Comparison of Synchrophasor Estimation Methods in Simulation Environment and Real Hardware Implementation

Pablo Leibovich, Fernando Issouribehere, *Member, IEEE* and Juan Barbero
IITREE – LAT Universidad Nacional de La Plata
La Plata, Argentina

***Abstract*— This paper presents a description and performance comparison of two synchrophasor estimation algorithms proposed in the literature. The theoretical error performance is analyzed and compared with a practical implementation. Both synchrophasor estimators were implemented in a low cost hardware architecture proposed in the paper, with the purpose of showing the inherent estimator errors and the external error factors such as noise, quantization errors and sampling clock affected by jitter. In order to test the estimation algorithms, analysis under steady-state and dynamic conditions were performed. The tests were made under the conditions specified in the IEEE Standard C37.118.1-2011 and the total vector error of the algorithms was considered as the performance index. The studied and implemented algorithms were chosen by its nature, so one is based on frequency-domain analysis making an Interpolated-Discrete Fourier Transform and the other is based on time-domain analysis, implementing frequency mixing and a low pass FIR filter.**

***Index Terms*-- Error analysis, microcontroller, phasor measurement unit, signal processing, synchrophasor.**

I. INTRODUCTION

In regional or national interconnected power systems, Smart power grids and others bigger power systems, reliability is a main objective to achieve. To this aim, amplitude, phase and frequency of the electrical waveforms must be measured in order to monitor, protect and control the entire system [1].

For years, Supervisory Control And Data Acquisition (SCADA) system was used to provide measurements of the power system stationary state. However, to monitor fast dynamic behavior of wide-area electrical systems, SCADA has important limitations due to its poor temporal resolution and high latency. To obtain measurements at higher rates and lower latency, Phasor Measurement Units (PMUs) are used.

A PMU is a device that provides instantaneous phasors related to the location of the system where it is installed. These phasors are synchronized to the universal coordinated time (UTC), so they have common phase relationship with phasors provided by other PMUs installed in other places. This is why the measured phasors are called Synchrophasors. In order to define accuracy requirements, the main PMU required parameters are specified in the IEEE Standard C37.118.1-2011 [2] and its amendments, which will be referred as the Standard in the rest of this paper.

There are several synchrophasor estimation techniques proposed and evaluated in the state-of-art of PMUs implementation. They can be classified by the domain where the synchrophasor estimation is made. There are time-domain techniques and frequency-domain techniques [3].

The time-domain estimation techniques are based on the analysis of signal samples, by implementing time-domain frequency mixers and filters or other processing algorithms to obtain the phasor. This type of analysis is proposed in the Standard as the reference processing model. The frequency-domain estimation techniques are based on different variants of the Discrete Fourier Transform (DFT), calculated with the Fast Fourier Transform (FFT) algorithm. These techniques are applied to a set of samples, representing a half, one, or multiple waveform cycles.

In this paper, two synchrophasor estimation techniques for voltage and current measurements in three-phase electric power systems are simulated and compared. One, a time-domain algorithm, is based on the method described in the Standard and is called "*Filter based algorithm*" in the rest of the paper. The other, a frequency-domain algorithm, is based on the Interpolated DFT algorithm presented in [4] and is called "*FFT based algorithm*" in the rest of the paper. These techniques are implemented taking into account the effects of signal sampling and quantization errors.

Additionally, this paper shows an implementation of these techniques in real-hardware, comparing the results in terms of Total Vector Error. This comparison also shows how the algorithms are sensible to external errors such as clock jitter and other problems that are typical of real measuring systems implementations.

The structure of the paper is as follows. Section II provides a brief description of the analyzed techniques. Section III presents the simulation of these algorithms, showing the obtained results and errors if the tests proposed in the

978-1-5386-7703-2/18/$31.00 ©2018 IEEE

Standards are made. Then, Section IV describes the real-hardware implementation techniques and Section V shows the experimental results. Finally, Section VI concludes this paper.

II. SYNCHROPHASOR ESTIMATION TECHNIQUES

A. FFT based algorithm

The frequency-domain estimation technique used in this paper is based on the algorithm described in [4]. The power system node voltage can be modeled as a sinusoidal signal with a nominal frequency f_n (i.e. 50 or 60 Hz) and a frequency deviation Δf, rms-value A and initial phase φ. This signal is sampled by the PMU at a fixed rate f_s multiple of f_n and a set of samples is obtained during a time window $T = N/f_s$ ($N \in \mathbb{N}$). T values must be large enough to obtain a good frequency resolution and small enough to be able to consider the signal is in a stationary state.

$$v[n] = \sqrt{2}A\cos\left(2\pi(f_n+\Delta f)\frac{n}{f_s}+\varphi\right),\quad n\in[0,N-1] \quad (1)$$

Typically, the time window T corresponds to one or two nominal signal cycles. This algorithm supposes that frequency deviation Δf satisfies the inequality shown in (2), where δ_f is the DFT frequency resolution.

$$|\Delta f| < \delta_f/2,\qquad \delta_f = 1/T \quad (2)$$

With these considerations, the DFT of the acquired signal is as expressed in (3) where w[n] is the window sequence used to reduce the spectral leakage. In this case, the Hanning window is used.

$$V[k] = \sum_{n=0}^{N-1} v[n]w[n]e^{-\frac{j2\pi nk}{N}},\quad k\in[0,N-1] \quad (3)$$

In order to estimate the amplitude and phase, the Interpolated DFT method must, first, estimate the frequency by interpolating the amplitude of DFT bins. This is done by taking into account the supposition shown in (2) and a sampling rate f_s multiple of f_n. The actual frequency of the signal, $f_n+\Delta f$, may fall between two subsequent DFT bins, where one of them corresponds to the nominal frequency of the signal. With these considerations, the signal frequency can be expressed as:

$$f_n + \Delta f = (k_1 + \delta_{bin})\delta_f \quad (4)$$

where $-0.5 < \delta_{bin} \leq 0.5$ is the deviation from the nominal frequency bin with index $k_1 = Tf_s$.

If sampling rate f_s is much higher than the nominal signal frequency, δ_{bin} can be expressed as shown in (5):

$$\delta_{bin} = \varepsilon(2-\alpha)/(1+\alpha) \quad (5)$$

where α is the ratio between the amplitude of nominal frequency bin (the one with index k_1), which is expected to be the highest bin of the DFT, and the second higher bin:

$$\alpha = |V[k_1]|/|V[k_1+\varepsilon]| \quad (6)$$

ε value can be +1 or -1 and defines the second higher bin:

$$\varepsilon = \text{sign}\left(|V[k_1+1]| - |V[k_1-1]|\right) \quad (7)$$

With these parameters, the amplitude and phase of the synchrophasor can be estimated with the following expressions.

$$A = [2/(N/2)]|V[k_1]|\pi\delta_{bin}(1-\delta_{bin}{}^2)/\sin(\pi\delta_{bin}) \quad (8)$$

$$\varphi = \underline{|V[k_1]|} - \pi\delta_{bin} \quad (9)$$

For final frequency estimation, direct sequence of the three-phase estimation is used. Once the direct sequence D is obtained, frequency deviation is estimated with (10).

$$\Delta f = (\underline{|D[n]|} - \underline{|D[n-1]|})f_n/2\pi \quad (10)$$

B. Filter based algorithm

The standard IEEE C37.118.1 [2] proposes a time-domain synchrophasor estimation technique based on frequency mixers, filters and fixed sampling rate f_s. Each voltage signal is sampled and digitalized with an analog to digital converter. As shown in the FFT based algorithm, the voltage signal sampled at a fixed rate f_s multiple of nominal frequency f_n can be modeled with (1).

The obtained samples are mixed with a local oscillator in phase and quadrature which has a frequency equal to the nominal frequency of the voltage signal. The resulting signal is shown in (11) and (12).

$$v_p[n] = \frac{A}{\sqrt{2}}\left(\cos\left(2\pi\Delta f\frac{n}{f_s}+\varphi\right)+\cos\left(2\pi(2f_n+\Delta f)\frac{n}{f_s}+\varphi\right)\right) \quad (11)$$

$$v_q[n] = \frac{A}{\sqrt{2}}\left(\sin\left(2\pi(2f_n+\Delta f)\frac{n}{f_s}+\varphi\right)-\sin\left(2\pi\Delta f\frac{n}{f_s}+\varphi\right)\right) \quad (12)$$

The set of samples mixed with the oscillator is filtered with a low-pass FIR filter, so the real and imaginary part of synchrophasor is obtained. The standard defines the low pass filter that can be used for a P-class PMU [2]. If the signal is sampled obtaining N samples per cycle of nominal frequency, the filter is an N-order FIR filter with coefficients following the next equation, where m is the coefficient's index.

$$W[m] = 1 - \frac{2}{N+2}|m|,\quad m\in[-N/2, N/2] \quad (13)$$

In Fig. 1, a block diagram of the processing system is shown.

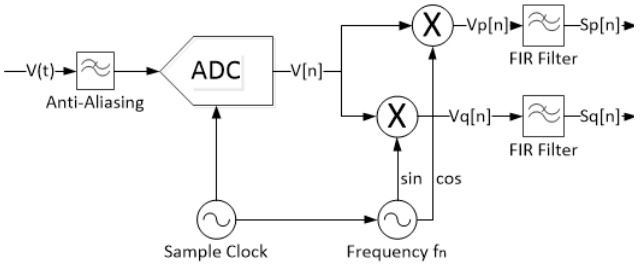

Figure 1. FIR Filter based processing diagram.

The filter, as the Standard states, works well for frequency and phase estimation for both nominal frequency and off-nominal frequency. However, the magnitude estimation must be compensated when off-nominal frequency signal is estimated because of the filter response. To estimate the amplitude, the Standard defines the next expression, where G is the sum of coefficients of the filter.

$$A = \frac{\sqrt{2}}{G}\sqrt{s_p{}^2 + s_q{}^2}\,\frac{2f_n}{\sin\left(\pi\left(f_n + 1.625\Delta f\right)\right)} \qquad (14)$$

The phase estimation is obtained with the following expression.

$$\varphi = \underline{\left|s_p + js_q\right.} \qquad (15)$$

Finally, for frequency estimation, direct sequence of the three-phase estimation is used. Once the direct sequence D is obtained, frequency deviation is estimated with (10).

III. SIMULATION OF ESTIMATION TECHNIQUES

In this section, described synchrophasor estimation techniques are simulated and evaluated under the influence of steady-state conditions, dynamic conditions, and some other tests defined by the Standard. The techniques were designed in a Simulink environment, implementing the processing stage and, also, the acquisition state with sampling, conversion and quantization.

For the FFT based algorithm, the sample rate was configured to 12800 Hz, so 256 samples per nominal cycle of 50 Hz were acquired and the FFT was applied over a 2-cycle length window, so a 512 samples FFT is calculated, For the Filter based algorithm, 800 Hz sample clock was used, therefore 16 samples per cycle of nominal 50 Hz frequency were acquired. The Analog to Digital converter (ADC) was simulated as a 12-bit resolution ADC.

For all cases, the Total Vector Error, defined in (16) and specified by the Standard, was used to analyze and compare the techniques.

$$TVE = \left|V_{Estimated} - V\right| / \left|V\right| \qquad (16)$$

where $V_{Estimated}$ is the estimated synchrophasor and V is the real synchrophasor. Simulated tests were run for 4 seconds and 200 synchrophasor estimations were obtained.

A. Steady-state conditions

This test evaluates the estimation techniques when phases have nominal amplitude and frequency in a range of ±2Hz. Also, it evaluates the techniques under the influence of a single harmonic with amplitude equal to 1% of fundamental signal. In this case, the second harmonic was selected because it represents the worst condition for the estimation. The TVE obtained for both studied techniques under these tests is shown in Fig. 2. The Standard defines that TVE must not be greater than 1% in all of these cases.

B. Measurement bandwith - Modulation

This test evaluates the estimation techniques when amplitude and phase of measured signals are modulated by sinusoidal signals. Modulation frequency was configured to be 2 Hz and a modulation index value of 0.1 was adopted. Both modulations were evaluated separately, as defined by the Standard. The TVE obtained for both studied techniques under these tests is shown in Fig. 3. The Standard defines that TVE must not be greater than 3% in all of these cases.

C. Performance under system frequency ramps.

This test evaluates the estimation techniques when system frequency is affected by a linear ramp of ±1 Hz/s. The variation was applied to the entire three-phase system and the range of variation was ±2 Hz. The TVE obtained for both studied techniques under this test is shown in Fig. 4. The Standard defines that TVE must not be greater than 1% in all of these cases.

In Table I, a summary of results is presented, considering the maximum TVE value obtained in all cases.

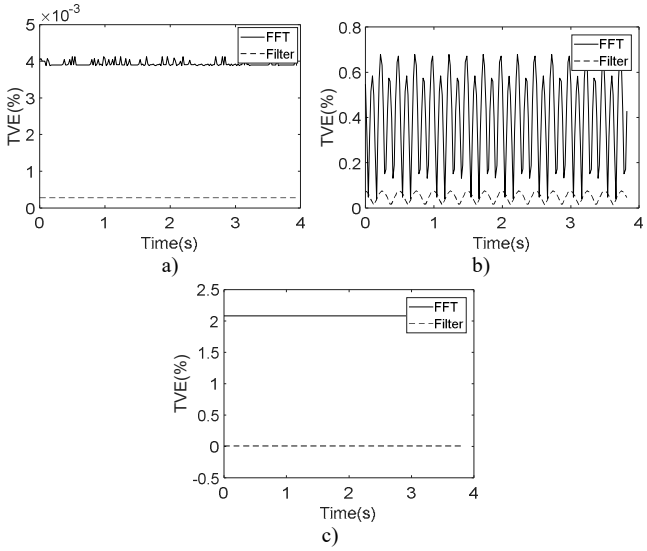

Figure 2. TVE under steady-state conditions. a) nominal amplitude and frequency; b) -2 Hz off-nominal frequency and c) presence of 2nd harmonic.

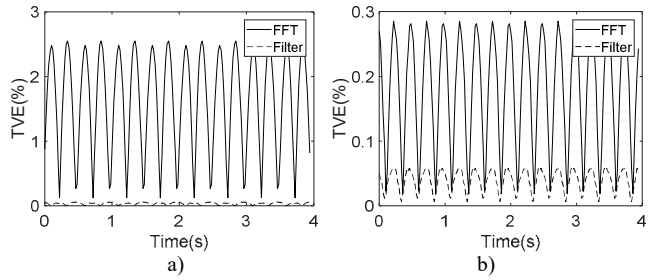

Figure 3. TVE associated to the estimation techniques under modulation conditions. a) amplitude modulation and b) phase modulation.

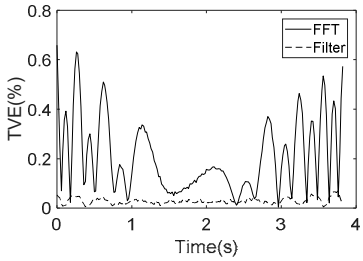

Figure 4. TVE for each method under +1 Hz/s ramp frequency.

TABLE I. Simulation results.

Test		Total Vector Error (%)	
Type	***Characteristics***	***FFT Based***	***Filter Based***
A	Nominal Conditions	0.004	0.0002
	-2 Hz Frequency Deviation	0.68	0.08
	2nd order Harmonic	2	0.006
B	Amplitude Modulation	2.55	0.06
	Phase Modulation	0.29	0.06
C	1 Hz/s Ramp	0.65	0.07

Analyzing the results, it can be concluded that the Filter based estimation technique is compliant with the expected TVE for each test that was analyzed. In the case of the FFT based technique, the obtained Total Vector Error is compliant for most cases but it shows non-compliant error with 2nd order harmonic interference.

IV. Real-Hardware Implementation

Based on the techniques considerations, and with the aim of obtaining real results of the proposed algorithms, the Phasor Estimation Techniques described earlier were implemented with real hardware to sample and process the signals in order to obtain the desired synchrophasors. The most important parts that were used in this design are listed below.

- Microcontroller Unit (MCU) STM32F407VG [5]. It's a high-performance ARM™ Cortex™-M4 from STMicroelectronics Company with 32-bit Reduced Instruction Set Computer (RISC) core operating at a frequency of 64 MHz. It has a single precision floating point unit (FPU) used for FFT or Filter processing and three simultaneous 12-bit successive approximation ADC.
- Raspberry Pi Model 3B [6]. It's a low cost embedded computer designed by the Raspberry Pi Foundation and is based on a Microprocessor Unit BCM2837, an ARM™ Cortex ™-A53 processor running at 1.2 GHz.

The design is based on three independent ADCs integrated on the STM32F407VG Microcontroller. Triggered by an external clock with the corresponding frequency, the microcontroller samples and converts simultaneously [5] the three-phase signals and stores them in a buffer for each phase. Once the buffer received the correct number of samples (256 for the FFT and 16 for the Filter), the microcontroller calculates the FFT or the Filter output for the set of samples.

In the case of the FFT based technique, the microcontroller applies the Hanning window and calculates the FFT. Finally, it transmits the 2nd, 3rd, and 4th bin of the FFT (corresponding to 25Hz, 50 Hz and 75 Hz respectively), with its real and imaginary parts, to the Raspberry Pi via Inter-Integrated Circuit Protocol (I2C) [7].

In the case of the Filter based technique, the microcontroller performs the mixing between samples and an internal oscillator, and calculates the filter output for both the phase and quadrature parts. Finally, it transmits the last sample output of the filtered samples, with its real and imaginary parts, to the Raspberry PI via I2C.

Fig. 5 is a schematic diagram of the system that shows the processing steps and the tasks performed by each part of it.

Once the Raspberry Pi receives the data, it finishes the processing by calculating all the parameters and estimations described in Section II. By using the integrated Ethernet Interface that the Raspberry PI has, the embedded computer implements a server which a client can connect with, to transmit the generated packets of data with the results.

The anti-aliasing filter was implemented with operational amplifiers as a low pass filter with 160 Hz cutoff frequency.

In Fig. 6, a photo shows the STM32F407VG MCU board and the Raspberry Pi with its connections.

V. Experimental Results

The hardware described in previous section was analyzed with the same tests that were run in the simulation inside the Simulink environment. To generate the corresponding waveforms for each test, the Digital to Analog Converters (DAC) of two synchronized STM32F407VG microcontrollers were used. With an external sample clock signal of 25600 Hz, microcontrollers generate the three-phase waveforms saved in a circular buffer in the microcontroller memory.

In Fig. 7 to 9, TVE obtained from the same tests described in Section III are shown. In this case, all of the simulated tests except the ramp of frequency were run for 4 seconds, obtaining 200 syncrophasor estimations. In the case of the system frequency ramp test, 2 seconds were evaluated in order to generate a ramp of -1 Hz/s, from 50 Hz to 48 Hz.

In Table II, the same tests were performed during 10 seconds and the TVE was registered. A summary of results is shown, considering the maximum TVE obtained in all cases. Analyzing the results, it can be concluded that the Filter based estimation technique is compliant with the expected TVE in most of the tests that were analyzed. In most of tests performed with the FFT based technique, a TVE lower than the limit value was obtained. However, an out-of-band TVE was obtained in the off-nominal frequency test.

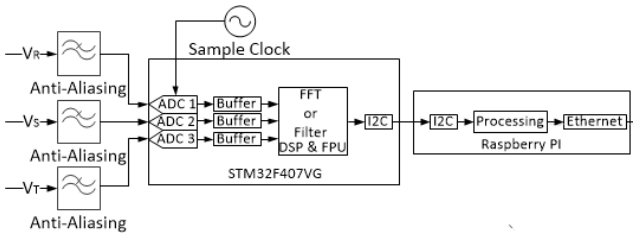

Figure 5. Hardware processing scheme.

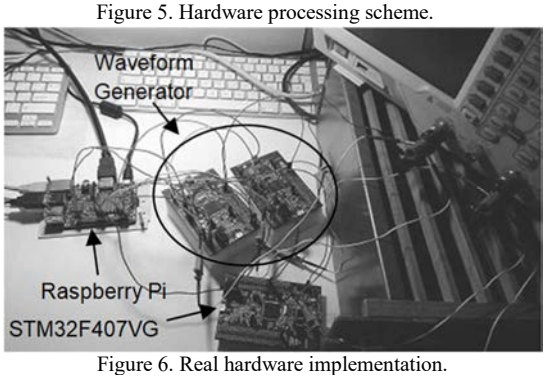

Figure 6. Real hardware implementation.

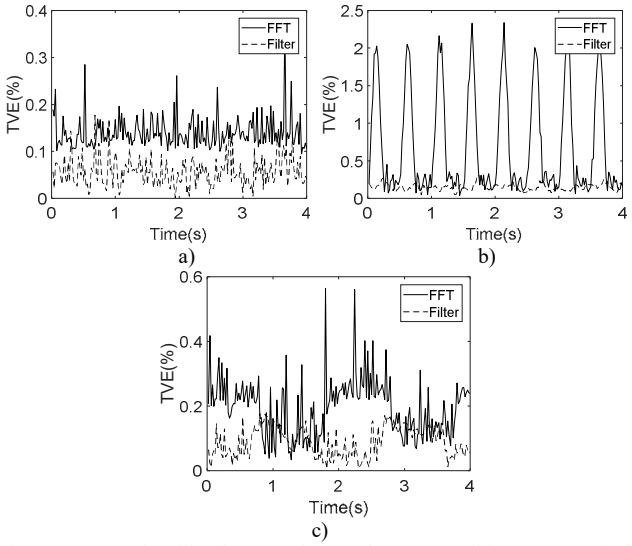

Figure 7. TVE of real hardware under steady-state conditions. a) Nominal amplitude and frequency; b) for -2 Hz off-nominal frequency and c) presence of 2nd order harmonic.

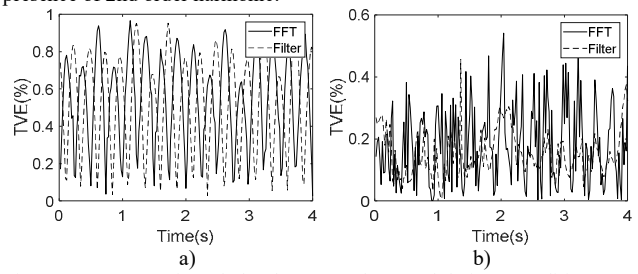

Figure 8. TVE of real hardware under modulation conditions. a) amplitude modulation and b) phase modulation.

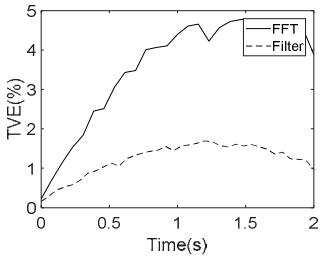

Figure 9. TVE of real hardware under -1 Hz/s ramp frequency.

TABLE II. Real Hardware results.

Test		Total Vector Error (%)	
Type	***Characteristics***	***FFT Based***	***Filter Based***
A	Nominal Conditions	0.37	0.22
	-2 Hz Frequency Deviation	2.34	0.35
	2nd order Harmonic	0.57	0.36
B	Amplitude Modulation	0.98	0.96
	Phase Modulation	0.65	0.45
C	1 Hz/s Ramp	4.80	1.69

In the off-nominal frequency test, an error was obtained when an integer number of signal cycles were not fitted into the processing window. This could be one of the error sources, generating spectral leakage not reduced by the Hanning window. However, taking into account that this effect is also detected in the Simulink simulations, the most important error source could be a jitter present in the sampling clock which affects the measurement.

The third factor is the available method for generating the test signal, which has a sampling frequency equal to the double of the measurement system sampling frequency. If a jitter is present, the generated signal will not be in steady state with the desired frequency and the algorithm will not obtain the expected samples. The last reason previously exposed can also explains the out-of-band TVE observed in the ramp of frequency test for both methods.

As it can be seen, an error in the clock frequencies can cause the algorithms to have higher errors, and this effect is much more important in the case of the FFT. The higher sensibility of the FFT based algorithm can be explained by the fact that the higher the frequency sampling, the more sensible to jitter the sampling results.

VI. CONCLUSIONS

This paper has presented a theoretical and practical comparison of two synchrophasor measurement techniques. The proposed algorithms cover the two main types of algorithms, showing its advantages and disadvantages. A complete procedure of phasors measurement is explained in, showing also real parameters like sampling frequency, window length, etc.

The analysis shows not only how to measure phasors. It shows, also, how to implement them and the results that can be obtained in a real application. The implemented simulations cover the algorithms and the quantization errors obtained from an Analog to Digital conversion.

To implement the measuring hardware, a low cost processing architecture was proposed, showing how the actual technology of microcontrollers and embedded systems can be used for phasor measurement applications. With the implemented hardware, relevant characteristics of estimation techniques were analyzed.

Results obtained with the two implemented techniques in the real hardware showed similar errors in most of the tests.

The comparison between both implementations showed that the FFT based technique is more sensible to external error factors such as noise, embedded FPU rounding, clock jitter, 2nd harmonic, etc. than the Filter based technique.

REFERENCES

[1] A. G. Phadke and J.S. Thorp, *Synchronized Phasor Measurements and Their Applications*, New York: Springer, 2008.

[2] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Std. C37.118.1, Dec. 2011.

[3] A. Monti, C. Muscas and F. Ponci, *Phasor Measurement Units and Wide Area Monitoring Systems*, London: Elsevier, 2016.

[4] P. Romano and M. Paolone, "Enhanced Interpolated-DFT for Synchrophasor Estimation in FPGAs: Theory, Implementation, and Validation of a PMU Prototype," *IEEE Trans. on instrumentation and measurement*, vol. 63, No. 12, Dec. 2014.

[5] STMicroelectronics, "STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced ARM®-based 32-bit MCUs Reference Manual", Rev 13, Sep. 2016.

[6] Raspberry Pi website. Get started with Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/learning/hardware-guide/

[7] Broadcom Corporation. "BCM2835 ARM Peripherals", Feb. 2012.