Hilbert Transformator IP Cores

Martin Kumm

December 27, 2008

1 Introduction

The Hilbert Transform is an important component in communication systems, e.g. for single sideband modulation/demodulation, amplitude and phase detection, etc. It can be formulated as filtering operation which makes it possible to approximate the Hilbert Transform with a digital filter. Due to the non-causal and infinite impulse response of that filter, it is not that easy to get a good approximation with low hardware resource usage. Therefore, different filters with different complexities have been implemented. The detailed discussion can be found in [KS08].

2 Mathematical Background

The Hilbert Transform can be defined as filter in the discrete frequency domain as

$$H_H(e^{j\Omega}) = \begin{cases} -j & \text{for } 0 < \Omega < \pi \\ j & \text{for } -\pi < \Omega < 0 \\ 0 & \text{for } \Omega = 0 \quad , \end{cases} \tag{1}$$

with $\Omega = 2\pi f/fs$, f_s is the sampling frequency in Hz and $j = \sqrt{-1}$. The magnitude is unity for all frequencies not equal to zero and the phase is $-90°$ for all positive, and $+90°$ for all negative frequencies. The inverse Fourier transform leads to the impulse response

$$h_H(n) = \frac{1 - \cos(\pi n)}{\pi n} = \begin{cases} 2/(\pi n) & \text{for odd } n \\ 0 & \text{for even } n. \end{cases} \tag{2}$$

A so-called analytic signal can be generated from a real valued signal by extending the signal with its own Hilbert transform as the imaginary part:

$$\underline{x}(n) = x(n) + j\mathcal{H}\{x(n)\} \tag{3}$$

This signal has the property that no spectral components exist for the lower half of the z-plane ($-\pi < \Omega < 0$). The operation in (3) can be formulated as a filter operation with

$$\underline{H}_A(e^{j\Omega}) = 1 + jH_H(e^{j\Omega}) \quad . \tag{4}$$

The realizations are filters that generates an analytic signal from a real valued signal, therefore these filters are called analytic filter ($\underline{H}_{Ax}$). The implementations described in the next section realize analythic filters.

3 Implementation

The first analytic filter is implemented as FIR filter

$$\widetilde{H}_{H1}(z) = z^{-N/2} \sum_{k=-N/2}^{N/2} b_k z^{-k} \tag{5}$$

Matlab (`firls()`) was used to calculate the impulse response for an Hilbert filter with an order of $N = 10$ to be b_1 =163/256, b_3 =54/256, b_5 =32/256 and zero for all even coefficients. An additional delay is needed to form the analytic filter

$$\underline{H}_{A1}(z) = z^{-5} + j\widetilde{H}_{H1}(z) \quad . \tag{6}$$

Due to the zero coefficients and the asymmetric impulse response, only three real multipliers are needed. The structure of the implementation is shown in Figure 1. Two pipeline registers are implemented after the multiplication and the addition in the last stage. Therefore, the delay of the real part must be set to z^{-7}. Detailed plots of the filter are shown in Figure 4.

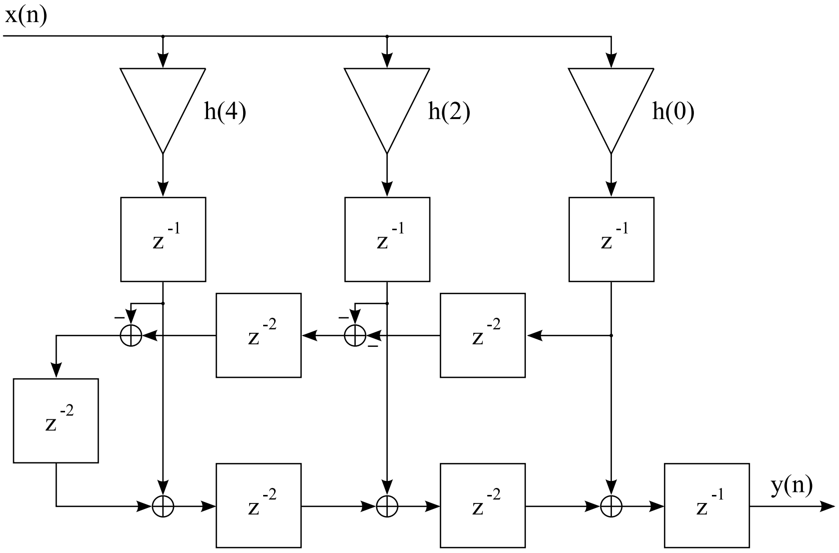


Figure 1: Structure of $\widetilde{H}_{H1}(z)$

The following analytic filters are based on frequency sampling filters (see [KS08] for further details), so only the transfer functions are given here:

$$\underline{H}_{A2}(z) = F_{z4}(z)C_{+90}(z) \tag{7}$$

with

$$F_{z4}(z) = 1 - z^{-4} \tag{8}$$

and

$$C_{+90}(z) = \frac{z^{-1}}{1 - jz^{-1}} \quad . \tag{9}$$

Detailed plots of the filter are shown in Figure 5.

The third one is defined as

$$\underline{H}_{A3}(z) = \left(\underline{H}_{A2}(z)C_{0/180}(z)C^{-1}_{-30/-150}(z)\right)^2 \tag{10}$$

with

$$C_{0/180}(z) = \frac{z^{-2}}{1 - z^{-2}} \tag{11}$$

and

$$C^{-1}_{-30/-150}(z) = (1 + jz^{-1} - z^{-2}) \quad . \tag{12}$$

Detailed plots of the filter are shown in Figure 6. A block diagram of the implementation is shown in Figure 2.

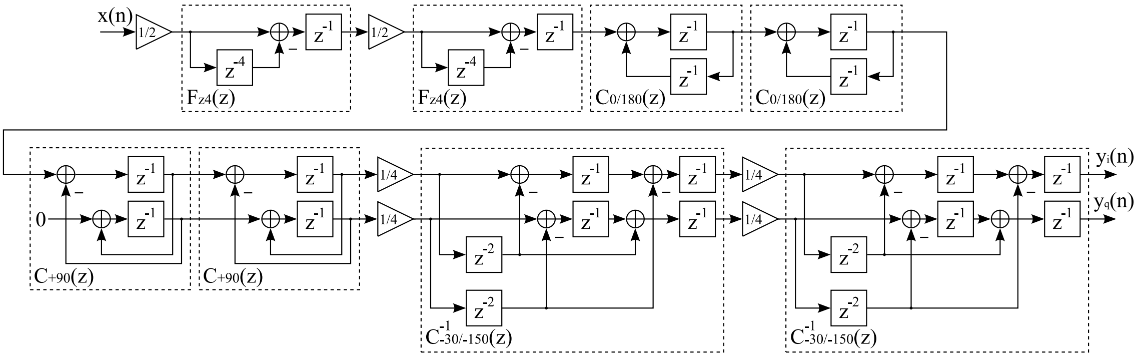


Figure 2: Block diagram of $\underline{H}_{A3}(z)$

The last filter is an extension of $\underline{H}_{A3}(z)$ with an IIR component, so the total impulse response becomes IIR ($\underline{H}_{A1-A3}(z)$ have a finite impulste response even though IIR components are included). The transfer function is

$$\underline{H}_{A4}(z) = \underline{H}_{A3}(z)C_P^2(z) \tag{13}$$

with

$$C_P(z) = \frac{z^{-2}}{1 - \frac{1}{2}z^{-2}} \quad . \tag{14}$$

Detailed plots of the filter are shown in Figure 7. A block diagram of the implementation is shown in Figure 3.

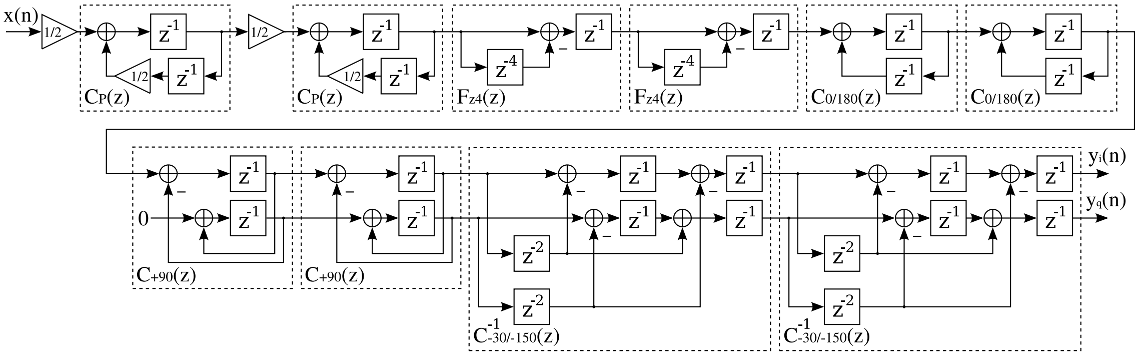


Figure 3: Block diagram of $\underline{H}_{A4}(z)$

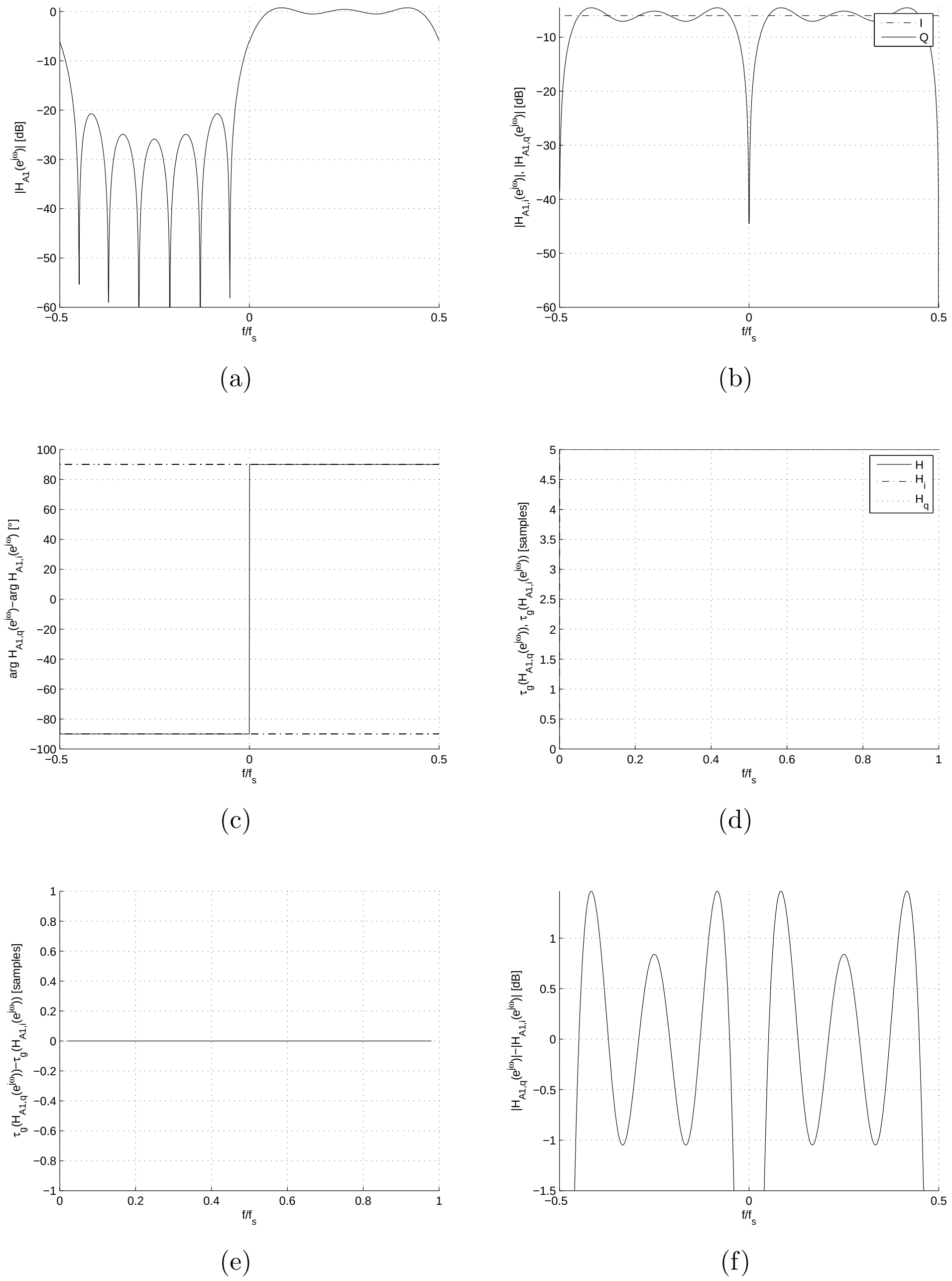
4 Comparison of the Filters

The resource usage of the filters is shown in Table 1. All filters have been compiled for an Altera Cyclone I FPGA (EP1C6Q240C8).

Filter	N_A	N_R	N_{LE}	f_{max} [MHz]
$\underline{H}_{A1}(z)$	10	6	337	266
$\underline{H}_{A2}(z)$	3	4	110	275
$\underline{H}_{A3}(z)$	16	18	562	191
$\underline{H}_{A4}(z)$	18	38	679	193

Table 1: Ressource usage of the implemented filters. Symbols: N_A, N_R and N_{LE}: required number of adders/subtracters (incl. output register), registers (incl. pipeline registers) and logic elements on Altera Cyclone I, f_{max}: maximum speed on Altera Cyclone I. The constant coefficient multipliers of $\underline{H}_{A1}(z)$ were represented by equivalent adders using the canonic signed digit system [MB07] for comparison.

5 Frequency Responses of the implemented Filters



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Plots of the filter $H_{A1}(e^{j\omega})$, (a) normalized magnitude response of the total filter, (b) normalized magnitude response of the real and imaginary parts (I and Q), (c) phase response, (d) group delay of the total filter and I and Q paths, (e) group delay error between I and Q, (f) amplitude error between I and Q

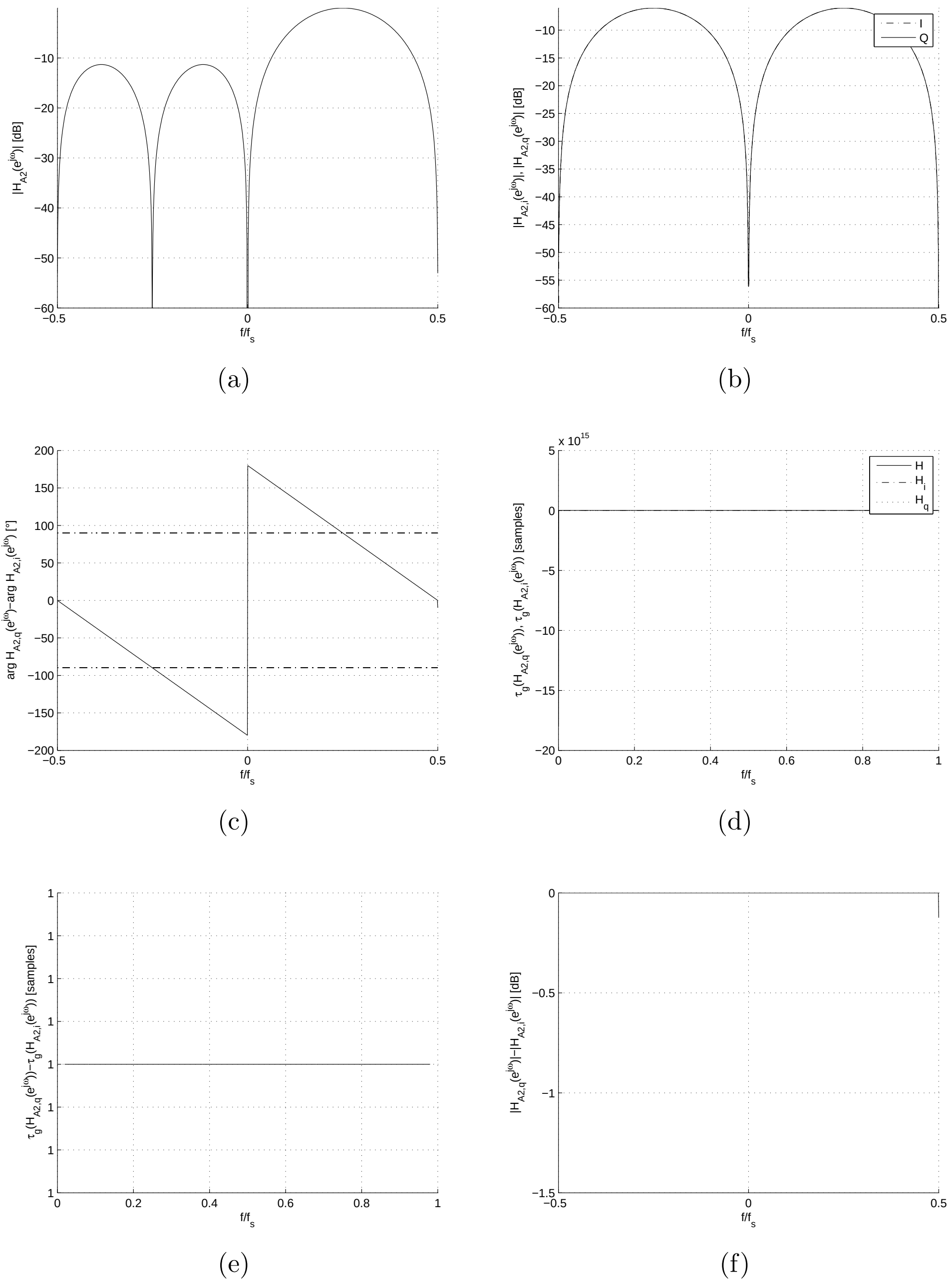
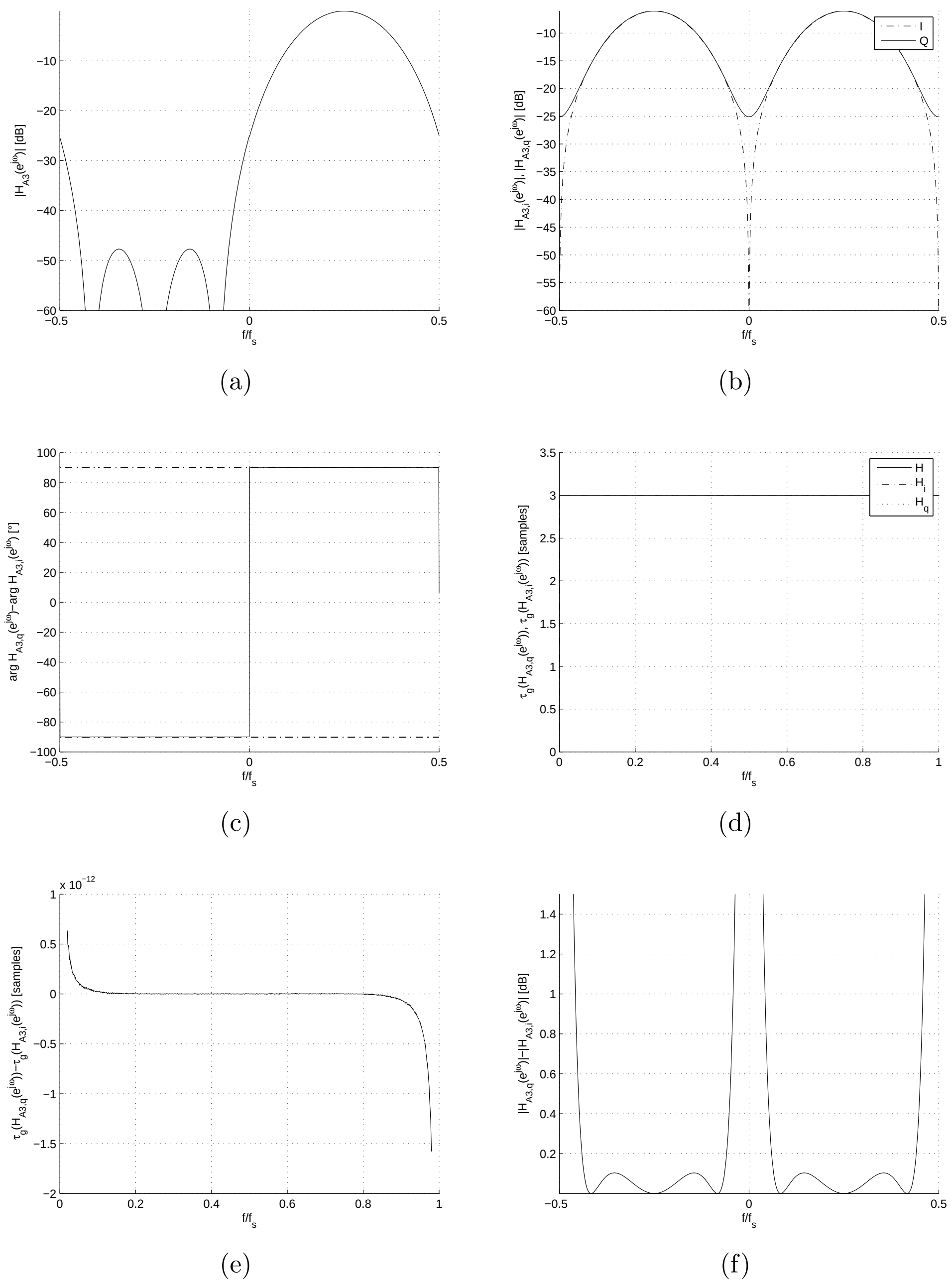


Figure 5: Plots of the filter $H_{A2}(e^{j\omega})$, (a) normalized magnitude response of the total filter, (b) normalized magnitude response of the real and imaginary parts (I and Q), (c) phase response, (d) group delay of the total filter and I and Q paths, (e) group delay error between I and Q, (f) amplitude error between I and Q



(a) (b)

(c) (d)

(e) (f)

Figure 6: Plots of the filter $H_{A3}(e^{j\omega})$, (a) normalized magnitude response of the total filter, (b) normalized magnitude response of the real and imaginary parts (I and Q), (c) phase response, (d) group delay of the total filter and I and Q paths, (e) group delay error between I and Q, (f) amplitude error between I and Q

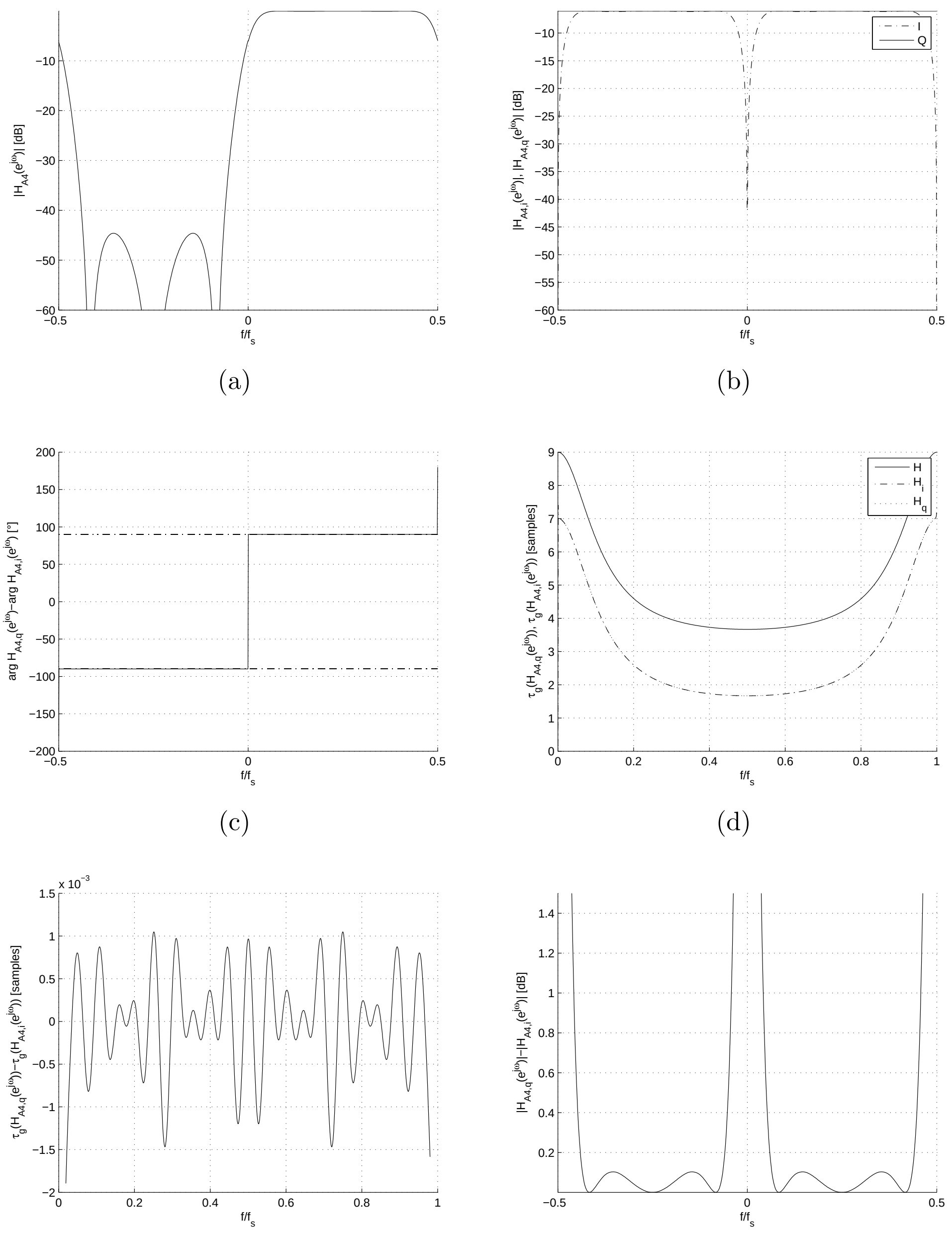


Figure 7: Plots of the filter $H_{A4}(e^{j\omega})$, (a) normalized magnitude response of the total filter, (b) normalized magnitude response of the real and imaginary parts (I and Q), (c) phase response, (d) group delay of the total filter and I and Q paths, (e) group delay error between I and Q, (f) amplitude error between I and Q

References

[KS08] Martin Kumm and Mohammad Shahab Sanjari. Digital Hilbert Transformers for FPGA-based Phase-Locked Loops. In *International Conference on Field Programmable Logic and Applications 2008*, pages 251–256, 2008.

[MB07] Uwe Meyer-Baese. *Digital Signal Processing with Field Programmable Gate Arrays, 3rd Edition.* Springer, Berlin, 2007.