Integration of an IEEE Std. C37.118 Compliant PMU into a Real-Time Simulator

Paolo Romano, Marco Pignati and Mario Paolone
École Polythecnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
paolo.romano@epfl.ch, marco.pignati@epfl.ch, mario.paolone@epfl.ch

Abstract—**The simulation of Phasor Measurement Units (PMUs) into real-time simulators (RTSs) is typically limited by the complexity of the synchrophasor estimation (SE) algorithm. This is especially true when dealing with distribution network PMUs due to the more demanding accuracy requirement and, for the case of class-P PMUs, for the limited latency. In this respect, if the SE algorithm is too simplistic, the performances of the simulated PMU might not match the specific application needs. On the other hand, when higher precision of the synchrophasors estimations is required, an increased computational complexity of the SE algorithm is obtained and, as a consequence, few devices can be simulated into a RTSs at the same time. The work presented in this paper illustrates the design and the deployment of a C37.118 class-P compliant PMU into the Opal-RT RTS. The RTS-deployed PMU has demonstrated to match the requirements of both transmission and distribution networks. The simulated PMU has been experimentally validated and demonstrated to be well suited for its integration into any RTS.**

Index Terms—**Phasor Measurement Units, synchrophasors, real-time simulation, enhanced-Interpolated Modulated Sliding Discrete Fourier Transform, IEEE Std. C37.118.**

I. Introduction

According to the IEEE Std. C37.118.1-2011 [1], Phasor Measurement Units (PMUs) are measurement devices capable of providing UTC time-tagged estimation of frequency, Rate-of-change-of-frequency (ROCOF), voltage/current phasors and streaming them to a concentration point. The high reporting rates and accuracies of the streamed information make these devices a gold standard for transmission and distribution network operators aiming at performing advanced real-time monitoring, control and protection functionalities (e.g., [2-4]).

The validation of PMU-based real-time monitoring, protection and control processes is typically performed by means of Hardware-in-the-loop (HIL) setups. These usually adopt Real-Time Simulators (RTSs) to reproduce the behavior of a given electrical network and test the functionality under study in steady-state, dynamic and, also, faulted conditions. The targeted RTS might be interfaced with real PMUs through the low/high power signals. Then, PMUs stream data over a Local Access Network (LAN) to a Phasor Data Concentrator (PDC).

The connection of real PMU devices to the RTS might presents technical and economical drawbacks typically represented by: (i) the limited number of available analog outputs to interface large number of PMUs (e.g., several tens/hundreds) and (ii) the potential high cost of such a kind of devices. To overcome these obstacles, PMUs might be virtualized and directly embedded inside the model running into the RTS. This enables the usage of a number of PMUs limited only by the computational resources of the RTS with respect to the selected integration time-step and the model complexity.

Despite the evident advantages of such an approach, its application is limited by the computational resources typically required by the RT simulation of the majority of Synchrophasor Estimation (SE) algorithms. In this respect, in the current literature, few works explored the possibility of virtualizing PMUs in RTSs [5-8]. The majority of them focused on the development of test platforms for PMU-based wide-area monitoring and control applications [6,8]. However, to the best of the Authors' knowledge, none of them has deployed into a RTS a virtual PMU characterized by a SE algorithm that has demonstrated its compliance to the requirements imposed by the IEEE Std. C37.118.1-2011 [1] and its latest amendment IEEE Std. C37.118.1a-2014 [9]. In this respect, this work presents (i) the formulation, (ii) the deployment and (iii) the metrological characterization of the so-called e-IpMSDFT SE algorithm into the Opal-RT eMEGAsim RTS. The algorithm has been first formulated by the Authors in [10] and further extended in [11] to reduce the measurement reporting latencies and, consequently, increase the reporting rates. As discussed in [10], the algorithm, and the associated PMU prototype, satisfies every requirement defined in [1] for class-P PMUs and the majority of class-M requirements (with the exception of the Out-Of-Band Interference tests). Moreover, the PMU prototype has been also validated in real electrical grids (i.e., distribution and sub-transmission networks) and it is currently installed in several pilot projects in Europe [12, 13].

The remainder of the paper is structured as follows: Section II presents the theoretical background needed to define the e-IpMSDFT SE algorithm. Section III describes its implementation in the Opal-RT eMEGAsim RTS. Section IV summarizes the experimental validation of the presented solution together with the assessment of its computational requirements with respect to the adopted RTS platform. Section V concludes the paper with the final remarks and future work.

II. THEORETICAL BACKGROUND

The majority of SE algorithms are based on the use of the Discrete Fourier Transform (DFT) and the SE algorithm presented here belongs to this category too. Based on the specific SE technique, different DFT bins are required to be computed. In general, if this number is relatively small, the so-called "sliding" DFT algorithms (see for instance [14] and [15]) are preferred due to the relatively low number of operations needed to update a single DFT bin (see [16] for an extensive analysis of the computational advantages and limits of this category of algorithms).

Based on the above considerations, this Section provides the theoretical background to understand the e-IpMSDFT SE algorithm. The first part presents the Modulated Sliding DFT (MSDFT) algorithm adapted to the case of SE; the second part couples the MSDFT method with the enhanced-Interpolated DFT (e-IpDFT) approach presented by the Authors in [10].

A. MSDFT for Synchrophasor Estimation

Let consider a generic power system quantity (branch/nodal current or node voltage) modeled as a signal characterized by a main tone fluctuation around the rated frequency of the system f_0 (i.e., 50 or 60 Hz). The signal is sampled by the PMU each $T_s = 1/F_s$ (being F_s the PMU sampling rate), and collected over the sliding window of length M sufficiently short so that the signal can be assumed stationary within it:

$$x(m) = A\cos(2\pi f m T_s + \varphi), \quad m \in [0, M-1] \tag{1}$$

where A, f and φ are the amplitude, frequency and phase of the main tone of the spectrum that are supposed to be constant over the observation interval of length M .

At every time-step n , the DFT can be calculated based on the most recent set of samples $\{x(n-M+1), x(n-M+2), \ldots, x(n)\}$:

$$X_k(n) = \sum_{m=0}^{M-1} x(q+m) \cdot W_M^{-km} \tag{2}$$

being k the DFT-bin index, $q = n - M + 1$ and $W_M^{-km} = e^{-j2\pi km/M}$ the DFT complex twiddle factor. As it can be noticed, (2) is a computationally-inefficient way to update the DFT spectrum since it requires, for every new sample, the re-processing of *M-1* older ones.

In order to increase the PMU throughput by reducing the number of operations to update a single DFT bin, the required portion of the DFT spectrum can be calculated via recursive algorithms (e.g., [15]). Despite this evident advantage with respect to the class of non-recursive DFT algorithms, the two categories do not generally have identical performances. In particular, as demonstrated in [15], the well-known Sliding-DFT algorithm suffers from accumulated errors due to numerical rounding. Reference [17] shows that, for $k = 0$, the calculation of the bin does not involve the complex twiddle factor and is, therefore, stable. Taking advantage of this peculiarity, and by making use of the so-called Fourier modulation property [18], the generic k-th DFT-bin can always be shifted to the position $k = 0$ by the complex twiddle factor W_M^{-km} :

$$X_0(n) = X_0(n-1) + W_M^{-km} \cdot (-x(n-M) + x(n)) \tag{3}$$

where (3) is obtained thanks to the intrinsic periodicity of the modulating sequence W_M^{-km} .

The twiddle factor modulation only introduces a phase shift that is changing with the index m : it is equal to zero for $m = 0$, it increases by the $\angle W_M^{-k}$ factor at each iteration and is periodically reset to 0 every M samples. In view of the above, the k-th bin of the DFT can be derived from equation (3) as:

$$X_k(n) = W_M^{k(m+1)} \cdot X_0(n) \tag{4}$$

where $W_M^{k(m+1)}$ compensates for the phase-shift due to the modulating sequence.

Equations (3) and (4) define the MSDFT method for the update of the value of a single bin of the entire DFT spectrum.

B. Enhanced-IpMSDFT Algorithm

IpDFT algorithms apply specific windowing functions and DFT interpolation schemes to reduce the spectral leakage effects due to power-system dynamics. During power-system dynamics, the spectral leakage effects decrease the accuracy levels of the majority of DFT-based synchrophasor estimation algorithms well above the IEEE Standard C37.118.1-2011 limits [19]. IpDFT algorithms try to reduce this bias by sequentially applying specific windowing functions and DFT interpolation schemes. However, these algorithms still suffer from errors when the frequency of the signal drifts from the rated one. Indeed, every IpDFT algorithm is based on the assumption that $F_s \gg f_0$. As a consequence, the positive and negative images of the main tone are typically very close in the DFT spectrum and the tails of their envelopes might eventually corrupt the neighboring image. In this respect, the enhanced-IpDFT (e-IpDFT) scheme presented in [10] extends the classical IpDFT approach by compensating the spectral interference produced by the negative image of the spectrum. It can be described in terms of the following simplified algorithm exposed in a verbose way:

```
Procedure e-IpDFT ($x(m), m \in [0, M-1]$)
    apply Hanning window
    compute DFT
    estimate signal parameters via 2-points IpDFT
    estimate spectral interference
    compensate for spectral interference
    estimate signal parameters via 2-points IpDFT
return signal parameters
```

In [10] it has been shown that, by adopting a time window T containing 3 periods of a signal at the rated power system frequency (i.e., 50 or 60 Hz) and sampling rate F_s of some tens of kHz, the IpDFT algorithms performs well under most of the conditions dictated by [1]. In particular, the developed PMU prototype based on the e-IpDFT is capable of passing every class P test defined by [1] and every class M test with exception of the out-of-band ones. Additionally, the Total Vector Error (TVE) obtained by this PMU prototype in steady state is sufficiently low to make this device suitable for distribution systems applications.

The measurement reporting latencies and achievable reporting rates of the e-IpDFT algorithm are mainly limited by the time needed to compute the relevant portion of the DFT spectrum. In particular, the e-IpDFT algorithm only needs to compute the 3 DFT bins associated to indices $k_{\max}+\{-1,0,1\}$, where $k_{\max}\in\mathbb{N}$ is the index of the DFT maximum that is fixed for a typical PMU operating conditions and equal $k_{\max}=\lfloor f_0 M/F_s \rceil$ (being $\lfloor\,\rceil$ the nearest integer function).

In this respect, the MSDFT seems to fit well in the e-IpDFT scheme that might benefit from its fast refresh rates without disrupting the previously defined procedure. The only operation that needs careful consideration is the signal windowing (i.e. step 2 of the e-IpDFT procedure), since its application in time-domain would compromise the whole MSDFT formation. As a consequence, the signal windowing needs to be moved after the MSDFT update and replaced by a frequency-domain convolution that results in a linear combination of adjacent $X_k(n)$ values. In the particular case of the Hanning window, the windowed k-th bin can be computed as:

$$X_k(n) = -0.25\cdot X_{k-1}(n) + 0.5\cdot X_k(n) - 0.25\cdot X_{k+1}(n) \qquad (5)$$

From Equation (5), it is clear that, in order to compute 3 windowed DFT bins, we need to compute 5 MSDFT bins, namely those associated to indices $k_{\max}+\{-2,-1,0,1,2\}$. Therefore, the MSDFT does not modify the precision of the e-IpDFT algorithm but only improve its measurement reporting latencies and achievable reporting rates (see [11] for further details).

III. e-IpMSDFT Deployment into the eMEGAsim Real Time Simulator

To correctly simulate a PMU into a RTS, the following set of minimal functionalities needs to be included: (a) a RT implementation of the adopted SE algorithm; (b) the synchronization to a UTC traceable time-source (like the one provided by the Global Positioning System); (c) a data streaming module that encapsulate and streams the PMU data according to one of the available Standards (for instance IEEE C37.118.2-2011, IEC-61850 etc.).

In order to test the proposed PMU design based on the e-IpMSDFT algorithm, we have adopted the Opal-RT eMEGAsim PowerGrid Real-Time Digital Simulator [20]. Such a RT platform consists of a multi-core processor hardware able to perform operations within an integration time-step generally of some tens of microseconds. The setup used for the development is shown in Figure 1 and it consists of one industrial PC (12 cores), a Spartan3 FPGA board and a Dolphin DXE410 Expansion Chassis. The RT simulated model runs in the industrial PC, the FPGA is used to lock the integration time-step to a more stable clock. The DXE410 module enables the communication between the two elements. In addition to this, in order to have access to a stable and reliable UTC-time reference needed to simulate any PMU, the industrial PC has been equipped with a hardware GPS-synchronization module from Spectracom (Tsync-PCIe express board [21]). This board is used to (i) provide the UTC timestamp for PMU data-frames (ii) provide the Pulse-per-Second (PPS) to a clock adapter in order to generate a GPS synchronized clock (see Section III.B for further details).

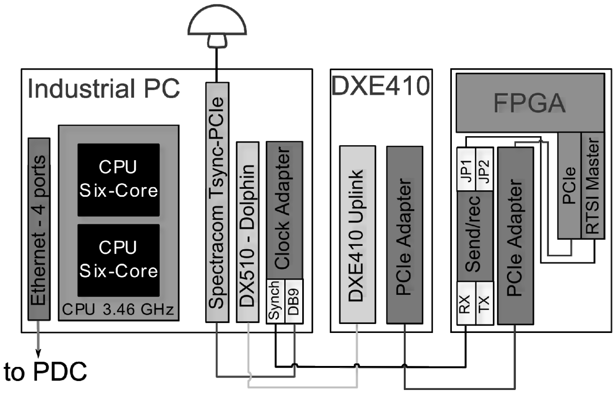

Figure 1: Setup of the eMEGAsim PowerGrid RTS.

What follows analyzes in details the aspects related to deployment of the e-IpMSDFT algorithm into the adopted RTS and the synchronization of its estimations to a traceable UTC-time reference.

A. Simulated-PMU design

The e-IpMSDFT-based PMU has been developed starting from the block scheme given in Figure 2 where the macro-functionalities of a 1-channel simulated PMU are presented (indeed, the derivation of a *n*-channel PMU is straightforward by simply replicating *n* times the depicted block-scheme).

The inputs of the PMU model are essentially two: (i) a digital signal representing either a voltage or current waveform sampled by the simulation integration time-step T_s; (ii) the UTC timestamp, provided, in our case, by the GPS.

At every integration time-step, the input signal is processed by the MSDFT algorithm to compute the 5 DFT bins surrounding the DFT maximum. The MSDFT outputs are given to the e-IpDFT algorithm that updates the estimation of the signal parameters (namely its frequency, ROCOF, amplitude and phase), based on the technique explained in Section II.B, for every new acquired sample. These values are then used, together with the UTC timestamp, to build up the C37.118 data frame that is then streamed out of the Opal-RT RTS and reported to an external PDC at specific reporting times defined by the PMU nominal frequency f_0 and reporting rate F_r. In this respect, the C37.118 data encapsulation library built-in the Opal-RT RTS has been used to encapsulate and stream the time-stamped estimated quantities in compliance with the most recent version of the standard [22].

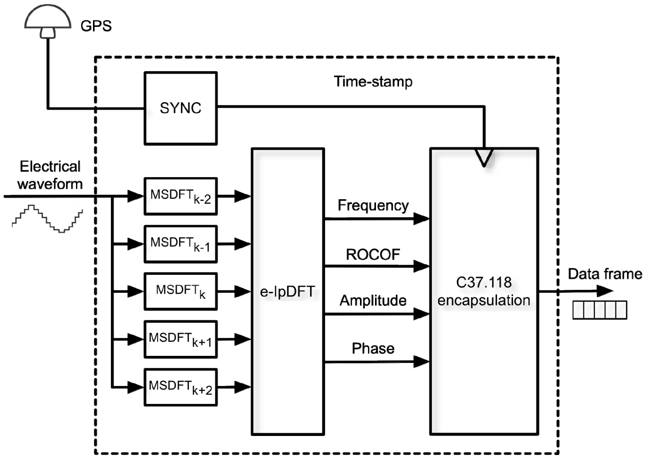

Figure 2: Block scheme of the proposed e-IpMSDFT-based virtual PMU.

B. UTC-time synchronization of the simulated PMU

As known (see Figure 3.a), PMUs are measurement devices connected to the electrical grid that synchronously take and report their measurements to a PDC [18]. PMUs need therefore to be synchronized to a stable and traceable UTC time reference (typically the one provided by the Global Positioning System – GPS) to report to the outside world the estimated values at regular time-intervals defined by the adopted reporting rate and at reporting times aligned with the UTC-second rollover. The PDC needs to operate synchronously as well and the synchronization can either be given by the synchronous PMU data flow itself or by properly synchronizing the PDC to a traceable UTC-time source [23].

The experimental validation of PMU-based monitoring and control schemes is typically carried out using HIL setup that needs to include at least the following two components:

- a RTS integrating the RT-models of the simulated electrical grid and PMUs;
- a PDC coupled with the monitoring or control algorithms under test.

In such context, it is clear that the RTS needs to integrate a proper GPS receiver, or equivalent UTC-time source, to bring synchronization to the simulated PMUs and, in this respect, two main possibilities exist.

The first one refers to a sort of ideal operating condition for the RTS that wants to integrate one or more virtual PMUs. In such a setup, the CPU clock used to derive the RTS integration time-step is disciplined by a UTC-stable time reference (see Figure 3.b). This condition allows to considerably simplify the PMU design by directly synchronizing the PMU sampling and reporting processes to UTC and guarantees that the RTS simulation time would not drift in time.

In the case of the adopted Opal-RT eMEGAsim PowerGrid Real-Time Digital Simulator, such synchronization can be achieved by taking advantage of the setup previously described and shown in Figure 1. The driver of the Spectracom board reads the integration time-step configured in the simulation model and generates a pulse at the corresponding frequency. The pulse is constantly aligned with the PPS of the GPS source so that it does not drift in time and it is fed to the FPGA board. In correspondence of each pulse, a FPGA counter register is incremented. The model developed by the Authors polls on the same FPGA counter and uses its increment to determine the exact instant to proceed to the next calculation step. In other words, since the FPGA counter increment is driven by a GPS disciplined pulse, the model integration time-step will not drift over time.

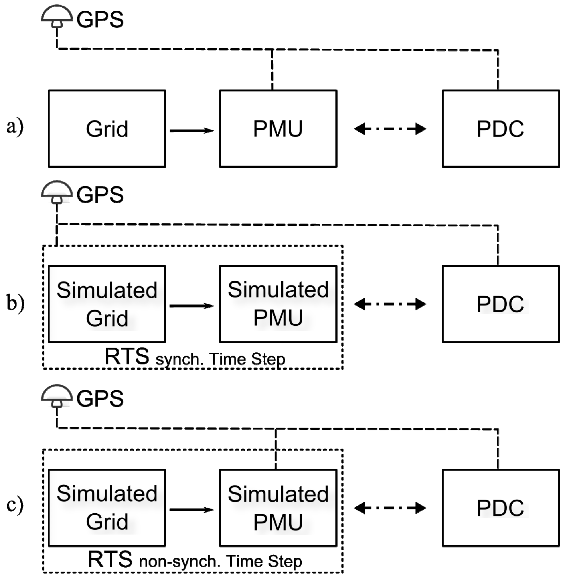

Figure 3: Possible time synchronizations of PMUs: (a) physical PMU device, (b),(c) virtual PMU deployed into a RTS.

The second possibility (see Figure 3.c) refers to a very common configuration where the CPU clock, and therefore the RTS integration time-step, cannot be disciplined to any stable time reference. This condition might happen, for instance, when the Opal-RT setup does not include any FPGA board. As a consequence the integration time-step can only be directly derived from the free-running CPU clock whose characteristic accuracy is in the order of few tens of ppm (part-per-million) and will therefore be affected by a slow but relevant time-drift (in the order of some tens of microseconds-per-second).

On the other hand, the PMU measurement reporting is automatically synchronized to UTC by means of the built-in C37.118 data encapsulation library by Opal-RT. More specifically, every time the GPS-derived UTC time hits a specific reporting time, the data frame is composed using the e-IpMSDFT estimations and streamed out of the RTS. As a consequence, the PDC receives a synchronous PMU data flow from the RTS but the estimated synchrophasors are obtained by means of a CPU clock that is drifting with respect to a generic UTC time-reference. In order to illustrate the effects of this drift, let consider, for the sake of simplicity, a simple setup where a single-channel PMU is directly connected to a simulated 50 Hz steady-state signal generator. Since the signal generation and the PMU sampling processes are running at the same speed, the simulated PMU will not notice the CPU clock time drift and will measure consistent data with respect to the signal generation settings. Nevertheless, the frequency and phase estimation received at the PDC will not be consistent due to the RTS's clock drift with respect to the UTC time reference. The consequence is that the PDC will receive a drifting phase from the simulated PMU with a constant 50 Hz frequency estimation.

In order to cope with the drift in time of the RTS, the PMU estimations of frequency and phase have been opportunely compensated. In particular, the RTS sampling clock error is computed in real time inside the simulated PMU as follows:

$$\varepsilon(n) = \frac{t_p - t_{p-Z}}{Z \cdot T_s} - 1 \tag{6}$$

where T_s is the integration time-step and the equivalent PMU sampling time, Z is the measurement window length (expressed as number of input samples) over which the clock error is computed (for the specific case of this simulated PMU we have adopted a 10 seconds window length) and t_p is the UTC start time of the *p*-th integration time step. Every time the CPU clock error is computed, the frequency estimation can be properly updated (see [10] for more details about this aspect).

In addition to this, another correction takes place into the simulated PMU when the integration time step is derived from the free-running CPU clock. This second correction compensates the phase uncertainty related to the PMU sampling time T_s by measuring the time distance between the specific reporting time and the first sample of the time-window used to estimate the synchrophasor. Such an uncertainty corresponds, for a sampling rate of 10 kHz and a rated system frequency of 50 Hz, to 10π mrad (1.8 deg), a value that definitely needs to be compensated particularly for PMUs conceived for distribution networks (see [10] for further details).

IV. Experimental Validation

Once deployed in the Opal-RT eMEGAsim PowerGrid Real-Time Digital Simulator, the simulated PMU has been experimentally validated with respect to: (i) the accuracy of the algorithm during the static and dynamic testing conditions dictated by the IEEE Std. C37.118.1-2011 [1]; (ii) the computational load of the developed PMU model.

In what follows, the Authors have selected to validate the configuration (b) of Figure 3 since it allowed a straightforward integration of the UTC-time synchronization of the RTS and particularly of the PMU model.

A. Accuracy assessment

Typically the validation of the performance of any synchrophasor estimation algorithm deployed in a real hardware always includes additional sources of errors dependent on (i) the PMU A/D converters resolution, (ii) the sampling process time-stability and (iii) the adaptation of the developed SE algorithm to the adopted hardware platform. In the case of the simulated PMU model these error sources are not included. In particular, the sampling process is perfectly synchronous with respect to the simulated power system and its jitter can be, therefore, neglected. Similarly the influence of the A/D converters and of the specific implementation of the SE algorithm are minor since the PMU model is directly coupled with the simulated voltage/current signals and the SE algorithm has been implemented using floating point precision (64 bits).

As a consequence, the PMU accuracy assessment, carried out on the RTS, allowed an analysis of the performance of the e-ipMSDFT synchrophasor estimation algorithm only (thus, eliminating all the sources of additional uncertainty). In this respect, the validation is performed by applying static and dynamic signals to the PMU as defined in [1] and verifying its compliance with the same standard. TABLE I. provides the performance assessment in terms of Total Vector Error (TVE), Frequency Error (FE) and response time.

TABLE I. Performance assessment of the e-IpMSDFT algorithm (the meaning of the symbols is the one in [1])

Test type	Test range	Maximum Error	
		TVE[%]	**FE[Hz]**
Single tone	f_0 ± 5 Hz	5.71e-4	6.90e-4
Multi tone	10 %, each harmonic up to 50th	9.24e-12	4.83e-13
Ampl. Modulation	Mod.frequency 0.1 to 5 Hz, k_x = ± 0.1	6.29e-1	7.96e-2
Phase Modulation	Mod.frequency 0.1 to 5Hz, k_a = ± 0.1 rad	5.69e-1	1.85e-2
Frequency ramp	±1Hz/s ramp, f_0 ±5Hz	3.71e-2	6.95e-4
Test type	**Test Range**	**Response Time [s]**	
		TVE	**FE**
Ampl. Step	Magnitude = ±10%, k_x = ±0.1	0.031	0.051
Phase Step	Angle = ±10°, k_a = ±π/18	0.026	0.048

For each test type, the results shown in Table I, were obtained by using the more demanding testing conditions among those defined for class P or class M PMUs. The specified range refers to a PMU with a reporting rate of 50 frame per second but other reporting rates were tested as well. The errors and response times were always within the limits and the TVE in steady state conditions is compatible with distribution system applications. The measurement accuracy was verified to be compatible with the one presented in [10].

B. Algorithm's computational requirements

Further tests were performed to verify the computational resources of the proposed SE algorithm when running in RT in the eMEGAsim target. In this respect, when selecting the integration time-step, it should be taken into consideration the fact that the MSDFT methods assumes that, for each new sample, every DFT bin needs to be updated in order not to compromise the next DFT estimation. This computation must be performed before the acquisition of the next sample, over the whole set of PMU input channels, in order to correctly estimate the corresponding synchrophasors. Following this considerations, the integration time-step (i.e. the PMU's sampling time) must be carefully selected to avoid overruns in the RTS target. In what follows, this parameter has been set to 100 µs.

Indeed, a virtual PMU is seldom simulated alone. Typically, a simulated electrical grid might integrate several tens of PMUs. In this respect, several virtual PMUs were included in the same model in order to assess the maximum number of PMU that can be simulated by a single core of the RTS. The test results are presented in TABLE II. and refer to an integration time-step of 100 µs and to 3 different virtual

PMU models, equipped with 1, 6 and 12 input channels respectively.

TABLE II. COMPUTATIONAL REQUIREMENTS OF THE PROPOSED VIRTUAL PMU INTO THE OPAL-RT EMEGASIM RTS

Integration time step	**Number of virtual PMUs per RTS core**		
	1-channel	**6-channels**	**12-channels**
100 µs	16	9	5

From the results of Table II we can conclude that, in the case of the adopted eMEGAsim RTS, the available 12 cores per Industrial PC easily allow the simulation of a realistic, thus complex, electrical grid that might be potentially equipped with several tens of virtual PMUs. Moreover, these results do not depend on the adopted reporting rate that, based on the developed e-IpMSDFT algorithm, can be set as high as the adopted sampling rates (see also [11]).

V. CONCLUSIONS

The paper has shown a synchrophasor extraction (SE) algorithm that is both compliant with the accuracy requirements of the IEEE Std. C37.118 and deployable into a real-time simulation platform. In particular, the proposed SE algorithm, named Enhanced Interpolated Moving Sliding DFT (e-IpMSDFT), is characterized by accuracy levels compliant with all the requirements of the class-P PMUs and the majority of the class-M (with the only exception of the out-of-band interference ones). It is also worth noting that the accuracy of the obtained virtual PMU in steady state conditions are extremely high and compatible with its use in distribution systems applications.

In terms of computation complexity, the proposed virtual PMU is characterized by a relatively low computational complexity allowing to deploy up to 9 PMU per simulator core (in this case, each PMU assesses 6 synchrophasors).

Finally, it is worth observing that the e-IpMSDFT algorithm is capable to provide a SE for each raw sample processed by the virtual PMU. Therefore, the proposed solution is capable to stream the estimated synchrophasors with a reporting rate equal to the inverse of the simulation time-step of the model executed in the real-time simulator.

ACKNOWLEDGMENT

The research leading to these results has received funding from the NanoTera Swiss National Science Foundation project *S*^{3}-Grids. The authors alone are responsible for the content of this paper.

REFERENCES

[1] "IEEE Standard for Synchrophasor Measurements for Power Systems," IEEE Std C37.118.1-2011, 2011.

[2] J. De La Ree, V. Centeno, J.S. Thorp, A.G. Phadke, "Synchronized Phasor Measurement Applications in Power Systems," *IEEE Transactions on Smart Grid*, vol.1, no.1, pp. 20-27, June 2010.

[3] A. Borghetti, C.A. Nucci, M. Paolone, G. Ciappi, A. Solari, "Synchronized Phasors Monitoring During the Islanding Maneuver of an Active Distribution Network," *IEEE Transactions on Smart Grid*, vol.2, no.1, pp. 82-91, March 2011.

[4] R.J. Best, D.J. Morrow, D. M. Laverty, and P. A. Crossley, “Synchrophasor Broadcast Over Internet Protocol for Distributed Generator Synchronization”, *IEEE Trans. Power Delivery*, Vol. 25, No. 4, pp. 2835-2841, October 2010.

[5] D. Ouellette, M. Desjardine, R. Kuffel, Yi Zhang, E. Xu, "Using a Real Time Digital Simulator with Phasor Measurement Unit technology," *International Conference on Advanced Power System Automation and Protection* (APAP), 2011, vol.3, no., pp.2472,2476, 16-20 Oct. 2011.

[6] D. R. Gurushinghe, A. D. Rajapakse, D. Muthumuni, "Modeling of a synchrophasor measurement units into an electromagnetic transient simulation program," *International Conference on Power Systems Transients (ISPT)*, Vancouver, Canada July 18-20, 2013.

[7] A. T. Al-Hammouri, L. Nordstrom, M. Chenine, L. Vanfretti, N. Honeth, R. Leelaruji, "Virtualization of synchronized phasor measurement units within real-time simulators for smart grid applications," *IEEE Power and Energy Society General Meeting (PES)*, 2012, pp.1,7, 22-26 July 2012.

[8] Kun Zhu, S. Deo, A. T Al-Hammouri, N. Honeth, M. Chenine, D. Babazadeh, L. Nordstrom, "Test platform for synchrophasor based wide-area monitoring and control applications," *IEEE Power and Energy Society General Meeting (PES),* 2013 , pp.1,5, 21-25 July 2013.

[9] "IEEE Standard for Synchrophasor Measurements for Power Systems – Amendment 1: Modification of Selected Performance Requirements", IEEE Std C37.118.1-2014, 2014.

[10] P. Romano, M. Paolone, "Enhanced interpolated-DFT for synchrophasor estimation in FPGAs: theory, implementation, and validation of a PMU prototype," *IEEE Transactions on Instrumentation and Measurement*, vol.63, no.12, pp.2824-2836, Dec. 2014.

[11] P. Romano, M. Paolone, "An enhanced interpolated-modulated sliding DFT for high reporting rate PMUs," presented at the *International workshop on Applied Measurements for Power Systems (AMPS),* Aachen, Germany, 24-26 Sept. 2014.

[12] M. Pignati et al. "Real-time state estimation of the EPFL-campus medium voltage grid by using PMUs," presented at the *IEEE Power and Energy Society Conference (ISGT)*, 2015, 17-20 February 2015.

[13] Cyber-Secure Data And Control Cloud For Power Grids (C-DAX) FP-7 Project [Online]. Website: http://www.cdax.eu

[14] K. Banks, “The Goertzel algorithm,” Embedded Syst. Programming Mag., vol. 15, no. 9, pp. 34–42, Sept. 2002.

[15] E. Jacobsen, R. Lyons, "The sliding DFT," Signal Processing Magazine, IEEE , vol.20, no.2, pp.74-80, Mar. 2003.

[16] Park, C.; Ko, S., "The Hopping Discrete Fourier Transform [sp Tips&Tricks]," *IEEE Signal Processing Magazine*, vol.31, no.2, pp.135-139, Mar. 2014

[17] K. Duda, "Accurate, Guaranteed Stable, Sliding Discrete Fourier Transform [DSP Tips & Tricks]," Signal Processing Magazine, IEEE , vol.27, no.6, pp.124-127, Nov. 2010.

[18] A. V. Oppenheim, R. W. Schafer, J. R. Buck et al., Discrete-time signal processing. Prentice-hall Englewood Cliffs, 1989, vol. 2.

[19] D. Belega, D. Petri, "Accuracy Analysis of the Multicycle Synchrophasor Estimator Provided by the Interpolated DFT Algorithm," *IEEE Transaction on Instrumentation and Measurement*, vol.62, no.5, pp.942,953, May 2013

[20] Opal-RT eMEGAsim PowerGrid Real-Time Digital Hardware in the Loop Simulator [Online]. Website: http://www.opal-rt.com/product/emegasim-powergrid-real-time-digital-hardware-in-the-loop-simulator

[21] Spectracom PCI Express Slot Cards [Online]. Website: http://www.spectracomcorp.com/ProductsServices/TimingSynchronization/Bus-levelTiming/PCIexpressslotcards/tabid/1296/Default.aspx

[22] "IEEE Standard for Synchrophasor Data Transfer for Power Systems," IEEE Std C37.118.2-2011, 2011.

[23] "IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring" IEEE Std C37.244-2013, 2013.