International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 23 (2018) pp. 16333-16341

An FPGA-BASED implementation of a Hilbert filter for Real-time Estimation of Instantaneous Frequency, Phase and Amplitude of Power System Signals

Q. Bart and R. Tzoneva

Department of Electrical, Electronic and Computer Engineering,
Cape Peninsula University of Technology, Cape town, 7535, South Africa.

Abstract

The instantaneous parameters, amplitude, phase and frequency of power system signals provide valuable information about the status of the power system, particularly when these parameters can be obtained simultaneously from locations that are remotely located from each other. Phasor Measurement Units (PMU's) placed at these remote locations can perform these simultaneous time synchronized measurements. In this work digital signal processing techniques and Field Programmable Gate Array (FPGA) technology was used for the real-time estimation of power system signal parameters. These parameters were estimated based upon an approach utilizing the Hilbert transform and formation of an analytic signal as a basis for the estimation of instantaneous phase, frequency and amplitude of power system signals. However, the emphasis in this work is upon the accurate estimation of instantaneous phase and frequency because of the level of complexity to accurately achieve this practically. Three different methods for frequency estimation were implemented and compared, based upon the criterion of accurate frequency measurement, to assess the efficacy of the three techniques. The MATLAB/SIMULINK simulation environment together with Xilinx System Generator, a high level digital signal processing development tool for FPGA's, was used to practically implement the estimation techniques.

Keywords: FPGA, Hilbert Transform, Instantaneous phase, Instantaneous frequency, Instantaneous Amplitude

INTRODUCTION

In power systems the phasor has always been the formal mathematical description of voltage and current signals. In 1988 Arun G. Phadke and James S. Thorp developed the first phasor measurement unit (PMU), which used an accurate time source and the global positioning system, to synchronise these phasor measurements. These real-time phasor measurements were time-tagged to UTC time and as multiple phasor measurements were synchronised they were known as synchrophasors. Since the advent of synchrophasors and PMU technology many algorithms have been proposed. The literature contains a plethora of algorithmic techniques, with the discrete fourier transform (DFT) and derivatives [1],[2],[3] thereof being most popular. Complications and erroneous results however will be obtained from DFT estimators when the power system frequency changes because the DFT requires the power system frequency as apriori knowledge for a particular fixed sampling rate. The DFT is periodic in nature and assumes that the window of sampled data contains an integer number of waveform cycles. During frequency perturbations the sampled data may not comprise an integer number of cycles resulting in discontinuities at the endpoints. This is one of the disadvantages of the DFT for real-time estimation of synchrophasors and this phenomenon is commonly known as spectral leakage. In order to accommodate the dynamic nature of power system signals an extension of the DFT known as the Taylor-Fourier Transform [4],[5] has also been utilized as a synchrophasor estimator. Alternative techniques such as Kalman Filtering [6] and Enhanced Phase Locked Loop techniques [7] have also been proposed. In this paper although signal amplitude, phase and frequency can be estimated using the proposed technique, depicted in Figure 4, the focus is upon the accurate estimation of phase and frequency as the process to accomplish this is more complex than amplitude estimation. The forthcoming sections in this paper are organized as follows. First a brief overview of the synchrophasor reference algorithm and the concept upon which it is based is discussed. Then an alternative approach based upon the Hilbert transform and the analytic signal is presented. The concept of instantaneous phase and frequency follows next and then a technique for phase and frequency estimation is described as depicted in Figure 4. After this the practical implementation using Xilinx System Generator is discussed and implementation results presented. To put the work done in context the next section describes the theoretical concept of demodulation as it applies to synchrophasor estimation based upon the synchrophasor reference model [8].

SYNCHROPHASOR REFERENCE MODEL

The block diagram in Figure 1 depicts the reference signal processing model found in the synchrophasor standard [8].

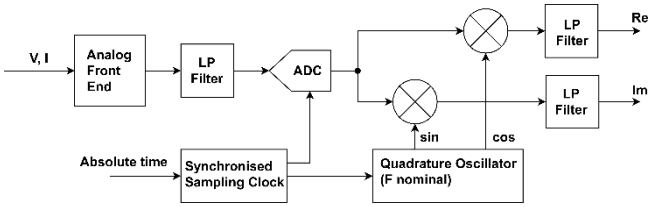

Figure 1: Single phase section of the PMU phasor signal processing model

The standard does not dictate the use of this reference design however it may be used as a technique for estimation. This is a demodulation based technique with the assumption that power system signals, voltage and current, are modelled by a band pass signal

$$x(t) = A(t)\cos(\omega t + \varphi(t)) \quad (1)$$

Where $A(t)$ is the amplitude, $\varphi(t)$ the phase and $\omega = 2\pi f_0$ with f_0 being the frequency in cycles per second. This sinusoidal waveform is commonly expressed as a phasor

$$X(t) = A(t)e^{j\varphi(t)} \quad (2)$$

In another way it can be expressed trigonometrically in polar form as

$$X(t) = |X(t)|[cos\varphi(t) + jsin\varphi(t)] \quad (3)$$

Equation (3) is composed of a real and an imaginary quantity where $|X(t)|cos\varphi(t)$ and $|X(t)|jsin\varphi(t)$ represents the real and imaginary components respectively.

These components are essentially the outputs that are obtained from the reference signal processing model. This representation is particularly useful as the parameters of instantaneous amplitude and phase can be obtained as

$$A(t) = \sqrt{(|X(t)|cos\varphi(t))^2 + (|X(t)|jsin\varphi(t))^2} \quad (4)$$

where $A(t)$, is the waveform amplitude. For a signal that has an amplitude that is changing as a result of disturbances within the power system, the amplitude also traces the envelope of the signal. In a similar manner the phase, $\varphi(t)$, of the signal can be given as

$$\varphi(t) = \tan^{-1}\frac{sin\varphi(t)}{cos\varphi(t)} \quad (5)$$

Phasor estimation based upon the Hilbert Transform

The Hilbert transform is defined by

$$\hat{x}(t) = H[x(t)] = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}dt \quad (6)$$

Although this process performs a transformation on the input signal, $x(t)$, the derived Hilbert signal, $\hat{x}(t)$, remains in the time domain. The Hilbert transform on a signal, $x(t)$, will produce a signal that is shifted by $\frac{\pi}{2}$ radians to yield $\hat{x}(t)$. For positive frequencies the shift will be $-\frac{\pi}{2}$ and for negative frequencies, $+\frac{\pi}{2}$. The analytic signal can then be formed by a summation in quadrature of the original signal, $x(t)$, and its Hilbert transform, $\hat{x}(t)$, to form a complex valued function, which is the analytic signal as indicated in Equation (7).

$$x_a(t) = x(t) + j\hat{x}(t) \quad (7)$$

This analytic signal has the same form as Equation (3) and hence the amplitude, $A(t)$ and phase, $\varphi(t)$ can be derived in a similar way to Equation (4) and (5) respectively by using the signal, $x(t)$ and its Hilbert transform signal, $\hat{x}(t)$, to compute the amplitude and phase as

$$A(t) = \sqrt{x^2(t) + \hat{x}^2(t)} \quad (8)$$

$$\varphi(t) = \tan^{-1}\frac{\hat{x}^2(t)}{x^2(t)} \quad (9)$$

FIR Hilbert Transformer

The practical implementation of the Hilbert Transformer is based upon a finite impulse response (FIR) filter design. From a practical perspective the Hilbert transform is nothing more than a filter that can provide a $\frac{\pi}{2}$ phase shifted signal as an output. According to [9] the ideal Hilbert transform impulse response, $h(n)$ is

$$h(n) = \begin{cases} \frac{2}{n\pi}sin^2\left(\frac{n\pi}{2}\right), & n \neq 0 \\ 0, & n = 0 \end{cases} \quad (10)$$

Where n represents the nth sample for

$n = 0\ to\ N-1$

and N represents the window of samples

From Equation (10) it can be seen that for positive and negative values of n the impulse response is anti-symmetric. This implies that a FIR realization of a Hilbert transformer can be accomplished with filters that have either Type III or Type IV symmetry.

The filter length is of prime importance in any FIR filter design. This filter parameter affects the steepness of the transition band as well as the ability to suppress unwanted frequencies in the stop band. Longer filter lengths provide better frequency response and reduced ripple in the pass band. However, the delay introduced by the filter increases proportionally with the filter length. For even filter length M, the delay is $\frac{M}{2}$ samples and for odd filter length M, the delay is $\frac{M-1}{2}$ samples.

In this work filters are implemented on a FPGA platform. A necessary requirement is to minimize the usage of the logic elements on the FPGA fabric. The advantage of this would be a reduction in the processing time as well as a reduction in the power consumption. Based upon the implementation requirements and a desire to have as close as possible an ideal filter response, two methods for the derivation of the

Hilbert transformer filter coefficients were employed. This is the 'optimal' method for designing filter coefficients called the Parks-Mclellan algorithm [10] and the second method is based upon the derivation of coefficients using half-band filters. To compare the two aforementioned techniques in terms of delay, analytic signal generation and FPGA resource utilization the same filter orders were used in each technique to derive the filter coefficients. Table 1 shows the results obtained when using the two techniques for deriving the filter coefficients. From Table 1 it can be seen that although the amplitude response of the Hilbert filters are similar the Hilbert filter derived from the half-band filter technique provides a more accurate response in the pass band for frequencies that span a range from $40Hz\ to\ 60Hz$, where the input signal has a normalized amplitude input of 1 unit. Also for the same filter order, N, the half band filter technique results in a more reduced ripple, δ, in the pass band than the Parks-Mclellan technique. Higher order implementations, not shown here, has shown that the 0dB gain is maintained. It was also determined that in order to achieve the same accuracy using the Parks-Mclennan technique much higher order filters are required. This would imply larger FPGA resource utilization and hence larger throughput delays. Hence for optimal implementation the half-band Hilbert filter was used because utilizing the half-band filter affords more accuracy in terms of signal amplitude magnitude for the range of frequencies from $40Hz\ to\ 60Hz$. Also with reduced ripple, δ, in the pass band a more desirable response is obtained. Figure shows the frequency response of the designed Hilbert transform filter, where the order, N, of the filter is 150 and ripple, δ, is 0.0023. From Table 1 this provides the best response required.

Table 1. Tabulated results comparing Hilbert filter magnitude responses of the two selected techniques

Parks-Mclellan	**N=50 / δ = 0.1830**			**N=100 / δ = 0.0471**			**N=150 / δ = 0.0116**		
Input Frequency	**40Hz**	**50Hz**	**60Hz**	**40Hz**	**50Hz**	**60Hz**	**40Hz**	**50Hz**	**60Hz**
Output Amplitude	± 0.9917	±1.1815	±1.1590	± 1.0437	±0.9803	±0.9537	± 1.0437	±0.9803	±0.9537
Half-Band technique	**N=50 / δ = 0.1126**			**N=100 / δ = 0.0127**			**N=150 / δ = 0.0023**		
Input Frequency	**40Hz**	**50Hz**	**60Hz**	**40Hz**	**50Hz**	**60Hz**	**40Hz**	**50Hz**	**60Hz**
Output Amplitude	± 0.9932	±1.0802	±1.1000	± 1.0002	±0.9980	±0.9980	± 1.0000	±1.0000	±1.0000

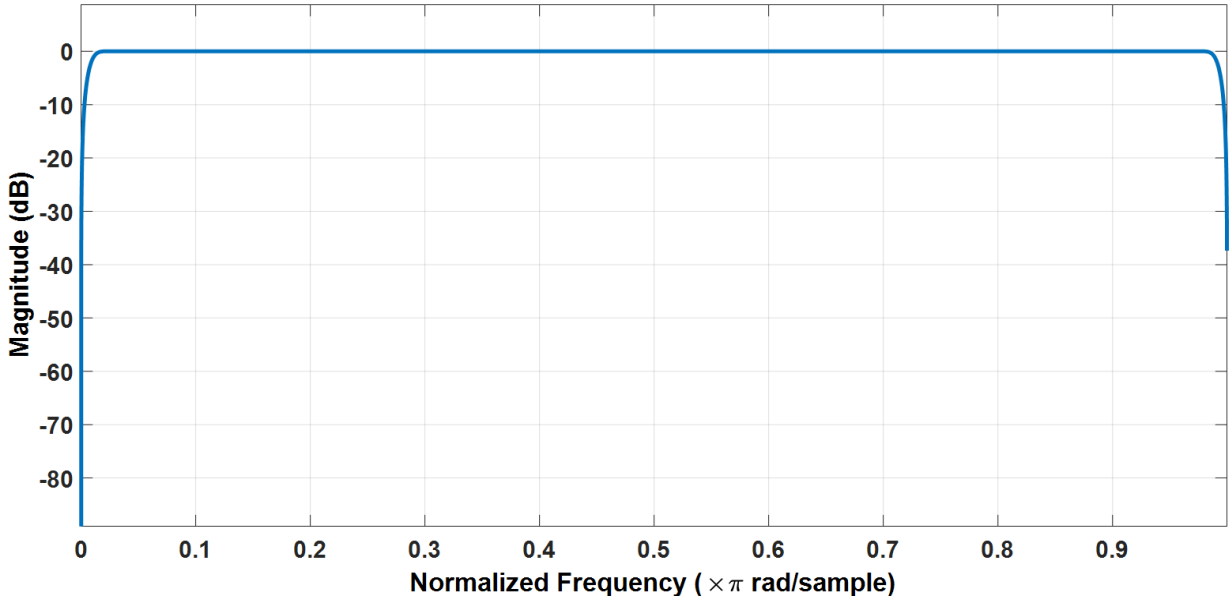

Figure 2: Frequency response of designed Hilbert transform filter

The concept of instantaneous phase and instantaneous frequency

When synchrophasors are estimated knowledge of the instantaneous frequency (IF) is very important. As an example, having the IF as apriori information can improve the accuracy of a fast fourier transform (FFT) reducing the erroneous effects of leakage. The concept of IF has been well researched and many researchers [11],[12] ,[13] have contributed towards the theory and techniques for estimating instantaneous frequencies. Coupled with IF is instantaneous phase, as IF is essentially derived from the derivative of instantaneous phase, where $\varphi_i(t)$ is the instantaneous phase and instantaneous frequency, f_{0_i} is

$$f_{0_i} = \frac{1}{2\pi}\frac{d\varphi_i(t)}{dt} \tag{11}$$

This definition of IF was first derived by Van der Pol in 1946 [14] based upon a real signal and has been adopted by the synchrophasor standard [8] as the definition for the measurement of the instantaneous frequency. Two years later in 1948, Ville [15] related the instantaneous frequency to the analytic signal, which is a complex valued function and has the same form as Equation (11)

$$f_0 = \frac{1}{2\pi}\frac{d}{dt}[\arg(x_a(t)] \tag{12}$$

Where

$x_a(t)$ is the analytic signal.

Software simulations to estimate phase and frequency can be easily achieved, however hardware implementation to perform estimations based upon calculations such as indicted in Equations (5), (9) for phase and (11), (12) for frequency may not be a straightforward implementation. In particular, when utilizing embedded platforms such as FPGA's, consideration should be given to the fact that digitally based systems manipulate discrete data that may be affected by quantization and sampling. Consideration should also be given to how mathematical operations will be performed, especially with regards to complex formulae that require differentiation and integration. In the next section techniques for estimating phase and frequency will be discussed which will lead to the process and procedure for practical implementation.

Technique for phase and frequency estimation

In this work three different approaches for determining the frequency has been adopted. The first is based upon the simple technique of zero crossing and the second and third technique is dependent upon the instantaneous phase. As a start the phase dependent techniques are approached first. From the previous section Equations (11) and (12) are clearly dependent upon finding the derivative of the phase value which can be easily implemented in a software simulation environment or coded in a numerical computing language such as Matlab, Octave or Scilab. It becomes a bit more challenging when some form of practical realization on a digitally based hardware platform such as an FPGA is required. As mathematical differentiation is key to obtaining the instantaneous phase and frequency, effective methods and techniques are required to accurately achieve this practically. This can be accomplished by numerically calculating the derivative of the phase which is dependent upon differentiating the signal,$x(n)$ and its Hilbert counterpart, $\hat{x}(n)$.

$$\dot{\varphi}(n) = \frac{x(n).\dot{\hat{x}}(n) - \dot{x}(n).\hat{x}(n)}{x^2(n) + \hat{x}^2(n)} \tag{13}$$

Where $\dot{\varphi}(n), \dot{x}(n)$ *and* $\dot{\hat{x}}(n)$ are the differentiated versions of phase, discrete input signal and its Hilbert signal respectively.

Then by multiplying Equation (13) by $\frac{1}{2\pi}$ the instantaneous frequency can be calculated as

$$f_i(n) = \frac{1}{2\pi}\frac{x(n).\dot{\hat{x}}(n) - \dot{x}(n).\hat{x}(n)}{x^2(n) + \hat{x}^2(n)} \tag{14}$$

Equation (14) forms the basis for the design of the signal parameter estimator which is shown in Figure 4. Essentially, with the exception of the blocks labelled 'Phase and Frequency Estimator' and 'Zero Crossing Detector' the block diagram shows the model implementation of Equation (14). This represents the first approach towards frequency estimation.

The second frequency estimation method is also related to the phase. This is accomplished on the basis of utilizing forward or backward differencing equations where the following discrete time representations can be used. The forward differencing equation is

$$f_i(n) = \frac{1}{2\pi}[\varphi(n+1) - \varphi(n)] \tag{15}$$

And backward differencing equation is

$$f_i(n) = \frac{1}{2\pi}[\varphi(n) - \varphi(n-1)] \tag{16}$$

Where n represents the nth sample for

$n = 0\ to\ N-1$

and N is the window width of samples of phase $\varphi(n)$.

From a digital computation perspective Equations (15) and (16) are more efficient and not prone to the effects of noise as that of Equation (14). However a mechanism to obtain a window of phase samples is required. It is proposed here to use the **CO**ordinate **R**otation **DI**gital **C**omputer (CORDIC) algorithm which was developed by Jack Volder in 1959 [16]. Broadly stated this is an iterative algorithm that can be used to calculate trigonometric functions using only bit shifting and addition. This enables obtaining a window of phase

samples and using the differencing equations, the frequency can be obtained. This technique is represented by the block labelled 'Phase and Frequency Estimator' in Figure 4.

The third proposed frequency estimation technique is based upon simple zero crossing where the signal is monitored to observe its crossing on the zero axis. This method represents a very direct way of frequency estimation. The implementation essentially requires the observation of the change of the sign of the signal and measurement of the time difference between zero crossing points. The time difference can be made between the crossing points for half period or full period of the waveform cycle. Once the value for the full time period is obtained the frequency can be calculated as

$$f_i(n) = \frac{1}{T} \qquad (17)$$

where T represents the period of the waveform. The implementation is very simple and practical and will be explained further in the section dealing with the actual hardware implementation. In Figure 4 this is shown as the block labelled 'Zero Crossing Detector'.

The reason for employing three different frequency estimation methods is to determine which one is more accurate and efficient. The technique based upon Equations (13) and (14) depends upon differentiating the signals $x(n)$ and $\hat{x}(n)$. Generally, differentiators have a tendency to amplify any noise within a signal which can result in inaccuracies in estimation. However, with the practical implementation using FIR filter differentiators great care has be given to ensure that the signal $x(t)$, has been sufficiently pre-filtered to remove any noise from it before it is processed by the analogue to digital converter (ADC) on the front end indicated in Figure 4. The frequency response of the FIR differentiators is shown in Figure 3 and are represented by the 'differentiator 'blocks in Figure 4

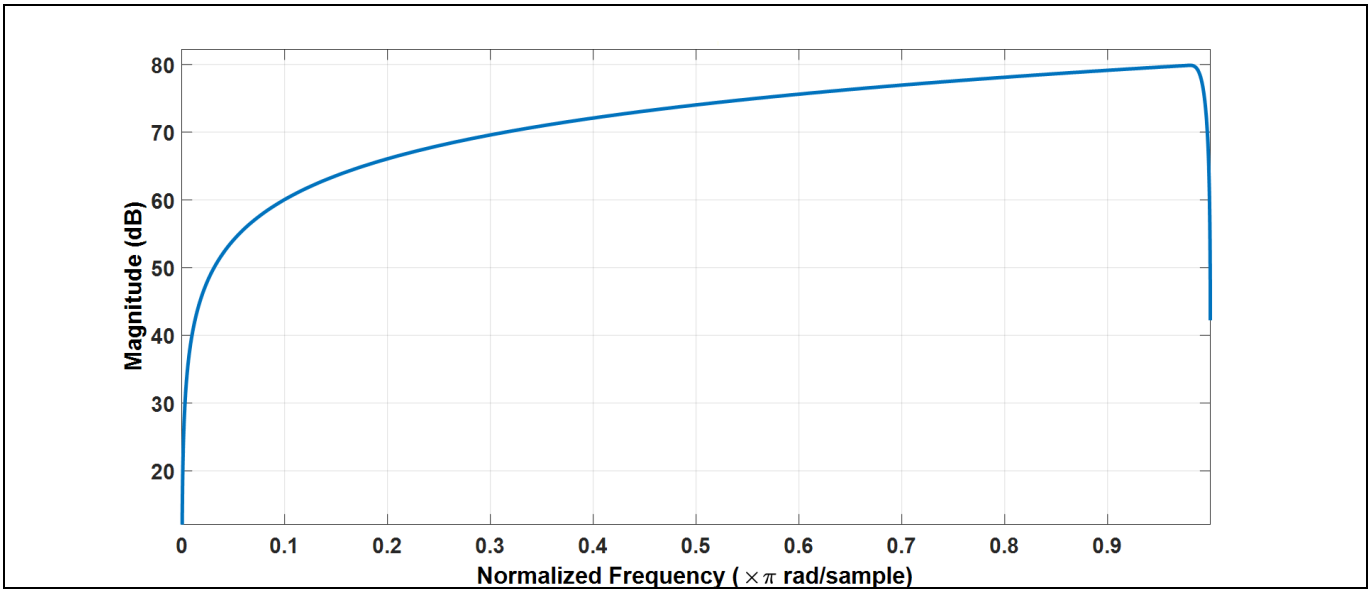

Figure 3: Frequency response of the FIR differentiator

Block diagram description

Figure 4 describes the implementation of Equations (13) and (14), (15) or (16) and Equation (17). As previously mentioned the Hilbert signal as well as the differentiators are implemented as FIR filters. In section 3.1 it was mentioned that filters will impose a delay on any signal that it processes. Therefore, to ensure that unfiltered signals are aligned with filtered signals, delay units should be introduced in the path of unfiltered signals. Note that this is not explicitly shown in Equations (13) and (14) but it is implied that this is the case when practically implemented in a system with digital filters. Input and output signals of certain selected blocks in the diagram are indicated and provides an idea of the signal flow through the system.

Implementation of the System

The practical implementation of all the signal processing elements were actualized on a relatively new type of FPGA based device known as a System-on-Chip (SoC).

More specifically it was implemented on an embedded platform known as the Zedboard which has the Zynq®-7000 System-on-Chip (SoC) on board. This chip contains standard FPGA logic for custom hardware design as well as dual ARM CORTEX A9 processors. The system modeling and design was performed using Xilinx System Generator™ that enables the realization of the signal processing system within the MATLAB/SIMULINK® environment as well as the generation of Very High Speed

Integrated Circuit (VHSIC) hardware description language (HDL) code commonly referred to as VHDL code and Intellectual Property (IP) cores that can be used in a larger FPGA based system. Figure 5 shows the custom designed functional units that implement the design indicated in Figure 4. These units are the Analytic Signal Generator (ASG), Amplitude Estimation Square Root Sum (ASQS),

Discrete Phase Differentiator and Frequency Estimator (DPDF), Signal Differentiator Unit (SDU), Phase Estimator Cordic Algorithm (PCA), Phase Unwrap Difference and Frequency Estimator (PUDF) and the Zero Crossing (ZC) unit. The functionality of each of these units is achieved by using functional units contained in the Xilinx System Generator library. The next sections describe these functional units.

Analytic Signal Generator (ASG)

This block implements the Hilbert transformer as previously described with the generation of the analytic signal. This is composed of a System Generator Filter block with filter coefficients derived from the half-band Hilbert transform design and a delay unit for the input signal. Additional supporting components such as up-sample and down-sample blocks were added to complete the block design.

Signal Differentiator Unit (SDU)

This unit provides the differentiation of the Hilbert signal and the delayed input signal. It consists of System Generator FIR filter blocks with coefficients to enable the differentiation of the input signals. It also provides additional delay with the use of delay blocks to ensure that the signals are aligned upon exiting the differentiator unit.

Amplitude Estimation Square Root Sum (ASQS)

This unit essentially implements Equation (8). It is implemented using multipliers to square the input signals, a summation block and a functional unit to determine the square root of the output of the summation block. This produces the amplitude on the output of this unit. **Discrete**

Phase Differentiator and Frequency Estimator (DPDF)

This functional unit performs the numerical calculation of Equation (13) and calculates the frequency which represents F1 as indicated in Figure 4. This unit is composed of multiplication, summation, gain and a unit to perform division.

Phase Estimator Cordic Algorithm (PCA)

The estimation phase is achieved using the concept of the CORDIC algorithm. It is implemented with a System Generator CORDIC block, delay and gain blocks. The phase output of this unit is wrapped and therefore additional logic was required to unwrap the phase.

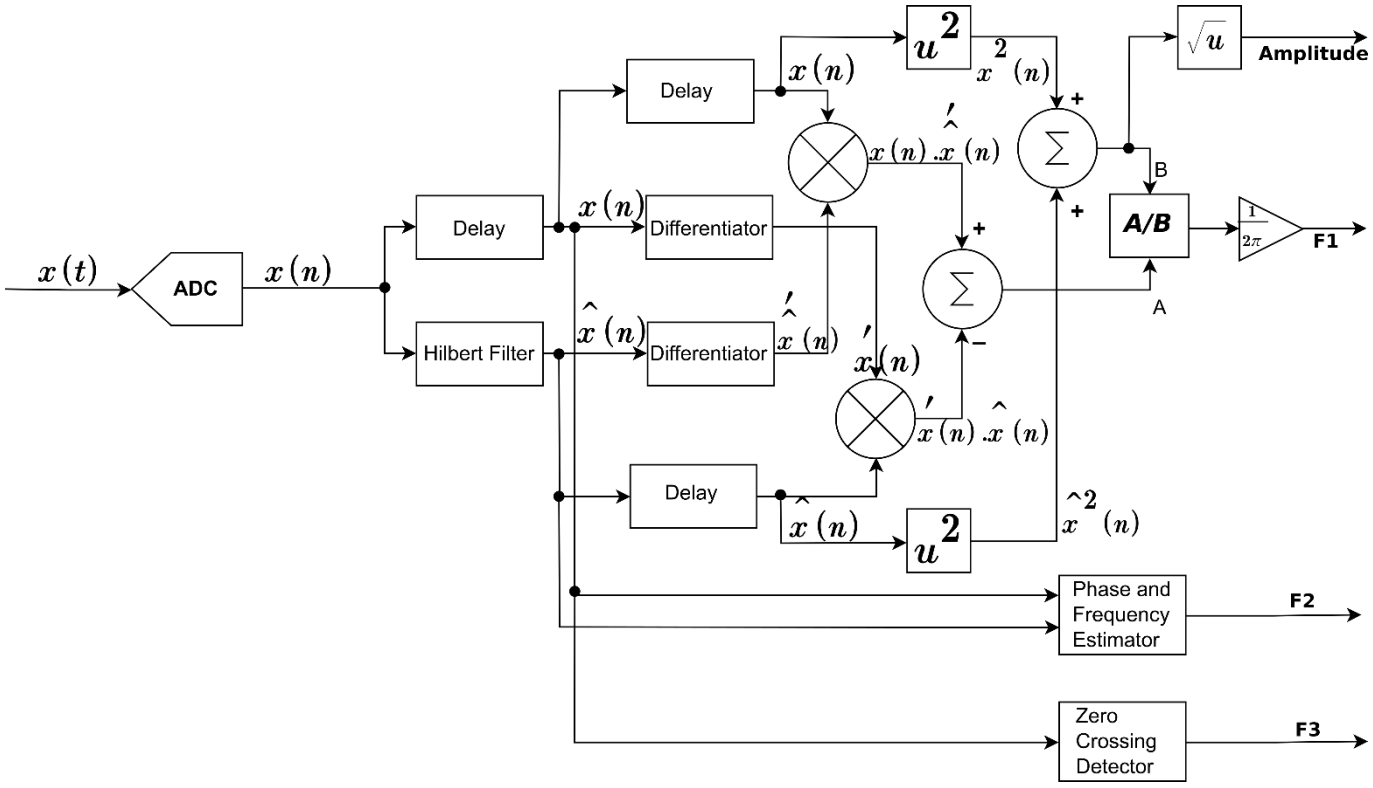

Figure 4: Block diagram of the parameter estimation technique

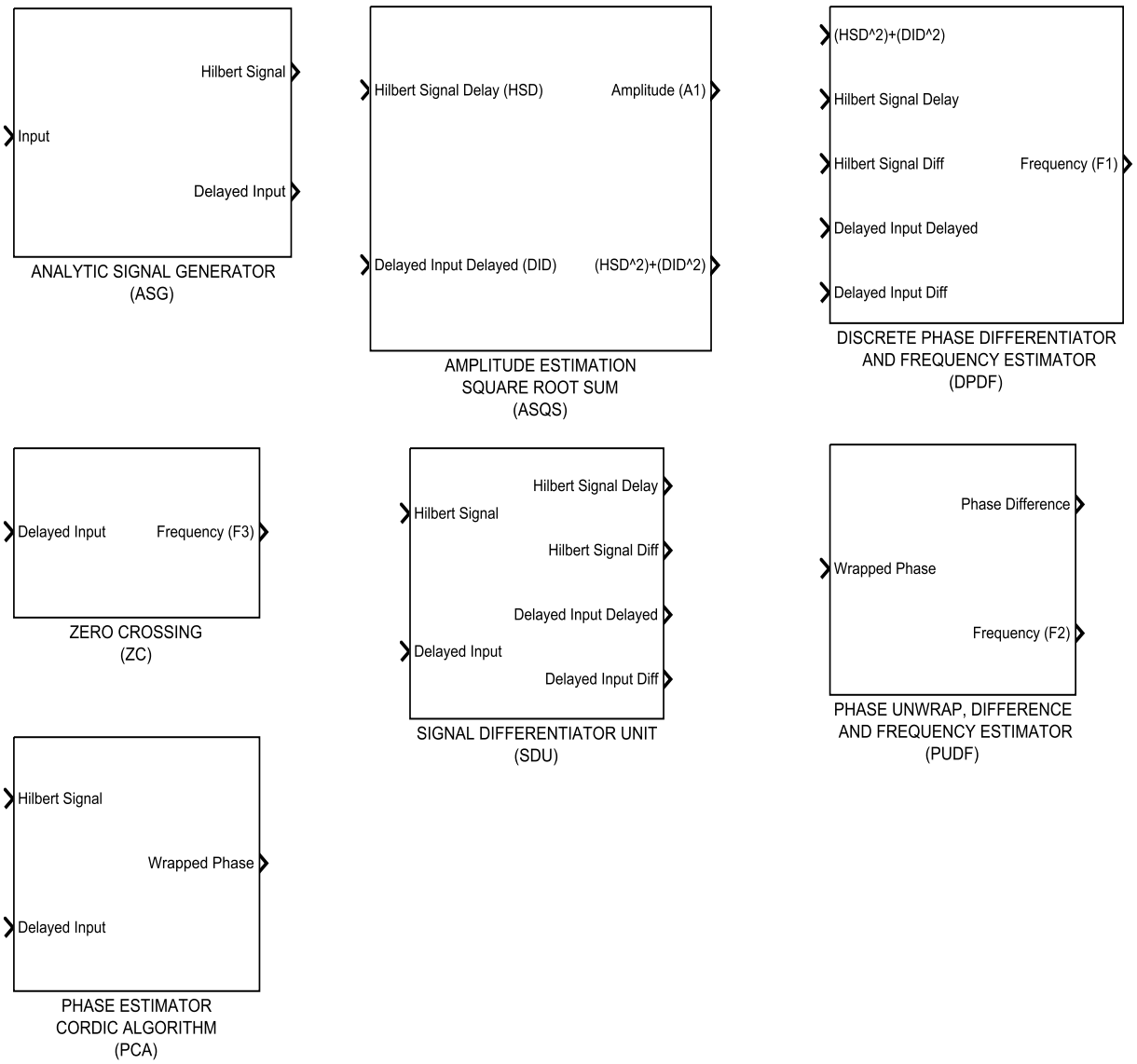

Figure 5: Custom designed System Generator components

Phase Unwrap, Difference and Frequency Estimator (PUDF)

This unit provides the logic elements to unwrap the phase output of the CORDIC block. It also implements the differencing equations as indicated in Equations (15) and (16). This block is user configurable and can either implement Equation (15) or (16) or both at the same time. This block is complex in implementation and consists of summation, gain, relational and delay units. It produces as output the change in phase as well as the frequency which is represented as F2 in Figure 4.

Zero-Crossing Detector (ZC)

The zero-crossing detector monitors the change in sign of the signal as it crosses the zero value. It accurately calculates the time period between these crossings at either half period or full period of a wave cycle. This unit consists of counter, delay, threshold, relational and gain blocks. The output of this block is the frequency that represents F3 in Figure 4.

Table 2.Estimated frequency values obtained for the three frequency outputs

Frequency	40	42	44	46	48	50	52	54	56	58	60	ACTUAL
F1	39.52	41.67	43.83	46.05	48.37	50.56	52.77	54.86	56.92	58.90	60.78	**MEASURED**
F2	39.94	41.96	43.99	45.97	47.93	49.98	51.97	53.96	56.02	58	60.03	
F3	40	42.02	44.03	46.02	48.06	50	52.01	54.03	56.06	58.01	60.04	

Table 3. Percentage error of estimated frequencies

Frequency	40	42	44	46	48	50	52	54	56	58	60	ACTUAL
												AVERAGE % ERROR
F1	1.20	0.79	0.39	0.11	0.77	1.12	1.48	1.59	1.64	1.55	1.30	0.6541
F2	0.15	0.10	0.02	0.07	0.15	0.04	0.06	0.07	0.04	0	0.05	0.0514
F3	0	0.05	0.07	0.04	0.13	0	0.02	0.06	0.11	0.02	0.07	0.0500

Comparison of results obtained

As previously mentioned the designed system was implemented on a SoC device, the Zynq®-7000 hosted on the ZEDBOARD hardware development kit and a test signal was applied to the input of the system. The test signal was varied from 40Hz to 60Hz in increments of 2Hz. The results of the frequency estimation as calculated by the three frequency unit outputs F1, F2 and F3 are tabulated in Table 2. Table 3 shows the accuracy, in terms of frequency percentage error, obtained for each technique.

From Table 3 it can be seen that the frequency measurements obtained from F2 and F3 affords better precision, with the latter being slightly more accurate. The measurement F1 is least accurate and this could be as a result of the fact that differentiators are sensitive to noise and can amplify any spurious signals present in the signal being measured.

CONCLUSION

This paper has presented a design for instantaneous measurement of phase, amplitude and frequency. The emphasis in this paper was on the accurate measurement of frequency as it was found that measurement of this parameter posed more of a challenge than amplitude measurement. In particular three techniques were proposed to measure the frequency. The design of pertinent components were discussed and an overview of their design was given. The system was implemented using Xilinx System Generator which enabled rapid prototype development, simulation and VHDL code generation for the FPGA platform. The three frequency estimation techniques were run in parallel and the results obtained were compared to assess the accuracy and efficiency of the three techniques. It was shown that the techniques based upon zero crossing and the differencing equations afforded greater precision than the technique dependent upon differentiating signals. The project is ongoing and current work is interfacing a functional unit that will provide time synchronization and time stamping of the measured parameters.

REFERENCES

[1] P. Romano and M. Paolone, "Enhanced Interpolated-DFT for Synchrophasor Estimation in FPGAs: Theory, Implementation, and Validation of a PMU Prototype," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 12, pp. 2824–2836, Dec. 2014.

[2] P. Romano and M. Paolone, "An enhanced interpolated-modulated sliding DFT for high reporting rate PMUs," in *Applied Measurements for Power Systems Proceedings (AMPS), 2014 IEEE International Workshop on*, 2014, pp. 1–6.

[3] Z. Dong and P. Zhang, "Emerging Techniques in Power System Analysis," Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg, 2010.

[4] P. Castello, J. Liu, C. Muscas, P. A. Pegoraro, F. Ponci, and A. Monti, "A Fast and Accurate PMU Algorithm for P + M Class Measurement of Synchrophasor and Frequency," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 12, pp. 2837–2845, Dec. 2014.

[5] M. A. Platas-Garza and J. A. de la O Serna, "Polynomial Implementation of the Taylor-Fourier Transform for Harmonic Analysis," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 12, pp. 2846–2854, Dec. 2014.

[6] H. C. Wood, N. G. Johnson, and M. S. Sachdev, "Kalman Filtering Applied to Power System Measurements Relaying," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-104, no. 12, pp. 3565–3573, Dec. 1985.

[7] M. Karimi-Ghartemani, B.-T. Ooi, and A. Bakhshai, "Application of Enhanced Phase-Locked Loop System to the Computation of Synchrophasors," *Power Delivery, IEEE Transactions on*, vol. 26, no. 1, pp. 22–32, Jan. 2011.

[8] "IEEE Standard for Synchrophasor Measurements for Power Systems," *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–61, Dec. 2011.

[9] R. W. S. Alan V. Oppenheim, *Discrete-Time Signal Processing*. Prentice Hall, USA, 1989.

[10] J. H. McClellan and T. W. Parks, "A personal history of the Parks-McClellan algorithm," *IEEE Signal Processing Magazine*, vol. 22, no. 2, pp. 82–86, Mar. 2005.

[11] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 520–538, Apr. 1992.

[12] A. Akan, M. Yalcin, and L. F. Chaparro, "An iterative method for instantaneous frequency estimation," in *ICECS 2001. 8th IEEE International Conference on*

Electronics, Circuits and Systems (Cat. No.01EX483), 2001, vol. 3, pp. 1335–1338 vol.3.

[13] A. Ibrahim, F. Guillet, M. E. Badaoui, and F. Bonnardot, "Techniques to estimate the instantaneous frequency with an aim of diagnosis induction machines faults," in *2008 34th Annual Conference of IEEE Industrial Electronics*, 2008, pp. 391–396.

[14] B. van der Pol, "The Fundamental Principles of Frequency Modulation," *Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution of*, vol. 93, no. 23, pp. 153–158, May 1946.

[15] J. de Ville and others, "Théorie et applications de la notion de signal analytique," *Cables et transmission*, vol. 2, no. 1, pp. 61–74, 1948.

[16] J. E. Volder, "The CORDIC Trigonometric Computing Technique," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, 1959.