

Fixed-Point Kalman Filter on PYNQ-Z2 FPGA

Kalluri Bhavana¹; T. Satya Savithri²

^{1,2}Electronics and Communication Engineering Jawaharlal Nehru Technological University Hyderabad
Hyderabad, India.

Publication Date: 2025/12/09

Abstract: This paper presents the design of a Fixed-point Kalman Filter Bank Architecture and its implementation on the PYNQ-Z2 FPGA. The proposed architecture comprises Q1.15 fixed-point arithmetic, saturation logic and reciprocal-based safe division, which are utilized to ensure numerical stability and hardware efficiency. Initially, the architecture for a single filter is designed and simulated for a sinusoidal input, and then replicated 8 times to create a single Kalman filter bank module with a span of angular frequencies from 0.001 to 0.029 rad/sample. The constructed Kalman filter bank is created as an IP and implemented on the PYNQ Z2 FPGA board. The design communicates via AXI4-DMA interface between the Processing System (PS) and Programmable Logic (PL). Experimental results demonstrate effective denoising of sinusoidal signals under varying noise levels- low, medium and high noise. The obtained results show an average RMSE below 0.15 and a correlation coefficient above 0.95. Post place-and-route results on the device indicate a resource utilization of 37,596 LUTs (70.67%), 124 DSP slices (56.36%), and 2BRAMs (1.79%), with a maximum operating frequency of 21.4 MHz and total power consumption of 1.48 W.

Keywords: Kalman Filter, Fixed-Point Arithmetic, FPGA, PYNQ-Z2, Filter Bank, AXI4-Stream, Signal Denoising, Q1.15.

How to Cite: Kalluri Bhavana; T. Satya Savithri (2025). Fixed-Point Kalman Filter on PYNQ-Z2 FPGA. *International Journal of Innovative Science and Research Technology*, 10(11), 2816-2823. <https://doi.org/10.38124/ijisrt/25nov569>

I. INTRODUCTION

Filters are fundamental components in Digital Signal Processing (DSP) used to recover clean signals from noisy signals. However, signal denoising becomes challenging when dealing with unknown or time-varying frequencies, as traditional Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters rely on fixed coefficients. This static nature prevents them from effectively tracking or adapting to changes in the signal's spectral content.

The Kalman filter, introduced by Rudolf E. Kalman in 1960, provides an optimal framework for recursive state estimation using two sequential steps: Prediction and Correction. Research has shifted towards fixed-point realization to improve efficiency and enable real-time performance on resource-constrained platforms. However, these designs encounter significant challenges due to quantization errors, overflow, and numerical instability, especially in recursive operations involving covariance matrices and divisions.

Although a few FPGA-based Kalman filter implementations exist in the literature, these are not efficient in terms of area and power. This work is an attempt to address the area and power efficiency by proposing a DSP-optimized synchronous fixed-point Kalman filter core implemented using Q1.15 arithmetic (1 sign bit and 15 fractional bits), designed to efficiently use FPGA DSP slices. The proposed architecture is first developed as a single filter and then

replicated eight times to form a Kalman filter bank spanning angular frequencies from 0.001 to 0.029 rad/sample, enabling automatic frequency selection via a robust winner selection logic mechanism. The proposed work included the following key architectural features:

- Efficient state and covariance propagation using fixed-point arithmetic.
- Saturation logic and variance clamping for improved numerical stability.
- Adaptive measurement noise estimation derived from innovation.
- Parallel filter execution with hysteresis-based winner selection to avoid rapid switching.
- AXI4-stream DMA interface enabling seamless communication between the Processing System (PS) and Programmable Logic (PL).

The resulting architecture achieves a throughput of one sample per cycle, fully leveraging FPGA parallelism while maintaining robustness against quantization effects and resource efficiency.

Section II reviews the Kalman filter Fundamentals, Section III details the proposed FPGA Architecture, Section IV presents Implementation & Results, and Section V concludes the paper with future work.

II. KALMAN FILTER FUNDAMENTALS

The Kalman filter is a recursive estimation algorithm that provides optimal state estimates for linear dynamic

systems in the presence of process and measurement noise. It operates in two main steps – prediction and correction – to iteratively refine its estimate of the system using both prior knowledge and incoming measurements.

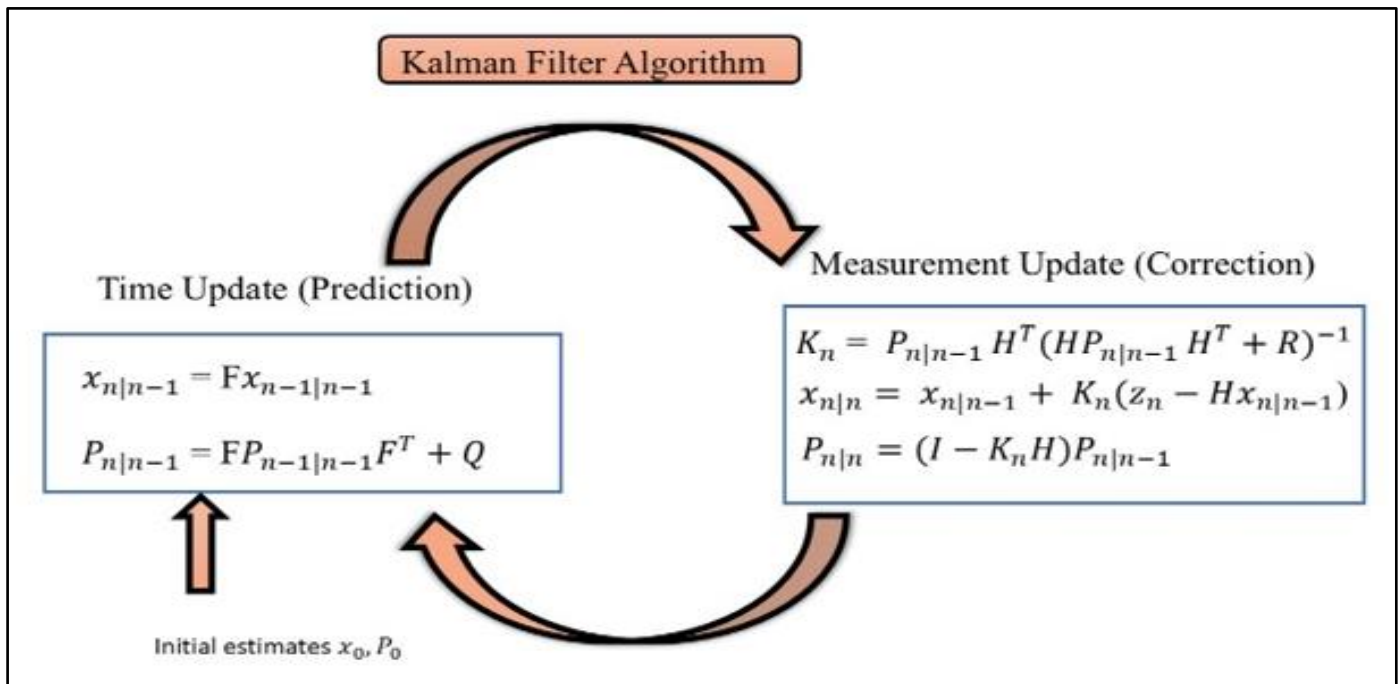


Fig 1 Block Representation of Kalman Filter Algorithm

In the first prediction step, the filter projects the previous state estimate and its covariance forward in time using the system's dynamic model:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} \quad (1)$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \quad (2)$$

Where $\hat{x}_{k|k-1}$ is the predicted state, F represents the state transition matrix, $P_{k|k-1}$ is the predicted covariance, Q denotes the process noise covariance, H is the observation matrix, and R represents the measurement noise covariance.

In the second correction step, the filter updates its prediction using the new measurement, z_k , where H maps the state to the measurement. the K_k Kalman gain determines the relative weight between the prediction and the measurement:

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \quad (3)$$

The updated state and covariance are given by:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \quad (4)$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \quad (5)$$

The Innovation term represents the discrepancy between the predicted and actual measurements:

$$y_k = z_k - H\hat{x}_{k|k-1} \quad (6)$$

Where z_k is the actual measurement at time k , and $H\hat{x}_{k|k-1}$ is the predicted measurement. These recursive equations ensure that innovation is minimized in a least-squares sense over time.

The Kalman filter is particularly effective for dynamic signal estimation where both the system model and noise statistics vary with time. In this work, the filter is adapted for sinusoidal state estimation, where the system matrix F encodes a rotational model parameterized by cosine and sine components of the signal's angular frequency. Efficient FPGA implementation of these equations requires fixed-point arithmetic, saturation logic and variance clamping to maintain numerical stability – explained in section III.

III. PROPOSED ARCHITECTURE

The proposed hardware architecture implements a fixed-point Kalman filter bank optimized for FPGA deployment on the PYNQ-z2 platform. The design translates the mathematical Kalman filter equations into pipelined and parallel Verilog modules using Q1.15 fixed-point arithmetic to balance computation precision and hardware efficiency.

➤ *The Architecture Comprises Two Hierarchical Components:*

- single Kalman filter core that estimates one sinusoidal component, shown in Fig.2 and

- A top-level filter array that spans multiple angular frequencies and automatically selects the best-performing filter in real time, shown in Fig.3.

➤ Kalman Filter Core

Each filter core shown in Fig.2 corresponds to one target angular frequency, defined by the state-transition matrix. Each clock cycle processes one new input sample and updates the estimated state. The filter implements the standard prediction and correction equations shown in Fig.1.

• Fixed Point Implementation:

All arithmetic operations use 16-bit signed fixed-point (Q1.15) format with parameters: defining the number representation as width=16, frac_bits=15, COS_DT_PARAM and SIN_DT_PARAM as precomputed constants for each frequency. Q_val and R_base as the process and measurement noise covariance. Dedicated DSP48 slices perform multiply-accumulate operations for state and covariance propagation.

The helper functions used ensure numerical stability, and saturation logic limits all results within the 16-bit signed range. Variance clamping limits the covariance values to prevent overflow or divergence. Fixed-point division and square root approximation are used for the computation of the kalman gain and signal amplitude.

• Adaptive Measurement Noise Estimation:

Each core includes an adaptive measurement noise estimator derived from innovation. The absolute magnitude $|y_k| = |z_k - H\hat{x}_{k-1}|$ is smoothed using an exponential moving average (EMA):

$$EMA_k = \frac{7 \cdot EMA_{k-1} + |y_k|}{8} \quad (7)$$

$$R_k = 1.25 \times EMA_k \quad (8)$$

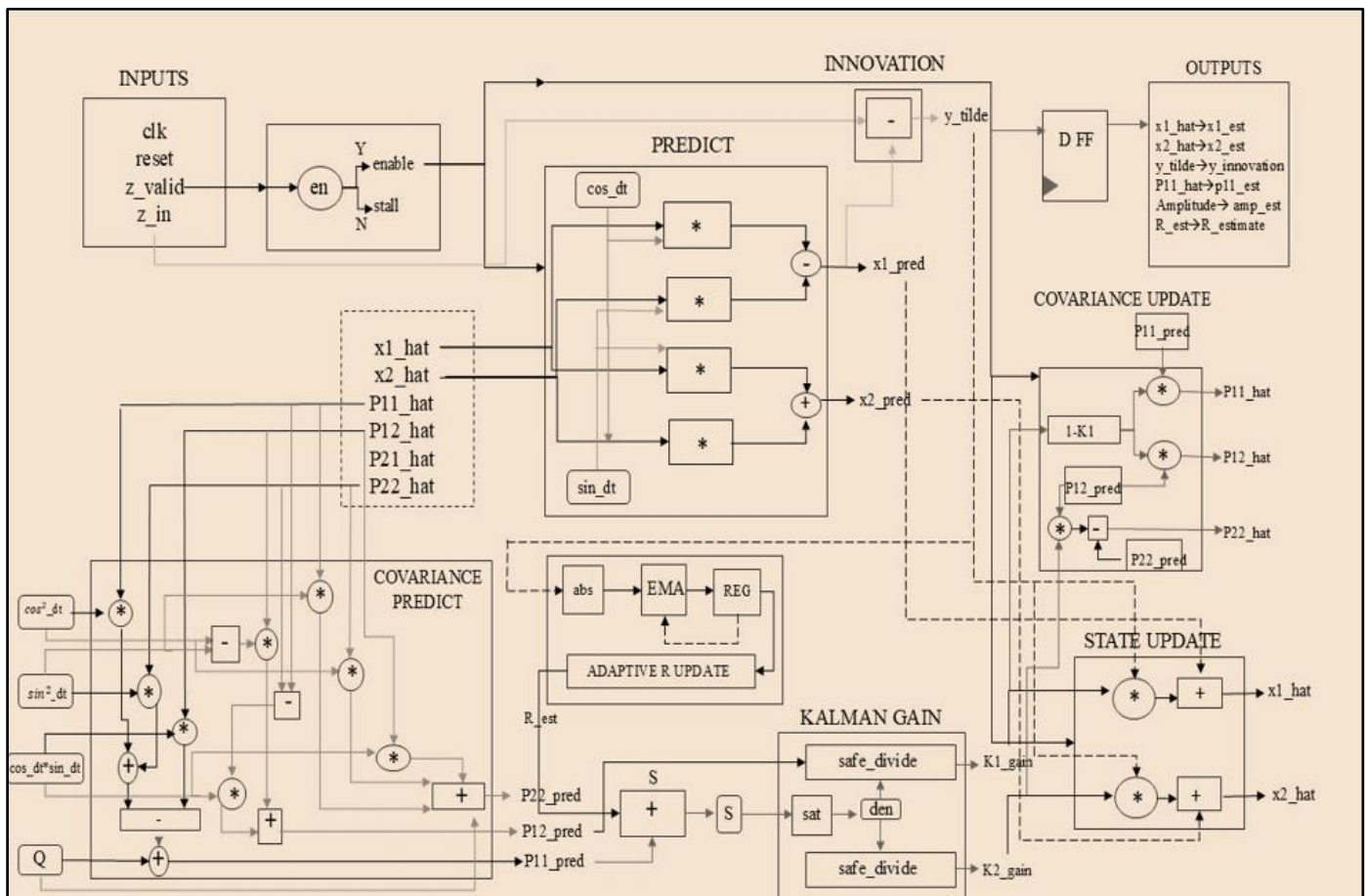


Fig 2 Kalman Filter Architecture

Here, consider the first 50 samples, R_k remains fixed at R_{base} . To allow convergence. Afterward, R_k Adapts dynamically to reflect measurement noise while avoiding overreaction to transients.

Each core outputs represented as: $\hat{x}_{n,1}, \hat{x}_{n,2}$ (state estimates), y_n (innovation), $P_{n,n,11}$ (posterior variance), \hat{A}

(estimated signal amplitude) and R_{est} (adaptive measurement noise estimate).

➤ Kalman Filter Bank

The top-level module instantiates $N=8$ parallel Kalman filter cores, each tuned to a distinct angular frequency spanning 0.001 – 0.029 rad/sample. All filters receive the same input and process it concurrently, exploiting FPGA parallelism to achieve one sample per clock throughput.

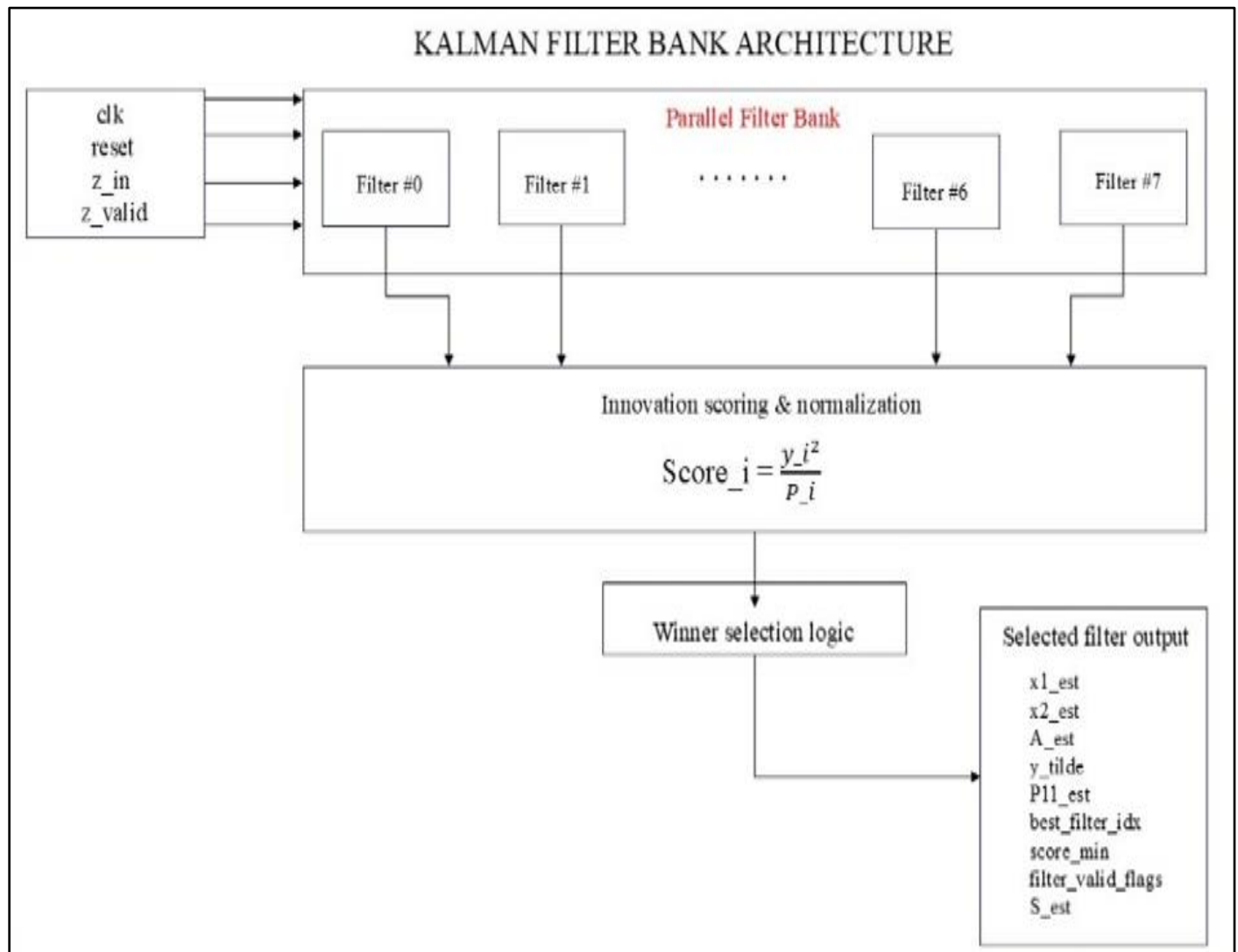


Fig 3 Kalman Bank Architecture

- **Innovation-Based Scoring:**

Each candidate filter of the filter bank computes a normalized innovation score that balances prediction error and confidence, where is the innovation and is the posterior variance.

$$\text{score}_i = \frac{y_i^2}{P_i} \quad (9)$$

The filter with the lowest average score over a defined observation window is considered the most reliable estimator for the given input.

Winner selection logic: A winner selection logic is a hysteresis-based selection mechanism that prevents rapid switching between filters. A candidate filter must maintain the lowest score for several consecutive samples before being declared the winner. This logic outputs: index of the dominant frequency, corresponding state estimates, amplitude estimate and normalized innovation score.

- **Output Timing and Interface**

When valid data are available, the selected state and amplitude are flagged with output_valid=1. All computations from input sample acquisition to state update are completed in a single clock cycle per filter. The system integrates with an AXI4-stream DMA interface to facilitate real-time data exchange between the processing system (PS) and programming logic (PL) on the PYNQ-Z2 platform.

IV. IMPLEMENTATION & RESULTS

- **Vivado Simulation**

A discrete-time noisy sinusoid was generated in the behavioral testbench to verify the Kalman filter's ability to track the frequency encoded in its state-transition matrix. Each core received the same noisy input, and the predicted and corrected state outputs are monitored to confirm convergence and noise suppression. The simulation was run for 2000 samples using a 100MHz clock, generating the waveform.

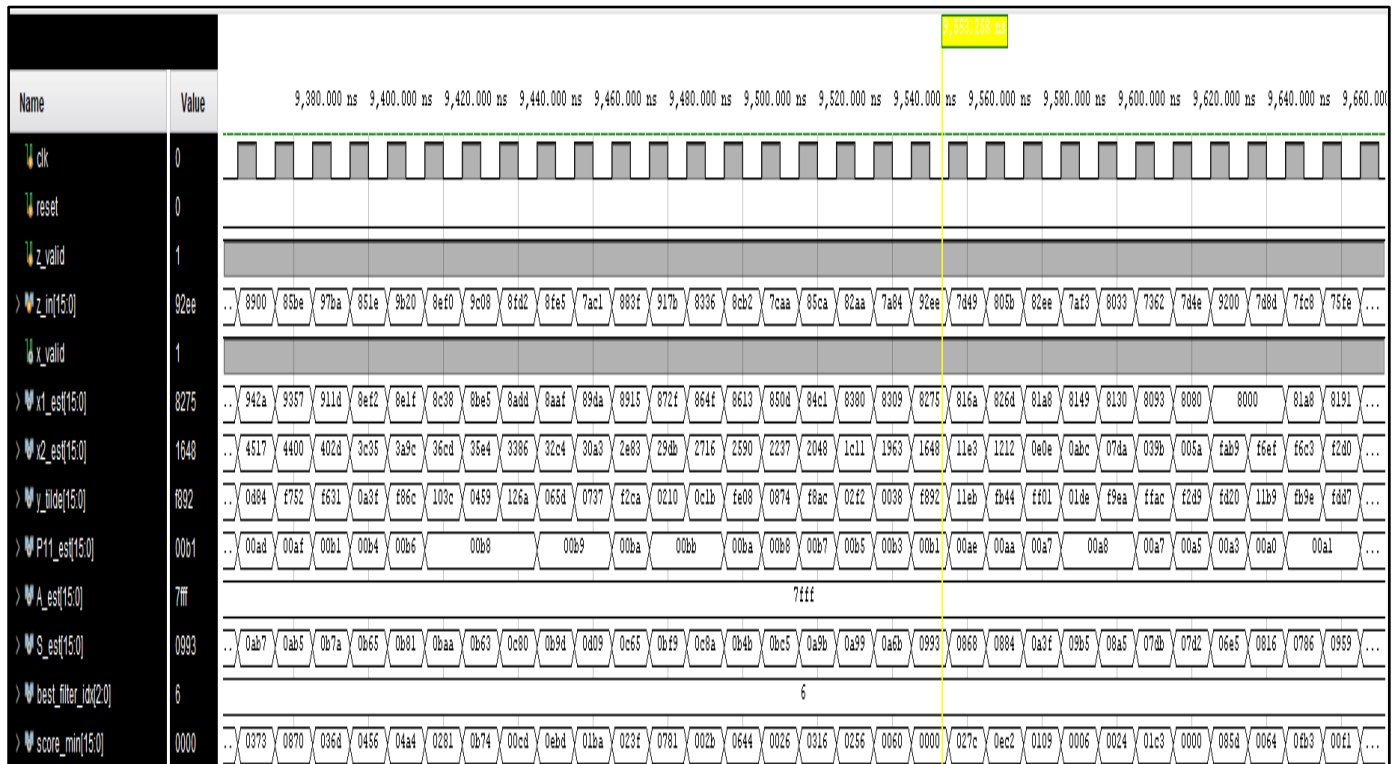


Fig 4 Vivado Simulation Waveform

Fig.4 shows the behavioral simulation output of the Kalman filter bank in Vivado. The signals $x1_est$, $x2_est$, and A_est converge smoothly after initialization, demonstrating that each filter tracks the true sinusoidal state. The $best_filter_idx$

The output indicates the winning filter corresponding to the frequency closest to the input signal. ($\omega = 0.025rad/sample$). The innovation term y_tilde reduces over time,

conforming to the filter's convergence and effective noise suppression.

➤ RTL Elaboration

The design is elaborated in Xilinx Vivado 2023.2, producing the RTL schematic that expands parameterized instances and reveals inter-module connectivity before synthesis.

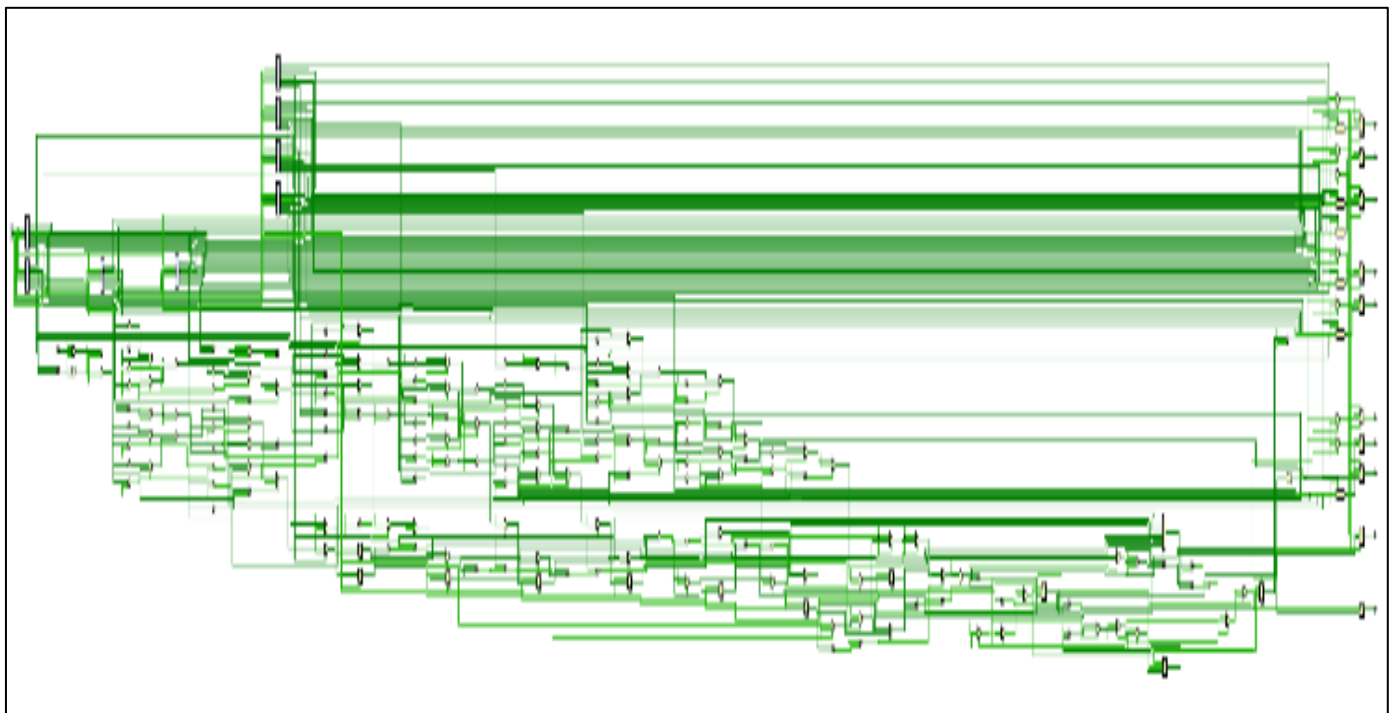


Fig 5 RTL Analysis of Kalman Filter Bank

➤ *Synthesis Report*

- *Bitstream Deployment & Testing*

Computation is divided between the processing system (PS) and programmable logic (PL). The PL hosts the parallel Kalman filter bank in Q1.5 arithmetic, while the PS manages data and visualization through a Jupyter interface.

An AXI DMA engine provides the streaming transfer: MM2S channel sends noisy inputs from PS memory to PL. S2MM channel returns filtered outputs for storage and plotting.

Handshake signals of AXI4 stream – tvalid/tready synchronize the dataflow, preventing underrun or overflow. The block design is illustrated in Fig. 6.

After synthesis and implementation, a bitstream for the block design is generated and loaded via overlay(“bitfile.bit”) on the pynq-z2. Data transfers were initiated using dma_send. Send channel. transfer(input_buf) and dma_recv. recvchannel. transfer(output_buf) to enable the filtering.

➤ *Performance Metrics*

Tracking accuracy was evaluated using root mean square error (RMSE) and correlation coefficient (r) between true and estimated signals. A low RMSE and high (r) indicate precise tracking and effective noise suppression, while degraded RMSE and r values reveal frequency mismatch or transient overload during estimation.

Table 1 Resource Utilization

Resources	Used	Available	Utilization (%)
LUTs	37596	53200	70.7
LUTRAM	630	-	3.6
Flip-flops	7354	106400	6.9
DSP48 slices	124	220	56.4
Block RAM	2	140	1.8

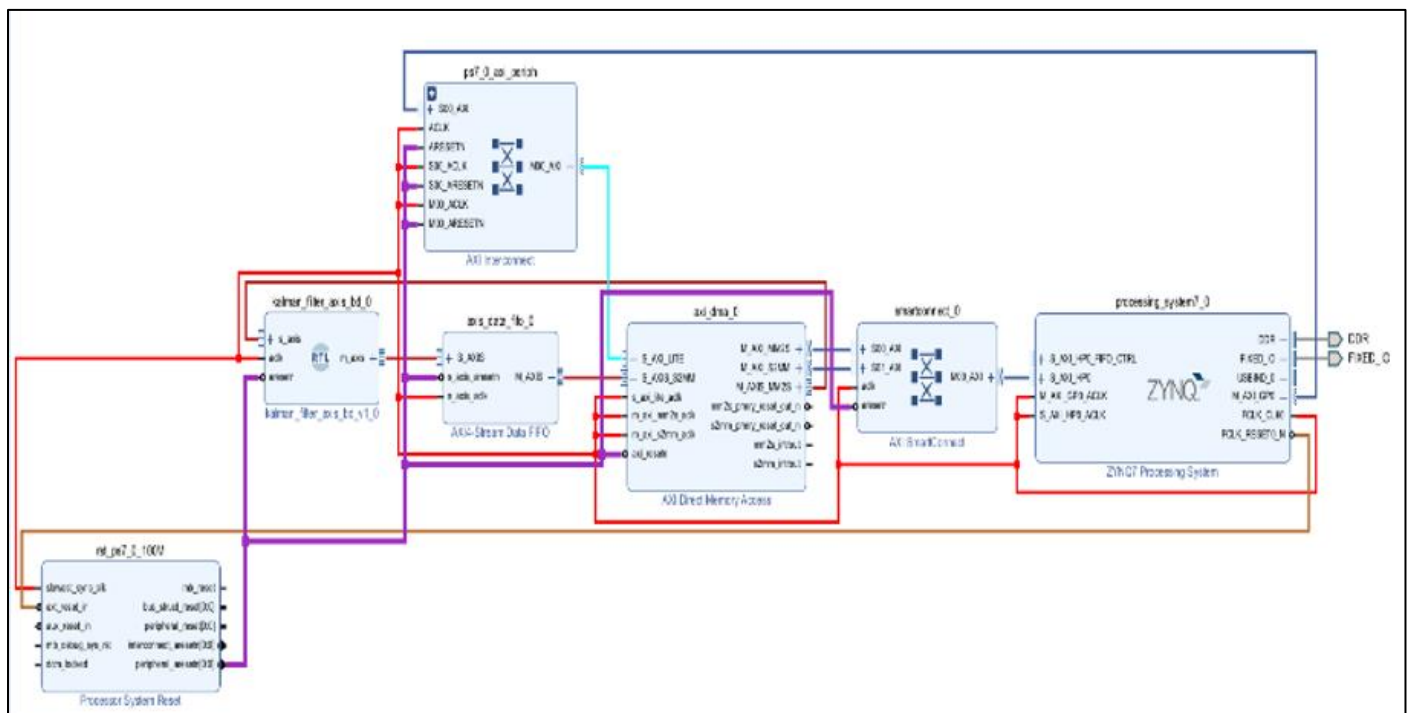


Fig 6 Vivado Block Design

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (x_k - \hat{x}_k)^2} \quad (10)$$

Where x_k and \hat{x}_k Denote the true and Kalman estimated samples, \bar{x} and $\bar{\hat{x}}$ represent their respective means, and N is the total number of samples.

$$r = \frac{\sum_{k=1}^N (x_k - \bar{x})(\hat{x}_k - \bar{\hat{x}})}{\sqrt{\sum_{k=1}^N (x_k - \bar{x})^2 \sum_{k=1}^N (\hat{x}_k - \bar{\hat{x}})^2}} \quad (11)$$

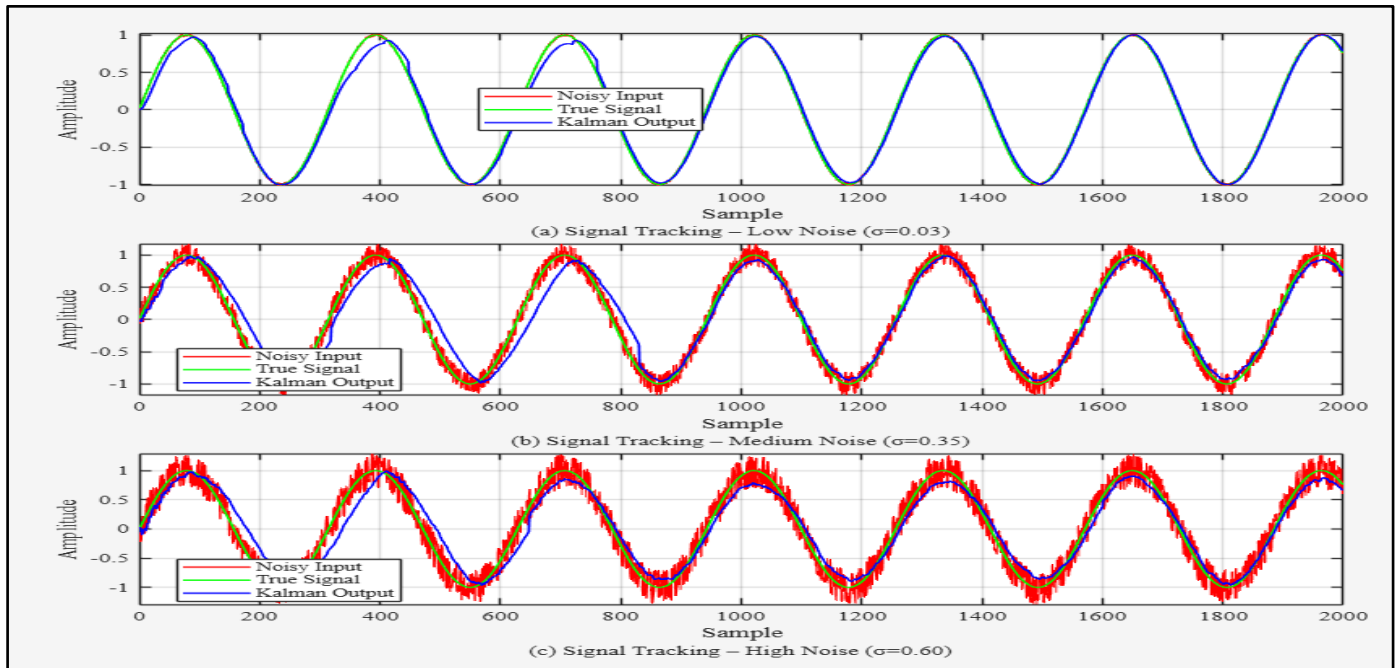


Fig 7 Kalman Filtering for Different Noise Levels

Fig.7 shows the time domain tracking performance of the filter bank for sinusoidal estimation under varying noise levels. The true signal (green), noisy input (red), and Kalman output (blue) are plotted for (a) low noise ($\sigma=0.03$), (b) medium noise ($\sigma=0.35$) and (c) high noise ($\sigma=0.60$). The

filter maintains phase coherence and accurate tracking even under severe noise conditions with correlation coefficients above 0.95 in all cases.

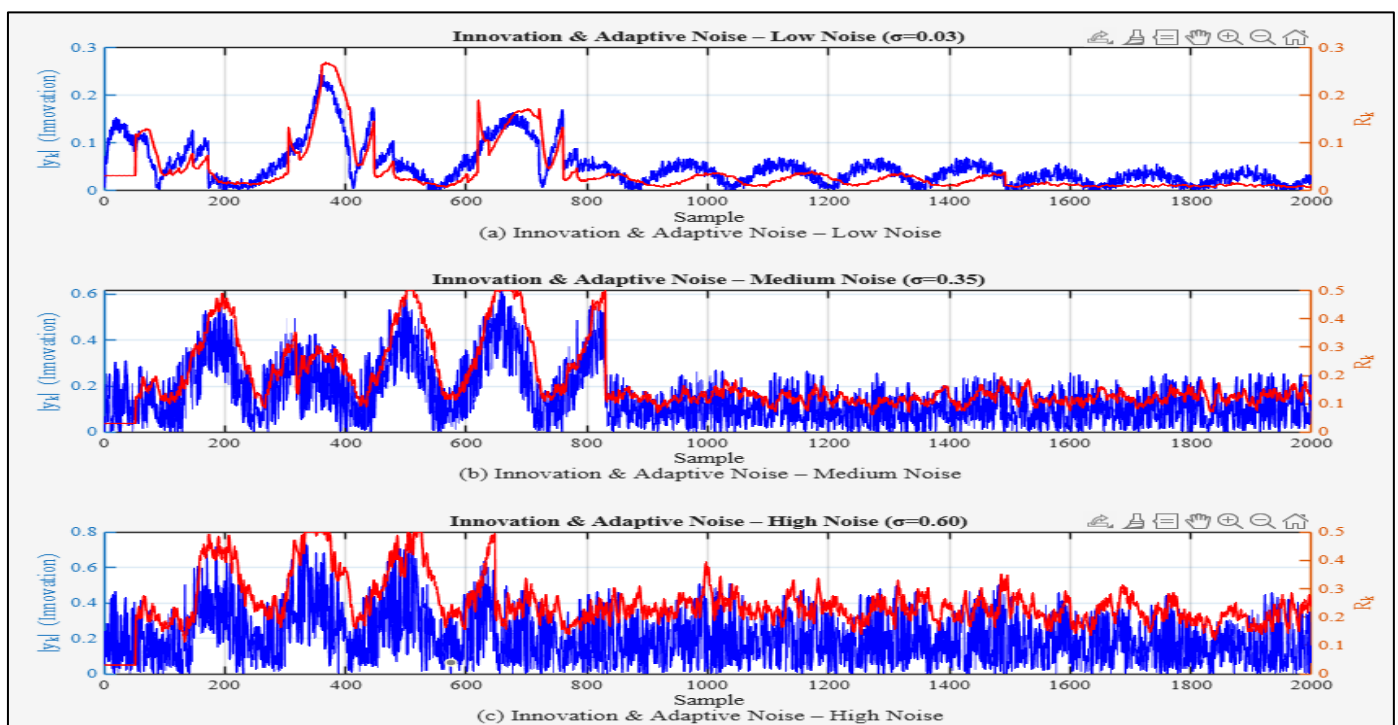


Fig 8 Innovation and Adaptive Noise Covariance Plot

Fig.8 illustrates the temporal evolution of the innovation term and adaptive noise covariance across different noise conditions. Under low noise, both signals remain stable, confirming strong confidence in measurements. As the input noise increases, the innovation magnitude and adaptive noise estimate grow, demonstrating the filter's capability to adaptively tune its gain and maintain robustness under varying noise environments.

V. CONCLUSION & FUTURE SCOPE

This work presented a fixed-point Kalman filter bank written in Verilog and implemented on the PYNQ-Z2 FPGA for denoising the noisy sinusoidal signals. The proposed architecture combines adaptive measurement noise estimation, saturation and variance clamping, and winner selection logic mechanism across eight parallel filters operating in Q1.15 arithmetic. The hardware results demonstrated $RMSE < 0.15$ and correlation coefficients > 0.95 , conforming to accurate estimation and noise suppression with moderate resource utilization and real-time throughput. These results validate the effectiveness of the fixed-point Kalman filtering for FPGA systems.

Future work will investigate dynamic partial reconfiguration on an FPGA to swap Kalman filter variants at runtime, allowing the system to adapt automatically to changing signal conditions. This flexibility would make the design suitable for audio and speech denoising.

ACKNOWLEDGMENT

The licenses for the Vivado tool and hardware tool used for this research paper, the PYNQ-Z2 board, are provided by MeitY under the C2S project titled "Development of SOC system with Vision-based UAV and Remote Mobile arm for Precision Agriculture", JNTUH CEH.

REFERENCES

- [1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME – Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- [2] G. Welch and G. Bishop, "An introduction to the Kalman filter," *UNC-Chapel Hill Department of Computer Science*, Tech. Rep. 95-041, 2006.
- [3] S. Haykin, *Adaptive Filter Theory*, 5th ed. Pearson, 2013.
- [4] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice-Hall, 1979.
- [5] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [6] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 4th ed. Wiley-IEEE Press, 2015.
- [7] S. Liu, M. Chan, and P. Li, "FPGA implementation of a fixed-point Kalman filter for real-time signal tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 8, pp. 5857–5866, Aug. 2020.
- [8] J. Lee and H. Kim, "Efficient hardware implementation of adaptive Kalman filters using fixed-point arithmetic," *IEEE Access*, vol. 9, pp. 148 750–148 763, 2021.
- [9] V. Kumar and S. Mishra, "FPGA-based implementation of fixed-point digital filters using Vivado and AXI interface," *IEEE Trans. Education*, vol. 63, no. 4, pp. 543–551, Nov. 2020.
- [10] M. A. El-Sayed, F. Mahmoud, and H. Abd-El-Kader, "Hardware efficient Kalman filter for sensor fusion applications," *IEEE Access*, vol. 10, pp. 49 320–49 333, 2022.
- [11] Xilinx Inc., *AXI4-Stream Interface Protocol Specification*, UG761, Ver. 1.0, 2023.
- [12] PYNQ Community, "PYNQ-Z2 board and overlay development documentation," *pynq.io*, 2024. [Online]. Available: <https://pynq.io>
- [13] H. Gao, X. Zhu, and L. Wang, "Resource-optimized Kalman filter for embedded FPGA sensor systems," *IEEE Embedded Systems Letters*, vol. 16, no. 3, pp. 112–115, 2024.
- [14] A. Singh and T. Patel, "Real-time FPGA-based adaptive Kalman filter for dynamic signal estimation," *IEEE Access*, 2023.