# Implementation and Validation of a Dual P- and M-Class Compliant PMU Prototype Based on the Delayed In-Quadrature Interpolated DFT

César García-Veloso
José María Maza-Ortega
Department of Electrical Engineering
University of Seville
Seville, Spain
{cgveloso, jmmaza}@us.es

Mario Paolone
Alexandra Karpilow
Distributed Electrical Systems Laboratory
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
{mario.paolone, alexandra.karpilow}@epfl.ch

***Abstract*—In this paper, the design and experimental validation of a prototype phasor measurement unit (PMU) that simultaneously meets the P and M class requirements of the IEC/IEEE Std 60255-118-1-2018 are presented. The device is based on a synchrophasor estimation (SE) algorithm, previously formulated by the authors, which exploits the generation and use of a delayed in-quadrature complex signal to mitigate the self-interference of the fundamental tone. The method is deployed to a NI CompactRIO-9039 platform, requiring a total use of 19.2% flip-flops, 50.6% LUTs, 77.9% DSPs and 13.7% BRAM for a single-channel configuration and operating under a 50-fps report rate, 50 kHz sampling rate, and three nominal cycle observation windows. A PMU calibrator is used to perform a comprehensive metrological analysis of the device and verify its compliance with the standard. The results show that the prototype meets the requirements of both P and M classes with a worst-case measurement reporting latency of 36.21 ms.**

***Index Terms*—Field programmable gate array (FPGA), IEC/IEEE Std 60255-118-1-2018, interpolated DFT (IpDFT), phasor measurement unit (PMU), quadrature signal generator (QSG).**

## I. Introduction

The ability to cope with out-of-band interferences (OOBI) is one of the key differences that separates the two PMU performance classes (P and M) defined in the IEC/IEEE Std 60255-118-1-2018 [1]. The range of disturbances in this category, as defined in [1], depends on the reporting rate $F_r$ of the PMU and covers interferences within the subharmonic and interharmonic range around the fundamental tone[1]. As known, while harmonic components refer to spectral interferences at integer multiples of the fundamental frequency, interharmonics represent those that occur at noninteger values [2], and subharmonics simply denote a specific case for the latter when their frequencies fall below the fundamental [3].

Cycloconverters, time-varying loads and double conversion systems, i.e. asynchronous AC systems coupled by power electronic equipment through a DC link, are among the main sources of interharmonics [2]. Furthermore, adjustable-speed drives, which are widely present in industry, are considered the main source of harmonics and interharmonics in the grid [4]. Moreover, PV systems also cause interhamonic emissions, as shown with field measurements [5] and experimental laboratory tests [6]. In [7], one of the causes of these emissions has been shown to be the perturbation from maximum power point tracking (MPPT), and a model is proposed to predict the interharmonic characteristic in PV systems.

The prevalence of such interferences demonstrates the need for resilient measurement devices capable of providing reliable estimates. From this perspective, the advantage of a single PMU capable of complying with the requirements of both PMU classes is twofold. It not only provides better performance, but also reduces the total cost of the sensing infrastructure for the network operator. This is crucial, especially in distribution networks, where the lack of measurement infrastructure, radial nature, and larger number of branches render a more complex system that would require a higher number of PMUs compared to the transmission level. In fact, there are many studies that work on the optimal placement of PMUs in distribution systems, e.g. [8]–[10].

An example of a practical implementation that would benefit from the usage of P- and M-class-compliant PMUs can be found in [11], where a two-layer control for the dispatch of active distribution networks is proposed and experimentally validated on a real MV distribution grid in Aigle, Switzerland. The proposed control operates based on the data provided by a real-time state estimator that uses the measurements delivered by a monitoring system consisting of 17 P-class certified

The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 58 100010434). The fellowship code is LCF/BQ/DR20/11790026. César García-Veloso received financial support from the International Mentoring Foundation for the Advancement of Higher Education to cover his personal expenses, travel expenses and health insurance over a 3-month period during his research stay at the Distributed Electrical Systems Laboratory (EPFL).

[1]The IEC/IEEE Std [1] defines the OOBI range as [10 Hz - $(f_n - \frac{F_r}{2})$ Hz]$\cup$[$(f_n + \frac{F_r}{2})$ Hz- $2f_n$ Hz], where $f_n$ denotes the nominal frequency of the system. All M-class PMUs must meet the accuracy limits set in [1] for signals affected by this kind of disturbance.

PMUs. Replacing the current P-compliant PMUs with dual P- and M-class-compliant devices would ensure the correct operation of such control even in the presence of OOBI disturbances, resulting in a more resilient estimation of the system state and the consequent enforcement of the grid's constraints.

Despite the benefits of joining the strengths of both PMU classes in a single device capable of simultaneously meeting the requirements set in [1], very few works in the literature are capable of achieving this [12]–[15]. In [14] the authors presented an algorithm based on the use of interpolated DFT (IpDFT) which exploits the generation and use of a delayed in-quadrature complex signal to mitigate the self-interference of the fundamental tone. Furthermore, the technique, named time-delay IpDFT (TD-IpDFT), allows one to estimate and iteratively remove the impacts of an interference tone within the OOBI range with a magnitude greater than or equal to 4% of that of the fundamental. This work seeks to provide a complete and comprehensive experimental validation of the method described in [14], extending the simulation benchmark and preliminary testing presented in [14] and including an evaluation of reporting latency to demonstrate its readiness and suitability for synchrophasor estimation (SE). To achieve this, a PMU prototype, powered with the TD-IpDFT algorithm at its core, is deployed to a commercial National Instruments CompactRIO-9039 embedded control and acquisition system [16]. Compliance with [1] is evaluated using the PMU calibrator described in [17] and applied in [18]. Both the PMU prototype and the calibrator are synchronized to Coordinated Universal Time (UTC) using the time information provided by a stationary Global Positioning System (GPS) unit.

Thus, the three main contributions of this paper are:

- The adaptation of the SE algorithm proposed in [14] for its implementation on industry-grade time deterministic hardware.
- The actual deployment of the algorithm in a PMU prototype using a commercial National Instruments CompactRIO-9039 unit [16].
- Its metrological analysis with state-of-the-art calibration equipment [17] previously used for the validation of other PMU designs.

The remainder of the paper is structured as follows. Section II summarizes the fundamentals of the TD-IpDFT method formulated in [14]. Section III describes the design and implementation of the method on FPGA hardware and its associated computational cost. Section IV presents the experimental setup and the results that prove the compliance of the PMU with the standard. Finally, Section V concludes the paper.

## II. The Time-Delay IpDFT

This section summarizes the fundamentals of the TD-IpDFT method formulated in [14]. The technique combines the generation and use of a delayed in-quadrature complex signal to mitigate the self-interference of the fundamental tone with a three-point DFT interpolation to counteract the effects of long-range leakage [19]. Moreover, it adopts a Hanning window function as a good balance between sidelobe decay and mainlobe width [20]. For a detailed description of the method, the reader can refer to [14].

### A. Three point (3p) IpDFT based on the Hanning Window

The DFT spectrum $X(k)$ of a set of $N$ samples $x(n)$ (1) taken from a single tone steady-state signal $x(t)$ with a sampling period $T_s$ can be calculated according to (2).

$$x(n) = A_0 \cos\left(2\pi f_0 n T_s + \varphi_0\right), \quad n \in [0, N-1] \tag{1}$$

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \mathcal{W}_N^{-nk}, \quad k \in [0, N-1] \tag{2}$$

In (1) $f_0$, $A_0$ and $\varphi_0$ denote the fundamental frequency, amplitude and initial phase while in (2) $\mathcal{W}_N = e^{j2\pi/N}$ represents the Twiddle factor. The Hanning windowed DFT spectrum $X_H(k)$ can be obtained from $X(k)$ by operating directly in the frequency domain:

$$X_H(k) = 0.5X(k) - 0.25(X(k-1) + X(k+1)) \tag{3}$$

Given $X_H(k)$, the 3p Hanning IpDFT provides an analytical formulation to determine $f_0$ (5a), $A_0$ (5b) and $\varphi_0$ (5c), based on the spectral shift of the fundamental tone with respect to the maximum bin $k_m$ ($\delta_H$) (4):

$$\delta_H = 2\varepsilon \frac{|X_H(k_m+\varepsilon)| - |X_H(k_m-\varepsilon)|}{|X_H(k_m-\varepsilon)| + 2|X_H(k_m)| + |X_H(k_m+\varepsilon)|} \tag{4}$$
$$\delta_H \in [-0.5, 0.5)$$

where $\varepsilon = \pm 1$ if $|X_H(k_m+1)| \gtrless |X_H(k_m-1)|$ and $k_m$ is the index of the highest bin. Note that $\delta_H$ equals 0 for the particular case of coherent sampling ($NT_s f_0 \in \mathbb{Z} \Rightarrow k_0 = k_m$).

$$f_{0_H} = (k_m + \delta_H)\Delta_f = k_0 \Delta_f \tag{5a}$$

$$A_{0_H} = 2|X_H(k_m)| \left| \frac{\pi\delta_H}{\sin(\pi\delta_H)} \right| |\delta_H^2 - 1| \tag{5b}$$

$$\varphi_{0_H} = \angle X_H(k_m) - \pi\delta_H \tag{5c}$$

In (5a) $\Delta_f$ represents the frequency resolution of the DFT spectrum, which is reciprocal to the considered window length $T = NT_s$. It is important to note that (4)-(5), hereafter referred to as `IpDFT` and which will result in a set of estimates $(\hat{f}, \hat{A}, \hat{\varphi})$, are only valid if the bins $X_H(k)$ are not affected by aliasing and correspond to the positive image of the tone to be calculated.

### B. Fundamental Negative Image Supression: TD-QSG

As detailed in [14] a two-step delayed in-quadrature complex signal generation method based on the IpDFT, named TD-QSG, is adopted. The technique, described in Algorithm 1 can be seen as a 'filter' that acting on the original real signal $y(n)$ produces a complex output $\bar{y}(n) = y(n) + jy(n - d_\theta)$ with $d_\theta$ being the applied delay. Since the method relies on the delay of input $y(n)$ to generate the imaginary in-quadrature component, its application is intrinsically limited to suppress the negative

image of a specific range of tones[2]. This is because the same delay $d_\theta$ in terms of samples results in different phase shifts $\theta_\alpha = \omega_\alpha d_\theta T_s$ depending on the frequency $\omega_\alpha = 2\pi f_\alpha$ of each tone '$\alpha$'. Thus the filter exhibits a periodic frequency response, as shown in Fig. 1 (a), centered on the tone whose frequency $f$ has been considered for generation of the delay $d_\theta$. The plot is shown for a normalized frequency $f_0$[pu] $= f_0/f$ and $\sigma_+$ and $\sigma_-$ denote the complex positive and negative delay gains and describe, respectively, the positive and negative frequency responses of the filter:

**Algorithm 1** TD-QSG Algorithm [14]

**Input:** $[x(n)]$

1: $d_0 = \left\lfloor \frac{f_s}{4f_n} \right\rceil$
2: $\bar{x}_o(n) = x(n) + jx(n - d_0)$
3: $X_o(k) = \texttt{DFT}[\bar{x}_o(n)]$
4: $X_{o_H}(k) = 0.5X_o(k) - 0.25(X_o(k-1) + X_o(k+1))$
5: $\{\hat{f}_0\} = \texttt{IpDFT}[X_{o_H}(k)]$
6: $d_f = \left\lfloor \frac{f_s}{4\hat{f}_0} \right\rceil$
7: $\bar{x}_f(n) = x(n) + jx(n - d_f)$

**Output:** $\{\bar{x}_f(n)\}$

$\lfloor ... \rceil$ denotes the round-to-the-nearest integer function.

$$\sigma_\pm = 1 + e^{j(\pi/2 \mp \omega_0 d_\theta T_s)} \tag{6}$$

To illustrate the suppression effect on the negative image of the TD-QSG technique, Fig. 1 (b) and (c) compare the spectrum at the input and output, respectively, of the filter, when analyzing a signal $y(n)$ corrupted by an interfering tone.

### C. The TD-IpDFT Technique

The TD-IpDFT algorithm assumes a static signal model so that the samples within the analysis window ($n \in [0, N-1]$) are the result of a fundamental tone and a potential interference tone, respectively, characterized by the parameter sets $\{A_0, f_0, \varphi_0\}$ and $\{A_i, f_i, \varphi_i\}$ denoting their amplitude, frequency, and initial phase. The method, which is summarized using the pseudocode in Algorithm 2, provides estimates of the fundamental tone's parameters $\{\hat{A}_0, \hat{f}_0, \hat{\varphi}_0\}$ as well as those of an interfering tone $\{\hat{A}_i, \hat{f}_i, \hat{\varphi}_i\}$ if detected.

First, the DFT spectrum $(X_f(k))$ (line 2) of the delayed in-quadrature complex signal $(\bar{x}_f)$ (line 1) resulting from the TD-QSG filter (Algorithm 1) is calculated and subsequently windowed in the frequency domain through the Hanning window $(X_{f_H}(k))$ (line 3). The resulting spectrum is then used to obtain a first estimate of the signal parameters (line 4) using an IpDFT. After initializing a set of auxiliary variables $\{\hat{X}^0_{i+}, \hat{X}^0_{i-}, R^0_e, \tau_{R_e}, \tau_i\}$ (line 5) the iterative detection and compensation loop is entered (line 6). Three stop conditions are defined to exit the iterative routine. A maximum number of iterations $Q$, a residual energy exit flag $(\tau_{R_e})$, which is activated if the evolution of the total normalized residual spectral energy $R_e$ falls below a predefined threshold level $(\lambda_{R_e})$, and the non-activation of the interference detection flag $(\tau_i)$ if no interferences are found.

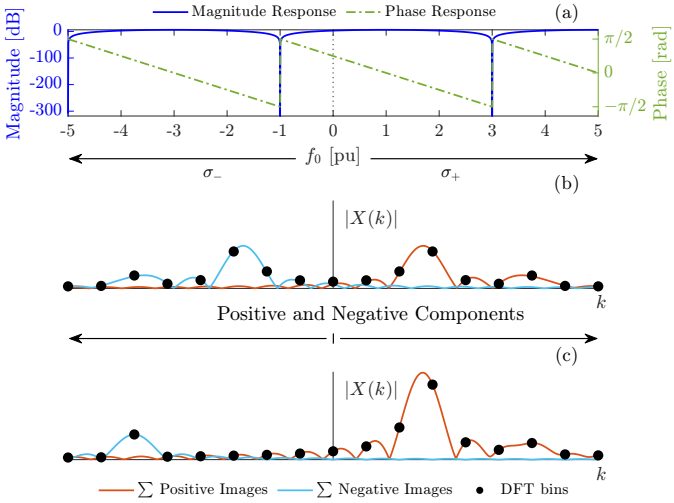


Fig. 1. Effect on the frequency domain of the TD-QSG 'filter': (a) Frequency response of a 'filtered' delayed in-quadrature complex signal $\bar{y}(n)$ generated considering a frequency $f$ from a signal $y(n)$ with normalized frequency $f_0$[pu] $= f_0/f$; (b) spectrum of $y(n)$ affected by additive white Gaussian noise with a signal-to-noise ratio equal to 80 dB [$y(n) = A_0 \cos(2\pi f_0 n T_s) + A_i \cos(2\pi f_i n T_s)$, $n \in \mathbb{Z}$; where $A_0 = 1$, $f_0 = 45$ Hz, $A_i = 0.35$ and $f_i = 100$ Hz] and (c) spectrum of $\bar{y}(n)$. Note that a rectangular window is used to represent (b) and (c) so that the effects of long-range spectral leakage are noticeable. A 35% harmonic distortion is selected for the same reason.

If $\tau_{R_e}$ has not yet been triggered, which is the case when the loop is first entered (current iteration '$q$' $= 1$), Algorithm 3 is used to estimate the spectral contribution of the fundamental tone $(\hat{X}^{q-1}_0)$. Algorithm 3, named Time-Delay Spectral Reconstruction (TD-SR), determines the spectral contribution of any tone '$\alpha$' within $\bar{x}_f$ given estimates of its frequency $(\hat{f}_\alpha)$ and of its uncorrected positive image amplitude $(\hat{A}_{\alpha+})$ and phase $(\hat{\varphi}_{\alpha+})$. `wf` within Algorithm 3 refers to (10) defined in Section II-D. $\hat{X}^{q-1}_0$ is then removed from $X_{f_H}(k)$ together with the spectral contribution from the negative image of the interference tone $(\hat{X}^{q-1}_{i-})$, which has been initialized to 0 for the first iteration $\hat{X}^0_{i-}(k) = 0$. The resulting bins correspond to the positive image of a potential interference $\hat{X}^q_{i+}(k)$ (line 9) and are used to assess whether an interference is present (lines 10-16). This is only done during the first iteration by calculating the energies $E_c$, $E_o$, and $E_i$ and comparing their ratios with three heuristically defined threshold levels ($\lambda^l_o$, $\lambda^u_o$, $\lambda_i$). $E_c$ (8) only considers the range in which a potential OOBI is expected[3] by grouping the highest magnitude bin $k_c$ (7) together with its two adjacent bins.

$$k_c = \arg\max_k |\hat{X}_{i+}(k)|; \quad k \in [0, 2] \cup [4, 7] \tag{7}$$

[2]These are the tone of interest '$a$' and those other tones '$b$' that fulfill $f_b/f_a = 1 + 4n \quad \forall n \in \mathbb{N}$.

[3]For $T = 3/f_n$ most of the energy of a potential OOBI is within bins $[0-2] \cup [4-7]$, i.e. frequency range $[0-2]\Delta_f \cup [4-7]\Delta_f$ Hz.

$$E_c = \sum_k |\hat{X}_{i+}(k)|^2; \begin{cases} k \in [0,2], & \text{if } k_c = 0 \\ k \in [5,7], & \text{if } k_c = 7 \\ k \in [k_c \pm 1], & \text{otherwise} \end{cases} \quad (8)$$

If no interference is detected ($\tau_i = 0$) the process is exited by removing the amplitude and phase alterations introduced by the filter (line 27-28). Otherwise, $R_e$ is calculated as the normalized energy of the residual spectrum ($\hat{X}_r^q(k)$) (lines 18-19), where again for the first iteration $\hat{X}_{i+}^0(k) = 0$, and its variation is evaluated to trigger $\tau_{R_e}$. The parameters corresponding to the positive image of the detected interference $\{\hat{f}_i^q, \hat{A}_{i+}^q, \hat{\varphi}_{i+}^q\}$ are obtained via a second IpDFT (line 23) and subsequently used to approximate its full spectrum (line 24). Finally, a refined fundamental spectrum ($X_{f_H}(k) - \hat{X}_i^q(k)$) is used to enhance the estimates of the main tone (line 25). When either condition is met ($q = Q$ or $\tau_{R_e} = 0$) Algorithm 4 named Time-Delay Amplitude and Phase correction (TD-APc), is used to remove the amplitude and phase alterations introduced by the filter (lines 31 and 36)[4]. Lastly, the rate of change-of-frequency (ROCOF) at the reporting time $m$ is calculated with a first-order backward approximation of a first-order derivative based on two successive frequency estimates at the reporting times $m$ and $m-1$.

$$\hat{\dot{f}}_0(m) = \Big(\hat{f}_0(m) - \hat{f}_0(m-1)\Big) F_r \quad (9)$$

The reader is referred to [14] where the selection and tuning of Q, $\lambda_o^l$, $\lambda_o^u$, $\lambda_i$ and $\lambda_{Re}$ are thoroughly justified and a more in-depth description of Algorithms (1-4) is also provided. In addition, a refinement of the selection of $\lambda_o^l$, $\lambda_o^u$ and $\lambda_i$ based on improved testing conditions is presented in the Appendix.

### *D. Adaptation for Hardware Deployment*

Although the TD-IpDFT [14] was designed from the very beginning with its implementability in mind, its full deployment on industry-grade hardware has required some non-trivial considerations, adaptations, and modifications. The most relevant are now described.

*1) Adaptation to fixed-point arithmetic:* In [14] the simulation benchmark was performed based on double precision floating point format, while for the preliminary tests only the signal acquisition and the TD-QSG (Algorithm 1) including the DFT calculation (done through the mSDFT [21]) were implemented at the FPGA level and thus translated to fixed-point precision. The current implementation presents the full deployment of the method, which requires careful adaptation to ensure that accuracy can be preserved without exceeding the available computational resources.

*2) Alternative spectra reconstruction method:* A simpler and more efficient way to estimate the spectral contributions (`wf`) of the positive or negative image of a tone '$\alpha$' is adopted compared to the one used in [14]. For this, expressions (10a)-(10c) are used that rely on the estimated parameters of the

[4]Contrary to line 27 where there is no need to recalculate $\sigma_{+_0}$ as it is already obtained when Algorithm 3 is applied in line 8.

**Algorithm 2** TD-IpDFT Algorithm [14]

**Input:** $[x(n)]$
1: $\{\bar{x}_f(n)\} = \texttt{TD-QSG}[x(n)]$
2: $X_f(k) = \texttt{DFT}[\bar{x}_f(n)]$
3: $X_{f_H}(k) = 0.5X_f(k) - 0.25(X_f(k-1) + X_f(k+1))$
4: $\{\hat{f}_0^0, \hat{A}_{0+}^0, \hat{\varphi}_{0+}^0\} = \texttt{IpDFT}[X_{f_H}(k)]$
5: $\hat{X}_{i+}^0(k) = 0; \hat{X}_{i-}^0(k) = 0; R_e^0 = 0; \tau_{R_e} = 1; \tau_i = 0$
6: **for** $q = 1$ **to** $Q$ **do**
7: **if** $\tau_{R_e} = 1$ **then**
8: $\{\hat{X}_0^{q-1}(k), \sigma_{+_0}^{q-1}\} = \texttt{TD-SR}[\hat{f}_0^{q-1}, \hat{A}_{0+}^{q-1}, \hat{\varphi}_{0+}^{q-1}]$
9: $\hat{X}_{i+}^q(k) = X_{f_H}(k) - \hat{X}_0^{q-1}(k) - \hat{X}_{i-}^{q-1}(k)$
10: **if** $q = 1$ **then**
11: Apply (7)-(8)
12: $E_o = \sum |X_{f_H}(k)|^2$; $E_i = \sum |\hat{X}_{i+}(k)|^2$
13: **if** $(\frac{E_c}{E_o} \in [\lambda_o^l, \lambda_o^u]$ **and** $\frac{E_c}{E_i} \geq \lambda_i)$ **or** $\frac{E_c}{E_o} > \lambda_o^u$ **then**
14: $\tau_i = 1$
15: **end if**
16: **end if**
17: **if** $\tau_i = 1$ **then**
18: $\hat{X}_r^q(k) = \hat{X}_{i+}^q(k) - \hat{X}_{i+}^{q-1}(k)$
19: $R_e^q = \sum |\hat{X}_r^q(k)|^2 / E_o$
20: **if** $|R_e^q - R_e^{q-1}| < \lambda_{R_e}$ **then**
21: $\tau_{R_e} = 0$
22: **end if**
23: $\{\hat{f}_i^q, \hat{A}_{i+}^q, \hat{\varphi}_{i+}^q\} = \texttt{IpDFT}[\hat{X}_{i+}^q(k)]$
24: $\{\hat{X}_i^q(k), \text{-}, \hat{X}_{i+}^q(k), \hat{X}_{i-}^q(k)\} = \texttt{TD-SR}[\hat{f}_i^q, \hat{A}_{i+}^q, \hat{\varphi}_{i+}^q]$
25: $\{\hat{f}_0^q, \hat{A}_{0+}^q, \hat{\varphi}_{0+}^q\} = \texttt{IpDFT}[X_{f_H}(k) - \hat{X}_i^q(k)]$
26: **else**
27: $\hat{\varphi}_0^{q-1} = \hat{\varphi}_{0+}^{q-1} - \angle\sigma_{+_0}^{q-1}$; $\hat{A}_0^{q-1} = \frac{\hat{A}_{0+}^{q-1}}{|\sigma_{+_0}^{q-1}|}$
28: **break**
29: **end if**
30: **else**
31: $\{\hat{A}_0^q, \hat{\varphi}_0^q\} = \texttt{TD-APc}[\hat{f}_0^q, \hat{A}_{0+}^q, \hat{\varphi}_{0+}^q]$
32: **break**
33: **end if**
34: **end for**
35: **if** $q = Q$ **then**
36: $\{\hat{A}_0^Q, \hat{\varphi}_0^Q\} = \texttt{TD-APc}[\hat{f}_0^Q, \hat{A}_{0+}^Q, \hat{\varphi}_{0+}^Q]$
37: **end if**

**Output:** $\{\hat{f}_0, \hat{A}_0, \hat{\varphi}_0\}$

**Algorithm 3** TD-SR Algorithm [14]

**Input:** $[\hat{f}_\alpha, \hat{A}_{\alpha+}, \hat{\varphi}_{\alpha+}]$
1: $\varphi_{d_\alpha} = 2\pi \hat{f}_\alpha d_f T_s$
2: $\sigma_{+_\alpha} = 1 + e^{j(\frac{\pi}{2} - \varphi_{d_\alpha})}; \sigma_{-_\alpha} = 1 + e^{j(\frac{\pi}{2} + \varphi_{d_\alpha})}$
3: $\hat{\varphi}_{\alpha-} = -(\hat{\varphi}_{\alpha+} - \angle\sigma_{+_\alpha}) + \angle\sigma_{-_\alpha}$
4: $\hat{A}_{\alpha-} = \hat{A}_{\alpha+} \frac{|\sigma_{-_\alpha}|}{|\sigma_{+_\alpha}|}$
5: $\hat{X}_{\alpha+}(k) = \texttt{wf}[\hat{f}_\alpha, \hat{A}_{\alpha+}, \hat{\varphi}_{\alpha+}]; \hat{X}_{\alpha-}(k) = \texttt{wf}[-\hat{f}_\alpha, \hat{A}_{\alpha-}, \hat{\varphi}_{\alpha-}]$
6: $\hat{X}_\alpha(k) = \hat{X}_{\alpha+}(k) + \hat{X}_{\alpha-}(k)$

**Output:** $\{\hat{X}_\alpha(k), \sigma_{+_\alpha}, \hat{X}_{\alpha+}(k), \hat{X}_{\alpha-}(k)\}$

**Algorithm 4** TD-APc Algorithm [14]

**Input:** $[\hat{f}_\alpha, \hat{A}_{\alpha+}, \hat{\varphi}_{\alpha+}]$

1: $\varphi_{d_\alpha} = 2\pi \hat{f}_\alpha d_f T_s$; $\sigma_{+_\alpha} = 1 + e^{j(\frac{\pi}{2} - \varphi_{d_\alpha})}$;

2: $\hat{\varphi}_\alpha = \hat{\varphi}_{\alpha+} - \angle\sigma_{+_\alpha}$; $\hat{A}_\alpha = \frac{\hat{A}_{\alpha+}}{|\sigma_{+_\alpha}|}$

**Output:** $\{\hat{A}_\alpha, \hat{\varphi}_\alpha\}$

tone $\{\hat{f}_\alpha, \hat{A}_\alpha, \hat{\varphi}_\alpha\}$. These are simply obtained by rearranging expressions (5a) - (5c).

$$\delta_\alpha = (k - \hat{f}_\alpha/\Delta_f) \tag{10a}$$

$$|X_H(k)| = \frac{\hat{A}_\alpha}{2|\delta_\alpha^2 - 1|} \left| \frac{\sin(\pi\delta_\alpha)}{\pi\delta_\alpha} \right| \tag{10b}$$

$$\angle X_H(k) = \hat{\varphi}_\alpha + \pi\delta_\alpha \tag{10c}$$

*3) Processes distribution in the FPGA:* Another important aspect is the distribution of functions in the FGPA, meaning how they are implemented and share its resources. Within the FPGA, a dedicated section of hardware is reserved for each function. Thus, multiple instances can be run by either parallelizing or serializing them, resulting in a trade-off between computational speed and hardware usage. Therefore, efficient use of equipment requires a proper distribution and allocation of functions calls. Similarly, common memories, such as those used to buffer DFT bins (see Section III-A), could be accessed by processes executed at different rates, causing overlapping and unintended data overwriting. Thus, proper access coordination as well as an adequate sizing of the dedicated memories are also essential.

## III. Hardware Implementation

This section describes the design and implementation of the TD-IpDFT described in [14] on FPGA hardware and its associated computational cost. As in other previous implementations of PMU prototypes [18], [22], a CompactRIO platform is chosen, as its versatility allows the incorporation of all essential functions of a PMU. For this paper the prototype is deployed on a National Instruments CompactRIO-9039 [16] which includes a (i) 1.91 GHz quad-core Intel Atom E3845 real-time processor with 16 GB nonvolatile solid state drive data storage and 2 GB of DDR3L memory together with a (ii) reconfigurable Xilinx Kintex-7 325T FPGA equipped with a 40 MHz on-board clock and consisting of 407,600 flip-flops, 203,800 look-up tables (LUTs), 16,020 kbits of BRAM and 840 DSP slices.

An overview of the overall architecture of the proposed PMU prototype is presented in Fig. 2. As shown, the input signal is acquired using a NI 9215 module, which is equipped with an independent 16 bit $\pm$10V successive approximation register (SAR) analog-to-digital converter (ADC) capable of sampling frequencies up to 100 kHz [23]. Measurement of high-voltage signals in the field would require the use of dedicated instrumentation transformers, whose contribution and potential impact are beyond the scope of this work. Although the 9215 module has four input channels and thus

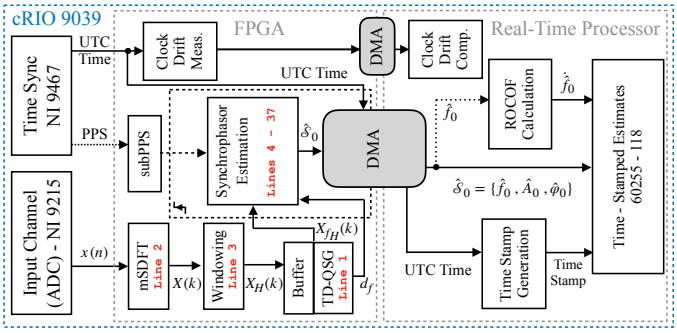


Fig. 2. Overall architecture of the proposed PMU prototype. The correspondence between processes and the pseudocode lines in Algortihm 2 are highlighted within each respective block.

allows for further expandability, only one is used in the current implementation. As in [18], [22], a free-running sampling process is adopted and synchronization to UTC is achieved later on. For synchronization with UTC, GPS technology is chosen due to its precision, stability, and low cost [22]. A stationary GPS unit with an uncertainty $3\sigma$ of $\pm$100 ns is used.

The different processes are distributed between the FPGA and the Real-Time processor, with the core functionality of the PMU being executed at the FPGA level. These main processes include the acquisition of the signal, the calculation of the DFT bins through the mSDFT and their buffering, the TD-QSG, the synchrophasor estimation, the time synchronization, and the measurement of the drift of the onboard clock. The remaining minor tasks are completed at the Real-Time processor. These include the compensation of the on-board clock drift, the estimation of the ROCOF as the finite backward derivative from consecutive frequency estimates, and the correction of the time-stamp by accounting for the maximum group delay introduced by the TD-QSG filter as well as that caused by the observation window. Data transfers between the FPGA and Real-Time processor are managed through dedicated Direct Memory Access (DMA) channels, which allow for the transfer of the estimated synchrophasors, the time-stamp and the clock drift. The complete PMU prototype is programmed using the LabVIEW programming language.

Reliable and deterministic behavior is sought by implementing the core functions at the FPGA level, as this ensures that all vital processes are executed according to its internal clock. This is critical, as the main design constraint is the measurement reporting latency, defined in [1] as the delay between the time of occurrence of an event in the power system and the moment in time of it being reported. The standard [1] limits the reporting latency of the PMU based on the performance class and the chosen reporting rate $F_r$. The P class represents the most demanding case, only allowing two reporting periods, which for a $F_r = 50$ fps translates into 40 ms. However, the computational margin is, in reality, much smaller as further delays are introduced in the estimation process both through the filtering and windowing stages. The TD-IpDFT [14] uses a Hanning observation window of three

TABLE I
FPGA COMPILATION RESULTS FOR EACH CONSTITUENT FUNCTION AND THE TOTAL IMPLEMENTATION.

| Flip-Flops | LUTs | DSP | Latency | Function | Config. / Lines |
|---|---|---|---|---|---|
| 6,731 | 7,085 | 0 | 3.2 $\mu s$ | mSDFT | 9 bin / 2 |
| 9,471 | 9,861 | 0 | 0.8 $\mu s$ | Wind. | 9 bin / 3 |
| 10,967 | 12,011 | 22 | 4.3 $\mu s$ | TD-QSG | - / 1 |
| 9,006 | 10,561 | 35 | 4.5 $\mu s$ | IpDFT | 3p / 4 |
| 29,736 | 53,764 | 424 | 6.9 $\mu s$ | TD-SR | - / 8 |
| 12,558 | 12,011 | 38 | 10.4 $\mu s$ | Int. Dect. | - / 10-16 |
| 46,741 | 71,916 | 529 | 1.19 $ms$ | OOBI Comp. | - / 17-37 |

| Total FPGA Allocation | | | | Latency | |
|---|---|---|---|---|---|
| Flip-Flops | LUTs | DSP | BRAM | no int. | OOBI |
| 78,126 | 103,217 | 654 | 61 | 42.6 $\mu s$ | 1.23 $ms$ |
| 19.2% | 50.6% | 77.9% | 13.7% | | |

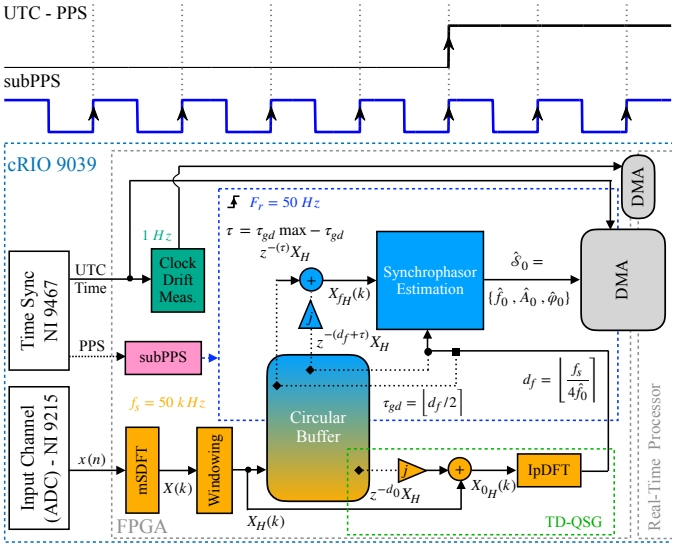


Fig. 3. Overview of the processes implemented at the FPGA level.

nominal cycles (60 ms) and a TD-QSG filter to generate the complex in-quadrature signal. The first introduces a delay equal to half of the window length ($\tau_w = 30$ ms), and the latter a maximum of 2.78 ms ($\tau_{g_{td}}$ max) for a 45 Hz input signal, leaving just 7.22 ms to perform all calculations.

One way to limit the computational cost is to adopt recursive techniques for the calculation of DFT bins, which, as indicated in [18] are generally more efficient when only a small number of bins are required. In this work, the mSDFT [21] is adopted, which represents an accurate and guaranteed stable recursive calculation technique. The implementation is the same as that proposed in [24] and also adopted in [18]. In particular, for the same required number of DFT bins and windowing function used in this work, the mSDFT was shown to be over 90 times faster in [18] compared to the use of a canonical DFT [25]. The selection of a recursive technique is also twofold since for the generation of the in-quadrature signal used by the TD-IpDFT algorithm a continuous stream and buffering of DFT bins are needed.

### A. FPGA Functions

An overview of the different processes running at the FPGA is shown in Fig. 3 and the compilation results for each step of the estimation, as well as those of the complete process, are summarized in Table I. The results indicate the number of flip-flops, LUTs, DSPs and the execution time of each stage as well as those of the total process. For the latter, the required BRAM is also indicated. Compilation is performed using the Vivado v2019.1.1_AR73110 64-bit compiler.

Signal samples acquired at a 50 kHz rate through the NI-9215 module are stored in a circular buffer so that the difference $x(n) - x(n-N)$ can be calculated for every new sample by accessing the right memory register. This is done by using a looping counter updated for each iteration. The sample difference is then fed into the mSDFT calculation block which is executed at the sampling rate and is based on the implementation described in [24] and used in [18]. The implementation exploits the N periodicity of $\mathcal{W}_N^{-nk} = e^{-j2\pi kn/N}$ and thus relies on the pre-calculation and pre-allocation of the real and imaginary components of $\mathcal{W}_N^{-nk}$ in dedicated memories. Then, based on the index $k$ of the calculated bin, the right memory registers are accessed. A detailed explanation of the technique can be found in [24]. The Hanning window is then applied in the frequency domain.

The resulting bins $X_H(k)$ are stored in another circular buffer again in terms of real and imaginary components, which is used to generate the delayed in-quadrature complex signal spectrum. To avoid coupling issues, the size of this buffer $D_f$ has been increased over its required theoretical dimension. This is done to avoid data overwriting as the buffer is a shared resource between two loops running at different rates, i.e., the mSDFT and TD-QSG loop running at $f_s$ and the synchrophasor estimation loop running at $F_r$. The execution of the SE loop is triggered by the rising edge of the 'subPPS' signal (Fig. 3), which is a square waveform generated at the desired reporting rate $F_r$ and in sync with UTC-PPS. The same trigger technique is used in [18], [24]. The SE process then uses the refined delay calculated by the TD-QSG to complete the estimate. As shown in Algorithm 2 the process is made of repetitive tasks that allow us to take advantage when distributing the FPGA resources, especially when the iterative correction is triggered.

The use of the TD-QSG filter introduces a group delay ($\tau_g(\omega)$) in all frequency components. Group delay is the time delay introduced to a frequency component $\omega = 2\pi f$ as it passes through the filter and is defined as [26]:

$$\tau_g(\omega) = -\frac{d\Theta(\omega)}{d\omega} \tag{11}$$

where $\Theta(\omega)$ denotes the phase response of the filter. In the case of filters with a linear phase response, as is the case of the TD-QSG, the group delay is constant, and all frequencies experience the same delay. For the TD-QSG, the group delay

equals half of the refined delay $d_f$. This can be easily proved by applying (11) to the phase response shown in Fig. 1. As it has a linear phase response, its derivative can be determined simply by looking at its slope. Thus:

$$\tau_{g_{td}} = -\frac{d\Theta_{td}(\omega)}{d\omega} = \frac{\pi}{4(2\pi f)} = \frac{1}{8f} \tag{12}$$

where $f$ here denotes, like in Fig. 1, the frequency considered to generate the delayed in-quadrature complex signal $\bar{y}(n)$ from $y(n)$. This is done by applying a delay $d_f$ calculated based on $f$:

$$d_f = \left\lfloor \frac{f_s}{4f} \right\rceil. \tag{13}$$

By omitting the round-to-the-nearest integer function and substituting (13) into (12), the following expression is obtained:

$$\tau_{g_{td}} = \frac{4d_f}{8f_s} = \frac{1}{2} d_f T_s \tag{14}$$

with the sampling time $T_s$ simply relating $d_f$, which is expressed in number of samples, with $\tau_{g_{td}}$ canonically expressed in units of time.

As shown in Algorithm 1, $d_f$ is adjusted based on an initial estimate of the frequency of the input signal, which in practice results in a variable delay. To ensure that a constant and consistent $F_r$ is maintained, the group delay introduced by the TD-QSG filter must be taken into account when selecting the bins from the circular buffer used to generate $X_{f_H}(k)$. This is done by delaying the bin selection by $\tau$, which is the difference between the maximum expected group delay ($\tau_{g_{td}}$ max) and the current group delay ($\tau_{g_{td}}$), i.e. $X_{f_H}(k) = z^{-(\tau)} X_H(k) + j z^{-(d_f+\tau)} X_H(k)$. This effectively guarantees that the same delay ($\tau_{g_{td}}$ max) is always applied with respect to the 'subPPS' rising edge.

The last process executed on the FPGA corresponds to the measurement of the clock drift. This is a phenomenon that might affect free-running clocks and refers to the deviation of the sampling frequency $f_s$ from its nominal value. This drift depends on many factors, such as ambient conditions, manufacturing characteristics, or even the quality of the device [22], and, if not taken into account, can severely degrade the estimates. However, by using an accurate time reference, such as the available GPS signal, this deviation can be corrected. The same procedure used in [18], [22] is adopted. Given two distant samples, $x(m)$ and $x(m-M)$, collected a few seconds apart, an ideal time interval equal to $MT_s$ should exist between them. By measuring the actual elapsed time $\Delta t_M(m) = t_m - t_{m-N}$ and calculating the normalized difference with the ideal interval $MT_s$, the updated frequency resolution ($\hat{\Delta}_f(m)$) can be determined as:

$$\hat{\Delta}_f(m) = \Delta_f \left(1 - \frac{\Delta t_M(m) - MT_s}{MT_s}\right) \tag{15}$$

The execution of this task is divided between the FPGA and the Real-Time processor. Measurement of $\Delta t_M(m)$ and calculation of $\Delta t_M(m) - MT_s$ is done in the FPGA, while the remaining calculations are completed in the real-time processor. An ideal interval equal to 1 s and corresponding to an execution rate of 1 Hz is selected and controlled by means of a counter of size $M$. The counter is updated with each incoming sample $x(n)$ and after each full turn, the time stamp of the latest sample $t_m$ is compared with that of the previous turn $t_{m-M}$. Lastly, the estimated synchrophasor and frequency $\hat{\mathcal{S}}_0$, the time-stamp of the latest sample, and the clock drift measurement are sent to the Real-Time processor through dedicated DMA channels written at their corresponding rate as shown in Fig. 3.

### B. Real-Time Functions

The remainder of the processes are completed in the real-time processor. These are the compensation of the on-board clock drift (as discussed above), the estimation of the ROCOF and the adjustment of the time-stamp. As indicated by (9), the ROCOF is calculated with a first-order backward approximation of a first-order derivative using two consecutive frequency estimates. The maximum group delay must be accounted for and used to correct the time-stamp of the obtained estimates, together with the half-window length delay introduced by the observation window. This is because the time-stamp in the FPGA is assigned to the latest sample within the window. These allow to ensure the synchronization to UTC of the free-running sampling process.

One difference from the implementations in [18], [22] is that they apply one additional correction to the estimated phase $\hat{\varphi}_0$ to account for the potential delay between the rising edge of the 'subPPS' signal and the time instant when the first sample of the window is actually captured. However, in this work, this delay is grouped with those introduced by the signal generation (PMU calibrator) and acquisition processes (PMU prototype) (see Section IV-A) and is used instead to correct the time-stamp in the postprocessing of the estimates at the error calculation stage.

## IV. Metrological Validation

This section presents: (i) the experimental setup used in the laboratory for the validation (Section IV-A) and (ii) the results that demonstrate the compliance of the PMU with the standard (Section IV-B).

### A. Laboratory Setup

The same PMU calibrator [17] previously used for the validation of [18] is also used to verify the compliance of the current PMU prototype with IEC/IEEE Std requirements [1]. In [17], the calibrator was experimentally shown to deliver reference synchropahsors characterized by a TVE with uncertainty levels of 0.00x% and 0.0x% respectively for static and dynamic conditions. It consists of a National Instruments PXI 1042Q chassis [27] equipped with: (i) a NI PXI-8110 controller, (ii) a NI PXI-6682 GPS timing and synchronization module, and (iii) a NI PXI-6289 data generation and acquisition board. Here, a variant of the experimental setup used in [18] is adopted and summarized in Fig. 4. As shown in Fig.

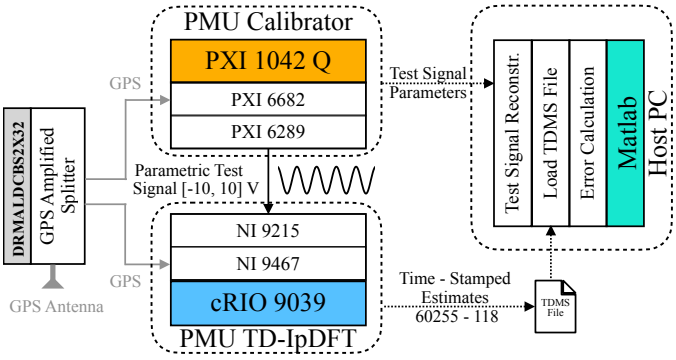


Fig. 4. Experimental setup used for the metrological validation of the PMU.

TABLE II
TD-IPDFT PARAMETERS

| Parameter | Variable | Value |
|---|---|---|
| Nominal System Frequency | $f_n$ | 50 Hz |
| Window Type | - | Hann |
| Window Length | $T$ | 60 ms ($3/f_n$) |
| Sampling Rate | $f_s$ | 50 kHz |
| PMU Reporting Rate | $F_r$ | 50 fps |
| mSDFT bins | $K+1$ | 9 |
| DFT bins | $K$ | 8 |
| Max Number of Iterations | $Q$ | 36 |
| IpDFT Interpolation Points | - | 3 |
| Lower Spectral Energy Threshold | $\lambda_o^l$ | $4.6 \cdot 10^{-4}$ |
| Upper Spectral Energy Threshold | $\lambda_o^u$ | $2.3 \cdot 10^{-3}$ |
| Spectral Energy Concentration Threshold | $\lambda_i$ | 0.765 |
| Residual Energy Variation Threshold | $\lambda_{R_e}$ | $6.9 \cdot 10^{-11}$ |

4, the calibrator operating in an open loop generates a user-defined parametric test waveform among an available library of the different IEC/IEEE Std tests [1] and generates a data stream using a sampling rate of 500 kHz and with a maximum amplitude of 10 V.

The test signals are then transmitted over a negligible-length coaxial cable and acquired by the NI 9215 module of the PMU prototype under evaluation. These are then processed by the PMU according to the processes described in Section III and used to generate the time-stamped estimates. Finally, they are written to a distinctive TDMS file and sent over to a host PC for post-processing. Similarly, all the information required to fully characterize the user-generated test signal, including its initial time-stamp, is also sent to the host PC for reconstruction. Thus, the reference parameters can be derived at the same instants reported by the PMU and used to assess the total estimation errors.

Time synchronization is performed for both the PMU prototype and the calibrator via the time information provided by a stationary GPS unit with an uncertainty $3\sigma$ of $\pm 100$ ns. The time synchronization signal coming from the antenna is routed with a GPS splitter [28] and then fed to each device. This information is crucial for assessing the PMU reporting latency. In this work, the measurement latency is evaluated as the cumulative contributions of the delays introduced in the estimation process, i.e. from the data acquisition stage to the moment the estimates are ready to be written to the TDMS file. Since the delays introduced by the observation window and the filter are known, and those of the FPGA processes have been characterized in Section III, the remaining part is caused by the real-time processor. The results of the latency assessment, together with all other validation results, are reported in Section IV-B.

Lastly, it is important to note that the processes of generating and acquiring the test signal through the NI PXI-6289 board and the NI 9215 module, respectively, result in the introduction of distortions both in magnitude and in time delay that must be characterized and compensated for. Thus, an initial test is run over an almost four and a half-hour period and used to calibrate the equipment and characterize the total magnitude distortion and time delay introduced by the generation and acquisition modules. For this calibration test, a simple undisturbed steady-state signal at nominal frequency is generated so that the resulting estimates can be used to determine the correction parameters. The values resulting from this single calibration test are then subsequently used to correct the estimates in all validation assessments conducted. This characterization is a common practice with calibration equipment, which is affected by ambient conditions as discussed in [29].

### B. Validation Results & Discussion

Compliance with the IEC/IEEE standard [1] has been evaluated by validation tests with a duration of just below 10 s through the setup shown in Fig. 4. For all tests, the PMU prototype is configured according to the parameters provided in Table II, which are taken directly from [14] with the exception of $\lambda_o^l$ and $\lambda_o^u$ for which an enhanced parameterization, presented in the Appendix, is used. The interested reader is referred to [14] for a detailed explanation and justification on their selection. Throughout the test and according to the selected $F_r$, the PMU estimates are recorded and compared to the reconstructed reference signal to derive the total vector error (TVE), frequency error (FE) and ROCOF error (RFE) as well as the response ($R_t$), delay times ($D_t$) and overshoots (OS) for the step tests. The maximum registered values in the different tests are summarized in Table III and Table IV. Furthermore, the limits set in [1] for the P and M classes are also indicated. The following tests have been conducted:

- *Signal frequency range test (SF):* Three tests are conducted that consider fundamental frequencies equal to 45 Hz, 50 Hz, and 55 Hz.
- *Harmonic distortion test (HD)*: Three tests have been analyzed, each considering one of the closest and thus most severe harmonic tones ($2^{\text{nd}}$, $3^{\text{rd}}$, and $4^{\text{th}}$). Each test is carried out at two different distortion levels equal to 1% and 10%, and all are evaluated for a signal with a fundamental frequency equal to 51 Hz to avoid synchronous sampling.

TABLE III
MAXIMUM TVE, FE AND RFE OBTAINED BY THE PMU PROTOTYPE AND MAXIMUM LIMITS ALLOWED BY [1].

| | TVE [%] | | FE [mHz] | | RFE [Hz/s] | |
|---|---|---|---|---|---|---|
| | P / M | PMU | P / M | PMU | P / M | PMU |
| SF | 1 / 1 | 1.2E-3 | 5 / 5 | 8.0E-2 | 0.4 / 0.1 | 5.7E-3 |
| HD 1% | 1 / 1 | 1.2E-3 | 5 / 25 | 6.5E-2 | 0.4 / - | 6.1E-3 |
| HD 10% | 1 / 1 | 2.7E-3 | 5 / 25 | 7.6E-2 | 0.4 / - | 6.5E-3 |
| OOBI 4%[a] | - / 1.3 | 3.0E-3 | - / 10 | 3.5E-1 | - / - | 2.9E-2 |
| OOBI 10% | - / 1.3 | 4.0E-3 | - / 10 | 4.3E-1 | - / - | 3.6E-2 |
| AM 2Hz | 3 / 3 | 1.1E-1 | 60 / 300 | 6.9E-2 | 2.3 / 14 | 5.5E-3 |
| AM 5Hz | 3 / 3 | 6.5E-1 | 60 / 300 | 8.0E-2 | - / 14 | 6.9E-3 |
| PM 2Hz | 3 / 3 | 9.8E-2 | 60 / 300 | 1.3E+0 | 2.3 / 14 | 3.2E-1 |
| PM 5Hz | 3 / 3 | 5.8E-1 | 60 / 300 | 1.9E+1 | - / 14 | 4.8E+0 |
| FR | 1 / 1 | 4.0E-2 | 10 / 10 | 8.3E-2 | 0.4 / 0.2 | 7.0E-3 |

[a] Maximum limit values taken from [1] for the 10% case.

TABLE IV
MAXIMUM $R_t$, $D_t$ AND OS OBTAINED BY THE PMU PROTOTYPE AND MAXIMUM LIMITS ALLOWED BY [1].

| | TVE $R_t$ [ms] | | FE $R_t$ [ms] | | RFE $R_t$ [ms] | |
|---|---|---|---|---|---|---|
| | P / M | PMU | P / M | PMU | P / M | PMU |
| AS | 40 / 140 | 28.8 | 90 / 280 | 47.5 | 120 / 280 | 73.0 |
| PS | 40 / 140 | 35.1 | 90 / 280 | 52.9 | 120 / 280 | 75.9 |
| | $D_t$ [ms] | | OS [%] | | | |
| | P / M | PMU | P / M | PMU | | |
| AS | 5 / 5 | 1.6 | 5 / 10 | 7.6E-3 | | |
| PS | 5 / 5 | 1.6 | 5 / 10 | 5.2E-3 | | |

[a] Maximum limit values taken from [1] for the 10% case.

- *Out-of-Band Interference test (OOBI)*: Two tests covering the cases with the closest spectral proximity within the subharmonic and interharmonic range are evaluated. These correspond to the frequency pairs of fundamental and interference tones of $f_0 = 47.5$ Hz – $f_i = 25$ Hz and $f_0 = 52.5$ Hz – $f_i = 75$ Hz. Both tests are evaluated for a 4% and 10% OOBI level compared to the main tone.
- *Amplitude and phase modulation tests (AM and PM)*: Two tests are carried out considering modulating frequencies of 2Hz and 5Hz for amplitudes of 10% (AM) and $\pi/18$ (PM). Modulating frequencies of 2Hz and 5Hz are selected because they are the maximum, respectively, for the P and M class.
- *Frequency ramp test (FR)*: Two tests have been considered for ramp rates equal to $\pm 1$ Hz/s.
- *Amplitude and phase step tests (AS and PS)*: Two sets of 10 tests are performed, respectively, for AS and PS with corresponding magnitudes of 0.1 p.u. and $\pi/18$ rad. $R_t$, $D_t$ and OS are evaluated using the shifted repeated signal method (SRS) [30], where the location of the step is shifted relative to the reporting time by $\Delta t = n/(10F_r)$ for $n = 0...9$. The initial phase of the signal is also appropriately adjusted by $2\pi f_0 \Delta t$ so that each test is performed on the exact translation of the signal. The estimates are shifted back and interleaved to construct a higher resolution response curve, as shown in Fig. 5.

Test selection is justified based on the results obtained in [14]. As shown in Table III, the results obtained fully meet the combined requirements of the standard classes P and M. The ability of the method to detect and remove the effects of interfering tones equal to or greater than the 4% range within the OOBI range is a prominent feature of the TD-IpDFT algorithm. This is a characteristic that goes beyond the requirements of [1], since no performance limits are imposed for interfering signals with a magnitude other than 10% of the fundamental. Thus, the same values are used as a reference for the 4% case. These allow consolidating and further validating the results of [14]. The AS and PS tests shown in Fig. 5 are represented in terms of TVE, FE, and RFE as functions of their $R_t$, i.e., the time is centered at the instant when the error first exceeds the most demanding standard requirement, respectively. Equivalently, for $D_t$, the reconstructed amplitude and phase are centered at the instant the step occurs.

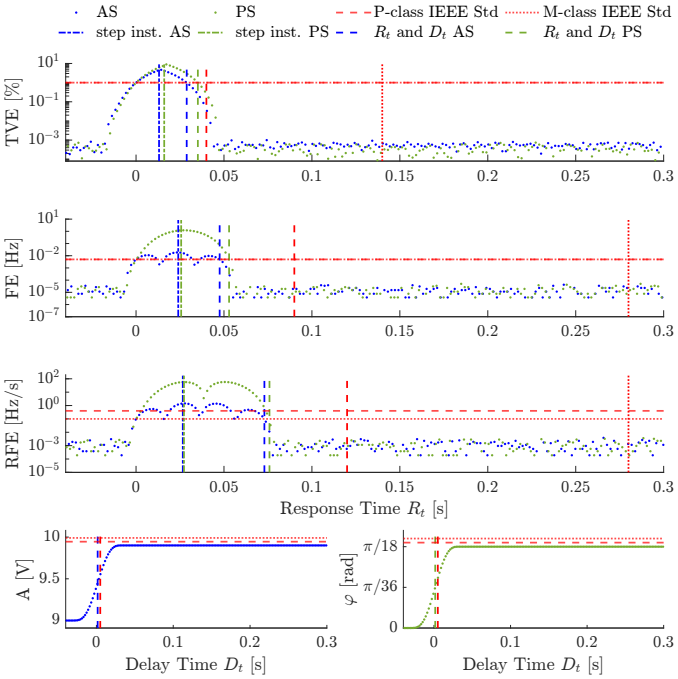


Fig. 5. Step tests: AS of 10% (blue) and PS of $\pi/18$ (green) [1].

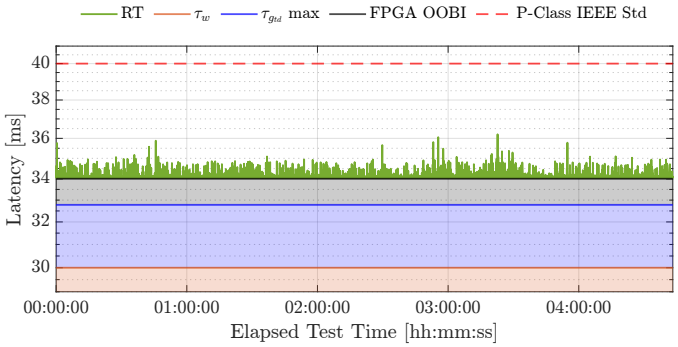


Fig. 6. PMU Measurement Reporting Latency Evaluation.

Finally, the last aspect to be evaluated is the method measurement reporting latency. To assess this, an over four

and a half-hour duration test is conducted under the most challenging conditions, i.e., forcing the activation of the iterative correction mechanism to compensate for an interfering tone and forcing the maximum number of iterations. The test results are summarized in Fig. 6. The figure depicts the total aggregated reporting latency obtained discriminated by source, i.e. windowing ($\tau_w$), TD-QSG group delay ($\tau_{g_{td}}$ max), FGPA (OOBI), and real-time processor (RT). It can be seen how the method can meet the most demanding conditions set in [14] for the P class, demonstrating the suitability of the technique for SE.

## V. Conclusions

This paper has presented the design and experimental validation of a dual IEC/IEEE Std 60255-118-1-2018 P- and M- class compliant PMU prototype. For SE the TD-IpDFT technique has been adopted since it allows to mitigate the self-interference of the fundamental tone and provides the device with resilience against interference tones within the OOBI range greater than or equal to 4% of that of the fundamental tone. The technique has been adapted and deployed to a NI CompactRIO-9039 platform in a single-channel configuration with a 50-fps report rate, 50 kHz sampling rate, and three nominal cycle observation windows, requiring a total use of 19.2% flip-flops, 50.6% LUTs, 77.9% DSPs and 13.7% BRAM.

A comprehensive metrological characterization of the prototype was conducted by means of a PMU calibrator capable of accurately re-producing the test signals required by the IEC/IEEE Std. 60255-118-1-2018 standard. Results have shown that the PMU can indeed meet the combined requirements of both the P and M classes even for interfering tones equal to or greater than the 4% range within the OOBI range. Finally, the latency assessment returned a worst-case measurement reporting latency of 36.21 ms for the case when an interfering tone is detected, well within the most demanding 40 ms requirement of the P class.

## References


[1] "IEEE/IEC International Standard - Measuring relays and protection equipment - Part 118-1: Synchrophasor for power systems - Measurements," *IEC/IEEE 60255-118-1:2018*, pp. 1–78, 2018.

[2] A. Testa, M. F. Akram, R. Burch, *et al.*, "Interharmonics: Theory and modeling," *IEEE Transactions on Power Delivery*, vol. 22, no. 4, pp. 2335–2348, 2007.

[3] A. Testa and R. Langella, "Power system subharmonics," in *IEEE Power Engineering Society General Meeting, 2005*, 2005, 2237–2242 Vol. 3.

[4] H. Soltani, P. Davari, F. Zare, P. C. Loh, and F. Blaabjerg, "Characterization of input current interharmonics in adjustable speed drives," *IEEE Transactions on Power Electronics*, vol. 32, no. 11, pp. 8632–8643, 2017.

[5] V. Ravindran, S. K. Rönnberg, T. Busatto, and M. H. J. Bollen, "Inspection of interharmonic emissions from a grid-tied pv inverter in north sweden," in *2018 18th International Conference on Harmonics and Quality of Power (ICHQP)*, 2018, pp. 1–6.

[6] R. Langella, A. Testa, J. Meyer, F. Möller, R. Stiegler, and S. Z. Djokic, "Experimental-based evaluation of pv inverter harmonic and interharmonic distortion due to different operating conditions," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 10, pp. 2221–2233, 2016.

[7] A. Sangwongwanich, Y. Yang, D. Sera, H. Soltani, and F. Blaabjerg, "Analysis and modeling of interharmonics from grid-connected photovoltaic systems," *IEEE Transactions on Power Electronics*, vol. 33, no. 10, pp. 8353–8364, 2018.

[8] K. Chauhan and R. Sodhi, "Placement of distribution-level phasor measurements for topological observability and monitoring of active distribution networks," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 3451–3460, 2020.

[9] M. Picallo, A. Anta, and B. de Schutter, "Comparison of bounds for optimal pmu placement for state estimation in distribution grids," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4837–4846, 2019.

[10] A. von Meier, "Synchrophasor monitoring for distribution systems: Technical foundations and applications," NASPI-2018-TR-001, Tech. Rep., 2018.

[11] R. Gupta, A. Zecchino, J.-H. Yi, and M. Paolone, "Reliable dispatch of active distribution networks via a two-layer grid-aware model predictive control: Theory and experimental validation," *IEEE Open Access Journal of Power and Energy*, vol. 9, pp. 465–478, 2022.

[12] A. Derviškadić, P. Romano, and M. Paolone, "Iterative-interpolated dft for synchrophasor estimation: A single algorithm for p-and m-class compliant pmus," *IEEE Trans. Instrum. Meas.*, vol. 67, 3 2018, ISSN: 00189456.

[13] J. Song, A. Mingotti, J. Zhang, L. Peretto, and H. Wen, "Fast iterative-interpolated dft phasor estimator considering out-of-band interference," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.

[14] C. García-Veloso, M. Paolone, and J. M. Maza-Ortega, "Computationally efficient synchrophasor estimation: Delayed in-quadrature interpolated dft," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–16, 2024.

[15] C. García-Veloso, M. Paolone, and J. M. Maza-Ortega, "Attenuated self-interference synchrophasor estimation: The sogi interpolated dft," in *2023 IEEE Belgrade PowerTech*, 2023, pp. 1–9.

[16] National Instruments, "CompactRIO cRIO-9039 Specifications," [Online]. Available: https://www.ni.com/docs/en-US/bundle/crio-9039-specs/page/specs.html (visited on 09/21/2023).

[17] G. Frigo, A. Derviškadić, D. Colangelo, J. P. Braun, and M. Paolone, "Characterization of uncertainty contributions in a high-accuracy pmu validation system," *Measurement*, vol. 146, pp. 72–86, Nov. 2019.

[18] A. Derviškadić and M. Paolone, "Design and experimental validation of an fpga-based pmu simultaneously compliant with p and m performance classes," *Electr. Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106650.

[19] D. Agrež, "Weighted multipoint interpolated dft to improve amplitude estimation of multifrequency signal," *IEEE Trans. Instrum. Meas.*, vol. 51, 2 2002, ISSN: 00189456.

[20] T. Grandke, "Interpolation algorithms for discrete fourier transforms of weighted signals," *IEEE Trans. Instrum. Meas.*, vol. 32, 2 1983, ISSN: 15579662.

[21] K. Duda, "Accurate, guaranteed stable, sliding discrete fourier transform [dsp tips & tricks]," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 124–127, 2010.

[22] P. Romano and M. Paolone, "Enhanced interpolated-dft for synchrophasor estimation in fpgas: Theory, implementation, and validation of a pmu prototype," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 12, pp. 2824–2836, 2014.

[23] National Instruments, "NI 9215 C Series Voltage Input Module," [Online]. Available: https://www.ni.com/docs/en-US/bundle/ni-9215-specs/page/specs.html (visited on 09/21/2023).

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

[25] A. Derviškadić, "Synchronized sensing for wide-area situational awareness of electrical grids in non-stationary operating conditions," Ph.D. dissertation, EPFL, 2019.

[26] J. G. Proakis and D. K. Manolakis, *Digital signal processing*, eng, Fourth edition, Pearson new international edition. Harlow, England: Pearson, 2014, ISBN: 1-292-03816-0.

[27] National Instruments, "NI PXI-1042Q chassis," [Online]. Available: https://www.ni.com/docs/en-US/bundle/pxi-1042-1042q-seri/resource/371088a.pdf (visited on 09/21/2023).

[28] GPS Networking, "DRMALDCBS2X32 Dual Antenna Rack Mount Amplified 2X32 GPS Splitter," [Online]. Available: https://www.gpsnetworking.com/system/datasheets/148/original/DRMALDCBS2X32-2018.pdf?1610392915 (visited on 09/21/2023).

[29] D. Colangelo, L. Zanni, M. Pignati, *et al.*, "Architecture and characterization of a calibrator for pmus operating in power distribution systems," in *2015 IEEE Eindhoven PowerTech*, 2015, pp. 1–6.

[30] P. Castello, C. Muscas, P. Attilio Pegoraro, and S. Sulis, "Critical analysis of pmu testing procedures for step response evaluation," in *2015 IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, 2015, pp. 37–42.


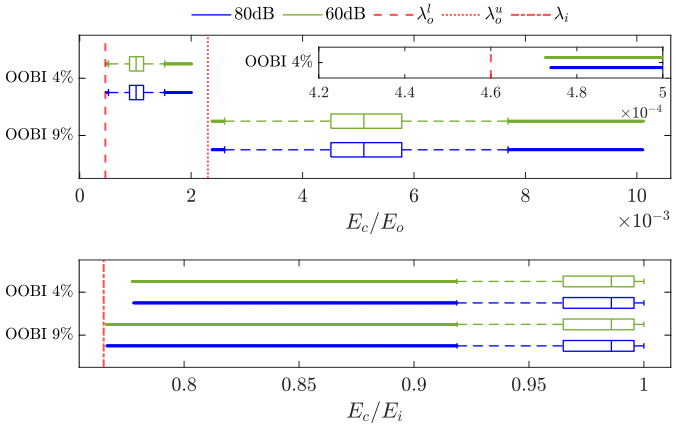


Fig. 7. Boxplot representation of $E_c/E_o$ (top) and $E_c/E_i$ (bottom).

## APPENDIX
## PROPOSED REFINEMENT OF OOBI THRESHOLDS

After a closer examination of the OOBI thresholds proposed in [14] by using a MATLAB simulated environment with an increased reporting rate of 0.2 ms ($F_r = 5000$), some outliers were found where the suggested value for $\lambda_o^l$ would not ensure correct detection of 4% OOBIs. Thus, the analysis conducted in [14] to set appropriate values for $\lambda_o^l$, $\lambda_o^u$ and $\lambda_i$, which allow the identification and correction of OOBIs below the limit of 10% set by [1], is here revisited and improved.

To enhance the previous analysis conducted, each test is now carried out 101 times (instead of 20) considering different initial phase angles for the generation of the reference signal evenly distributed between 0 and $2\pi$. At the same time, an increased reporting rate equal to 0.2 ms (instead of 20 ms) is used to examine a wider range of analysis windows. All tests are performed in a MATLAB simulated environment, considering additive white Gaussian noise with a signal-to-noise ratio equal to 60 and 80 dB. Although the analysis in [14] considered all standard tests, this revision is limited to the two OOBI levels used to define the $\lambda_o^l$, $\lambda_o^u$ and $\lambda_i$ thresholds, i.e. the 4% and 9% OOBI. A detailed examination of all remaining standard test conditions under the currently revised testing conditions and their potential impact on the selected thresholds is left for future work. However, it is important to remember that in [14] all test conditions defined in [1] were superimposed with a 10% amplitude modulation, with the exception of the OOBI tests and the AM itself, to derive more robust thresholds. Thus, no impact is found under normal standard conditions with the current selection.

Fig. 7 shows the statistical distributions of $E_c/E_o$ and $E_c/E_i$ using a boxplot representation. The results show that, to correctly identify all potential 4% OOBIs, a value of $\lambda_o^l$ of $4.6 \cdot 10^{-4}$ (instead of $4.9 \cdot 10^{-4}$ as suggested in [14]) is necessary. Similarly, a reduction of $\lambda_o^u$ from $2.4 \cdot 10^{-3}$ to $2.3 \cdot 10^{-3}$ is suggested given the current results. The compliance and adequacy of $\lambda_i$ is finally verified.