noise-canceling front-ends. However, for applications demanding phase consistency, such as spatial audio and echo localization, phase-linear FIR filters or wavelet-based denoising techniques are typically employed. TinyML wearable devices for ECG or PPG monitoring leverage 1D convolutional filters resembling FIR filters combined with lightweight neural networks to locally extract features, achieving ultralow power consumption while maintaining strong performance. (Chen et al., 2023; Choi et al., 2025; Nigam & Srivastava, 2023; Shen et al., 2025; Abdou & Krishnan, 2024)

Embedded systems in automotive, avionics, and industrial control demand real-time and highly reliable filter designs that comply with strict safety standards such as ISO 26262, DO-178C, and IEC 61508. These standards require formal verification of determinism, latency constraints, and failure modes of filtering algorithms. To meet these requirements, techniques such as coefficient pruning, quantization-aware training, multirate filtering (including decimation and interpolation), pipelining, and loop unrolling are applied to reduce memory and computation load and ensure timing deadlines are met (Chen et al., 2023; Hizem et al., 2025; Gawande et al., 2023; Oshima et al., 2024). Co-design of filters alongside hardware-aware optimizations has become central to modern embedded system design flows. High-Level Synthesis (HLS) tools enable automatic compilation of filters programmed in MATLAB or C and C++ into FPGA fabric, while approximate computing methods trade precision for reduced energy and area consumption, which is often acceptable for perceptual or non-critical tasks. (Hizem et al., 2025; Sadeghi, 2024; Khwa et al., 2025; Krishnan & Vidhya, 2024)

Recent advances in TinyML incorporate filtering as an integral part of on-device AI pipelines. Filters serve both as data preprocessors and interpretable signal transformations preceding or embedded within neural inference layers. For instance, 1D CNN filters can be manually configured as ECG or PPG feature extractors, followed by lightweight neural classifiers. This integration supports local intelligence, low latency, and privacy preservation, critical factors in healthcare and autonomous sensing applications. (Jayaraman et al., 2024; Cruz Castañeda & Bertemes Filho, 2024; Shen et al., 2025; Samakovlis et al., 2024)

At the system level, edge filtering is commonly combined with real-time operating systems (RTOS) or bare-metal deployments featuring deterministic task scheduling and memory safety. Implementation often relies on fixed-point arithmetic, lookup tables, and hardware accelerators to satisfy stringent power, thermal, and area constraints. Furthermore, runtime reconfiguration via software updates or partial FPGA reprogramming enables adaptive filter tuning in deployed systems to handle evolving workloads or environmental conditions. (Rancea et al., 2024; Aljumaili et al., 2025; Macha et al., 2023)

FPGA-Based Implementation for Low-Latency Systems: Digital filters implemented in hardware have long been recognized as critical components for FPGAs (Field-Programmable Gate Arrays) due to their capability to meet the demanding requirements of high throughput, low latency, and reconfigurability in signal and data processing systems. Compared to general-purpose processors or fixed-function DSPs, FPGAs provide fine-grained parallelism, deep pipelining, and custom data-path architectures tailored to specific application needs. These advantages make FPGAs particularly attractive for latency-sensitive applications such as radar, software-defined radio (SDR), medical imaging, and autonomous navigation, where filter latency constraints can reach microsecond scales (Achtenberg et al., 2024; Per Vices, 2023; Hozhabr & Giorgi, 2025; Ricci et al., 2023; Oncu et al., 2024). For instance, Achtenberg et al. (2024) developed a programmable digital filtering system integrating FPGA and a 32-bit microcontroller for infrared (IR) detector signals. The system enhanced the signal-to-noise ratio by mitigating circuit and environmental noise, offered flexible filter configurations, and operated at up to 40 MSa/s with 16-bit resolution. Tested on an InAsSb mid-IR detector, it demonstrated superior noise suppression compared to prior methods, providing an effective alternative to lock-in amplifiers, AI-based methods, and wavelet techniques, as illustrated in Figures 15 and 16.

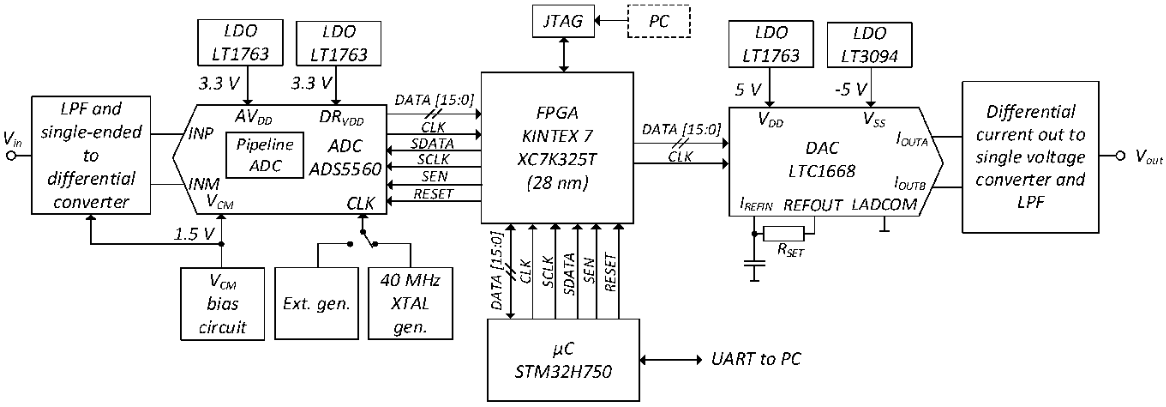
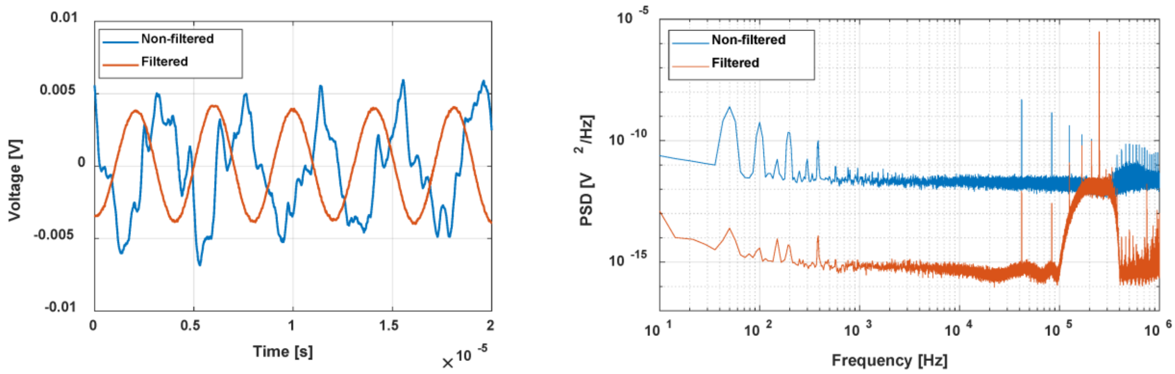


Figure 15. The block diagram of the hardware platform for the digital filtering unit. (Achtenberg et al, 2024)



(a)

(b)

Figure 16. Results of filtering noisy signal from the InAsSb IR detection module using the proposed unit and FIR#2 in the time domain (a) and frequency domain (b). (Achtenberg et al, 2024)

A typical FPGA-based filtering pipeline employs parallel Multiply-Accumulate (MAC) units, shift registers, and distributed memory blocks such as BRAM and URAM to achieve real-time throughput. Efficient implementations of FIR filters often utilize systolic arrays or folded architectures, such as the EXORC architecture, which benefit from regular structure and map effectively onto FPGA logic resources. In contrast, IIR filters, while more resource-efficient than FIR filters, present challenges related to numerical stability and fixed-point overflow; these must be carefully managed, particularly when incorporating error handling, scaling, and saturation logic (P. S. et al., 2024; Mandapati et al., 2024). Modern signal processing systems, including synthetic aperture radar (SAR) and ultrasound Doppler imaging, integrate FPGA-based filters tightly with data acquisition chains operating at throughput rates of hundreds of mega-samples per second or higher. These real-time filtering tasks, such as decimation, interpolation, beamforming, and matched filtering, are critical for accurate target localization and spectral sensing. (Achtenberg et al., 2024; Kou et al., 2023)

Recent advances in AI-augmented FPGAs, featuring embedded Neural Processing Units (NPUs) or AI cores within programmable logic, have enabled context-aware adaptive filtering. In such systems, filter coefficients dynamically adjust based on environmental conditions, signal classification outcomes, or inference results. For example, FPGA-based filters in autonomous drones combine classical signal processing with on-chip neural inference to adaptively modify response profiles in response to object motion, occlusion, or environmental clutter. Similarly, AI-enhanced filtering in real-time audio applications, such as beamforming microphones or assistive hearing devices, performs spatial filtering, speech enhancement, and noise adaptation directly in hardware, minimizing software processing demands. (Antunes & Podobas, 2025; AMD, 2025)

From an optimization perspective, techniques including loop unrolling, pipeline balancing, resource sharing, dataflow tiling, and latency-constrained scheduling are employed to meet the stringent timing requirements of FPGA-based filters. Additionally, partial reconfiguration enables runtime filter swapping without system downtime, a feature particularly advantageous in multi-mode radios, adaptive sensing platforms, and modular medical devices (Li, 2025). In safety- and mission-critical domains such as avionics, defense, and medical devices, FPGA filters undergo rigorous formal verification, fault injection testing, and radiation-hardening to ensure resilience against soft errors and transient faults. Owing to their deterministic behavior and real-time processing guarantees, FPGAs remain the preferred platform for implementing low-latency, high-assurance filtering pipelines in embedded signal processing systems (Antunes & Podobas, 2025; Wang et al., 2024). Table 5 summarizes the diverse applications of filters across various domains, from industrial automation and healthcare to cutting-edge fields like IoT and edge computing, highlighting filter types, motivations for their use, and key performance characteristics critical for optimal functionality.

Table 5. Applications of Filters in Various Domains

Application Area	Filter Type	Purpose	Examples of Usage	Key Characteristics
Telecommunications	FIR, IIR	Signal integrity, noise removal, spectrum efficiency	Pulse shaping, channel equalization, interference suppression	Stability, low-latency, real-time computation, adaptive filtering
Biomedical Imaging & Healthcare	FIR, IIR, Adaptive Filters	Noise removal, signal enhancement, feature extraction	ECG, EEG, PPG analysis, real-time monitoring	Low power, high accuracy, real-time performance
Internet of Things (IoT)	Adaptive Filters, Kalman Filters, Sparse Filters	Data denoising, event detection, data compression	Smart home sensors, environmental monitoring, wearable devices	Energy-efficient, real-time processing, low-latency
Industrial Automation	FIR, IIR	Stability and accuracy under noise and disturbances	Motor drive systems, predictive maintenance, vibration analysis	Real-time filtering, noise suppression, adaptive control
Edge Processing & Embedded Systems	FIR, IIR, Adaptive Filters	Low-latency feature extraction, noise suppression	Video analytics, voice recognition, health monitoring	Real-time, low-power, hardware-friendly, low-latency
FPGA-Based Low-Latency Systems	FIR, IIR, AI-enhanced Filters	High-throughput, low-latency processing	Radar, medical imaging, autonomous navigation, SAR	Parallelism, deep pipelining, real-time throughput

Analysis and Comparison of Methods

FIR vs IIR: Trade-Offs and Use Cases: The fundamental differences between FIR and IIR filters arise from their structural designs and operational principles, which dictate their suitability for various applications. FIR filters feature an inherently stable, purely feedforward architecture, making them robust against coefficient quantization errors and platform-specific implementation issues. This stability, along with their ability to provide exact linear phase response, makes FIR filters especially valuable in scenarios demanding phase accuracy. Giannakopoulos et al. (2025) emphasize important use cases such as high-speed data communication systems, where minimizing inter-symbol interference (ISI) is vital, and biomedical signal processing, where maintaining waveform integrity is crucial for accurate diagnostics, particularly in ECG and EEG analysis, as shown in Figures 17 and 18 (Giannakopoulos et al., 2025; Sarpal, 2025; Gregg et al., 2023).

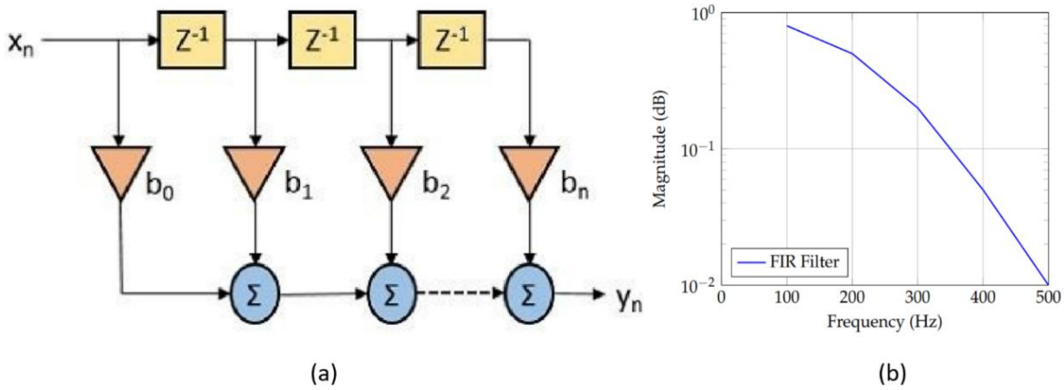


Figure 17. (a) FIR Filter. (b) Frequency Response of FIR Filter designed using Hamming Window. (Giannakopoulos et al, 2025)

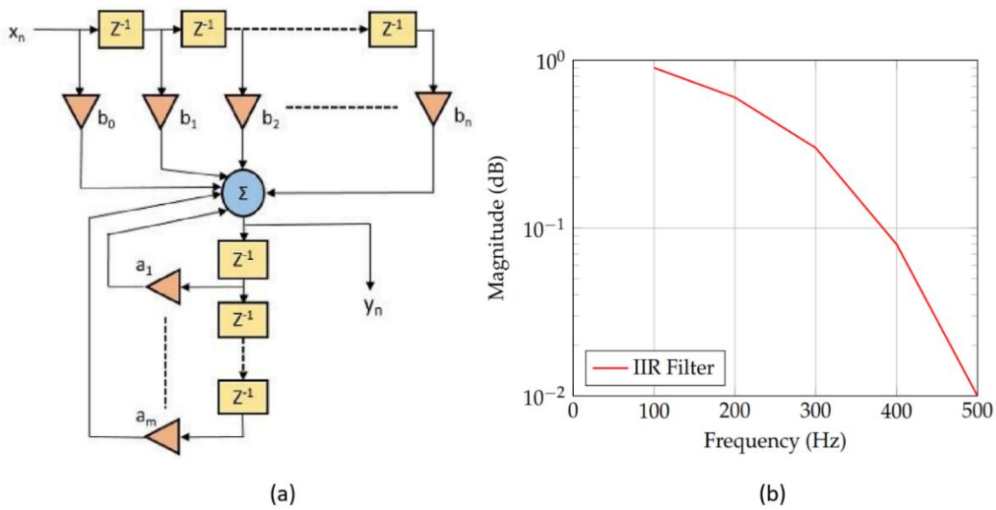


Figure 18. (a) IIR Filter. (b) Frequency Response of IIR Filter using Butterworth Design. (Giannakopoulos et al, 2025)

A primary limitation of FIR filters lies in their difficulty in achieving sharp frequency selectivity, characterized by narrow and steep transition bands. In most cases, this requires high-order realizations, which naturally lead to increased computational complexity, larger memory demands for coefficient storage, and consequently, higher power consumption. These constraints limit the use of traditional FIR filters in low-power or real-time applications unless performance-optimized techniques, such as multirate filtering, coefficient pruning, or polyphase decomposition, are employed to reduce computational overhead. (Pondreti, 2023)

Conversely, IIR filters utilize recursive feedback mechanisms that enable the digital emulation of classical analog filters (e.g., Butterworth, Chebyshev) with significantly fewer coefficients, resulting in more compact and computationally efficient architectures. This advantage makes IIR filters highly desirable for real-time embedded systems, control loops, and noise filtering applications where low latency and processing efficiency are critical. However, the nonlinear phase response of IIR filters can induce numerical instabilities, especially under finite word-length arithmetic and coefficient quantization. Therefore, careful design considerations, including stability analysis, scaling, and limit-cycle suppression, are essential. (Zhuang & Liu, 2023)

The trade-off between FIR and IIR architectures has given rise to hybrid filtering strategies in modern digital signal processing. By leveraging the strengths of both filter types, these hybrid designs efficiently meet increasingly demanding system requirements. For example, wireless mixed-signal receivers often implement phase-critical operations such as pulse shaping or channel equalization using baseband FIR filters, while relying on IIR filters for downstream broadband noise suppression and adaptive gain control. This approach optimizes both power consumption and latency for the overall receiver design. Similarly, wearable biomedical devices frequently employ FIR filters to remove artifacts and preserve signal morphology, while using IIR filters for smoothing data streams within constrained computational resources. Furthermore, advancements in design automation flows and high-level synthesis (HLS) tools now enable comprehensive exploration of the FIR-IIR trade-off space with unprecedented flexibility. These automated design frameworks support multi-objective optimization, minimizing latency, power, hardware usage, and spectral error, thus allowing customized filter topologies tailored to specific application requirements, surpassing traditional one-size-fits-all solutions. (Kaur et al., 2022; Fernández de Bulnes et al., 2020)

Latency and Real-Time Constraints: Latency constitutes a critical performance metric in a broad spectrum of real-time digital signal processing (DSP) systems, including radar-based sensing, autonomous robotics, industrial automation, and voice-driven user interfaces. In such systems, delay constraints directly impact system responsiveness, stability, and usability. However, the contribution of digital filters to overall latency is not a trivial function of filter order or structure alone; it is governed by an interplay of algorithmic design, hardware implementation characteristics, pipelining strategies, and system-level integration. Accurate estimation and optimization of latency must account for the full processing pipeline, including sampling rates, filter throughput, hardware memory access patterns, and scheduling delays, especially in tightly coupled systems such as radar signal chains and 5G edge nodes. For example, in vehicular radar applications, low-latency FIR filters are often preferred to meet stringent timing constraints for ego-motion estimation and object tracking (Sen et al., 2025). Likewise, centralized 5G processing units built on programmable data planes (e.g., Blink) employ optimized real-time DSP blocks to support ultra-reliable low-latency communication (URLLC) services (Rouili & Boutaba, 2025). These scenarios highlight the necessity of latency-aware filter design as a cornerstone of time-critical DSP deployments. (Carnevale, 2024)

FIR filters inherently exhibit long group delays proportional to their order, as they rely on a fixed number of past input samples defined by the length of their impulse response. High-order FIR designs, often necessary to achieve stringent frequency selectivity and exact linear phase characteristics, incur significant latency, which may be prohibitive in time-critical applications. While parallelism and pipelining in FPGA and ASIC platforms can reduce apparent delay by overlapping computations, the fundamental linear relationship between filter order and latency remains a key design constraint. (Achtenberg et al., 2024; Benois et al., 2018; Hu et al., 2023; Kishor & R, 2024)

In contrast, IIR filters, owing to their recursive structure and shorter effective impulse responses, generally exhibit lower nominal latency than FIR counterparts. However, this advantage comes with a trade-off: IIR filters introduce nonlinear phase characteristics, resulting in group delay dispersion across frequencies. This property renders them unsuitable for time-domain applications requiring phase consistency, such as sonar detection or high-resolution medical imaging, where accurate temporal alignment is critical. (Achtenberg et al., 2024; Maclair et al., 2006; MathWorks, n.d.)

Latency constraints are particularly prominent in edge computing, 5G and 6G communication, and autonomous systems, where real-time responsiveness is essential and system latencies must remain below the sub-millisecond threshold. In these domains, filter architecture decisions are guided by a complexity–latency–power trade-off, especially under tight resource budgets on embedded processors and FPGAs. To meet such requirements, designers increasingly employ adaptive and reconfigurable filter structures, optimized through high-level synthesis (HLS) and automated design space exploration. These frameworks support fine-grained parameterization of filter order, coefficient precision, and parallelism granularity, facilitating a tailored balance between latency and resource efficiency. (Sen et al., 2025; Hu et al., 2023; Kishor et al., 2024; Maclair et al., 2006; Rahman et al., 2012; MathWorks, n.d.)

Additionally, employing low-latency architectural approaches, such as cascade and lattice structures for IIR filters, and frequency-domain processing techniques for FIR filters, has become critical in next-generation digital front-end designs. Furthermore, multirate filtering techniques enhance latency efficiency by undersampling signals in less critical frequency bands, effectively reducing both computational complexity and processing delay. (Gadawe et al., 2024; Yousif & Abdulnabi, 2023)

Computational Complexity and Power Efficiency: Computational complexity constitutes a pivotal factor in the selection of filter architectures, especially in systems constrained by stringent real-time performance and limited computational resources. In the case of finite impulse response (FIR) filters, the complexity increases linearly with the filter order N, as each output sample entails (N+1) multiply-accumulate (MAC) operations. This linear relationship often becomes a performance bottleneck in real-time software-defined radio (SDR) systems, thereby necessitating the use of reconfigurable platforms and parallel multi-MAC architectures to enhance processing efficiency (Arivalagan et al., 2024). The computational burden is particularly significant in scenarios demanding sharp spectral transitions or wideband signal processing.

To address these challenges, several algorithmic optimizations have been proposed:

- **Symmetry Exploitation:** The symmetric nature of coefficients in linear-phase FIR filters can be strategically leveraged to nearly halve the number of required multiplications. This structural property facilitates a substantial reduction in computational load without compromising filter accuracy. This optimization is widely utilized in the implementation of interpolated and polyphase FIR filter structures. (Rao et al., 2023)
- **Polyphase Decomposition:** Partitioning the filter into a bank of parallel sub-filters, i.e., polyphase components, improves the efficiency of multirate signal processing. This decomposition significantly reduces computational overhead by enabling operations at a lower effective sampling rate, a particularly valuable feature in decimation and interpolation applications. (Matei & Chiper, 2024)
- **Multirate Signal Processing:** Given that different segments of a signal often exhibit distinct spectral properties, uniform-rate processing results in suboptimal resource utilization.

Multirate processing techniques dynamically adjust the sampling rate in accordance with the local spectral content, thus allowing computational resources to be allocated more efficiently to the most critical frequency bands. (Reddy & Singh, 2024)

In contrast, infinite impulse response (IIR) filters are capable of achieving similar frequency response characteristics using significantly fewer coefficients, which translates to a lower number of operations per output sample. However, this computational advantage comes with increased sensitivity to numerical effects. The recursive feedback structure of IIR filters amplifies quantization noise and round-off errors, potentially leading to instability. To ensure numerical robustness, designers must apply precise fixed-point arithmetic techniques, conduct comprehensive stability analysis, apply coefficient scaling, and implement safeguards against limit cycles and overflow conditions. The efficiency gains offered by IIR filters can only be fully realized when such numerical vulnerabilities are rigorously addressed. (Giannakopoulos & Perez, 2025; Mugisha et al., 2024; Yi et al., 2025)

Ensuring power efficiency is essential for modern embedded systems, particularly those deployed in battery-operated, wearable, and Internet of Things (IoT) devices. Achieving low-power operation requires a co-optimized approach that integrates both algorithm-level and hardware-level strategies. Several established techniques have been proposed to minimize energy consumption while maintaining high performance:

- **Multiplier-less Architectures:** Multipliers are among the most energy-intensive components in digital filters. To address this, multiplier-less architectures rely on shift-and-add operations and specialized coefficient encoding schemes, such as the Canonical Signed Digit (CSD) representation. These techniques eliminate explicit multiplication, thereby reducing both power consumption and hardware complexity. Such designs are particularly advantageous in energy-constrained and high-throughput systems. (Song et al., 2024)
- **Coefficient Quantization and Pruning:** Reducing the bit-width of filter coefficients through quantization, and eliminating insignificant coefficients via magnitude-based pruning, are effective strategies for lowering dynamic power. These approaches reduce memory footprint and switching activity, making them ideal for low-power digital signal processing (DSP) applications. Nevertheless, maintaining acceptable filter performance requires careful trade-off analysis to mitigate the impact of reduced numerical precision. (Mugisha et al., 2024)
- **Hardware-Aware Optimization:** Advanced hardware-aware techniques, such as loop unrolling, pipelining, and resource sharing, can be customized for specific computing platforms (e.g., CPUs, DSPs, FPGAs) to improve both throughput and energy-per-operation metrics. For example, Liao et al. (2025) introduced a system-level high-level synthesis (HLS) design space exploration (DSE) framework that integrates end-to-end (EtoE) latency constraints into the optimization process. Their methodology combines latency modeling and a path-finding algorithm to estimate EtoE delays, followed by frequency-based segmentation, pruning, and latency-constrained optimization to explore the design space effectively. Applied to an autonomous driving subsystem, this approach yielded up to an 89.26% improvement in energy–area Pareto-optimal outcomes compared to prior HLS DSE techniques (Liao et al., 2025), as shown in Figures 19–21 and Table 6.

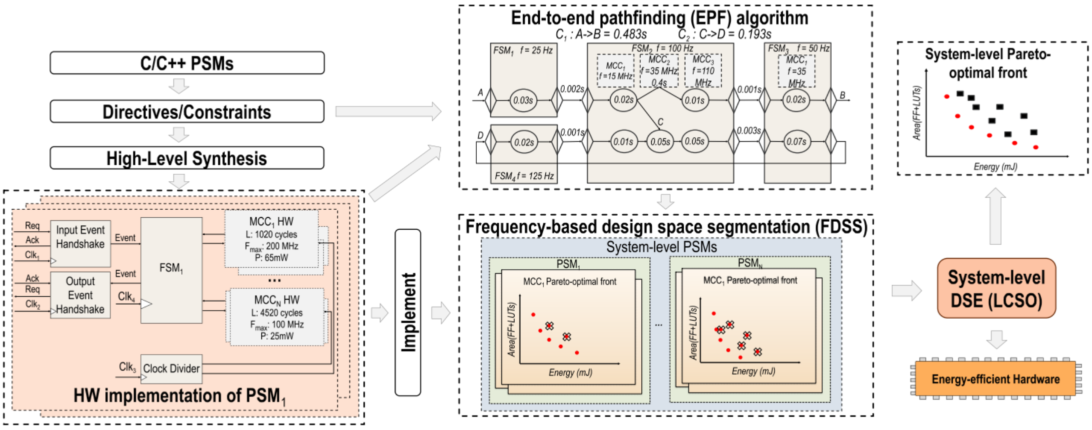


Figure 19. The proposed End-to-End Design Space Exploration (EtoE-DSE) methodology to find system-level Pareto-optimal configurations while meeting the variable timing and end-to-end (EtoE) latency constraints. (Liao et al, 2025)

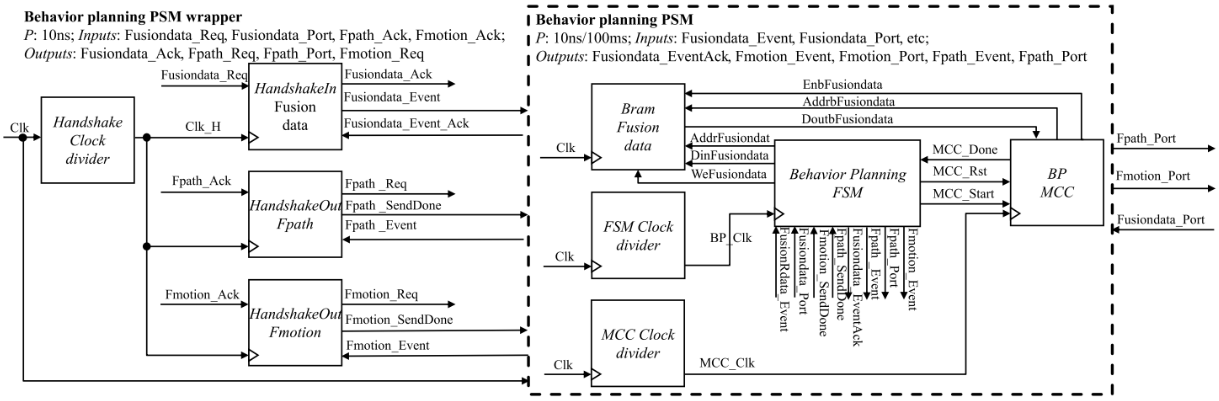


Figure 20. A block diagram for the behavior planning PSM in an autonomous driving subsystem to illustrate the relationship between PSM, FSM, MCC, and both HandshakeIn and HandshakeOut components. (Liao et al, 2025)

Table 6. ADS's design space sizes after applying FDSS algorithm with different segment threshold. The original design space is 5.291e+128. (Liao et al, 2025)

Segment	After FDSS	Improvement
44	2.469e+73	2.142e+55x
22	2.456e+73	2.154e+55x
11	2.443e+73	2.165e+55x
Geo. mean	2.45629e+73	2.15389e+55x

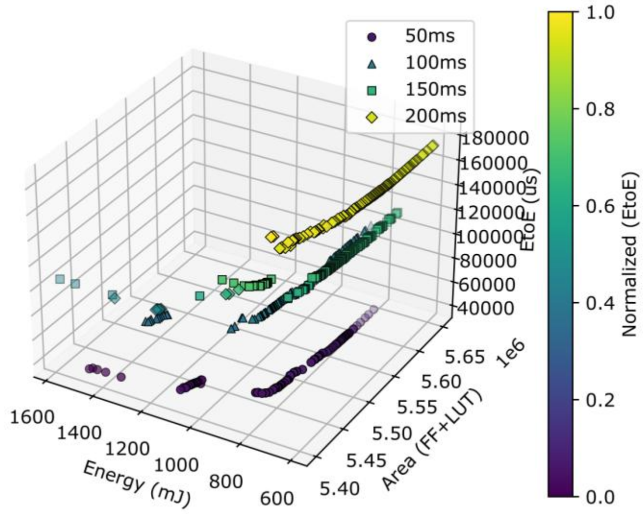


Figure 21. Visualized 3D comparison of the system-level Pareto-optimal fronts of EtoE-DSE under 50 ms, 100 ms, 150 ms, and 200 ms end-to-end (EtoE) latency constraint. Different level of color in color bar represents the EtoE latency for one of the paths between GPSsensor and Control PSMs. (Liao et al, 2025)

Implementation on CPU, DSP, and FPGA

CPU Implementations: General-purpose CPUs provide a highly flexible software environment that facilitates rapid prototyping and execution of a wide range of filtering algorithms without the need for hardware-level reconfiguration. This flexibility is enhanced by vectorized SIMD instruction sets, such as Intel AVX-512 and ARM NEON, that enable partial parallelization of filtering computations, thereby increasing throughput especially for filters of moderate order. However, due to their fundamentally sequential execution model and dependence on complex cache hierarchies, CPUs are susceptible to pipeline stalls, memory access latencies, and non-deterministic low-level behaviors. These limitations pose significant challenges in real-time signal processing. Additionally, operating system–induced jitter and context-switching overhead further complicate the fulfillment of stringent real-time requirements. Consequently, while CPUs offer ease of programmability, they often fail to meet the ultra-low-latency and high-throughput demands inherent to modern digital filtering tasks. (Michon et al., 2025)

DSP Implementations: Digital Signal Processors (DSPs) are architected specifically for signal processing applications, featuring multiple multiply-accumulate (MAC) units, zero-overhead looping, and circular buffering. These capabilities reduce instruction counts and processing latency, resulting in predictable, low-latency execution suitable for hard real-time deadlines. Although most DSPs target fixed-point arithmetic, some support floating-point operations. Their compact form factor and energy efficiency make DSPs ideal for portable audio devices, instrumentation, and sensor-interface conditioning. However, limited parallelism and relatively small architectural scale constrain DSPs when scaling to multi-channel or ultra-high-order filtering, often requiring higher clock frequencies or multicore designs to meet increased computational demands. (Samakovlis et al., 2024)

FPGA Implementations: Field-Programmable Gate Arrays (FPGAs) exploit fine-grained hardware-level parallelism to accelerate data and digitized video filtering at data rates reaching gigabits per second, making them especially suitable for complex video filter designs. For instance, FIR filters with fully unrolled feedforward coefficients or systolic MAC arrays can compute all filter taps concurrently, reducing latency to the logic propagation delay and eliminating the bottleneck of sequential instruction execution. Designers can select arbitrary datapath widths and precisions, including fixed-point or floating-point arithmetic, balancing resource utilization against numerical accuracy. The reconfigurability of FPGAs also enables dynamic, adaptive filtering architectures with time-varying parameters. IIR filter implementations on FPGAs require careful pipelining to avoid timing feedback loops and ensure numerical stability. Resource-sharing techniques, such as time-multiplexing pipeline stages, maximize logic utilization without sacrificing throughput. Modern FPGAs incorporate dedicated DSP slices (e.g., Xilinx DSP48) optimized for fast MAC operations and accumulation. High-Level Synthesis (HLS) tools accelerate development by translating high-level languages (C/C++) into optimized hardware descriptions, enabling design space exploration of pipeline depths, loop unrolling factors, and bit-widths to optimize latency, area, and power trade-offs.

For example, Ibrahim et al. (2025) demonstrated a Direct Form II IIR filter implementation on a Cyclone IV GX FPGA utilizing a BCD-multiplier architecture. This design outperformed array, Booth, Vedic, and MUX multiplier architectures, using only 188 logic elements, achieving the lowest propagation delay (7.89 ns) and highest clock frequency (282.8 MHz), as illustrated in Figures 22-24 and Tables 7, 8 (Ibrahim et al., 2025; Achtenberg et al., 2024; Michon et al., 2025).

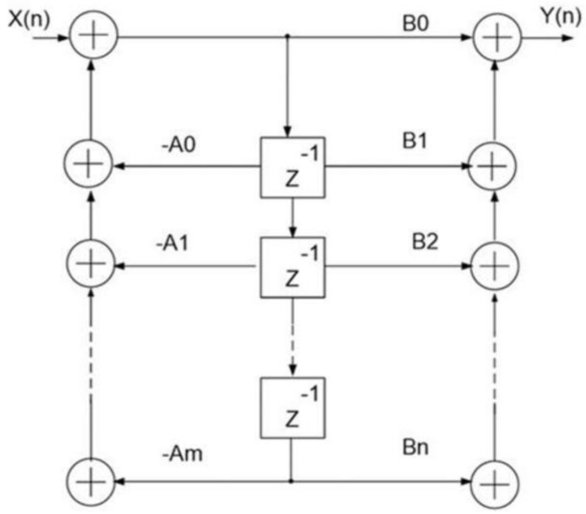


Figure 22. Realization digital IIR system using direct form. (Ibrahim et al, 2025)

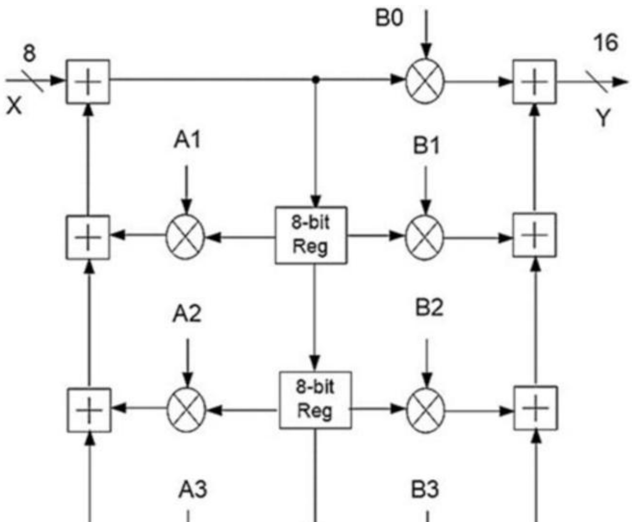


Figure 23. Block circuit diagram of the digital IIR filter. (Ibrahim et al, 2025)

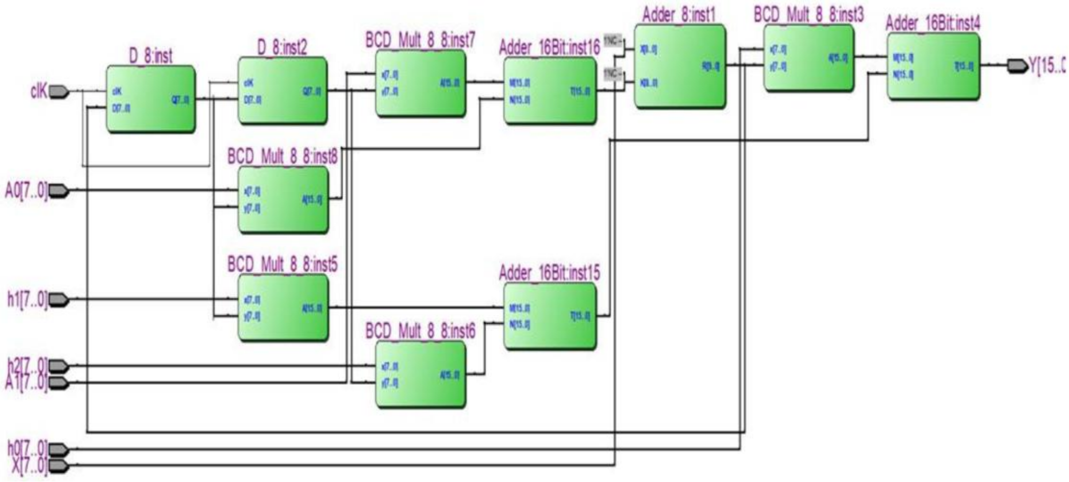


Figure 24. Register transfer level (RTL) viewer of the designed IIR filter. (Ibrahim et al, 2025)

Table 7. Multiple multipliers circuit applied to the IIR filter design. (Ibrahim et al, 2025)

Multiplier	Technique	Logic Elements	Freq. Operation (MHz)	Power Dissipation (mW)	Delay (ηSec)
Array multiplier	Multi-layer adders	839	88.5	89.23	10.51
Parallel multiplier using two circuits	Booth & parallel	828	121.05	89.13	10.39
Vedic multiplier	Vedic mathematics	661	131.2	88.37	9.38
Mux-multiplier	Approx. multiplexing	686	78.24	89.23	10.71
BCD-multiplier	BCD decoding	188	282.8	88.32	7.89

Table 8. Comparison of the proposed IIR filter based on BCD-multiplier compared to other works. (Ibrahim et al, 2025)

Multiplier	Technique	Freq. Operation (MHz)	Logic Elements	Delay (ηSec)
[35]	Virtex-5	32.63	854	38.02
[36]	NA	NA	1582	30.74
[37]	Zynq	53.51	1799	18.68
Proposed	Cyclone IV	282.8	188	7.89

Comparative Characteristics: The choice of the best filter type and realization platform is a complex and multi-criteria evaluation of both technical and practical criteria. Every parameter will make a difference to system performance, and these can be of different order of importance depending on the domain of application. (Khan et al., 2024)

- **Stability:** Stability ensures that a filter's output is predictable and free from oscillations or divergence. FIR filters are inherently stable due to their feedforward nature, while IIR filters have conditional stability that depends on design and coefficient quantization, especially important in hardware implementations with limited precision. Careful design and quantization are vital to ensure IIR filter stability in practice. (Lai & Zhao, 2024; Yi et al., 2025; Jaiswal & Mitra, 2024)

- **Phase Response:** In applications where waveform shape and timing integrity are critical, such as biomedical signal processing and telecommunications, a linear phase response is essential. FIR filters inherently provide an accurate linear phase, while IIR filters generally introduce nonlinear phase distortions. (Achtenberg et al., 2024; Jaiswal & Mitra, 2024)
- **Computational Complexity:** Computational complexity refers to the number of operations required per output sample. Due to their recursive structure, commonly used IIR filters require fewer coefficients and operations, leading to better computational efficiency. In contrast, FIR filters, especially those with sharp frequency responses, are generally more computationally demanding unless optimized design techniques are applied .(Giannakopoulos & Perez, 2025)
- **Latency:** In real-time applications, latency is a critical parameter. FIR filters generally exhibit longer impulse responses, resulting in higher latency. In contrast, IIR filters typically offer lower latency due to their feedback loop structures. (Chang et al., 2025; Michon et al., 2025)
- **Power Consumption:** Power consumption is closely linked to the number of operations performed by the filter. FPGA-based designs typically offer lower power consumption and reduced costs compared to CPU implementations, as FPGAs can be hardware-optimized for specific tasks. Digital Signal Processors (DSPs) generally fall between CPUs and FPGAs in terms of power efficiency. (Giannakopoulos & Perez, 2025; Khan et al., 2024; Zheng et al., 2025)
- **Reconfigurability:** Dynamic adaptability of filter parameters and structures is essential for modern adaptive systems. Among various implementation platforms, FPGAs offer the highest degree of reconfigurability. While processors such as CPUs provide software flexibility, they are limited by processing power constraints. Digital Signal Processors (DSPs) offer a moderate level of configurability, falling between CPUs and FPGAs in this regard. (Giannakopoulos & Perez, 2025; Khan et al., 2024)
- **Flexibility of development:** The ease and speed of algorithm development and subsequent enhancements depend largely on the implementation platform. CPUs offer the highest flexibility due to their software-centric environment but are limited by overhead in processing real-world signals. DSPs provide a balanced compromise between development complexity and achievable performance. FPGAs, while the most challenging platform to develop for, deliver the greatest raw computational power. (Giannakopoulos & Perez, 2025; Zheng et al., 2025)
- **Filter Parallel Scalability:** Filter parallel scalability refers to the extent to which a platform supports parallel processing of filtering operations to improve speed. FPGAs excel in this area due to their inherently parallel hardware architecture, while CPUs and DSPs have limited parallel processing capabilities constrained by their design. (Sadeghi, 2024; Michon et al., 2025)
- **Deterministic Timing:** Deterministic timing ensures precise and consistent input-to-output delays, which is critical for hard real-time applications. FPGAs deliver the highest timing determinism due to their predictable hardware behavior. DSPs provide moderate

determinism, while CPUs typically exhibit the lowest due to operating system overhead and task scheduling. (Michon et al., 2025; Khan et al., 2024)

Tables 9 and 10 offer a detailed comparison of FIR and IIR filters based on key technical and practical criteria, showing how CPUs, DSPs, and FPGAs influence their performance. This summary aids in making informed choices for specific application requirements.

Table 9. Comparative Characteristics of Filters and Implementation Platforms

Criterion	FIR Filters	IIR Filters	CPU Implementation	DSP Implementation	FPGA Implementation
Stability	Always stable	Conditionally stable	High	Medium	High
Phase Response	Exact linear phase	Nonlinear phase	Yes	Yes	Yes
Computational Efficiency	Lower (higher order)	Higher (fewer coefficients)	Medium	High	Very high
Latency	Higher (proportional to order)	Lower (feedback-based)	High	Medium	Very low
Power Consumption	Higher	Lower	High	Medium	Low
Reconfigurability	Moderate	Moderate	High	Medium	Very high
Development Flexibility	High	High	Very high	High	Moderate

Table 10. Explanations of Filter Characteristics and Implementation Platforms

Criterion	FIR Filters	IIR Filters	CPU Implementation	DSP Implementation	FPGA Implementation
Stability	Always stable due to non-recursive structure.	Conditionally stable, can become unstable.	High precision, stable processing.	Medium stability, may have limitations.	High stability with hardware optimizations.
Phase Response	Exact linear phase, no timing distortion.	Nonlinear phase, potential signal distortion.	Controlled, but less precise.	Handles both linear and nonlinear, but with effort.	Precise control over phase responses.
Efficiency	Lower efficiency due to more coefficients.	More efficient, requires fewer coefficients.	Less optimized, lower efficiency.	Optimized for signal processing, more efficient.	Parallel processing greatly increases efficiency.
Latency	Higher latency, increases with filter order.	Lower latency, faster signal processing.	Higher latency due to software overhead.	Medium latency, optimized for real-time processing.	Minimal latency due to parallel hardware processing.
Power Consumption	Higher power consumption, especially with high orders.	Lower power due to fewer coefficients.	Higher power consumption for complex tasks.	More power-efficient than CPUs.	Low power consumption, highly efficient.
Reconfigurability	Limited flexibility, can change coefficients.	Requires careful tuning to avoid instability.	Highly reconfigurable with software.	Moderate reconfigurability, but less than CPUs.	Extensive reconfiguration for custom optimizations.
Design Flexibility	Easy to design and modify.	Complex design, higher flexibility.	Most flexible for implementing various algorithms.	Fairly flexible but less than CPUs.	Less flexible than CPUs but highly optimized.

Challenges and Emerging Trends

Real-Time and Hardware Implementation Challenges: Designing filters for real-time applications remains one of the most demanding areas in signal processing, particularly as latency constraints tighten and performance requirements escalate. Critical domains such as autonomous vehicles, telerobotics, defense radar systems, and ultra-reliable low-latency communications (URLLC) demand deterministic behavior with response times often within tens of microseconds. (Maghsoudnia et al., 2024; Yan & Roberts, 2025; Kim & Hao, 2025)

Achieving such responsiveness introduces several architectural and implementation-level challenges. General-purpose processors (GPPs), though flexible and programmer-friendly, suffer from limited parallelism and nondeterministic latency due to OS overhead and shared memory architectures. Consequently, they struggle to meet the real-time needs of high-throughput and adaptive filtering tasks. In contrast, FPGAs and ASICs are increasingly favored for their ability to deliver fine-grain parallelism, pipelined structures, and deterministic scheduling. Specifically, FPGAs support cycle-accurate filtering pipelines, making them ideal for tight control loops in embedded systems. However, these benefits come at the cost of increased design complexity, requiring hardware description languages (HDLs), stringent timing closure, and reliance on vendor-specific toolchains. (Michon et al., 2025; Dai et al., 2025)

Another pressing issue is seamless integration with surrounding components. Synchronization with ADC and DAC interfaces is crucial in multirate or oversampled systems; improper clock domain crossings or buffer mismatches can result in data loss or timing violations. Moreover, filter cores must be tightly coupled with control logic, such as PID controllers or state machines, necessitating a co-design methodology to ensure functional coherence and throughput guarantees (Choi et al., 2025). As system complexity grows, in-field debugging becomes more difficult, with simulation often failing to capture timing anomalies, thus requiring in-system profiling, logic analyzers, and hardware-in-the-loop (HIL) testing. Ensuring numerical stability and fixed-point precision, especially in recursive IIR filters, remains a persistent concern. (Herrou et al., 2024; Tang, 2025)

Recent developments have sought to alleviate some of these barriers:

- High-Level Synthesis (HLS) tools allow filter logic to be described in C/C++, then compiled into hardware with optimized performance.
- Real-time co-simulation environments facilitate iterative validation.
- Reconfigurable architectures with partial reprogramming capabilities balance adaptability and timing determinism.

Despite these advancements, a trade-off persists between development productivity and execution determinism. Hardware-aware filter design thus requires meticulous cross-disciplinary collaboration among DSP engineers, embedded software developers, and digital hardware architects. (Kim & Hao, 2025; Tang, 2025; Lahti & Hämäläinen, 2025; Alachiotis et al., 2025)

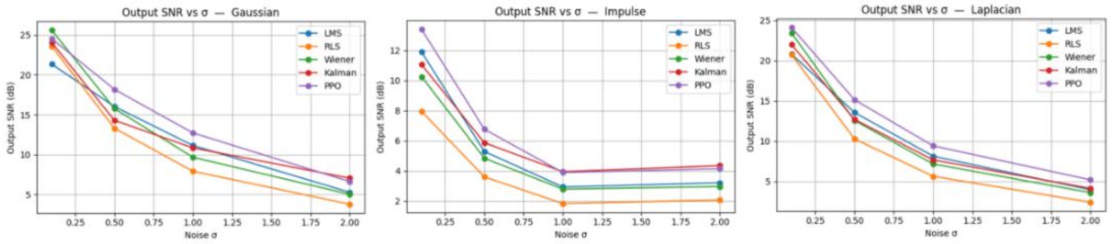
Adaptive Filtering and Machine Learning Integration: Adaptive filtering is essential in dynamic or non-stationary environments where fixed filters fail to track time-varying signal or noise characteristics. Traditional adaptive algorithms such as Least Mean Squares (LMS) and Recursive Least Squares (RLS) remain foundational due to their simplicity and rapid convergence (Igual et al., 2025). For example, RLS-based filters have proven effective in removing noise from physiological signals like ECG, achieving significant interference rejection (Mahmood et al., 2024). Classical adaptive filters continue to be widely used in real-time applications such as echo cancellation, channel equalization, and biomedical signal tracking owing to their efficiency and simplicity. (Zhang et al., 2025)

Modern deep learning models, including Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Temporal Convolutional Networks (TCNs), capture complex temporal dependencies beyond linear filter capabilities. An LSTM-based neural equalizer, for instance, has outperformed conventional LMS and RLS equalizers in challenging channel equalization tasks (Zhang et al., 2025). TCNs have enhanced EEG-based emotion recognition accuracy (Chen et al., 2025), and a TCN-based autoencoder designed for radar interference mitigation demonstrated superior noise suppression compared to traditional CNN autoencoders by explicitly modeling temporal correlations (Thornton et al., 2025). These data-driven models adapt to nonlinearities and non-stationary signals by learning underlying signal structures.

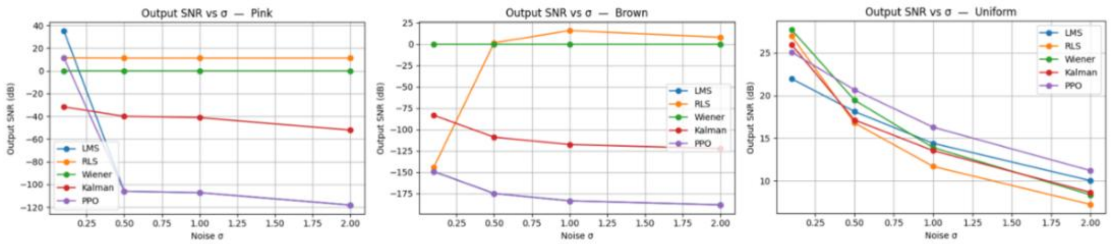
Deep denoising autoencoders and convolutional neural networks (CNNs) have been effective in reducing nonstationary noise in practical settings. A recent dual-pathway autoencoder (DPAE) outperformed previous deep models in EEG artifact removal with substantially lower error rates (Xiong et al., 2024). Similarly, neural multi-channel speech enhancers, typically based on CNNs, have significantly surpassed classical methods in audio processing (Xu et al., 2024). CNNs have also been applied in mechanical and biomedical domains; for example, adaptive CNN autoencoders improve vibration signal denoising, while CNN classifiers attain high accuracy in ECG anomaly detection under low signal-to-noise ratio (SNR) conditions. Integrating CNNs with wavelet transform layers facilitates frequency-aware feature extraction, yielding robust ECG reconstruction despite severe noise (Erada & Toyoura, 2025). Overall, data-driven filters better preserve essential signal features and reduce distortion compared to fixed filters, making them especially valuable for anomaly detection and signal enhancement on wearable and edge devices where signal conditions fluctuate. (Hizem et al., 2025)

Deploying deep models on embedded hardware is challenging due to high memory and computational demands. To address this, practitioners apply model compression techniques such as pruning (removing redundant weights) and quantization (reducing numerical precision) to shrink model size and speed up inference without degrading accuracy. For example, an on-device anomaly detector used aggressive pruning and quantization to minimize resource use while maintaining performance (Hizem et al., 2025). Hardware-aware optimizations, including post-training quantization and custom compiler toolchains, enable efficient execution of compact models on FPGAs, microcontrollers, or tensor processing units (TPUs), mitigating resource constraints for real-time deep filtering. (Khan et al., 2025)

Reinforcement learning (RL) has introduced new paradigms for self-optimizing filters, where agents dynamically adapt filter coefficients based on environmental feedback to maximize signal quality or system response. Unlike classical adaptive filters that rely on deterministic update rules, RL-based methods learn generalizable policies across tasks and domains. These approaches have been explored in intelligent hearing aids, adaptive antenna arrays, and cognitive radio front-ends. For instance, Bereketoglu (2025) proposed a real-time Proximal Policy Optimization (PPO)-based adaptive filter using a composite reward function to optimize signal-to-noise ratio (SNR), mean squared error (MSE), and smoothness, outperforming LMS, RLS, Wiener, and Kalman filters in non-stationary noise environments, as demonstrated in Figures 25–27.



(a) Gaussian noise (b) Impulse noise (c) Laplacian noise



(d) Pink noise (e) Brown noise (f) Uniform noise

Figure 25. Output SNR vs. noise level across various unseen noise types. PPO-based filter generalizes robustly beyond its training distribution. (Bereketoglu, 2025)

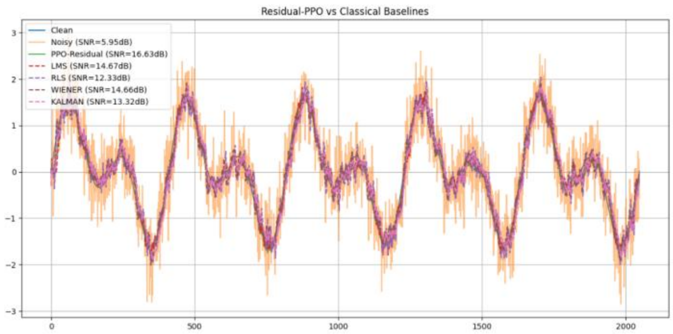
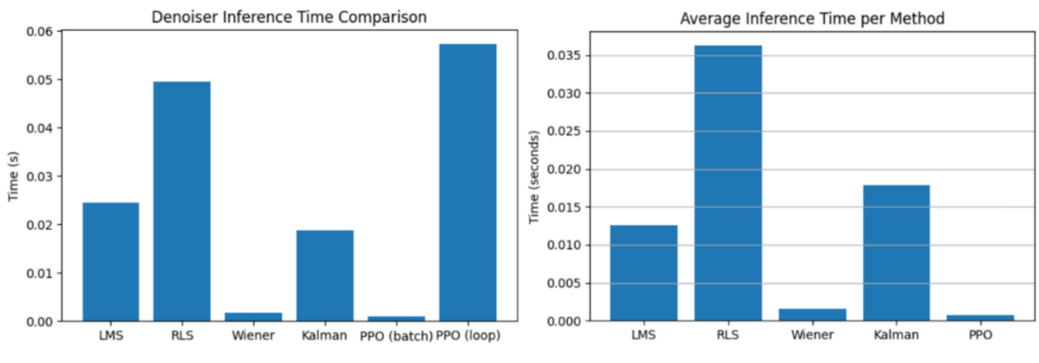


Figure 26. Residual comparison across methods. PPO achieves the most accurate reconstruction with the lowest residual noise. (Bereketoglu, 2025)



(a) Latency comparison of methods (b) Average inference time per method

Figure 27. Inference performance of different filters. PPO demonstrates fast real-time filtering capability across both comparison metrics. (Bereketoglu, 2025)

Few-shot and continual learning methods enable on-device adaptation using minimal new data, which is vital for personalized monitoring applications, such as biomedical or structural health monitoring, where devices must recognize new signal classes from very few examples (Denison et al., 2024). For instance, a recent wearable gesture-recognition system employed a few-shot continual learning approach to learn new hand gestures with only one to five labeled samples (Rafiq et al., 2024). Lightweight models like Support Vector Machines (SVMs) are preferred on edge devices due to their low memory footprint and rapid updatability with few support examples. These on-device learning strategies allow systems to adapt to individual users or novel conditions without relying on continuous cloud-based retraining, thereby enhancing responsiveness and preserving privacy.

As the field evolves, practical implementations increasingly blend classical adaptive filters with compact neural modules to balance interpretability and flexibility. For example, a hybrid CNN+LMS filter demonstrated superior EEG denoising compared to either component alone, improving signal quality while reducing silicon area by 77% and power consumption by 69.1% relative to prior designs (Nair et al., 2025). Such hybrid architectures typically use a baseline adaptive filter (e.g., LMS) for steady-state tracking, complemented by a lightweight neural network that corrects residual errors or dynamically adjusts the learning rate. These solutions offer low-latency updates with modest hardware costs, preserving much of the simplicity and stability inherent in classical filtering. (Arivalagan et al., 2024)

Reconfigurable Architectures: As digital systems increasingly operate within dynamic, multi-standard, and mission-critical environments, reconfigurability has become essential for ensuring long-term adaptability and sustainability. Reconfigurable architectures enable in-field updates to filter behavior, ranging from coefficient modifications to topological alterations, without the need for complete redesigns (Oliveira et al., 2024; Arivalagan et al., 2024). This architectural agility extends system longevity and facilitates real-time adaptability. For instance, FPGAs can dynamically adjust filter parameters at runtime, and more advanced implementations leverage Partial Reconfiguration (PR), wherein specific regions of the FPGA fabric are reprogrammed while the remainder continues to operate uninterrupted. Such capabilities support resource time-multiplexing, adaptive pipeline tuning, and dynamic switching between filter topologies (e.g., FIR and IIR) in response to changing operational requirements. (Fahmy et al., 2025; Uko et al., 2025)

In Software-Defined Radio (SDR) systems, reconfigurable filters are critical for enabling multi-standard operability. SDR platforms can dynamically modify modulation schemes, bandwidths, and noise suppression mechanisms by retuning or reconfiguring filter blocks in hardware or firmware. For example, a base station front-end can reprogram its filters to support LTE today and transition to 5G NR or 6G standards tomorrow, without requiring hardware replacement. Such multi-band flexibility is particularly vital in cognitive radio systems, where spectrum sensing and real-time adaptive filtering are necessary to avoid interference and maximize spectral efficiency. Similar adaptability is now pivotal in aerospace and defense domains, where mission profiles often change mid-operation. Satellites leverage in-orbit reconfiguration to optimize filtering performance based on positional dynamics or updated mission parameters. Likewise, UAVs and electronic warfare systems utilize dynamic filtering schemes to enhance resilience, stealth, and responsiveness. (Fahmy & Iyer, 2024; Oliveira et al., 2024)

Recent advances are also extending reconfigurability into the analog domain. Mixed-signal front-ends are being developed wherein digital controllers adjust analog filter behavior via DACs or switched-capacitor networks. For instance, a novel flash-DAC-based design implements a six-tap FIR filter entirely in the analog domain, using flash cells to store coefficients, enabling ultra-low-power and area-efficient channelization for IoT devices (Lee et al., 2025). In biomedical applications, tunable Gm–C filter banks have been demonstrated. One such CMOS analog biquad filter provides independent on-chip control over both center frequency and quality factor (Q), achieving a 0.5 V, 40 nW filter bank suitable for cochlear implants (Jendernalik & Jakusz, 2024). ADC architectures are also evolving; a recent voltage-controlled oscillator (VCO)-based ADC integrates reconfigurable quantization to emphasize neural spikes, functioning as a nonlinear analog preprocessor that enhances SNR. This analog pre-filtering stage significantly reduces downstream digital processing load. Emerging "analog-to-feature" front-ends further advance this trend by extracting key signal features through nonuniform sampling before digitization, offering new paradigms in low-power signal acquisition. (Manokhin et al., 2024)

To provide a clearer comparative perspective, Table 11 presents a detailed evaluation of various FPGA-based reconfigurable filtering architectures. It highlights how different design paradigms navigate trade-offs among flexibility, design complexity, power efficiency, and domain-specific applicability.

Table 11. Comparison of Reconfigurable Filter Design Paradigms in Modern Embedded Systems

Feature / Criterion	Static FPGA Design	Partial Reconfiguration	Dynamic Reconfiguration	Hybrid Mixed-Signal Architectures
Definition	Fixed hardware configuration	Ability to update part of the FPGA configuration at runtime	Real-time dynamic changes to hardware behavior without halting the system	Integration of digital control with reconfigurable analog filters
Flexibility	Low (fixed functionality, no runtime changes)	Moderate (can reconfigure part of the design)	High (supports real-time updates and dynamic changes)	High (flexible integration of digital and analog systems)
Reconfiguration Time	Long (requires full bitstream reload)	Fast (partial bitstream update)	Very fast (dynamic switching without halting the system)	Fast (quick updates due to hybrid nature)
Design Complexity	Low (simpler design, no runtime changes)	Moderate (requires management of partial reconfiguration)	High (requires advanced control systems for dynamic updates)	Very High (requires multidisciplinary expertise)
Power Consumption	Constant (power consumption remains the same)	Moderate (depends on reconfiguration frequency)	Optimized (due to dynamic changes adjusting power usage)	Optimized (multi-level control for power efficiency)
Reliability	High (stable and predictable performance)	High (partial reconfiguration is safe with proper validation)	Challenging (complex control mechanisms may introduce errors)	High (reliable with monitoring systems in place)
Typical Applications	Static systems, legacy protocols	Software-defined radios, military communications	UAVs, time-critical embedded systems	IoT devices, biomedical equipment, wearable systems

The Role of FPGA in Reconfigurable and Adaptive Filtering: Field-Programmable Gate Arrays (FPGAs) offer exceptional on-chip parallelism, reconfigurability, and ultra-low-latency execution, making them highly suitable for adaptive filtering. These devices support dynamic, real-time reconfiguration and enable massive parallel processing pipelines that outperform CPU-based implementations in latency and efficiency (Shi et al., 2024; Sravanthi et al., 2024). FPGAs are increasingly adopted for digital signal processing tasks, ranging from image and audio filtering to neural networks, owing to their reprogrammable fabric and dedicated DSP blocks, which facilitate energy-efficient adaptive filter designs. Unlike fixed-function ASICs or general-purpose GPUs, FPGAs allow for on-the-fly adaptation of filter parameters and even topologies, enabling real-time responsiveness to environmental changes such as noise fluctuations, channel dynamics, or evolving application requirements. (Achtenberg et al., 2024)

The complexity of FPGA-based adaptive filter design has been significantly reduced by the advent of High-Level Synthesis (HLS) tools. Modern HLS workflows translate C and C++ algorithms into optimized HDL, eliminating much of the manual low-level coding and accelerating the development cycle. Researchers have noted that HLS bridges the gap between software and hardware design paradigms, empowering software engineers to implement high-performance FPGA accelerators without deep expertise in HDL. Recent studies show that HLS can generate efficient, pipelined filter IPs with minimal effort, enabling rapid exploration of latency–throughput trade-offs and shortening time-to-deployment (Curzel, 2024; Berrazueta-Mena & Navas, 2025). This democratization of FPGA development opens access to hardware-level performance benefits while preserving fine-grained control through compiler directives and pragmas.

Contemporary FPGA platforms further enhance adaptability by integrating ARM CPUs, DSP slices, and AI accelerators on a single chip. System-on-Chip (SoC) FPGAs such as Xilinx Zynq and Intel Agilex combine programmable logic with general-purpose processors and domain-specific cores, allowing the development of hybrid adaptive filtering systems (Berrazueta-Mena & Navas, 2025). In practical use cases, the FPGA fabric can implement deterministic signal-processing pipelines, while the on-chip CPU or NPU manages control logic, coefficient adaptation, or learning-based signal adjustments. These hybrid architectures are now pervasive in application domains such as autonomous systems, edge AI, and IoT, where reconfigurable parallel processing is critical. Notably, recent reports indicate that FPGAs are beginning to surpass traditional DSPs in performance due to their vast parallelism and tightly integrated memory resources. Modern FPGAs house hundreds of DSP slices and embedded neural processing units (NPUs), enabling low-latency execution of complex filters and sensor-fusion algorithms. For instance, Berrazueta-Mena and Navas (2025) introduced the AHA IP Cores, five HLS-based accelerators (matrix multiplication, FFT, AES, backpropagation neural networks, and artificial neural networks) targeting Zynq-based SoCs. Their study showed that ANN and BPNN achieved the highest degrees of parallelism, AES demonstrated the most efficient resource usage, matrix multiplication offered promising optimization potential, and FFT was primarily limited by inherent data dependencies, as shown in Figures 28-31 (Berrazueta-Mena & Navas, 2025; Khan et al., 2024).

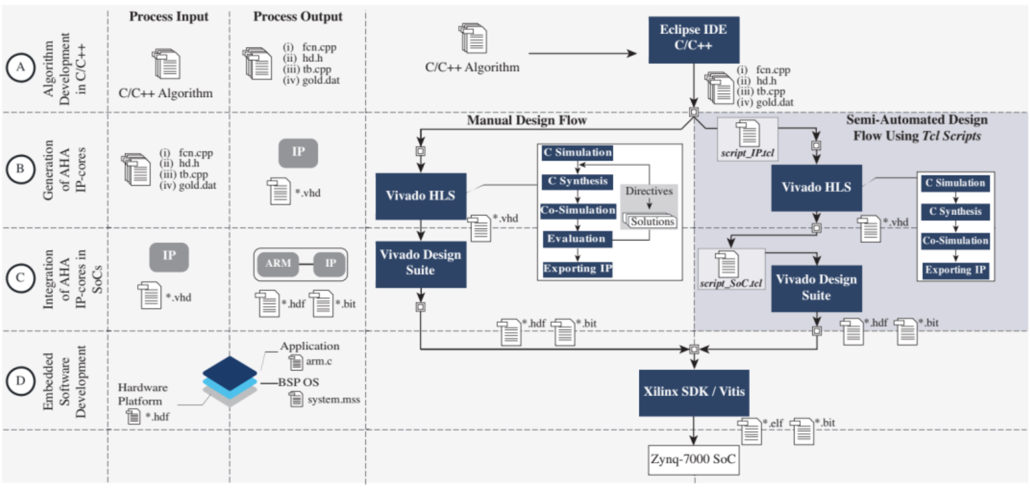


Figure 28. Design flow for developing Zynq SoCs with AHA IP-cores. This flow comprises four principal processes (**A**–**D**). (Berrazueta-Mena & Navas, 2025)

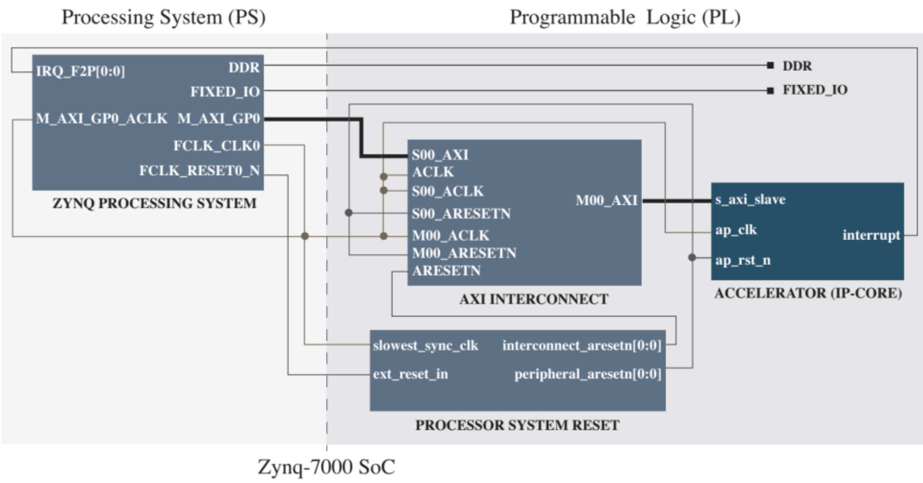


Figure 29. A simplified block diagram illustrating the fundamental heterogeneous SoC architecture used to evaluate the performance of the Zynq-7000 SoCs. (Berrazueta-Mena & Navas, 2025)

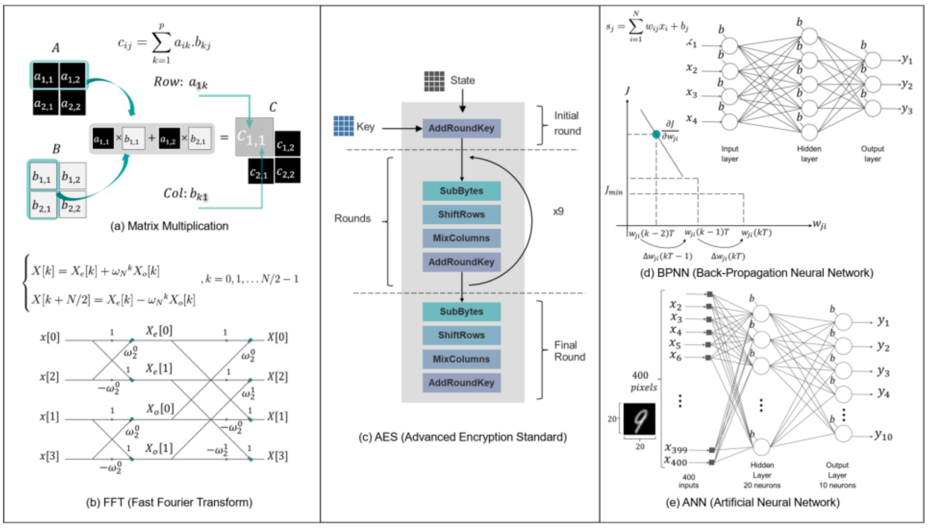


Figure 30. AHA IP core Suite. Basic principle of the algorithm implemented by each IP core. (Berrazueta-Mena & Navas, 2025)

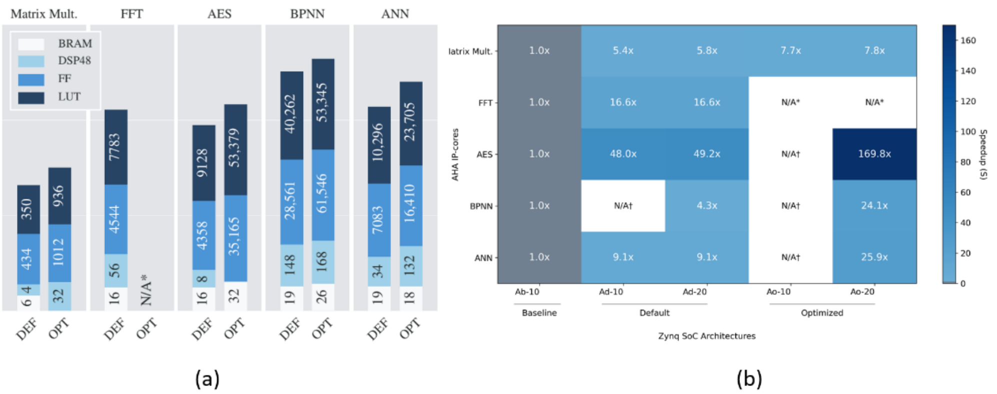


Figure 31. (a) Experiment 1—FPGA resource utilization of the default (DEF) and optimized (OPT) standalone IP cores generated in Vivado HLS. (b) Experiment 2—Speedup (S) of the Zynq SoC architectures with AHA IP-cores. (Berrazueta-Mena & Navas, 2025)

Sophisticated FPGA architectures leverage dynamic partial reconfiguration (DPR) to modify filter configurations at runtime without halting device operation. In this approach, a designated region of the FPGA fabric is reprogrammed while the remainder continues operating, enabling ultra-fast updates to filter coefficients or even entire topologies. Experts emphasize that DPR provides a powerful and flexible mechanism for deploying multiple accelerators and dynamically adapting system functionality on the fly (MDPI, 2025). This capability is particularly vital in mission-critical applications, for instance, radar and wireless communication systems can seamlessly swap filter kernels or tuning parameters in real time to track moving targets or adapt to dynamic channel conditions, while biomedical devices can reconfigure signal conditioning circuits in response to changes in patient physiology, all with minimal disruption. (Taghavirashidizadeh et al., 2024; El Bouazzaoui et al., 2023)

From a practical standpoint, implementing DPR-based adaptive filtering on FPGAs involves overcoming several technical challenges, including:

- Achieving deterministic timing closure during dynamic reconfiguration
- Ensuring efficient memory management for real-time coefficient storage and updates
- Applying power optimization strategies to maintain energy efficiency during frequent reconfigurations
- Developing robust verification and validation frameworks for dynamically evolving filter architectures

These concerns have been extensively addressed in recent FPGA literature. For example, real-time reconfigurable designs require predictable access patterns to I/O and memory to guarantee deterministic behavior (El Bouazzaoui et al., 2024). Moreover, power efficiency and system reliability are critical; studies in ML-driven reconfiguration demonstrate that redundancy mechanisms and scrubbing techniques enhance fault tolerance during dynamic modifications of filter structures (Khan et al., 2024; Taghavirashidizadeh et al., 2024).

Future Trends and Research Directions

The filter design landscape is changing fast with the impact of new technologies and growing complexity in applications. Some key trends are going to reshape the theoretical foundations as well as practical aspects of filtering systems:

Co-Design and Holistic Optimization Frameworks: Future filter design increasingly embraces holistic co-optimization frameworks that jointly refine algorithmic, model-level, and hardware-specific parameters. For instance, an RLS adaptive filter co-designed with FPGA hardware achieved an 89.78% SNR improvement in ECG denoising (Mahmood et al., 2024), highlighting the efficacy of algorithm–hardware synergy. Similarly, multi-objective optimization techniques, such as desirability functions and evolutionary algorithms, enable balanced trade-offs between conflicting criteria like power efficiency and spectral accuracy. In sparse MIMO array design, desirability-based approaches optimized both resolution and energy consumption (Tanyer et al., 2024), while particle swarm optimization was shown to effectively tune IIR filter coefficients under competing frequency-domain constraints (Su et al., 2024). These advancements underscore the shift toward end-to-end, hardware-aware filter development, where machine learning integration and FPGA and ASIC implementation constraints are co-considered to maximize performance and deployability.

Neuromorphic and Bio-Inspired Computing Paradigms: Biologically inspired hardware promises highly efficient adaptive filtering by leveraging sparse, event-driven computation. Spiking neural networks (SNNs) implemented on neuromorphic chips can process sensor data in an ultra-low-power, event-driven manner. For example, Bartels et al. (2024) demonstrated a mixed-signal neuromorphic SNN for on-chip EEG seizure detection, noting that such systems are well-suited for always-on biomedical monitoring with minimal energy consumption. Similarly, Wang et al. (2025) surveyed optoelectronic neuromorphic devices and showed that retina-inspired preprocessing, filtering out 80% of visual data before digital encoding, significantly reduces energy demands. Recent experimental advances include organic artificial neurons closely mimicking biological ion-channel dynamics, paving the way for bio-interfaced adaptive filters and sensors on flexible substrates (Belleri et al., 2024). Together, these studies underscore the potential of neuromorphic and bio-inspired designs for adaptive noise cancellation and anomaly detection in wearables and IoT devices operating under stringent power constraints.

Quantum-Inspired and Advanced Optimization Algorithms: Cutting-edge optimization techniques are being explored to enhance complex filter design. Xu et al. (2024) introduced a quantum-inspired genetic algorithm (QGA) that integrates principles of quantum mechanics into classical genetic algorithms, achieving faster and more robust convergence on planar photonic filter inverse design benchmarks compared to traditional methods. On the quantum computing front, Ekström et al. (2025) developed a variational quantum-classical algorithm for multi-objective optimization, where a quantum circuit encodes objectives and a classical optimizer tunes parameters to maximize the Pareto hypervolume, yielding efficient multi-criteria solutions. Additionally, Kim et al. (2025) demonstrated that a 5,000-qubit analog quantum annealer outperformed classical solvers, achieving approximately 0.013% higher solution accuracy and solving problems over 6,000 times faster. Finally, Majumdar et al. (2025) proposed quantum-based methods to implement classical discrete-time signal processing, specifically FIR filters, by encoding signals into quantum states and designing unitary operators. They also explored techniques for cascading filters and suggested future directions for classical processing of quantum states. These advances highlight the growing potential of hybrid quantum-classical and quantum-inspired methods to accelerate filter configuration and parameter optimization in high-dimensional spaces.

Cybersecurity and Trustworthy Filtering Architectures: Security is rapidly emerging as a critical concern in filter design, particularly for reconfigurable platforms such as FPGAs. Contemporary FPGA devices increasingly integrate on-chip cryptographic engines, enabling encrypted and authenticated bitstreams. For example, Intel's CertusPro series employs AES encryption and ECDSA-based authentication to prevent unauthorized reconfiguration (Dofe et al., 2024). Complementing hardware-level protections, machine learning techniques have been proposed to detect malicious modifications in FPGA-based filters. Notably, Abi-Karam et al. (2024) introduced a contrastive learning framework that analyzes power side-channel traces to identify hardware Trojans embedded in filter architectures. Similarly, Dofe et al. (2024) employed deep recurrent neural networks to inspect FPGA configuration bitstreams directly; their LSTM-based model achieved a Trojan detection accuracy of 93.5%. These studies underscore the potential of data-driven security approaches for safeguarding adaptive filters. In parallel, conventional fault-tolerance mechanisms, such as triplicated pipelines and error correction codes (ECC), remain essential for ensuring reliable filtering in the presence of soft errors or bit-flips. Overall, recent advancements, from bitstream encryption and authentication to machine-learning-based anomaly detection, demonstrate a concerted effort to secure reconfigurable filtering systems against evolving cyber-physical threats.

Standardization and Toolchain Development: A key trend in FPGA and DSP design is the enhancement and standardization of design toolchains, especially in high-level synthesis (HLS). For example, Berrazueta-Mena and Navas (2025) introduced the AHA suite, standardized IP cores and an optimized HLS flow for AMD-Xilinx Zynq SoCs, that accelerates filter prototyping through automated, reproducible workflows. Complementing this, Abi-Karam et al. (2024) developed HLSFactory, an extensible open framework that automates the generation of large HLS design datasets for machine-learning-based optimization, aiming to overcome the fragmentation of HLS benchmarks and enable reproducible cross-vendor experiments. More broadly, the industry is moving towards open tool integration, with machine-learning compilers like Xilinx Vitis-AI and Intel OpenVINO merging with DSP and HLS tools. Community-driven efforts such as RISC-V and MLIR further aim to unify heterogeneous hardware targets. Collectively, these advancements reflect a strong focus on shared, automated toolchains spanning from high-level languages to verification, essential for fast prototyping of intelligent filters.

Conclusion

This in-depth review has critically examined the evolution and state-of-the-art design techniques in analog and digital filters, emphasizing the pivotal role of FPGA-based implementations in fulfilling the stringent performance requirements of modern signal processing applications. Classical analog filters, such as Butterworth, Chebyshev, and Elliptic, remain foundational, offering robust design paradigms characterized by linear phase response and hardware simplicity. Conversely, digital filters, particularly FIR and IIR architectures, provide superior adaptability and scalability, essential for addressing the complexities of dynamic signal environments.

The integration of advanced optimization methods, artificial intelligence, and neural network-inspired adaptive filtering marks a significant paradigm shift toward intelligent, context-aware signal processing. Leveraging the inherent parallelism and reconfigurability of modern FPGAs enables ultra-low latency and high-throughput processing, making these platforms exceptionally suited for real-time embedded systems and edge computing applications. Through detailed comparative analyses, this review elucidated the critical trade-offs among latency, computational complexity, power efficiency, and hardware and software integration challenges. Persistent issues in achieving real-time performance, incorporating adaptive algorithms, and managing hardware design complexity were explored alongside emerging trends such as reconfigurable architectures and AI-enhanced filtering.

Looking forward, future research should focus on integrated co-design frameworks that synergistically optimize filter topologies, machine learning algorithms, and hardware platforms to meet evolving application demands. Dedicated efforts in energy-aware adaptive filtering, secure and robust reconfigurable systems, and fostering interdisciplinary collaboration among signal processing, hardware engineering, and AI experts will be vital to address the growing complexity and heterogeneity of forthcoming signal processing challenges. This review aspires to serve as a comprehensive guide for researchers and practitioners in developing efficient, scalable, and intelligent filtering solutions, thereby advancing a broad spectrum of fields including telecommunications, biomedical engineering, industrial automation, and the Internet of Things.

References

Tan, L., & Jiang, J. (2025). Digital signal processing: Fundamentals, applications, and deep learning (4th ed.). Amsterdam, Netherlands: Elsevier.

Maclean, (2023). Digital Filters: Fundamentals, Analysis, and Design (Kindle ed.). Amazon Digital Services.

Marzalo, C. F. (2025). Analog Filters Handbook using APL-2 (1st ed.). Cham, Switzerland: Springer.

Sundararajan, D. (2024). Digital Signal Processing: An Introduction (2nd ed.). Cham, Switzerland: Springer.

Williamson, T. G., & Ghalichechian, N. (2025). New reconfigurable mmWave filters enabled by vanadium dioxide. *IEEE Transactions on Microwave Theory and Techniques, 73*(6), 3263–3274.

Gattami, A. (2024). Deep reinforcement learning: A convex optimization approach. arXiv.

Khan, A., Shafi, I., Khawaja, S. G., & De la Torre Díez, I. (2023). Adaptive filtering: Issues, challenges, and best-fit solutions using particle swarm optimization variants. Sensors, 23(18), 7710.

John, A., Cardiff, B., & John, D. (2024). A review on multisensor data fusion for wearable health monitoring. arXiv preprint.

Mehrish, A., Majumder, N., Bhardwaj, R., Mihalcea, R., & Poria, S. (2023). A review of deep learning techniques for speech processing. arXiv preprint.

Abdelbaset, S. E., Kasem, H. M., Khalaf, A. A., Hussein, A. H., & Kabeel, A. A. (2024). Deep learning-based spectrum sensing for cognitive radio applications. Sensors, 24(24), 7907.

Achtenberg, K., Szplet, R., & Bielecki, Z. (2024). Advanced, real-time programmable FPGA-based digital filtering unit for IR detection modules. Electronics, 13(22), 4449.

Jana, J., Chowdhury, R. S., Tripathi, S., & Bhaumik, J. (2024). FPGA implementation of compact and low power multiplierless architectures for DWT and IDWT. Journal of Real-Time Image Processing, 21(1), Article 19.

Spagnolo, F., Corsonello, P., Frustaci, F., et al. (2024). Approximate bilateral filters for real-time and low-energy imaging applications on FPGAs. Journal of Supercomputing, 80, 15894-15916.

Choi, Y., Jeong, D., Lee, M., Lee, W., & Jung, Y. (2021). FPGA implementation of the range-Doppler algorithm for real-time synthetic aperture radar imaging. Electronics, 10(17), 2133.

Heo, J., Jung, Y., Lee, S., & Jung, Y. (2021). FPGA implementation of an efficient FFT processor for FMCW radar signal processing. Sensors, 21(19), 6443.

Lee, J., Jeong, D., Lee, S., Lee, M., Lee, W., & Jung, Y. (2023). FPGA implementation of the chirp-scaling algorithm for real-time synthetic aperture radar imaging. Sensors, 23(2), 959.

Magyari, A., & Chen, Y. (2022). Review of state-of-the-art FPGA applications in IoT networks. Sensors, 22(19), 7496.

Puranik, S., Barve, M., Rodi, S., & Patrikar, R. (2023). FPGA-based high-throughput key-value store using hashing and B-tree for securities trading system. Electronics, 12(1), 183.

Kao, Y.-C., Chen, H.-A., & Ma, H.-P. (2022). An FPGA-based high-frequency trading system for 10 Gigabit Ethernet with a latency of 433 ns. In Proc. 2022 Int. Symp. VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 1-6.

Wan, Z., Lele, A., Yu, B., Liu, S., Wang, Y., Reddi, V. J., Hao, C., & Raychowdhury, A. (2022). Robotic computing on FPGAs: Current progress, research challenges, and opportunities. In Proc. IEEE Int. Conf. Artificial Intelligence Circuits and Systems (AICAS), Incheon, Korea.

Wan, Z., Yu, B., Li, T. Y., Tang, J., Zhu, Y., Wang, Y., Raychowdhury, A., & Liu, S. (2021). A survey of FPGA-based robotic computing. IEEE Circuits and Systems Magazine (CAS-M), to appear.

Böttcher, A., & Kumm, M. (2024). Multiplier design addressing area-delay trade-offs by using DSP and logic resources on FPGAs. arXiv preprint, arXiv:2407.03962.

Ranjbar, A., Esmaeili, E., Rafieisangari, R., & Shiri, N. (2025). FPGA-based multiplier with a new approximate full adder for error-resilient applications. arXiv preprint, arXiv:2506.09596.

Sajja, A., & Rooban, S. (2024). FPGA implementation of automatic seizure detection in EEG signals using machine learning algorithm. Discovery Applied Science, 6, 383.

Ezilarasan, M. R., & Leung, M. F. (2024). An efficient EEG signal analysis for emotion recognition using FPGA. Information, 15(6), 301.

Kalapothas, S., Flamis, G., & Kitsos, P. (2022). Efficient edge-AI application deployment for FPGAs. Information, 13(6), 279.

Ney, J., & Wehn, N. (2024). Achieving high throughput with a trainable neural-network-based equalizer for communications on FPGA. In Proc. 27th Euromicro Conference Series on Digital System Design (DSD).

Li, Z. (2024). Chebyshev filter and Butterworth filters: Comparison and applications in different cases. Applied and Computational Engineering, 38(1), 102–111.

Lorenz, A., Matthus, C. D., & Ellinger, F. (2025). Microstrip-based anti-aliasing filters for beyond-50 GHz-bandwidth mix-filter-sample analog-to-digital-converters. IEEE Access, early access.

Wang, B. (2024). Signal processing based on Butterworth filter: Properties, design, and applications. Highlights in Science, Engineering and Technology, 97, 72–77.

Huang, C.-S. (2023). An adaptive cutoff frequency design for Butterworth low-pass filter pursuing robust parameter identification for battery energy storage systems. Batteries, 9(4), 198.

Yi, H., Chang, F., Tan, P., & Huang, B. (2025). Design of infinite impulse response maximally flat stable digital filter with low group delay. Scientific Reports, 15(1).

Patil, S. A., & Khot, U. P. (2024). An optimal design of fractional order Butterworth filter with optimized magnitude and phase response. SSRG International Journal of Electronics and Communication Engineering, 11(10), 161–175.

Dolecek, G. J., & Fernandez-Vazquez, A. (2024). Improving passband characteristics in Chebyshev sharpened comb decimation filters. Applied Sciences, 14(23), 11421.

Mahata, S., Kubanek, D., & Herencsar, N. (2024). Design of fractional-order transitional filters of the Butterworth-Sync-Tuned, Butterworth-Chebyshev, and Chebyshev-Sync-Tuned types: Optimization, simulation, and experimental verification. Computers & Electrical Engineering, 116, 109200.

Muñoz-Ferreras, J.-M., Malki, M., Yang, L., & Gómez-García, R. (2023). Digital-filter-based flat-group-delay RF analog circuits: Application to Dolph-Chebyshev windows. IEEE Transactions on Circuits and Systems II: Express Briefs, early access.

Harpe, P., Hershberg, B., & Naraghi, S. (2024). Session 22 Overview: High-speed analog-to-digital converters. Proc. 2024 IEEE Int. Solid-State Circuits Conf. (ISSCC).

Hennessy, B., Yardley, H., Debnam, R., Camilleri, T. A., Spencer, N. K., Holdsworth, D. A., Warne, G., Cheung, B., & Kharabash, S. (2025). Space surveillance with high-frequency radar. Sensors, 25(7), 2302.

Zhang, K., Liu, Y.-F., Wang, Z., Yuan, W., Keskin, M. F., Wymeersch, H., & Xia, S. (2025). Joint design of radar receive filter and unimodular ISAC waveform with sidelobe level control. arXiv preprint, arXiv:2503.15054.

Lorenz, A., Matthus, C. D., & Ellinger, F. (2025). Microstrip-based anti-aliasing filters for beyond-50 GHz-bandwidth mix-filter-sample analog-to-digital-converters. IEEE Access, 13, 1-11.

Yang, Y., Su, J., Lai, X., Chen, D., Li, D., & Yang, Y. (2025). Recent developments and perspectives on optimization design methods for analog integrated circuits. Symmetry, 17(4), 529.

Srinivasagan, R. (2025). Advanced low-power filter architecture for biomedical signals with adaptive tuning. *PLoS ONE, 20*(1), e0311768.

Elwakil, A. S., Elamien, M. B., Allagui, A., Maundy, B. J., & Psychalinos, C. (2024). On the behavior of a non-linear bandpass filter with self-voltage-controlled resistors. Electronics, 13(17), 3434.

Mugisha, S., Mohammed, W., Makiika, B. G., & Bashabe, J. (2024). Review and analysis on digital filter design in digital signal processing. KIU Journal of Science, Engineering, and Technology, 3(2).

Mix, D. (2024). Filter Design Techniques: Analog and Digital Filters, Frequency Selective and Matched Filters, Adaptive Matched Filter and Template Design (1st ed.). 251 pages.

Gao, P., Yu, T., Wang, F., & Yuan, R.-Y. (2024). Automated design and optimization of distributed filter circuits using reinforcement learning. Journal of Computational Design and Engineering, 11(5), 60–76.

Daylak, F., & Ozoguz, S. (2025). Automated neural network-based optimization for enhancing dynamic range in active filter design. Electronics, 14(4), 786.

Omar, A., Shpak, D., Agathoklis, P., & Moa, B. (2023). A nonlinear optimization design algorithm for nearly linear-phase 2D IIR digital filters. Signals, 4(3), 575–590.

Srivatsan, K., & Venkatesan, N. (2024). Improved meta-heuristic technique to FIR filter design and application. *IEEE Access*, *PP* (99), 1–1.

Samiappan, R. A., Prabukumar, M., & Fine-tuning, S. (2024). Fine-tuning digital FIR filters with gray wolf optimization for peak performance. Scientific Reports, 14, Article 12675.

Su, T.-J., Zhuang, Q.-Y., Lin, W.-H., Hung, Y.-C., Yang, W.-R., & Wang, S.-M. (2024). Design of infinite impulse response filters based on multi-objective particle swarm optimization. Signals, 5(3), 526–541.

Kadam, S. A., & Chavan, M. (2019). Design of digital filter by genetic algorithm. International Journal of Engineering and Advanced Technology, 8(6), 397–400, Aug.

Gokcen, A., & Boyaci, A. (2025). *Robust federated learning with confidence-weighted filtering and GAN-based completion under noisy and incomplete data* (arXiv:2505.09733). arXiv.

Abdelbaset, S. E., Kasem, H. M., Khalaf, A. A., Hussein, A. H., & Kabeel, A. A. (2024). Deep learning-based spectrum sensing for cognitive radio applications. Sensors, 24(24), Article 7907, Dec.

Shkil, A. S., Filippenko, O. I., Rakhlis, D., & Filippenko, I. (2024). Adaptive filtering and machine learning methods in noise suppression systems, implemented on the SoC. Radio Electronics Computer Science Control, 4, Dec.

Wang, Z., & Ou, Y. (2022). Learning human strategies for tuning cavity filters with continuous reinforcement learning. Applied Sciences, 12(5), Article 2409, Feb.

Javadi, S., Rezaee, B., Nabavi, S. S., Gadringer, M. E., & Bösch, W. (2025). Machine learning-driven approaches for advanced microwave filter design. Electronics, 14(2), Article 367, Jan.

Yadav, N. K., Dhawan, A., Tiwari, M., & Jha, S. K. (2025). Multistage cascaded LMS adaptive FIR filter and its application to multiple artifacts removal from ECG. Journal of the Institution of Engineers (India): Series B, published online Apr. 3.

Kim, J., & Hao, C. (2025). RealProbe: An automated and lightweight performance profiler for in-FPGA execution of high-level synthesis designs. arXiv preprint arXiv:2504.03879.

Ibrahim, S. H., Hameed, I. S., & Ali, H. H. 2025. High performance IIR filter design based on fast multiplier. *Diyala Journal of Engineering Sciences*, 18(1), 192–202.

El Bouazzaoui, A., Hadjoudja, A., & Mouhib, O. (2023). Real-time adaptive neural network on FPGA: Enhancing adaptability through dynamic classifier selection. arXiv preprint, arXiv:2311.09516, Nov.

Mugisha, S., Mohammed, W., Makiika, B. G., & Buhari, M. D. (2024). Review and analysis on digital filter design in digital signal processing. KIU Journal of Science Engineering and Technology, 3(2), Dec.

Sunny, M. N. M., Amin, M. M., & Hossain, K. M. S. (2024). Design of digital filters for noise reduction in RF communication systems. Nov.

Kamal, S., Khan, I. A., Ali, A., et al. (2025). Novel Nyquist-I filter to reduce PAPR for SC-FDMA scheme. Telecommunication Systems, 88, 34.

Somefun, C. T., Daramola, S. A., & Somefun, T. E. (2024). Advancements and applications of adaptive filters in signal processing. Journal Européen des Systèmes Automatisés, 57(5), 1259–1272, Oct.

Uko, M., Ekpo, S., Enahoro, S., Elias, F., Unnikrishnan, R., & Al-Yasir, Y. (2025). Highly adaptive reconfigurable receiver front-end for 5G and satellite applications. Technologies, 13(4), Article 124, Mar.

Xin, Y., Bao, T., Hua, J., & Zhang, H. (2024). Generalized filter bank orthogonal frequency division multiplexing systems for 6G. Electronics, 13(15), 3006, Jul.

Bartzoudis, N., Rubio Fernández, J., López-Bueno, D., Villarroel, A. R., & Antonopoulos, A. (2024). Agile FPGA computing at the 5G edge: Joint management of accelerated and software functions for open radio access technologies. Electronics, 13(4), Article 701, Feb.

Uko, M., Ekpo, S., Enahoro, S., Elias, F., Unnikrishnan, R., & Al-Yasir, Y. (2025). Highly adaptive reconfigurable receiver front-end for 5G and satellite applications. Technologies, 13(4), Article 124, Mar.

Tuninato, R., & Garello, R. (2024). 5G NTN primary synchronization signal: An improved detector for handheld devices. IEEE Open Journal of the Communications Society, 5, 3792–3803.

Jia, Y., Pei, H., Liang, J., Zhou, Y., Yang, Y., Cui, Y., & Xiang, M. (2024). Preprocessing and denoising techniques for electrocardiography and magnetocardiography: A review. Bioengineering, 11(11), Article 1109.

Khalili, M., GholamHosseini, H., Lowe, A., & Kuo, M. M. Y. (2024). Motion artifacts in capacitive ECG monitoring systems: A review of existing models and reduction techniques. Medical & Biological Engineering & Computing, 62(12), 3599–3622, Dec.

Jayaraman, J. R. D. K., Ganesh Babu, C., Priyadharsini, K., et al. (2024). Certain investigation on hybrid neural network method for classification of ECG signal with the suitable FIR filter. Scientific Reports, 14, Article 15087.

Dobrev, D., Neycheva, T., Krasteva, V., & Jekova, I. (2025). Design of high-pass and low-pass active inverse filters to compensate for distortions in RC-filtered electrocardiograms. Technologies, 13(4), Article 159, Apr.

Song, B.-G., & Kang, N. (2024). Removal of movement artifacts and assessment of mental stress analyzing electroencephalogram of non-driving passengers under whole-body vibration. Frontiers in Neuroscience, 18, Apr.

Haripriya, R., Khare, N., & Pandey, M. (2025). Privacy-preserving federated learning for collaborative medical data mining in multi-institutional settings. *Scientific Reports, 15*, 12482.

Nair, S., James, B. P., & Leung, M.-F. (2025). An optimized hybrid approach to denoising of EEG signals using CNN and LMS filtering. Electronics, 14(6), Article 1193, Mar.

Alqudah, A. M., & Moussavi, Z. (2025). A review of deep learning for biomedical signals: Current applications, advancements, future prospects, interpretation, and challenges. Computers, Materials & Continua, 83(3), 3753–3841, May.

Pedroso, A. F., & Khera, R. (2025). Leveraging AI-enhanced digital health with consumer devices for scalable cardiovascular screening, prediction, and monitoring. npj Cardiovascular Health, 2, 34.

Khan, M. I., & Silva, B. da. (2024). Harnessing FPGA technology for energy-efficient wearable medical devices. Electronics, 13(20), Article 4094, Oct.

Hudda, S., & Haribabu, K. (2025). A review on WSN-based resource-constrained smart IoT systems. Discover Internet of Things, 5, 56.

Lazea, D., Cioara, T., Hangan, A., & István, Z. (2024). Customizing pre-processing algorithms for streaming sensor data on embedded networked devices. In Proc. 2024 Int. Conf. Embedded Wireless Systems and Networks (EWSN), Feb.

Costa, H., Silva, M., Sánchez-Gendriz, I., Viegas, C. M. D., & Silva, I. (2024). An evolving multivariate time series compression algorithm for IoT applications. Sensors, 24(22), Article 7273, Nov.

Bobulski, J., Szymoniak, S., & Pasternak, K. (2024). An IoT system for air pollution monitoring with safe data transmission. Sensors, 24(2), Article 445, Jan.

Thakur, N., Mishra, V. P., Islam, S. M. N., Neyaz, Z., Jain, R., & Roy, S. (2023). IoT-based air pollution monitoring system. In Proc. 2023 Int. Conf. Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, UAE, Mar.

Lee, J., Lee, A. H., Leung, V., et al. (2024). An asynchronous wireless network for capturing event-driven data from large populations of autonomous sensors. Nature Electronics, 7, 313–324.

Misbahuddin, M., Cokrowati, N., Iqbal, M. S., Farobie, O., Amrullah, A., & Ernawati, L. (2025). Kalman filter-enhanced data aggregation in LoRaWAN-based IoT framework for aquaculture monitoring in Sargassum sp. cultivation. Computers, 14(4), Article 151, Apr.

Zhan, S., Huang, L., Luo, G., Zheng, S., Gao, Z., & Chao, H.-C. (2025). A review on federated learning architectures for privacy-preserving AI: Lightweight and secure cloud–edge–end collaboration. *Electronics, 14*(13), 2512

Chandran, L. R., Karuppasamy, I., Nair, M. G., Sun, H., & Krishnakumari, P. K. (2025). Compressive sensing in power engineering: A comprehensive survey of theory and applications, and a case study. Journal of Sensor and Actuator Networks, 14(2), Article 28, Mar.

Pazmiño Ortiz, L. A., Maldonado Soliz, I. F., & Guevara Balarezo, V. K. (2025). Advancing TinyML in IoT: A holistic system-level perspective for resource-constrained AI. Future Internet, 17(6), Article 257, Jun.

Saulquin, A., Fatahi, M., Boulet, P., & Meftali, S. (2025). ModNEF: An open-source modular neuromorphic emulator for FPGA for low-power in-edge artificial intelligence. *ACM Transactions on Architecture and Code Optimization, 22*(2), Article 73, 1–24.

Aljumah, A. (2025). Blockchain-inspired distributed security framework for Internet of Things. Scientific Reports, 15, Article 10066.

Mughal, F. R., He, J., Das, B., et al. (2024). Adaptive federated learning for resource-constrained IoT devices through edge intelligence and multi-edge clustering. Scientific Reports, 14, Article 28746.

Wang, S., Zhao, W., Zong, X., & Zhang, W. (2025). Optimal control model of electromagnetic interference and filter design in motor drive system. Electronics, 14(5), Article 980, Feb.

Kim, J., & Kim, H. (2024). Design of electronic filter for noise and vibration reduction in brushed DC motor. Machines, 12(3), Article 148, Feb.

Dursun, C., & Ozkilic, S. O. (2025). A novel filtering observer: A cost-effective estimation solution for industrial PMSM drives using in-motion control systems. Energies, 18(4), Article 883, Feb.

Samiappan, S. R., & Prabukumar, M. (2024). Fine-tuning digital FIR filters with gray wolf optimization for peak performance. Scientific Reports, 14, Article 12675.

Sanchis, R., & Peñarrocha-Alós, I. (2023). Optimal tuning of PID controllers with derivative filter for stable processes using three points from the step response. ISA Transactions, 143, 596–610, Dec.

Pacheco, J., Benitez, V. H., Pérez, G., & Brau, A. (2024). Wavelet-based computational intelligence for real-time anomaly detection and fault isolation in embedded systems. Machines, 12(9), Article 664, Sep.

Ye, T., Peng, T., & Yang, L. (2025). Review on sound-based industrial predictive maintenance: From feature engineering to deep learning. Mathematics, 13(11), Article 1724, May.

Ding, D., Han, Q.-L., Wang, Z., & Ge, X. (2019). A survey on model-based distributed control and filtering for industrial cyber-physical systems. IEEE Trans. Ind. Informat., 15(5), 2483–2499, May.

Altaf, S., Al-Anbuky, A., & Gheitasi, A. (2024). Enhancing fault detection in distributed motor systems using AI-driven cyber-physical sensor networks. In Proceedings of the 11th International Electronic Conference on Sensors and Applications (ECSA-11) (Vol. 82, No. 1, p. 78). Online.

El-Mahdy, M. H., Ali, A. O., Hassan, O. H., et al. (2025). Intelligent fault tolerance control using long short-term memory for efficient system performance under fault conditions. Scientific Reports, 15, Article 16259.

Su, C., Tang, X., Jiang, Q., et al. (2025). Digital twin system for manufacturing processes based on a multi-layer knowledge graph model. Scientific Reports, 15, Article 12835.

Manduchi, G. (2024). Overlaps between industrial informatics and control, data acquisition, and management in Big Science. Frontiers in Industrial Engineering, 2, Article 1342734.

Venkatesh, A., Naveen Kumar, V., & Sathisha, S. B. (2024). Collaborative edge computing in IoT: Audio signal processing with low-power VLSI implementation. Library Progress International, 44(3), 19270–19282.

Rancea, A., Anghel, I., & Cioara, T. (2024). Edge computing in healthcare: Innovations, opportunities, and challenges. Future Internet, 16(9), 329.

Cruz Castañeda, W. A., & Bertemes Filho, P. (2024). Improvement of an edge-IoT architecture driven by artificial intelligence for smart-health chronic disease management. Sensors, 24(24), 7965.

Ergen, M., Saoud, B., Shayea, I., El-Saleh, A. A., Ergen, O., Inan, F., & Tuysuz, M. F. (2024). Edge computing in future wireless networks: A comprehensive evaluation and vision for 6G and beyond. ICT Express, 10(5), 1151–1173.

Hu, M., Luo, Z., Pasdar, A., Lee, Y. C., Zhou, Y., & Wu, D. (2023). Edge-based video analytics: A survey. arXiv preprint arXiv:2303.14329.

Kim, B., Kim, S., Park, S., & Jeong, J. (2025). Person re-identification with attribute-guided, robust-to-low-resolution drone footage considering fog/edge computing. Sensors (Basel), 25(6), 1819.

Pandey, S., Jain, P. K., & Patel, P. (2025). Video stabilization using modified blurriness index employing de-blurring with temporal derivatives using motion smoothing. Traitement du Signal, 42(3), 1457–1479.

Romero-González, J.-A., Córdova-Esparza, D.-M., Terven, J., Herrera-Navarro, A.-M., & Jiménez-Hernández, H. (2024). Background subtraction for dynamic scenes using Gabor filter bank and statistical moments. Algorithms, 17(4), 133.

Oshima, Y., Yamaguchi, Y., Tsugami, R., Narikawa, S., et al. (2024). FPGA-based improved background subtraction for ultra-low latency. IEEE Access, Article 1.

Aljumaili, A. K., Hassan, R. F., Hamza, E. K., & Humaidi, A. J. (2025). FPGA implementation of lowpass FIR filter using single multiply-accumulate unit with dual-port RAM. Journal of Engineering Science and Technology, 20(1), 345–361.

Chen, Q., Chang, Y., Kim, K., Gao, C., & Liu, S. C. (2023). An area-efficient ultra-low-power time-domain feature extractor for edge keyword spotting. In Proceedings of the 56th IEEE International Symposium on Circuits and Systems (ISCAS).

Choi, J., Park, J., Lee, W., Lee, J.-J., & Pedram, M. (2025). ASAP-FE: Energy-efficient feature extraction enabling multi-channel keyword spotting on edge processors. arXiv preprint arXiv:2506.14657.

Nigam, H. K., & Srivastava, H. M. (2023). Filtering of audio signals using discrete wavelet transforms. Mathematics, 11(19), 4117.

Shen, Y., Wu, B., Straeussnigg, D., & Gutierrez, E. (2025). A KWS system for edge-computing applications with analog-based feature extraction and learned step size quantized classifier. Sensors, 25(8), 2550.

Abdou, A., & Krishnan, S. (2024). Enhancement of single-lead dry-electrode ECG through wavelet denoising. Frontiers in Signal Processing, 4.

Hizem, M., Bousbia, L., Ben Dhiab, Y., Aoueileyine, M. O., & Bouallegue, R. (2025). Reliable ECG anomaly detection on edge devices for Internet of Medical Things applications. Sensors, 25(8), 2496.

Gawande, G. S., Dhotre, D. R., Choubey, N., & Mate, D. S. (2023). A MILP based optimization and FPGA implementation of efficient polyphase multirate filters. International Journal of Intelligent Systems Applications in Engineering (IJISAE), 11(5s), 164–172.

Sadeghi, S. (2024). Classifying FPGA technology in digital signal processing: A review. International Journal of Engineering Technology and Sciences, 10.

Khwa, W. S., Wen, T. H., Hsu, H. H., Lin, T. Y., Yeh, C. H., & Chiu, Y. C. (2025). A mixed-precision memristor and SRAM compute-in-memory AI processor. Nature, 639(8003), 617–623.

Krishnan, S. S. H., & Vidhya, K. (2024). Distributed arithmetic-FIR filter design using approximate Karatsuba multiplier and VLCSA. Expert Systems with Applications, 249, 123488.

Samakovlis, D., Albini, S., Álvarez, R. R., Constantinescu, D.-A., Schiavone, P. D., Quirós, M. P., & Atienza, D. (2024). BiomedBench: A benchmark suite of TinyML biomedical applications for low-power wearables. arXiv preprint arXiv:2406.03886.

Macha, S., Oza, O., Escott, A., Caliva, F., Armitano, R., Cheekatmalla, S. K., Parthasarathi, S. H. K., & Liu, Y. (2023). Fixed-point quantization aware training for on-device keyword-spotting. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

Per Vices Corp. (2023). How software defined radio is advancing radar systems. Microwaves & RF, April 20.

Hozhabr, S. H., & Giorgi, R. (2025). A survey on real-time object detection on FPGAs. IEEE Access, Early Access.

Ricci, S., Caputo, S., & Mucchi, L. (2023). FPGA-based pulse compressor for ultra low latency visible light communications. Electronics, 12(2), 364.

Oncu, S., Karakaya, M., Dalveren, Y., Kara, A., & Derawi, M. (2024). Real-time radar classification based on software-defined radio platforms: Enhancing processing speed and accuracy with graphics processing unit acceleration. Sensors, 24(23), 7776.

S., et al. (2024). Low power and high throughput reconfigurable FIR filter architecture using RRNS for SDR applications. International Journal of Intelligent Systems Applications in Engineering, 12(18s), 99–110.

P. S., M., Kumari, C., S., R. U., M. T., T. G., & S. S. (2024). *Low power and high throughput reconfigurable FIR filter architecture using RRNS for SDR applications*. *International Journal of Intelligent Systems and Applications in Engineering, 12*(18s), 99–110.

Mandapati, S., Balss, U., & Breit, H. (2024). Real-time floating-point SAR focusing on FPGA. In Proceedings of the European Conference on Synthetic Aperture Radar (EUSAR). Garching, Germany.

Kou, Z., You, Q., Kim, J., Dong, Z., Lowerison, M. R., Sekaran, N. V. C., Llano, D. A., Song, P., & Oelze, M. L. (2023). High-level synthesis design of scalable ultrafast ultrasound beamformer with single FPGA. IEEE Transactions on Biomedical Circuits and Systems, 17(3), 446–457.

Antunes, P., & Podobas, A. (2025). FPGA-based neural network accelerators for space applications: A survey. arXiv preprint arXiv:2504.16173.

AMD. (2025). Versal™ adaptive SoCs: Heterogeneous acceleration from cloud to edge. AMD Official Website.

Li, R. (2025). Dataflow & tiling strategies in edge-AI FPGA accelerators: A comprehensive literature review. arXiv preprint arXiv:2505.08992.

Wang, W., Li, X., Chen, L., Sun, H., & Zhang, F. (2024). A review on soft error correcting techniques of aerospace-grade static RAM-based field-programmable gate arrays. Sensors, 24(16), 5356.

Giannakopoulos, G., Adegbenro, P., & Perez, M. A. (2025). Designing FIR and IIR digital filters for low-power modems using SIMULINK and VHDL. Preprints.

Sarpal, S. (2024). BLW filtering for ECG systems: Getting IEC/FDA compliance right. Advanced Solutions Nederland.

Gregg, R. E., An, J., Bailey, B., & Al-Zaiti, S. S. (2023). An efficient linear-phase high-pass filter for ECG. In Proceedings of the 2023 Computing in Cardiology Conference. November.

Sarpal, S. (2025). Difference between IIR and FIR filters: A practical design guide. ASN Filter Designer.

Pondreti, P. (2023). VLSI implementation of 3-parallel FIR filters through coefficient symmetry property and fast FIR algorithm. Tuijin Jishu/Journal of Propulsion Technology, 44(4), 966–977.

Zhuang, Y., & Liu, Y. (2023). A stable IIR filter design approach for high-order active noise control applications. Acoustics, 5(3), 746–758.

Kaur, M., Kaur, R., & Singh, N. (2022). Enhanced chimp optimization algorithm for high level synthesis of digital filters. Scientific Reports, 12(1), 21389.

Fernández de Bulnes, D. R., Maldonado, Y., & Trujillo, L. (2020). Development of multiobjective high-level synthesis for FPGAs. Scientific Programming.

Carnevale, J. (2024). Real-time digital signal processing: Challenges and solutions. Opinion Journal of Telecommunication System and Management, 13(2).

Sen, A., Chakraborty, S., Tripathy, S., & Chakraborty, S. (2025). RadarTrack: Enhancing ego-vehicle speed estimation with single-chip mmWave radar. arXiv preprint arXiv:2504.14495.

Rouili, M., & Boutaba, R. (2025). Blink: A P4-based 5G centralized unit. Presented at IEEE/IFIP Network Operations and Management Symposium (NOMS).

Benois, P. R., Nowak, P., Zölzer, U., Eckert, M., & Klauer, B. (2018). Low-latency FIR filter structures targeting FPGA platforms. ACM Transactions on Embedded Computing Systems, 1(1), 1–7.

Hu, M., Jiang, X., & Li, Z. (2023). A resource-efficient FIR filter design based on an RAG improved algorithm. In Proceedings of the 5th International Conference on Circuits and Systems (ICCS) (October, Huzhou, China).

Kishor, B., & R. S, K. (2024). Design and implementation of high throughput and efficient FIR filter architectures using pipelining. In Proceedings of the IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS). February 2024.

Maclair, G., Ries, M., & Moonen, C. (2006). Low latency vs. nonlinear phase: A balanced approach for UNFOLD filter design in quantitative real-time MRI. In Proceedings of the International Society for Magnetic Resonance in Medicine (ISMRM) 14th Scientific Meeting and Exhibition (p. 3186). Seattle, WA, USA, May.

MathWorks. (n.d.). Compensate for the delay introduced by an IIR filter. MathWorks Documentation.

Rahman, A. A.-H. A., Thavot, R., Brunet, S. C., Bezati, E., & Mattavelli, M. (2012). Design space exploration strategies for FPGA implementation of signal processing systems using CAL dataflow program. In Proceedings of the 2012 Conference on Design and Architectures for Signal and Image Processing (DASIP) (Karlsruhe, Germany, October).

Yousif, S. M., & Abdulnabi, S. H. (2023, June). *High performance FIR and IIR filters based on FPGA for 16 Hz signal processing*. Paper presented at the 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Istanbul, Turkey.

Gadawe, N. T., Hamad, R. W., & Qaddoori, S. L. (2024). Realization of IIR digital filter structures for ECG denoising. *Journal Européen des Systèmes Automatisés, 57*(2), 599–608.

Arivalagan, S., James, B. P., & Leung, M.-F. (2024). Reconfigurable frequency response masking multi-MAC filters for software defined radio channelization. Electronics, 13(21), 4211.

Rao, A., Chandra, S., & Purohit, N. (2023). Design of efficient polyphase interpolated-FIR filter using coefficient symmetry. In Proceedings of IEEE CICT.

Matei, R. C., & Chiper, D. F. (2024). Design and polyphase implementation of rotationally invariant 2D FIR filter banks based on maximally flat prototype. *Electronics, 13*(14), 2829.

Reddy, M. S., & Singh, M. K. (2024). FIR filter complexity analysis using multi-rate signal processing approach. EVERGREEN Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy, 11(2), 1263–1267.

Giannakopoulos, G., & Perez, M. A. (2025). Designing FIR and IIR digital filters for low-power modems using SIMULINK and VHDL. Preprints, 15 pages.

Song, J., Chen, L., Lu, Z., Li, B., Liu, Z., Xue, Z., Sun, G., & Liu, W. (2024). Cascaded multiplier-free implementation of adaptive anti-jamming filter based on GNSS receiver. Frontiers in Physics, 12, Space Physics.

Liao, Y., Adegbija, T., & Lysecky, R. (2025). System-level design space exploration for high-level synthesis under end-to-end latency constraints. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 44(4), 1354–1365.

Michon, R., Ducceschi, M., Cochard, P., Skare, T., Webb, C. J., & Russo, R. (2025). Evaluating CPU, GPU, and FPGA performance in the context of modal reverberation: A comparative analysis. Frontiers in Signal Processing, 5, Audio and Acoustic Signal Processing.

Samakovlis, D., Albini, S., Álvarez, R. R., Constantinescu, D.-A., Schiavone, P. D., Peón-Quirós, M., & Atienza, D. (2024). Enabling efficient wearables: An analysis of low-power microcontrollers for biomedical applications. arXiv preprint arXiv:2411.09534.

Khan, L., Isanaka, S. P., & Liou, F. (2024). FPGA-based sensors for distributed digital manufacturing systems: A state-of-the-art review. Sensors, 24(23), 7709.

Jaiswal, P., & Mitra, S. (2024). A holistic approach to designing nearly linear-phase IIR filters. Educational Administration: Theory and Practice, 30(1), 7148–7162.

Lai, C., & Zhao, R. (2024). An autocorrelation-based r-stability condition with application in the design of IIR filters. *IEEE Signal Processing Letters, 31*, 1389–1393.

Avital, N., Shulkin, N., & Malka, D. (2025). Automatic calculation of average power in electroencephalography signals for enhanced detection of brain activity and behavioral patterns. Biosensors, 15(5), 314.

Zheng, J., Lv, Z., Li, D., Lu, C., Zhang, Y., Fu, L., Huang, X., Huang, J., Chen, D., & Zhang, J. (2025). FPGA-based low-power high-performance CNN accelerator integrating DIST for rice leaf disease classification. Electronics, 14(9), 1704.

Maghsoudnia, A., Vlad, E., Gong, A., Dumitriu, D. M., & Hassanieh, H. (2024). Ultra-reliable low-latency in 5G: A close reality or a distant goal? In Proceedings of the 23rd ACM Workshop on Hot Topics in Networks (HotNets '24) (pp. 111–120).

Yan, B., & Roberts, I. P. (2025). Advancements in millimeter-wave radar technologies for automotive systems: A signal processing perspective. Electronics, 14(7), 1436.

Kim, J., & Hao, C. (2025). RealProbe: An automated and lightweight performance profiler for in-FPGA execution of high-level synthesis designs. arXiv preprint arXiv:2504.03879.

Dai, Y. (2024). Integrated signal processing for massive MIMO systems (Tech. Rep. UCB/EECS-2024-30). University of California, Berkeley.

Choi, S., Yang, H., Noh, Y., Kim, G., Kwon, E., & Yoo, H. (2024). FPGA-based multi-channel real-time data acquisition system. Electronics, 13(15), 2950.

Herrou, A., de Dinechin, F., Letz, S., Orlarey, Y., & Volkova, A. (2024). Towards fixed-point formats determination for Faust programs. arXiv preprint arXiv:2403.06527.

Tang, G. (2025). A digital signal processing-based multi-channel acoustic emission acquisition system with a simplified analog front-end. Sensors, 25(10), 3206.

Lahti, S., & Hämäläinen, T. D. (2025). High-level synthesis for FPGAs – A hardware engineer's perspective. IEEE Access.

Alachiotis, N., et al. (2025). FPGA innovation research in the Netherlands: Present landscape and future outlook. Frontiers in High Performance Computing, 3.

Igual, J., Gomes, H. M., Pfahringer, B., et al. (2025). Linear adaptive filtering for regression in data streams. International Journal of Data Science and Analytics.

Mahmood, A. F., Awny, S. N., & Alameer, A. (2024). RLS adaptive filter co-design for de-noising ECG signal. Results in Engineering, 24, 103563.

Zhang, C., Chen, W., & Chen, H. (2025). Denoising for ECG signals based on VMD and RLS. Journal of Measurements in Engineering, 13(1), 185–204.

Zhang, L. (2025). Channel equalization in ultraviolet communication based on LSTM-DNN hybrid model. Scientific Reports, 15(1), 17226.

Chen, J., Yu, K., Wang, F., Zhou, Z., Bi, Y., Zhuang, S., & Zhang, D. (2024). Temporal convolutional network-enhanced real-time implicit emotion recognition with an innovative wearable fNIRS-EEG dual-modal system. Electronics, 13(7), 1310.

Thornton, C. E., Sloop, J., Brown, S., Orndorff, A., Headley, W. C., & Young, S. (2025). Temporal convolutional autoencoder for interference mitigation in FMCW radar altimeters. arXiv preprint arXiv:2505.22783.

Xiong, W., Ma, L., & Li, H. (2024). A general dual-pathway network for EEG denoising. Frontiers in Neuroscience, 17.

Xu, Z., Aroudi, A., Tan, K., Pandey, A., Lee, J.-S., Xu, B., & Nesta, F. (2024). FoVNet: Configurable field-of-view speech enhancement with low computation and distortion for smart glasses. arXiv preprint arXiv:2408.06468.

Terada, T., & Toyoura, M. (2025). Wavelet integrated convolutional neural network for ECG signal denoising. In *Proceedings of MultiMedia Modeling (MMM, LNCS 15523)* (pp. 311–324). Springer.

Hizem, M., Bousbia, L., Ben Dhiab, Y., Aoueileyine, M. O.-E., & Bouallegue, R. (2025). Reliable ECG anomaly detection on edge devices for Internet of Medical Things applications. Sensors, 25(8), 2496.

Khan, Z., Yoon, S.-C., & Bhandarkar, S. M. (2025). Deep learning model compression and hardware acceleration for high-performance foreign material detection on poultry meat using NIR hyperspectral imaging. Sensors, 25(3), 970.

Bereketoğlu, A. B. (2025). *Composite reward design in PPO-driven adaptive filtering* (Version 1) [Preprint]. arXiv. arXiv:2506.06323

Chen, X., Yu, X., Chang, L., Huang, Y., He, J., Zhang, S., Li, J., Lin, L., Zeng, Z., Tu, X., & Zhang, S. (2025). A synergistic framework of nonlinear acoustic computing and reinforcement learning for real-world human-robot interaction. arXiv preprint arXiv:2505.01998.

Li, C., Denison, T., & Zhu, T. (2024). A survey of few-shot learning for biomedical time series. arXiv preprint arXiv:2405.02485.

Rafiq, R. B., Shi, W., & Albert, M. V. (2024). Wearable sensor-based few-shot continual learning on hand gestures for motor-impaired individuals via latent embedding exploitation. arXiv preprint arXiv:2405.08969.

Oliveira, M. S., Carvalho, M. B. S., Müller, C., Girardi, A., et al. (2024). Design and characterization of a digitally tunable Gm-C filter for multi-standard receivers. Electronics, 13(14), 2866.

Arivalagan, S., James, B. P., & Leung, M.-F. (2024). Reconfigurable frequency response masking multi-MAC filters for software defined radio channelization. Electronics, 13(21), 4211.

Fahmy, S. A., & Iyer, K. B. (2025). Dynamic and partial reconfiguration of FPGAs. In A. Chattopadhyay (Ed.), Handbook of computer architecture. Springer.

Uko, M., Ekpo, S., Enahoro, S., Elias, F., Unnikrishnan, R., & Al-Yasir, Y. (2025). Highly adaptive reconfigurable receiver front-end for 5G and satellite applications. Technologies, 13(4), 124.

Lee, C. Y., Khatri, S. P., & Ghrayeb, A. (2025). A novel mixed-signal flash-based finite impulse response (FFIR) filter for IoT applications. In Proceedings of the 30th Asia and South Pacific Design Automation Conference (ASP-DAC), Tokyo, Japan, January 2025 (pp. 1216–1222).

Jendernalik, W., & Jakusz, J. (2024). Fully tunable analog biquadratic filter for low-power auditory signal processing in CMOS technologies. Electronics, 13(16), 3132.

Shokri, R., Koolivand, Y., Shoaei, O., Caviglia, D. D., & Aiello, O. (2024). A reconfigurable, nonlinear, low-power, VCO-based ADC for neural recording applications. Sensors, 24(19), 6161.

Manokhin, M., Chollet, P., & Desgreys, P. (2024). Towards flexible and low-power wireless smart sensors: Reconfigurable analog-to-feature conversion for healthcare applications. Sensors, 24(3), 999.

Shi, Y. (2024). Unveiling the powerhouses of AI: A comprehensive study of GPU, FPGA, and ASIC accelerators (PhD thesis). University of Rochester.

Sravanthi, M., Gunturi, S. K., Chinnaiah, M. C., Lam, S.-K., Vani, G. D., Basha, M., Janardhan, N., Krishna, D. H., & Dubey, S. (2024). Adaptive FPGA-based accelerators for human–robot interaction in indoor environments. Sensors, 24, 6986.

Curzel, S. (2024). Modern high-level synthesis: Improving productivity with a multi-level approach. In F. Amigoni (Ed.), Special topics in information technology. Springer.

Berrazueta-Mena, D., & Navas, B. (2025). AHA: Design and evaluation of compute-intensive hardware accelerators for AMD-Xilinx Zynq SoCs using HLS IP flow. Computers, 14(5), 189.

El Bouazzaoui, A., Hadjoudja, A., Mouhib, O., & Cherkaoui, N. (2024). FPGA-based ML adaptive accelerator: A partial reconfiguration approach for optimized ML accelerator utilization. Array, 27, 100337.

Taghavirashidizadeh, A., Nouraei Chaleshtori, H., Rafat, A., & Bahramizadeh, A. (2024). Adaptive and reconfigurable FPGA-based systems architecture with approach machine learning model. Proceedings of Special Topics, 48(2).

Tanyer, S. G., Dent, P., Ali, M., Davis, C., Rajagopal, S., & Driessen, P. F. (2024). Multi-objective design and optimization of hardware-friendly grid-based sparse MIMO arrays. Sensors, 24(21), 6810.

Bartels, J., Gallou, O., Ito, H., et al. (2025). Event driven neural network on a mixed signal neuromorphic processor for EEG based epileptic seizure detection. Scientific Reports, 15, 15965.

Wang, X., Huang, H., Tang, J., et al. (2025). Bio-inspired optoelectronic devices and systems for energy-efficient in-sensor computing. npj Unconventional Computing, 2, 15.

Belleri, P., Pons i Tarrés, J., McCulloch, I., et al. (2024). Unravelling the operation of organic artificial neurons for neuromorphic bioelectronics. Nature Communications, 15, 5350.

Xu, Z., Shang, W., Kim, S., et al. (2024). Quantum-inspired genetic algorithm for designing planar multilayer photonic structure. npj Computational Materials, 10, 257.

Ekstrom, L., Wang, H., & Schmitt, S. (2025). Variational quantum multiobjective optimization. Physical Review Research, 7(2), 023141.

Kim, S., Ahn, S. W., Suh, I. S., et al. (2025). Quantum annealing for combinatorial optimization: A benchmarking study. npj Quantum Information, 11, 77.

Majumdar, A., Bakalov, B. N., Baron, D., & Liu, Y. (2025). *Implementing finite impulse response filters on quantum computers*. arXiv.

Dofe, J., Danesh, W., More, V., & Chaudhari, A. (2024). Natural language processing for hardware security: Case of hardware Trojan detection in FPGAs. Cryptography, 8(3), 36.

Abi-Karam, S., Sarkar, R., Seigler, A., Lowe, S., Wei, Z., Chen, H., Rao, N., John, L., Arora, A., & Hao, C. (2024). HLSFactory: A framework empowering high-level synthesis datasets for machine learning and beyond. In Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD (MLCAD '24). ACM.