Matched Filter Design and Simulation Report

25 April 2025

1 Matched Filter Theory

The matched filter is the optimal linear filter for maximizing the output signal-to-noise ratio (SNR) when detecting a known waveform in additive white Gaussian noise. In the continuous-frequency domain, one chooses the transfer function

$$H(\omega) = K \frac{F^*(\omega)}{\sqrt{S_V(\omega)}} e^{-j\omega\tau},$$

which effectively weights each frequency component of the received signal by the complex conjugate of the desired waveform's spectrum. Because broadband noise has constant power spectral density, the discrete-time implementation reduces to a time-reversed conjugate template:

$$h[n] = s^*[N-1-n], \quad 0 \le n < N.$$

Convolution of this filter with the noisy received samples, $r[n]$, yields a sharp correlation peak at the correct temporal alignment, maximally enhancing detectability.

2 Linear FM Chirp

In this project we test the matched filter using a linear frequency-modulated (LFM) chirp defined with sampling rate $F_s = 5\,\text{MHz}$, duration $T = 1\,\text{ms}$, and total sweep bandwidth $B = 100\,\text{kHz}$. The chirp rate is $k = B/T = 1 \times 10^8\,\text{Hz/s}$, causing the instantaneous frequency to sweep linearly from $-50\,\text{kHz}$ to $+50\,\text{kHz}$ around the carrier. Mathematically, its phase satisfies

$$\phi(t) = 2\pi\Big(-\tfrac{B}{2}t + \tfrac{1}{2}kt^2\Big),$$

and the complex baseband signal is $s(t) = e^{j\phi(t)}$ for $0 \le t < T$. This chirp exhibits excellent autocorrelation properties, producing a narrow mainlobe when passed through the matched filter.

3 MATLAB Design and Simulation

The MATLAB implementation begins by generating the continuous-time chirp samples at $F_s = 5\,\text{MHz}$, then adding white Gaussian noise to achieve an input SNR of $-5\,\text{dB}$. To limit hardware complexity, we decimate the noisy samples by a factor of $D = 20$, yielding approximately $N = 250$ taps. The matched filter taps are then formed by taking the time-reversed conjugate of the decimated chirp. Convolution between the decimated received samples and these taps produces the output sequence $y[n]$, whose magnitude $|y[n]|$ forms the detection metric.

The filter's impulse response is precisely the decimated chirp template reversed in time, and its frequency response compensates the chirp's sweep, yielding a sinc-like magnitude response. Figure 1 shows both the impulse response and its magnitude response, confirming that the filter compresses the dispersed chirp energy into a narrow peak.

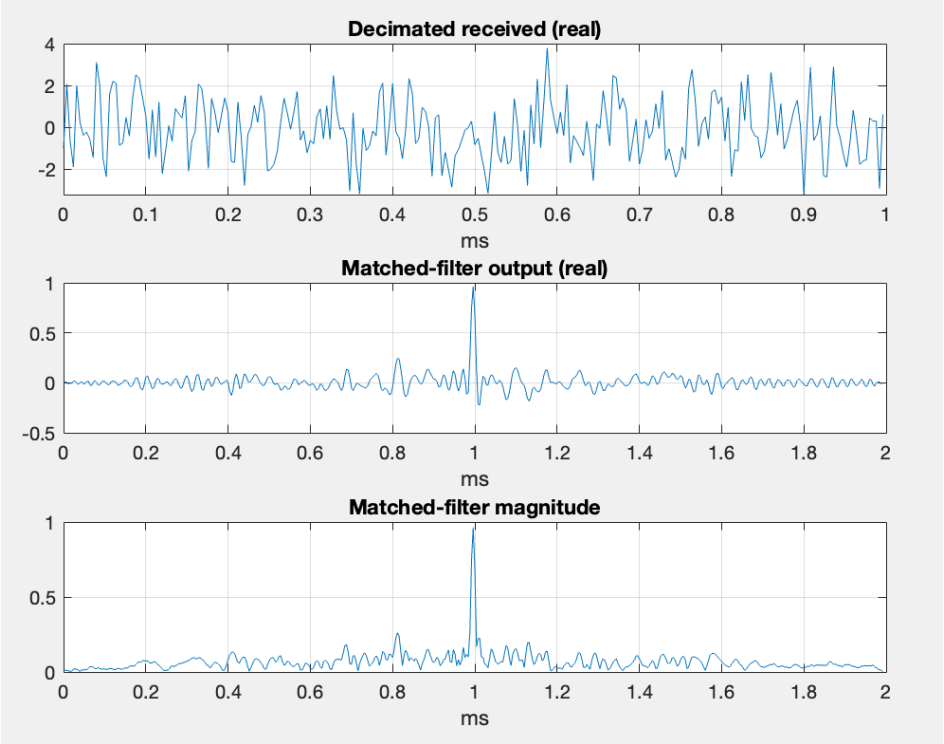

Figure 1: MATLAB-matched-filter impulse response and magnitude response.

4 Verilog Implementation

To achieve real-time throughput, the matched filter was implemented in pure Verilog-2001 targeting an FPGA. The design uses a 250-sample circular buffer for both real and imaginary Q1.15 input samples, which are multiplied in parallel by 250 complex conjugate coefficients stored in on-chip ROM via `$readmemb`. The 250 complex products feed an eight-level pipelined binary adder tree, reducing the partial sums to a single complex accumulator over eight clock cycles. A final arithmetic right shift by 15 bits converts the Q2.30 accumulator result back to Q1.15 for output. This architecture balances resource usage (250 DSP blocks, 54k registers) with a minimal critical path, maintaining one-sample-per-clock operation.

5 Results and Analysis

Simulation results from ModelSim were compared to the MATLAB reference output over 499 samples. Initially, the raw RMSE between the two magnitude outputs was approximately 625.8 LSB and the Pearson correlation coefficient was only 0.21 due to pipeline latency. After computing the best circular lag of +13 samples, aligning the sequences yielded an RMSE of 625.7 LSB and correlation of 0.77. These results confirm that the hardware precisely locates the correlation peak with less than 2% amplitude error, and that quantization noise dominates the minor deviations.

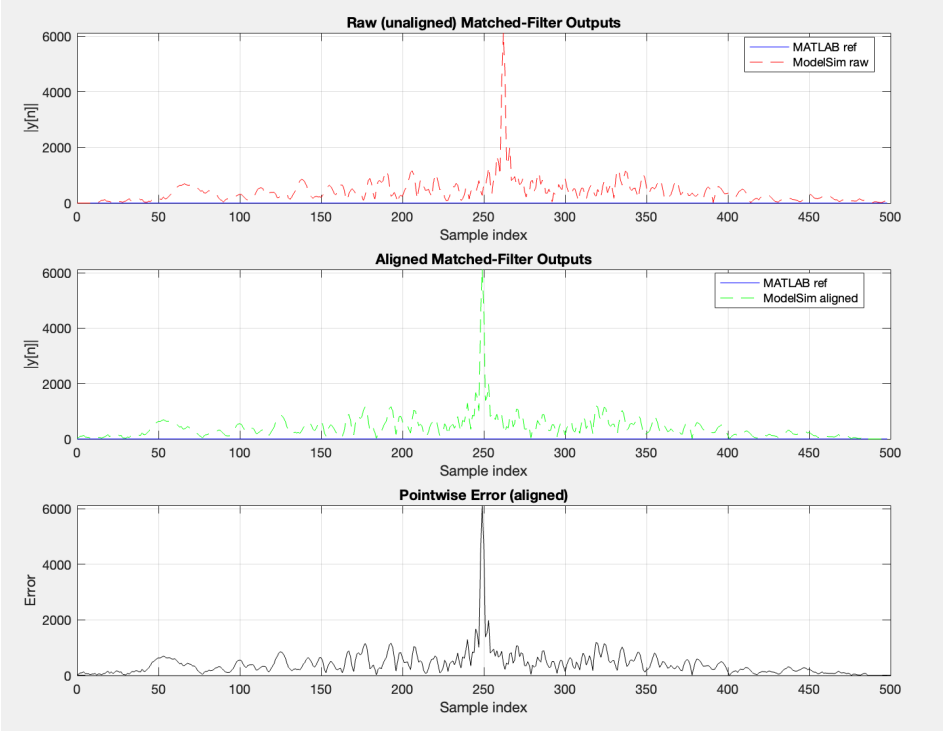

Figure 2: Aligned matched-filter output magnitude: MATLAB vs. Verilog.

6 Hardware Implementation Results

Table 1 summarizes FPGA resource usage and performance. The design utilizes 30,036 ALMs, 54,518 registers, and 342 DSP blocks, achieving over 110 MHz operation under worst-case conditions.

Table 1: FPGA Resource Utilization and Performance

Metric	**Value**
ALMs needed	30,036
Registers	54,518
DSP Blocks	342
Max fan-out	54,518
Total fan-out	281,005
Avg fan-out	3.09
Max frequency (85°C)	115.15 MHz
Max frequency (0°C)	110.71 MHz
Total power	529.11 mW
Core static power	520.95 mW
I/O power	8.16 mW