DIGITAL HILBERT TRANSFORMERS FOR FPGA-BASED PHASE-LOCKED LOOPS

Martin Kumm, M. Shahab Sanjari

Gesellschaft für Schwerionenforschung (GSI)
64291 Darmstadt
email: m.kumm@gsi.de, s.sanjari@gsi.de

ABSTRACT

The phase detector is a main building block in phase-locked loop (PLL) applications. FPGAs permit the realtime implementation of the CORDIC algorithm which offers an efficient solution for an accurate phase detection, provided that the signal is available as an analytic signal. Different architectures for generating analytic signals by approximating the Hilbert transform were analyzed. Thereby, the focus has been on the demands based on the PLL application and the efficient implementation on FPGAs. Two methods were implemented using either FIR or complex filters. The FIR method results in a remaining phase error that has a zero mean value in time domain. An efficient IIR low-pass structure is proposed to suppress this phase error. The complex filters were implemented using a novel method based on complex, multiplier-less frequency sampling filters. Structures with different complexities are presented. A better result was achieved compared to a standard IIR filter design.

1. INTRODUCTION

Recent advances in the technology of field programmable gate arrays (FPGA) enable the implementation of all-digital phase-locked loops (ADPLL) working directly in the high frequency range, unlike traditional micoprocessor-based approches. An offset-local oscillator (Offset-LO) PLL [1] was needed within a phase measurement system for the heavy-ion synchrotron (SIS18) accelerator at GSI. This system is used to determine the phase difference of RF cavities in relation to other cavities [2] or the beamphase [3] to compensate for deviations in a closed-loop control.

PLLs are widely used in communication systems. Many specialized integrated circuits exist that suit most of these applications. Due to the requirements of this Offset-LO, an FPGA-based implementation was realized instead, which particularly covers the relatively wide frequency range (f_{RF} $= 0.8-5.4$ MHz) and provides a phase accuracy better than 5° during the accelerating frequency ramp ($\dot{f}_{RF,max} = 68$ MHz/s).

The phase detector is a key component in a PLL. Most VLSI implementations use phase detection methods, such as flip-flop based phase frequency detectors or time-to-digital converters, that are not feasible on FPGAs with an accaptable phase accuracy. A sophisticated way in detecting the phase is provided by the coordinate rotation digital computer (CORDIC) algorithm which was presented by Volder in 1959 and was enhanced by Walther in 1971 [4]. An existing VHDL implementation of the algorithm from Guntoro [5] could be readily used. The CORDIC algorithm implies that the input is a (complex) analytic signal, which consists of in-phase (real part) and quadrature component (imaginary part). The connection between in-phase (I) and quadrature component (Q) is given by the Hilbert transform which will be described in the next section.

Different methods exist for generating the analytic signal from a real valued signal. They are described and compared in Section 3. The implementation of a Hilbert transform with a single real FIR filter is described in Section 4. Eventhough modern FPGAs include hardware multipliers, these are still a limited resource in affordable devices. A novel method for designing complex multiplier-less frequency sampling filters as Hilbert transformers is presented in Section 5.

2. PHASE DETECTION WITH ANALYTIC SIGNALS

A high resolution phase detector can be realized by using the concept of the analytic signal. An analytic signal can formally be generated from a real signal by using the Hilbert transform. The discrete Hilbert transform can be formulated as a convolution with a frequency response of

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

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

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

An 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

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

In the following we use the term Hilbert filter for H_H and analytic filter for $\underline{H}_A$ to distinguish between them.

The analytic signal representation of a real sinusoidal waveform

$$x(n) = a(n)\cos(\Omega n + \phi_0) \tag{5}$$

results in

$$\underline{x}(n) = a(n)e^{j(\Omega n+\phi_0)} \ . \tag{6}$$

The phase information $\varphi(n) = \Omega n + \phi_0$ can be obtained with the four-quadrant arc tangent, which is realized by the CORDIC algorithm:

$$\varphi(t) = \arctan\frac{Im\{\underline{x}(n)\}}{Re\{\underline{x}(n)\}} \tag{7}$$

3. ARCHITECTURES FOR ANALYTIC SIGNAL GENERATION

The structure of an ADPLL is shown in Figure 1. The phase is detected using the methods of last section. The phase difference will be compensated by the controller (or loop filter) $F(z)$ which tunes the frequency of a direct digital synthesizer (DDS). Further details about PLLs can be found in [6].

The main requirements to the Hilbert transform in relation to PLL applications follow from the phase accuracy of the PLL. Therefore, the resulting phase error should be kept as small as possible. The approximated analytic filter may damp the amplitude of the signal. One must ensure that the resulting signal amplitude is high enough for phase detection, provided that I and Q components are damped equally. A constant group delay is preferable as it can easily be compensated using a constant delay in the feedback loop (dotted line in Figure 1). A non-constant group delay can only be compensated using the same filter and phase detector for the PLL output signal which nearly doubles the effort for the phase detection (dashed line in Figure 1).

Common methods for the analytic signal generation are listed as follows:

A) A complex filter is used to directly cancel all negative frequency components ($-\pi < \Omega < 0$) while passing all positive frequency components ($0 < \Omega < \pi$).

B) Two filters are used to form a 90° phase splitting network producing a phase difference of 90° between the outputs.

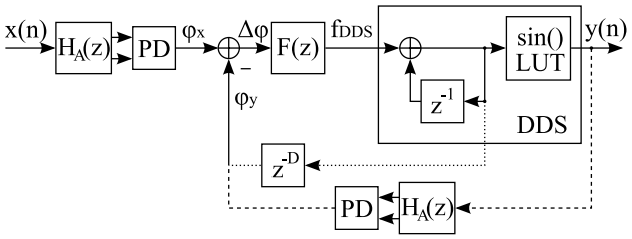


Fig. 1. Structure of an ADPLL with DDS, analytic Filter $H_A(z)$, arc tangent phase detector (PD) and loop filter $F(z)$

C) An FIR filter is used for the approximation of the Hilbert transform. The real part must be delayed by half of the order of the FIR Hilbert filter.

Method A is realized by an FIR or IIR approximation of the frequency response of (4). The coefficients are complex since the frequency response is asymmetric with respect to the imaginary axis. The hardware effort for complex multipliers is in principle much higher than for real multipliers. For real input signals, a complex filter can be divided into two filters, one for the real output and one for the imaginary output

$$\underline{H}_A(z) = H_i(z) + jH_q(z) \ , \tag{8}$$

resulting in method B. In the FIR case, all real and imaginary coefficients are equal to the coefficients of $H_i(z)$ and $H_q(z)$ respectively.

Method B is often implemented using all-pass IIR filters that usually result in a good approximation of the phase and amplitude related to the implementation effort. Nevertheless, the total group delay is not constant due to the nature of IIR filters. Since variations in group delay cause phase errors that can hardly be corrected by the PLL, the usage of this method is very limited for PLL applications. Rader and Jackson [7] proposed constant group delay IIR filters using the time reversal technique for Hilbert transforms. To use the time reversal operation, block processing of the input signal is unavoidable. This introduces unacceptable high delays for our application.

In method C, an FIR filter is used as an approximation to the Hilbert filter, e.g., by cutting and windowing the impulse response (2). The resulting filter $\widetilde{H}_H(z)$ has the property that the phase-approximation to $\pm 90°$ is perfect while the approximation of the amplitude depends on the order of the filter. The delay of the FIR Hilbert filter is $N/2$ clock cycles, where N is the order of the filter. Therefore, the real part of the signal must be delayed by an integer number of clock cycles for even N (or odd amount of filter taps) resulting in

$$\underline{H}_A(z) = z^{-N/2} + j\widetilde{H}_H(z) \ . \tag{9}$$

An implementation of this method is described in the next section. A proposal for the efficient implementation of method A is given in Section 5.

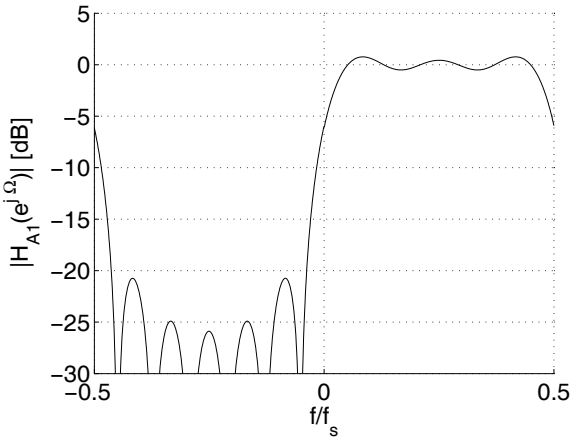


Fig. 2. Frequency response of the FIR analytic filter $\underline{H}_{A1}(z)$

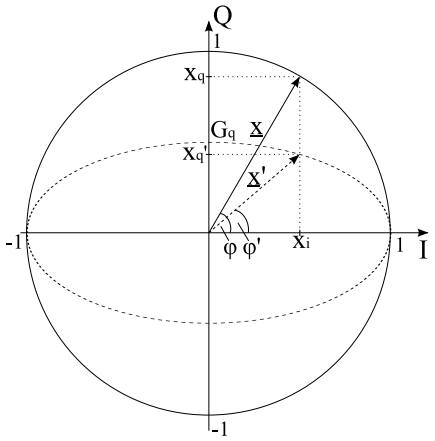


Fig. 3. Phase error due to amplitude errors

4. IMPLEMENTATION OF THE FIR ANALYTIC FILTER

The implementation consists of two sub-components, the FIR Hilbert filter for the generation of the imaginary part, and a delay that produces the delay of the Hilbert filter for the real part. A symmetric frequency range around $f_s/2$ for the Hilbert filter leads to an impulse response with an odd symmetry ($b_k = -b_{-k}$) and furthermore, the even coefficients become zero — both very useful for an efficient implementation.

The computation of the coefficients was optimized to the desired frequency range with the least square method [8] using the Matlab filter design toolbox (`firls()` - function). The design parameters were set to a frequency range from $0.005 f_s$ to $0.995 f_s$ ($f_s = 120$ MHz), a filter order of N=10, and a coefficient word size of 8 bit. The resulting filter has the causal representation of

$$\widetilde{H}_{H1}(z) = z^{-N/2} \sum_{k=-N/2}^{N/2} b_k z^{-k} \qquad (10)$$

with the coefficients b_1 =163/256, b_3 =54/256 and b_5 =32/256. The analytic filter is formed using (9), the corresponding frequency response is labeled $\underline{H}_{A1}(e^{j\Omega})$ and is shown in Figure 2.

4.1. Inspection of the Dynamic Phase Error

Even when the phase error of the FIR Hilbert filter is zero, the amplitude error leads to a dynamic phase error of the total analytic filter. This effect is illustrated in the complex plane in Figure 3. The imaginary part (Q axis) is attenuated at some frequencies due to the FIR Hilbert filter while the real part (I axis) has no attenuation at all as it is simply delayed. The output vector for a real sinusoidal input signal (5) with $a(n) = 1$ rotates on the unit circle for an ideal analytic filter. This path will be distorted to an elliptical curve when the gain ratio between imaginary part and real part is not equal to one ($G_e = G_q/G_i \neq 1$). The resulting vector is labeled

$$\underline{x}' = x_i + j x_q' \ , \qquad (11)$$

and its output phase in relation to the input phase and the gain error is

$$\varphi'(\varphi, G_e) = arg\ \underline{x}' = \arctan\left(G_e \tan\varphi\right) \ . \qquad (12)$$

The phase detector outputs for an IQ gain mismatch of 0 dB ($G_e = 1$, no mismatch), -3 dB ($G_e = 0.707$) and -20 dB ($G_e = 0.1$) are shown in Figure 4(a) for two periods of the input phase. The output phase can be separated into the undisturbed phase plus an additive phase error $\Delta\varphi = \varphi' - \varphi$ which is shown separately in Figure 4(b).

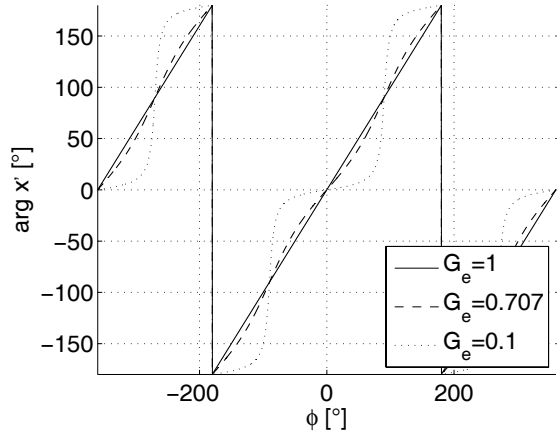
The phase error $\Delta\varphi$ has a zero mean value in time domain and its fundamental frequency component is twice the input frequency. It can be approximated by a shifted sine function for low IQ gain mismatch ($G_e \approx 1$) and will end up in a rectangular waveform for $G_e \to 0$. The maximum of $\Delta\varphi(\varphi)$ can be determined analytically to

$$\Delta\varphi_{max}(G_e) = \frac{\pi}{2} - 2\arcsin\sqrt{\frac{G_e}{1+G_e}} \ . \qquad (13)$$

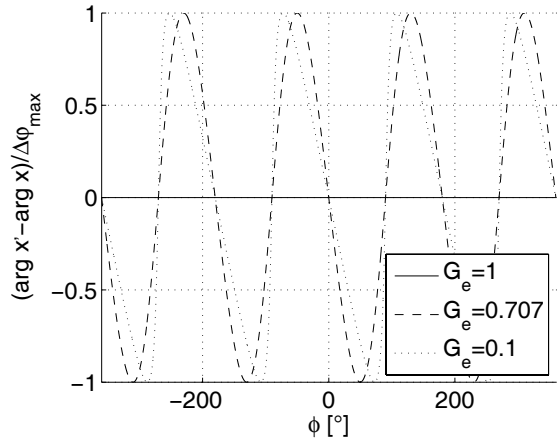
For small gain errors, this can be approximated by

$$\Delta\varphi_{max}(G_e) \approx -1/2(G_e - 1)\ [\text{rad}] \ . \qquad (14)$$

The phase error for a low IQ gain mismatch can be canceled by the PLL loop filter since this has usually an integrating or low-pass behavior with a cut-off frequency much lower than the signal frequency. Care must be taken in the choice of the loop filter to attenuate high frequencies sufficiently. The high frequency gain of a PI controller, which is a very common loop filter for second order (type II) PLLs, is approximately equal to the proportional gain, which is of course



(a)



(b)

Fig. 4. Detected phase for a sinusoidal signal (a) output phase $\varphi'(\varphi, G_e)$, (b) phase difference $\Delta\varphi(\varphi, G_e)$ normalized to $\Delta\varphi_{max}(G_e)$

greater than zero. The use of an additional low-pass filter is proposed so that frequency components far outside the loop filter bandwidth are attenuated and will not disturb the regular behavior of the PLL. The requirement for the former stages of a constant group delay is not necessary anymore since the phase difference is not dependent on the absolute delay. Therefore, an IIR filter can be used.

4.2. Implementation of a Multiplier-Less IIR Low-Pass Filter

The proposed low-pass filter is based on the discretization of a simple first order analog filter

$$F_{LP}(s) = \frac{1}{1 + sT_c} \tag{15}$$

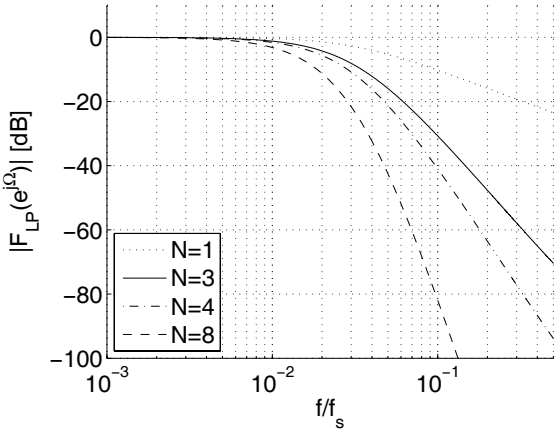


Fig. 5. Frequency response of the multiplier-less IIR low-pass filter $F_{LP}(z)$

with a 3-dB cut-off frequency of $f_c = 1/T_c$. A simple discrete approximation can be obtained by applying the rectangle method (also named backward difference [8]) substituting

$$s \leftarrow \frac{1}{T_s}(1 - z^{-1}) \ . \tag{16}$$

This results in the discrete transfer function

$$F_{LP}(z) = \frac{b_0}{1 + a_1 z^{-1}} \tag{17}$$

with $b_0 = 1/(1 + T_c/T_s)$ and $a_1 = -1/(1 + T_s/T_c)$.

The specification of the low-pass filter is relaxed as we need a cut-off frequency that lies in the band between the loop bandwidth (which is typically a few kHz up to some 100 kHz) and twice the minimum signal frequency. Hence, there is a degree of freedom in choosing the coefficients.

The multiplication with a power of two can be efficiently implemented using a simple shift operation. The approach $b_0 \stackrel{!}{=} 2^{-M}$ leads to $a_1 = 2^{-M} - 1$, which means that only one additional subtracter is needed to realize the filter. With this restriction, the 3-dB cut-off frequency is

$$f_c = \frac{fs}{2^M - 1} \ , \tag{18}$$

leaving enough room for different applications. A cascade of filters can be used to increase the stop-band attenuation. The different frequency responses for different filter orders N (which is equal to the number of cascaded first order filters) are shown in Figure 5 for $M = 4$ or $f_c = f_s/15$.

5. MULTIPLIER-LESS COMPLEX FREQUENCY SAMPLING FILTER AS HILBERT TRANSFORMER

A special class of multiplier-less frequency sampling filters (FSF) was described in [4] as a solution for band-pass dec-

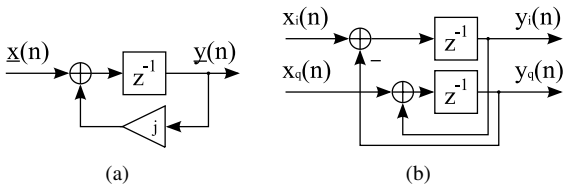


Fig. 6. Implementation of $C_{+90}(z)$ (a) formal structure with complex coefficient (b) structure of implementation

imators in multirate filters. These FSFs can be seen as generalization of the popular Combined Integrated Comb filters (CIC). Like the CIC, the FSF consists of a comb filter

$$F_{zN}(z) = 1 - z^{-N} \tag{19}$$

that produces equidistant zeros on the unit circle at multiple angles of $\Omega = 360°/N$. In contrast to the CIC, the FSF uses multiple types of IIR filters producing poles on the unit circle. The poles are chosen to cancel the zeros of the comb filter at desired frequencies. This cancellation can be realized exactly in two's complement, due to the precise coefficient representation. The IIR filters have the form

$$C_\Omega(z) = \frac{z^{-N}}{1 + a_1 z^{-1} + ... + a_N z^{-N}} \ . \tag{20}$$

For real coefficients, the angles of the poles on the z-plane are always symmetrical with respect to the real axis. Extending the range to complex coefficients allows a distinction for positive or negative frequencies. A set of possible IIR filter coefficients with the resulting pole locations is listed in Table 1. The upper part of the table consists of useful real coefficients which were mostly taken from [4]. Some useful complex coefficients are listed in the lower part of the table. A multiplication of a complex number by j can be effectively implemented by negating the imaginary part and swapping real and imaginary part afterwards ($(a+jb)\cdot j = -b+ja$). This principle is shown as example for filter $C_{+90}(z)$ in Figure 6.

Although the resulting filter structure is recursive, the final filter will have a finite impulse response. A linear phase and thereby a constant group delay can be achieved easily. The CIC filter, e.g., uses only $C_0(z)$ as an integrator, resulting in a low-pass filter. Real high-pass, band-pass or band-stop filters can be designed using poles from the upper half of the table. Complex filters like the analytic filter can be designed by adding additional complex IIR filters. The design procedure will be explained by the design of different analytic filters in the following.

An analytic filter has to cancel all negative frequency components, or all components between $-180°$ and $0°$. The comb filter $F_{z4}(z)$ produces zeros at e^{j0}, $e^{j90°}$, $e^{j180°}$ and $e^{-j90°}$. The zero at $+90°$ can be canceled by adding

$C_\Omega(z)$	a_1	a_2	a_3	Ω_1	Ω_2
$C_0(z)$	-1			$0°$	
$C_{180}(z)$	1			$180°$	
$C_{\pm60}(z)$	-1	1		$\pm60°$	
$C_{\pm90}(z)$	0	1		$\pm90°$	
$C_{\pm120}(z)$	1	1		$\pm120°$	
$C_{0/180}(z)$	0	-1		$0°/180°$	
$C_{+90}(z)$	$-j$			$+90°$	
$C_{-90}(z)$	j			$-90°$	
$C_{+30/+150}(z)$	$-j$	-1		$+30°$	$+150°$
$C_{-30/-150}(z)$	j	-1		$-30°$	$-150°$
$C_{0/+90/180}(z)$	$-j$	-1	j	$0°/180°$	$+90°$

Table 1. Coefficients of (20) with the corresponding angles of the pole location(s)

$C_{+90}(z)$ in series. Scaling the result to 0 dB, leads to

$$\underline{H}_{A2}(z) = \frac{1}{4}(F_{z4}(z)C_{+90}(z)) \ . \tag{21}$$

The result is a narrow-band analytic filter with a pass-band centered at $\Omega = 90°$, as shown in Figure 7. Better stop-band attenuation can be achieved by using several filters in series with the drawback of narrowing the pass-band. A wider pass-band can be achieved by canceling the zeros at $\Omega = 0°$ and $\Omega = 180°$ by multiplying the filter with $C_{0/180}(z)$. Further improvements can be achieved by using the inverse of $C_{-30/-150}(z)$ which adds additional zeros at $\Omega = -30°$ and $\Omega = -150°$. Using two filters in series (squaring the transfer function) and scaling the result to 0 dB results in

$$\underline{H}_{A3}(z) = \frac{4}{9}\left(\underline{H}_{A2}(z)C_{0/180}(z)C^{-1}_{-30/-150}(z)\right)^2 \tag{22}$$

whose frequency response is also shown in Figure 7.

Without the requirement for a constant group delay, the magnitude response can be further optimized using an additional IIR filter with poles inside the unit circle. The filter

$$C_P(z) = \frac{1}{(z - z_p)(z + z_p)} = \frac{z^{-2}}{1 - z_p^2 z^{-2}} \tag{23}$$

has two poles at $\pm z_p$. Extending $\underline{H}_{A3}(z)$ with two of these filters results in

$$\underline{H}_{A4}(z) = \frac{9}{4}\,\underline{H}_{A3}(z)C_P^2(z) \tag{24}$$

which has a flat (and equiripple) pass-band behavior for $z_p = 1/\sqrt{2}$, as shown in Figure 7. The coefficient $z_p^2 = 1/2$ can be realized by a simple bit shift.

Filter	f_c/f_s	G_{stop} [dB]	ΔG_{pass} [dB]	$G_{e,max}$ [dB]	$\varphi_{e,max}$ [°]	N_A	N_R	N_{LE}	f_{max} [MHz]
$F_{LP}(z)$	0...0.004	18	-	-	-	6	6	153	142
$\underline{H}_{A1}(z)$	0.018...0.0483	20.7	-0.51...0.76	-0.51...0.76	0	10	6	337	266
$\underline{H}_{A2}(z)$	0.137...0.364	11.3	3	0	± 40	3	4	110	275
$\underline{H}_{A3}(z)$	0.154...0.346	47.7	3	0...0.103	0	16	18	562	191
$\underline{H}_{A4}(z)$	0.0145...0.4865	44.6	-0.005...0	-0.1035...0	$\pm 4 \cdot 10^{-3}$	18	38	679	193

Table 2. Performance of the implemented filters. Used symbols: f_c: 3-dB cut-off frequency, G_{stop}: stopband attenuation, ΔG_{pass}: passband ripple, $G_{e,max}/\varphi_{e,max}$: maximum IQ gain/phase mismatch, N_A, N_R and N_{LE}: required number of adders/subtracters (incl. output register), registers (incl. pipeline registers) and logic elements, f_{max}: maximum speed. The constant coefficient multipliers of $\underline{H}_{A1}(z)$ were represented by adders using the canonic signed digit system [4].

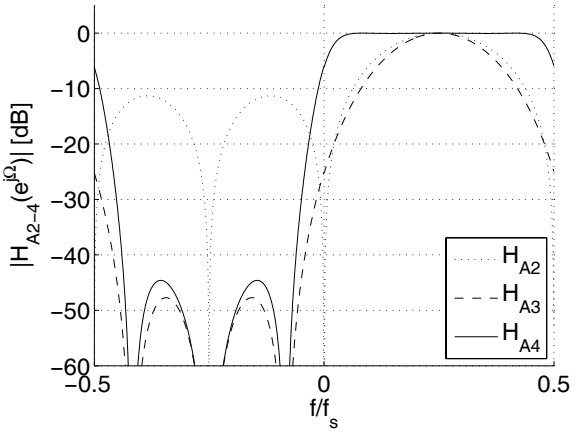


Fig. 7. Frequency response of the complex frequency sampling filters $\underline{H}_{A2-4}(z)$

6. RESULTS

The parameters of the developed filters are summarized in Table 2 together with the results from the synthesis. A standard IIR filter design with the specifications from Table 2 for $\underline{H}_{A4}(z)$ needs a minimum order of 10 resulting in 19 coefficients being either purely real or imaginary. In total 38 real multipliers, 36 real adders and 18 registers are needed, which is much more effort than the suggested $\underline{H}_{A4}(z)$.

The implementation of a PLL with $\underline{H}_{A1}(z)$ was tested on an available FPGA board equipped with an Altera Cyclone FPGA (EP1C6Q240C8). In addition to this FPGA board, an ADC/DAC board [9] was used for connecting analog signals. This board has two ADC (LTC2254, max. $f_s = 105$ MSPS, 14 bit) and two DAC channels (AD9744, max. $f_s = 210$ MSPS, 14 bit). Since the FPGA has only 5980 Logic Elements (LE) in total, filter $\underline{H}_{A1}(z)$ was used in the PLL prototype. The low-pass filter $F_{LP}(z)$ was used to suppress the remaining dynamic phase error. A better PLL performance could be expected though using twice the filter $\underline{H}_{A4}(z)$ according to the dashed line in Figure 1. This is planned for the final PLL design on a new hardware.

7. REFERENCES

[1] M. Kumm, "FPGA-Realisierung eines Offset-Lokaloszillators basierend auf PLL- und DDS-Technologien (in German)," Diploma Thesis, Darmstadt University of Technology, Juli 2007.

[2] H. Klingbeil, "A fast DSP-based phase-detector for closed-loop RF control in synchrotrons," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 3, pp. 1209–1213, 2005.

[3] H. Klingbeil, B. Zipfel, M. Kumm, and P. Moritz, "A digital beam-phase control system for heavy-ion synchrotrons," *IEEE Trans. Nucl. Sci.*, vol. 54, no. 6, pp. 2604–2610, 2007.

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

[5] A. Guntoro *et al.*, "Reconfigurable computing systems for digital high-frequency control of heavy ion accelerators - extending the phase detector module: Magnitude and phase information," Technical University Darmstadt, Tech. Rep., 2006.

[6] F. M. Gardner, *Phaselock Techniques, third edition*. New York, Chinchester, Brisbane, Toronto: John Wiley & Sons, Inc., 2005.

[7] C. M. Rader and L. B. Jackson, "Approximating noncausal IIR digital filters having arbitrary poles, including new Hilbert transformer designs, via forward/backward block recursion," *IEEE Trans. Circuits Syst. I*, vol. 53, pp. 2779–2787, 2006.

[8] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, New Jersey, United States: Prentice Hall Inc., 1975.

[9] M. S. Sanjari, "Hardware and software implementation of a radio frequency high-speed data conversion unit for digital control systems," Bachelor's Thesis, Darmstadt University of Technology, October 2006.