Virtualization of Synchronized Phasor Measurement Units within Real-Time Simulators for Smart Grid Applications

A.T. Al-Hammouri, L. Nordström, *Member, IEEE*, M. Chenine, *Student Member, IEEE*,
L. Vanfretti, *Member, IEEE*, N. Honeth, *Student Member, IEEE*, and R. Leelaruji, *Student Member, IEEE*

***Abstract*—Synchronized phasor measurement units (PMUs) provide GPS-time tagged high-sampling rate positive-sequence voltage and current phasors. When placed in high-voltage substations in power networks, PMUs can provide real-time information that is necessary for the development of Smart Transmission Grid software applications for improving power system monitoring, control and protection. The development of these applications, particularly for use within control centers for on-line purposes, is limited by the availability of and access to real-time PMU data and other information. One attractive approach for application development is the use of real-time simulators to which PMUs can be interfaced as hardware-in-the-loop (HIL) devices to harvest PMU data. However, this approach has technical and economical limitations, which can be tackled by the *virtualization* of PMU devices. This article describes the development of an entirely software-based synchronized phasor measurement unit for use within real-time simulators that will allow the emulation of a large number of real-life PMUs, which in turn can be used for creating new phasor-based applications.**

I. Introduction

Synchronized phasor measurement units, or shortly phasor measurement units (PMUs), are digital measurement instruments that by measuring three-phase voltage and current waveforms, are capable of providing high-sampling rate positive-sequence voltage and current phasors that are time-tagged by a GPS signal at the measurement source [1], [2], [3]. When placed in high-voltage substations in power networks, PMUs provide important real-time information that can be used within software applications to improve power system monitoring and control [4]. This is why synchronized phasor measurements and their supporting infrastructures are put in the front-line from the transmission system perspective as enablers of the Smart Transmission Grid [5]. This can be evidenced by the initiatives in North America, and elsewhere, that have created specialized systems exploiting measurements from PMUs with the aim of both enabling new PMU-data applications, and increasing the utilization of synchrophasors in power system operations and control [6].

There are different alternatives for developing new applications using synchrophasor data. First, PMU data can be harvested from PMUs installed at transmission or distribution networks through *ad hoc* WAMS systems [7]. This alternative has the drawback that researchers need to secure non-disclosure-agreements that allow the use of such data. In addition, researchers need to be provided with archives not only of phasor data but also of network topology and other SCADA information [8]. A second drawback of this approach is that the developed applications will be limited to off-line analysis methodologies, simply by the nature of the data. Therefore, with this approach, it is difficult to understand the impact of the supporting ICT infrastructure on power system applications. Hence, on-line applications for real-time monitoring are difficult to develop under this approach.

A second alternative for the development of phasor data applications is to use a simulation environment [9], [10]. To properly generate primary data for use with PMUs, the simulation environment needs to provide high-resolution three-phase data of voltage and current waveforms [9]. One suitable approach is to use real-time simulators, which can simulate three-phase with high resolution and can allow for the interfacing of hardware-in-the-loop devices. In this case, PMUs can be interfaced either through signal amplifiers or low-voltage outputs of these real-time simulators. This will at the same time allow the interfacing of the PMUs with Phasor Data Concentrators (PDCs) through a Local Access Network (LAN) or a Wide Area Network (WAN) by streaming data through their Ethernet ports using TPC/IP and the synchrophasor data protocol IEEE C37.118 [11].

The major drawbacks of this approach are both technical and economical. The technical limitations are related to the possible number of outputs that can be used to interface PMU devices, which can be very limited; and also the possible limits on small-time step computation due to high number of output signals. The economical constraints are mainly due to cost: for a research laboratory, only a limited number of units can be acquired due to budgeting issues.

To overcome these difficulties, the “virtualization” [12], i.e. the development of an entirely software-based synchrophasor measurement unit, presents an attractive approach. Such unit

Manuscript submitted to the IEEE PES General Meeting 2012.
The bulk of this work was performed while the first author was visiting KTH Royal Institute of Technology, Stockholm, Sweden, under an Erasmus Mundus Scholarship Programme.
This work was supported in part by the Swedish Power industry via ELEKTRA project 36084, and EKC^2 - The Swedish Centre of Excellence in Electric Power Engineering. Luigi Vanfretti is supported by the STandUP *for* Energy collaboration initiative and the KTH School of Electrical Engineering.

A.T. Al-Hammouri is with the Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid 22110, Jordan. E-mail: hammouri@just.edu.jo

L. Nordström, M. Chenine, and N. Honeth, are with the Industrial Information & Control Systems Division, School of Electrical Engineering, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden. E-mail: moustafac@ics.kth.se, nicholash@ics.kth.se, larsn@ics.kth.se

L. Vanfretti and R. Leelaruji are with the Electric Power Systems Division, School of Electrical Engineering, KTH Royal Institute of Technology, Teknikringen 33, SE-100 44, Stockholm, Sweden. E-mail: luigiv@kth.se, rujiroj@kth.se.

will be capable of delivering real-time data by harvesting the three-phase voltage and current waveforms from a real-time simulator, computing real-time phasors of voltage and current, and delivering them over a LAN or WAN.

A. *Purpose*

This article discusses the development of a software-based synchronized phasor measurement unit for use within real-time simulators. The *soft* PMU, is a *virtualized* device capable of exploiting real-time data generated by a real-time simulator, computing real-time phasors, and delivering them over a communication networks emulator. The continued improvement and development of such *virtualized device* will enable the development of PMU-data based applications that need large amounts of measurement devices to realize their potential. At the same time, these new devices can be used within a Wide-Area Monitoring System, which is part of a wider platform for “Smart Transmission Grid”.

B. *Outline*

The reminder of this paper is structured as follows. Section II presents an overview of the main characteristics of synchronized phasor-measurement units. Section III presents the overall architecture within which the *soft* PMU is used, while Section IV describes the implementation details of the *soft* PMU. In Section V, we validate the correct operation of the *soft* PMU and present the future planned experiments. Finally, Section VI concludes the paper.

II. Synchronized Phasor Measurement Units

A Phasor Measurement Unit is a digital measurement device capable of providing high-sampling rate positive-sequence voltage and current phasors that are time-tagged at the source. The time-tags associated with the measurements are calibrated by the Global Positioning System (GPS), thus allowing the synchronization of phasor measurements made across large geographical extensions of a power system. Synchronized measurements allow creates a comprehensive view of the system at the instants when the measurements are taken.

A functional diagram of a generic PMU is shown in Fig. 1. Although PMUs are made by many different manufacturers with different designs, this functional diagram encapsulates the important features common to most PMUs. This functional diagram is comprised of three stages: a Measurement Stage, a Computation Stage, and a Communications Stage.

In the Measurement Stage, the Analog Inputs corresponding to voltages and currents are obtained from potential transformers (PTs) and current transformers (CTs). All available three-phase voltage and currents are used to determine the positive-sequence phasors. Note that some PMUs are capable of processing current phasors for more than one line current in a single unit. Each of these analog signals is filtered using an anti-aliasing filter and sent to the Computation Stage. The sampling rate will dictate the frequency response of the anti-aliasing filters, which in most PMUs are analog filters. The selected cut-off frequency of each filter should satisfy the Nyquists criterion, i.e., the cut-off frequency is less than half the sampling frequency.

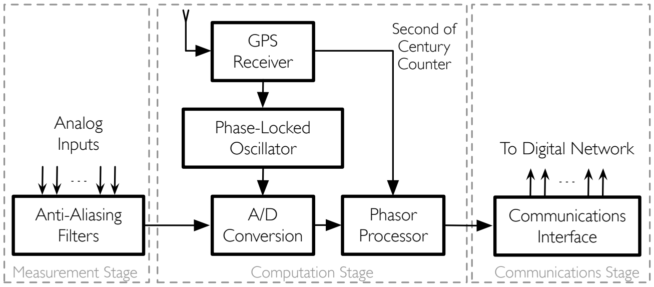


Fig. 1. Functional Diagram of a Generic PMU

In the Computation Stage, an analog-to-digital converter (A/D) samples the data, i.e., the signals are converted into digital samples. The sampler works in phase-lock with the GPS pulses, with sampling rates ranging from the initial 12 samples per cycle of the early developed devices [13] up to 128 in contemporary ones. A microprocessor receives the sampled data and the GPS time-tags, and calculates the positive-sequence components of all the voltage and current signals using different techniques [1]. In addition to voltage and current measurements, the microprocessor calculates an estimate of the frequency, f, and the rate of change of the frequency, df/dt, using the voltage angle. Positive-sequence phasor estimates are reported at a rate of 10 to 60 samples per second (i.e., interval of 100 to 16.67 ms)

Finally, at the Communications Stage, the GPS time-tagged measurements are encapsulated into packets and are transferred by the communication interface via suitable communication links [14] using the standard IEEE C37.118 protocol defined in [11]. Several PMUs also provide local storage for triggered disturbance events. Data streamed from different PMUs is collected in Phasor Data Concentrators (PDCs). It should be noted that many manufacturers are increasingly including software capable of performing PMU functions with other microprocessor based devices [15]. In the next sections, we describe how the *soft* PMU virtualizes the process described above.

III. The Overall Architecture

The platform within which the *soft* PMU is to be used consists of three (sub)systems (see Fig. 2):

1) The eMEGAsim Real-Time Digital Simulator [16]. The eMEGAsim is a commercial highly accurate power-systems simulator. Being a computationally powerful simulator, it can simulate large-scale power systems in real-time. Therefore, it can be used for hardware-in-the-loop simulations (HIL), where some parts of the power system can be real-life physical components [17]. The eMEGAsim simulates power system models that are constructed with the SimPowerSystems Toolbox in Simulink®[18].
2) Network Emulator (ModelNet) [19]. This is a wide-area-network (WAN) emulator. It can be introduced

between two or more machines that are on the same local-area-network (LAN) and are connected via, say, Ethernet, to emulate the behavior of a geographically long, slow, bandwidth-limited, and lossy communication link. Therefore, it can be used to study the effect of network impairments, e.g., delays, losses, and limited bandwidth, on the performance of networked applications and protocols.

3) KTH-PowerIT [20]. This is an application-level platform (or a middleware) that collects real-time synchrophasor measurements from geographically distributed phasor-measurement-units (PMUs). The platform is capable of analyzing the aggregated data off line. This is the user front-end that arranges and presents the synchrophasor measurements in a convenient manner.

These three subsystems are combined as in Fig. 2. In the figure, the Physical PMU is an actual PMU device that is connected to the eMEGAsim and is configured to obtain the phasor measurements from the simulation model and sends them out on the network. To an entity connected to the output of this PMU, the phasor measurements appear as if they are coming from an actual power system.

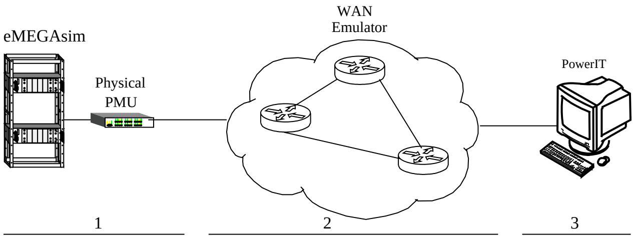


Fig. 2. The overall integrated simulation-emulation platform: interconnection of three subsystems, the eMEGAsim Real-Time Digital Simulator cascaded with the real-life PMU, the Network Emulator, and the PowerIT.

This setup can be used to study different scenarios, e.g., the influence of the network parameters on the effectiveness and efficiency of the power system monitoring and control. However, there are two limitations for this setup. First, an actual power system is often equipped with tens or hundreds of PMUs. Therefore, having a single or a few PMUs connected to the eMEGAsim provides limited benefits for considering and addressing more involved and interesting scenarios, such as, the existence of multiple measurement flows, and the interaction between different traffic flows belonging to different monitoring or control paths or loops. On the other hand, it is very costly to acquire and connect tens or hundreds of PMUs to the eMEGAsim. Second, with the physical PMU(s), one will be locked-in with the protocols and capabilities supported by the respective PMU (i.e., one will be unable to experiment with other protocols because the physical PMU represents a closed system). One possibility to overcome these limitations would be to cascade the PMU with a separate machine running a program that replicates the PMU output to several streams, and thus producing the illusion of multiple traffic flows with the possibility of tunneling or translating the PMU protocol(s) into arbitrary different ones. Nevertheless, the carried data would correspond to the same phasor quantities, and thus will be inappropriate if closed-loop feedback control is to be considered.

In light of these constraints, the alternative is to construct a *soft* PMU as a Simulink®module that executes on the eMEGAsim. The advantage of such choice is that several PMUs can be instantiated and plugged to different parts of the power system, where each act independently of others, i.e., each can measure a different phasor quantity. Also, other protocols that are not supported by the physical PMU can be investigated. In the next section, we elaborate more on the *soft* PMU.

IV. The *soft* PMU

The high-level architecture of the *soft* PMU is depicted in Fig. 3. The *soft* PMU is divided and structured into two different parts. First, there are the Simulink®blocks that can be connected to a power system model, which measure the three-phase voltage and current, compute phasors, and estimate other quantities such as the frequency and the frequency rate of change. These blocks are part of the power system simulation model, and thus run on the eMEGAsim (they can be regarded as the Measurement and Computation stages of the generic PMU shown in Fig. 1). Second, there is the *soft* PMU's outer interface daemon (i.e., the communication-related side). In this part, the measurements are wrapped into C37.118 protocol messages [11], and then inscribed within TCP or UDP packets that are ultimately put on the wire. This part executes on a separate machine. Every stream from each instance of the Simulink®measurement blocks is wrapped in IEEE C37.118 protocol with a distinct device ID (see [11]). Further, packets belonging to the same stream (i.e., coming from the same Simulink®measurement block) are forwarded with a distinct IP address, thus emulating several physical PMUs; see Fig. 3. This separation in two parts is aligned with newly yet-to-be-published IEEE C37.118 protocol that separates the measurement part from that of the data transmission, i.e., the communication part. The communication part is not only responsible for outputting C37.118 *data* packets but it also implements the full specification of the C37.118 protocol. In essence, it also outputs *configuration* and *header* messages

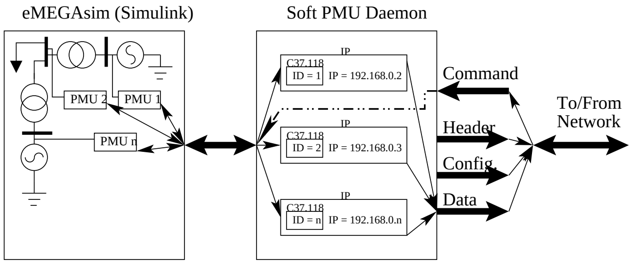


Fig. 3. The *soft* PMU architecture. The *soft* PMU is divided into two logical components: the measurement part and the communication part, where each part executes on a separate machine. Both parts making up the *soft* PMU correspond to subsystem 1 in Fig. 2.

[11], and is able to receive *command* messages [11] and takes proper actions based on such commands as is explained next.

A. Implementation of the IEEE C37.118 Protocol

We have implemented a library for the C37.118 Protocol [11] using the C++ programming language. There exist only some commercial libraries for the C37.118 [21] but not a single open-source one exists in C++. The only open source library for the C37.118 is the one that is part of the OpenPDC project [22], but unfortunately it was implemented using the C# programming language. The problem with C# is that the library is not portable to other platforms running non-Windows Operating Systems.

Our library is tailored for PMU operations. That is, it allows crafting C37.118 messages of the following types:

- Data messages carrying measurements data, e.g., current, voltage, and frequency quantities,
- Configuration messages carrying meta-data context for the data messages, i.e., carrying details or information about the phasor data in the data messages, and
- Header messages carrying user-defined information.

Since the PMU only receives command messages (carrying commands to control the operation and configuration of the PMU), and never sends them (see Fig. 3), the library does not support crafting command messages. However, the library contains the necessary functions to check the sanity and validity of command messages. These functions, which are used by the PMU before accepting and executing a received command, allow the validation of the following specific information

- The frame is well crafted, uncorrupted, and is sound. That is, the frame starts with a sync byte of 0xAA as the first byte, the frame type is indeed a command frame, the protocol number is the one that the PMU understands, the CRC code carried within the frame is identical to the computed one out from the received frame, and the carried frame size is correct.
- The frame is indeed addressed to the respective PMU, i.e., the ID code carried in the frame matches with the ID code assigned to and stored in the PMU.
- The intended command is a one that is supported by the PMU. Currently, the PMU understands only four commands, which are [11]
  - Turn OFF transmission of data,
  - Turn ON transmission of data,
  - Send the header frame,
  - Send the FIRST configuration frame, and
  - Send the SECOND configuration frame.

B. Implementation of the PMU Daemon

As mentioned above, this is the communication part of the PMU, i.e., the part that sends and receives C37.118 frames. This corresponds to the Communication Stage shown in the description of a generic PMU in Fig. 1. The PMU daemon obtains the calculated synchrophasor measurements—such as, phasor data (the magnitude and phase components, or the real and imaginary components of voltages and currents), other analog data (active and reactive power), digital data (switches' statuses), and frequency and rate of change of frequency—from the measurement part, i.e., the Simulink®blocks (explained next). Once the daemon starts up,

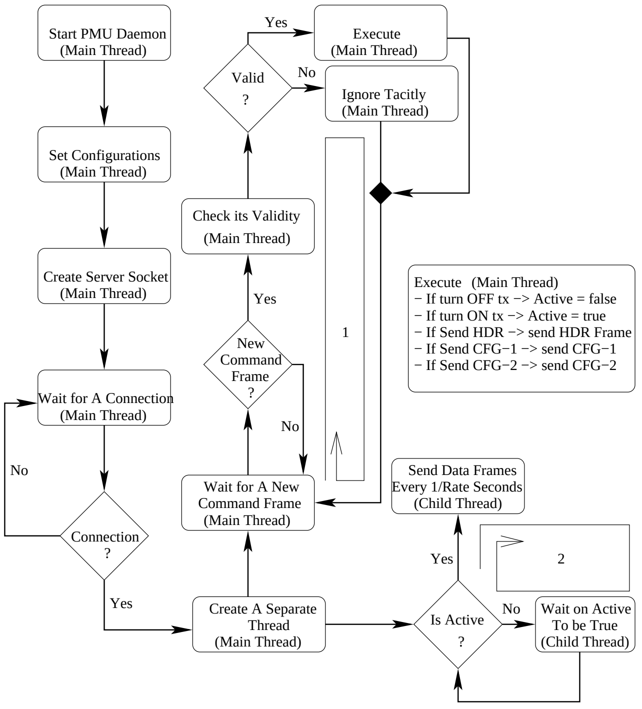

Start PMU Daemon (Main Thread)
Set Configurations (Main Thread)
Create Server Socket (Main Thread)
Wait for A Connection (Main Thread)
No
Connection ?
Yes
Create A Separate Thread (Main Thread)
Wait for A New Command Frame (Main Thread)
New Command Frame ?
No
Yes
Check its Validity (Main Thread)
Valid ?
Yes
No
Execute (Main Thread)
Ignore Tacitly (Main Thread)
1
Execute (Main Thread)
− If turn OFF tx −> Active = false
− If turn ON tx −> Active = true
− If Send HDR −> send HDR Frame
− If Send CFG−1 −> send CFG−1
− If Send CFG−2 −> send CFG−2
Send Data Frames Every 1/Rate Seconds (Child Thread)
Yes
2
Is Active ?
No
Wait on Active To be True (Child Thread)

Fig. 4. The PMU daemon execution steps. After a given PMU instance accept a remote connection, two threads are executing in parallel: one to accept and execute command and the other to stream out data frames whenever the PMU is put in the active state.

1) it populates the specific configurations for each PMU from a configuration file. These configurations include the PMU ID code, the IP address and a port number the PMU waits for connection on, whether the PMU is starting active or disabled, the station name, the reporting (transmission) rate, the data format, the number of phasors quantities and their descriptive names, the number of analog quantities and their descriptive names, and the number of digital quantities and their descriptive names [11].
2) it creates a server socket and starts listening at the given port number waiting for connections (usually coming from a PDC or a custom software designed for consuming the phasor measurements).

When a connection is established from a remote entity (e.g., a PDC),

- if the PMU is configured to start with an active state, the daemon streams data frames (on a different spawn thread) to the remote entity. Data frames are transmitted at the configured reporting rate. At the same time in the main thread, the daemon waits for arrival of command frames.
- if the PMU is configured to start disabled, the daemon stays in standby waiting for an arrival of a command frame.

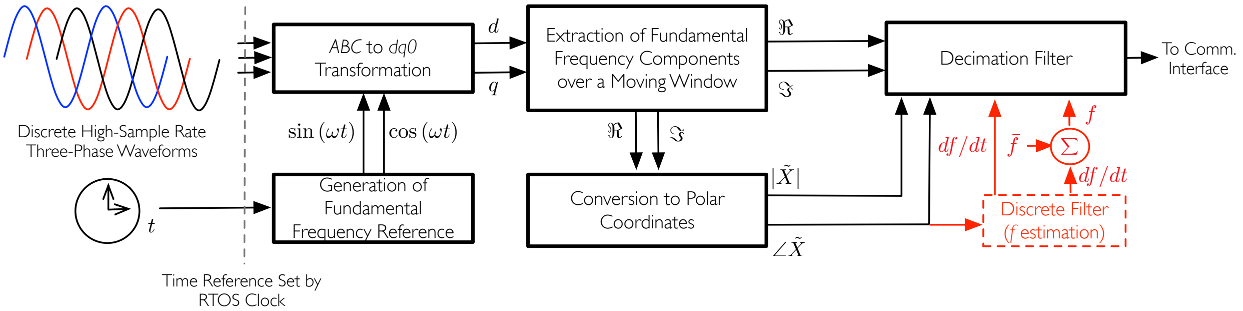

Discrete High-Sample Rate Three-Phase Waveforms
ABC to dq0 Transformation
d
q
Extraction of Fundamental Frequency Components over a Moving Window
$\Re$
$\Im$
Decimation Filter
To Comm. Interface
$\sin(\omega t)$
$\cos(\omega t)$
Generation of Fundamental Frequency Reference
t
$\Re$
$\Im$
Conversion to Polar Coordinates
$|\tilde{X}|$
$\angle\tilde{X}$
df/dt
$\bar{f}$
Σ
f
df/dt
Discrete Filter (f estimation)
Time Reference Set by RTOS Clock

Fig. 5. Phasor Computations Implemented in the *soft* PMU

Upon receiving a *valid* command, the PMU executes the command accordingly, e.g., turning on or off the transmission of data frames. If the command frame is asking for the header or for either of the configuration frames (1 or 2), the PMU crafts the corresponding frame and sends it out to the requester based on the specific configurations assigned to the PMU. If the received command, however, does not pass the validation checks mentioned above, the PMU just ignores the command tacitly after logging it to a file.

The logical order of execution of these steps is illustrated with the flowchart in Fig. 4. The two loops 1 and 2 keep executing until the remote entity closes the connection whereby the main thread returns to the 'wait for a connection' state.

C. Communication Between Simulink®and the PMU Daemon

Since the two parts of the *soft* PMU, the measurement and communication parts, are running on separate machines, the most convenient method to relay data between them is via the network, e.g., using socket programming. For the measurement part, we utilized specialized socket-communication modules that are provided by the OPAL-RT. These are Simulink®modules that are compliant with the eMEGAsim simulator unlike the general Simulink®modules that are part of the MATLAB's Instrument Control Toolbox [23]. The utilized modules send data in binary format. At the communication part, custom code is developed to receive and correctly interpret the data from the measurement part. The communicated phasor values are sampled at the Simulink®side at the same reporting rate of the PMU configuration, and are sent along with their respective sampling times to the PMU daemon. The communication process between the two parts takes place on the fly, i.e., in real-time.

D. Computation of Phasor Measurements from Three-Phase Real-Time Data

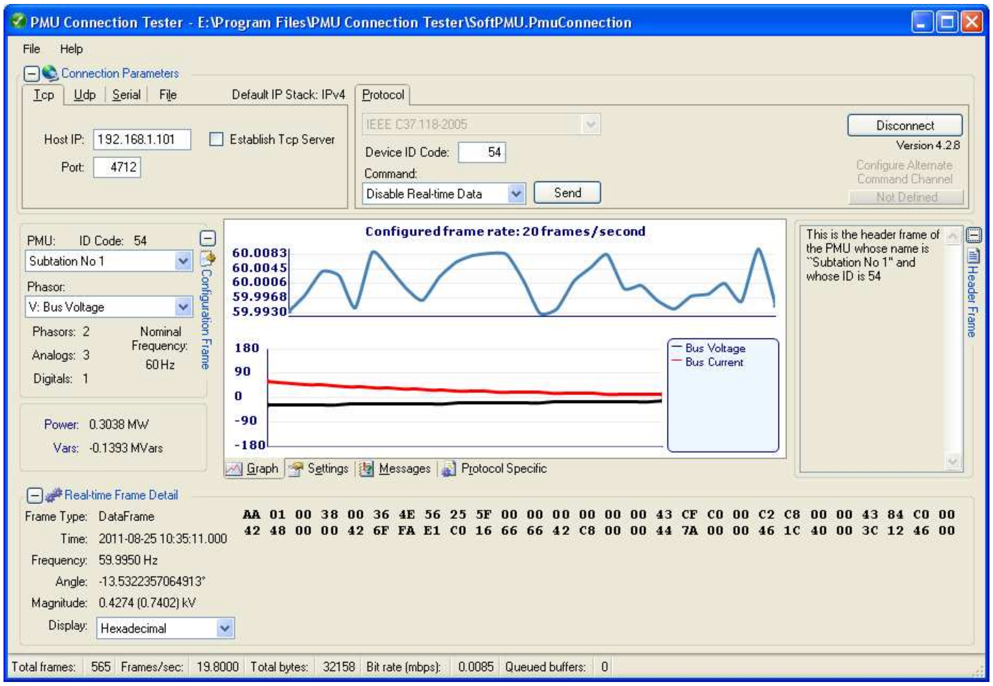
Up to this point the article has described how the communication end of this software-based PMU was developed and implemented. However, for this new device to properly emulate real-life PMUs, it is necessary to process the three-phase waveforms generated by the real-time simulator to produce positive-sequence voltage and current phasors, and other data, which will be transmitted to the communication networks emulator.

A block diagram of the computations process carried out by the *soft* PMU is shown in Fig. 5. With reference to Fig. 1, this implementation can be considered as the Computation Stage. The computations are carried out using Simulink®blocks compatible with the eMEGAsim real-time simulator. In this implementation, there is no need for the Measurement Stage, which comprises anti-aliasing filters or A/D conversion, as the data is already discretized at a fixed step resolution, e.g. 100 μs time-step.

The three-phase real-time simulated voltage and current waveforms are referred to the internal clock of the Real-Time Operating System (RTOS) of the real-time simulator. This same time reference is used to generate fundamental frequency reference signals, $\sin(\omega t)$ and $\cos(\omega t)$, which are used to transform the three-phase ABC signals to the $dq0$ plane. After each sample is transformed into $dq0$ components, the d and q components are used to generate a window of data. This moving window is used to extract the fundamental frequency component in both d and q signals, and thus generates the real ($\Re$) and imaginary ($\Im$) parts of the phasor. After these real and imaginary parts are computed, a conversion to polar coordinates is necessary so that the magnitude of the complex phasor $\tilde{X}$ can be divided into its magnitude, $|\tilde{X}|$, and phase, $\angle\tilde{X}$, where $\tilde{X}$ can be either a voltage or current phasor. In the case of current phasors, all computed phasors are passed through a decimation filter (in case the real-time simulation time-step is high) to extract redundant points. On the other hand, in the case of voltage phasors, there is an additional step that computes the bus frequency f and the df/dt, as shown by the red blocks in Fig. 5. This block uses a discrete filter to determine the df/dt from the voltage angle, and uses a recursive mean value of the frequency $\bar{f}$ to compute the frequency f. These quantities might also need to be passed through a decimation filter. After all these steps, the data is sent to the communications interface discussed above.

V. Experiments

The *soft* PMU was tested in a similar setup to the one shown in Figs. 2 and 3. We utilized the PMU Connection Tester [24], which is part of the OpenPDC project, by connecting remotely to the *soft* PMU daemon; see Fig. 6. As shown in the figure, the PMU Connection Tester is successfully receiving and deciphering the frames sent by the *soft* PMU



Default IP Stack: IPv4

Host IP: 192.168.1.101

Port: 4712

☐ Establish Tcp Server

IEEE C37.118-2005

Device ID Code: 54

Command: Disable Real-time Data

Version 4.2.8

PMU: ID Code: 54

Subtation No 1

Phasor: V: Bus Voltage

Phasors: 2

Analogs: 3

Digitals: 1

Nominal Frequency: 60 Hz

Power: 0.3038 MW

Vars: -0.1393 MVars

Configured frame rate: 20 frames/second

This is the header frame of the PMU whose name is ``Subtation No 1" and whose ID is 54

Frame Type: DataFrame

Time: 2011-08-25 10:35:11.000

Frequency: 59.9950 Hz

Angle: -13.5322357064913°

Magnitude: 0.4274 (0.7402) kV

Display: Hexadecimal

```
AA 01 00 38 00 36 4E 56 25 5F 00 00 00 00 00 00 43 CF C0 00 C2 C8 00 00 43 84 C0 00
42 48 00 00 42 6F FA E1 C0 16 66 66 42 C8 00 00 44 7A 00 00 46 1C 40 00 3C 12 46 00
```

Total frames: 565 | Frames/sec: 19.8000 | Total bytes: 32158 | Bit rate (mbps): 0.0085 | Queued buffers: 0

Fig. 6. A screen-shot of the PMU Connection Tester [24] while it is connected to the developed *soft* PMU. The IP address of the PMU is 192.168.1.101 and it accept connections on port number 4712, which is the default port number recommended by the C37.118 standard [11]. The PMU Connection Tester is successfully receiving and deciphering the data frames (the two plots and the Real-time Frame Detail section at the bottom), header frames (right), and configuration frames (left).

including data frames, header frames, and configuration frames of both types 1 and 2. Specifically, in the tested setup, the *soft* PMU sends phasor measurements in rectangular format (i.e., real and imaginary parts) while the PMU Connection Tester correctly displays the phasors in polar format (i.e., magnitude and angle components); see Fig. 6. In addition, the PMU is successfully receiving header and command frames from The PMU connection Tester and is correctly acting on the command frames.

A. Future Planned Experiments

The developed platform paves the way for unlimited number of interesting experiments. The intent in this paper is to present and report on the newly developed platform that targets the study of PMU based power system applications. We here highlight some of the experiments that we are currently undertaking as part of our short-term future work.

1) Network Infrastructure and Protocols: With the developed platform, it is very easy to assess the adequacy of different network infrastructures, topologies, and protocols on the Wide Area Monitoring and Control of the power system. Examples include physical media (e.g. wireless vs. wired), higher-level protocols (e.g., TCP vs. UDP), and bandwidth allocation schemes between different PMU traffic flows and between PMU traffic and multimedia traffic (e.g., VoIP and video traffic).

2) ICT Architecture: The supportive ICT architecture has large impact on the WAMC [14]. Utilizing our platform, one can study the impact of different ICT architectures on the WAMC, such as a single PDC vs. several PDCs, and a hierarchical PMU-PDC model vs. a flat PMU-PDC model.

3) Power System Control: The availability of massive number of phasor signals going through an ICT network provides an opportunity to evaluate the effect of network latencies and other issues on power system control applications, for example inter-area mode damping and voltage control.

VI. Conclusion

The development of Smart Grid applications that exploit synchronized phasor measurement data is limited by the availability of both real-time data streams and data archives. The use of real-time hardware-in-the-loop simulators interfaced with *real* PMUs is a suitable approach for generating phasor data in a controlled environment, but has both technical and economical limitations. In this article, we proposed, developed and implemented the concept of the *soft* PMU, a *virtualized* synchronized phasor measurement unit within real-time simulators that can help in overcoming the limitations of the HIL approach.

The paper discussed how both the communications end and the phasor computation end of this software-based PMU was developed and implemented. Although there are some limitations in the current implementation, this entirely software-based PMU is essential for the generation of data from a large number of PMUs, which in turn can be fed into a communications network emulator to carry out detailed studies on the impact of ICT on power system control applications.

Acknowledgment

A.T. Al-Hammouri would like to acknowledge the funding support of the Erasmus Mundus Scholarship Programme.

References

[1] A. Phadke, M. Ibrahim, and T. Hlibka, “Fundamental basis for distance relaying with symmetrical components,” *IEEE Transactions on Power Apparatus and Systems*, vol. 96, no. 2, pp. 635–646, Mar 1977.

[2] A. G. Phadke, J. S. Thorp, and M. G. Adamiak, “A New Measurement Techniques for Tracking Voltage Phasors, Local System Frequency, and Rate of Change Frequency,” *IEEE Transactions on Power Apparatus and Systems*, vol. 102, no. 5, pp. 1025–1038, May 1983.

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

[4] J. Hauer, D. Trudnowski, and J. DeSteese, “A perspective on WAMS analysis tools for tracking of oscillatory dynamics,” *IEEE PES General Meeting*, 2007.

[5] A. Bose, “Smart Transmission Grid Applications and Their Supporting Infrastructure,” *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 11–19, June 2010.

[6] M. Parashar and J. Mo, “Real Time Dynamics Monitoring System (RTDMS): Phasor Applications for the Control Room,” in *42nd Hawaii International Conference on System Sciences, 2009 (HICSS ’09)*, Jan 2009, pp. 1–11.

[7] J. Hauer and J. DeSteese, “Descriptive Model of Generic WAMS,” Pacific Northwest National Laboratory,Richland, WA., Tech. Rep., 2007.

[8] L. Vanfretti, V. H. D, L. Nordström, and J. Gjerde, “A smart transmission grid for Europe: Research challenges in developing grid enabling technologies,” in *IEEE Power and Energy Society General Meeting*, july 2011, pp. 1 –8.

[9] T. Rauhala and P. Järventausta, “Testing the Quality of PMU Output Data Based Subsynchronous Damping Analysis in Real-Time Simulation Environment,” in *International Conference on Power Systems Transients (IPST ’07)*, june 2007, pp. 1 –8.

[10] H. Kang, B. Cvorovic, C. Mycock, D. Tholomier, and R. Mai, “PMU simulation and application for power system stability monitoring,” in *IEEE/PES Power Systems Conference and Exposition, 2009 (PSCE ’09)*, march 2009, pp. 1 –7.

[11] “IEEE Standard for Synchrophasors for Power Systems,” *IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995)*, pp. 1–57, 2006.

[12] J. Stamp, V. Urias, and B. Richardson, “Cyber security analysis for the power grid using the virtual control systems environment,” in *IEEE Power and Energy Society General Meeting,*, july 2011, pp. 1 –4.

[13] Macrodyne, Inc., *Macrodyne Model 1690 PMU Disturbance Recorder*, 4 Chelsea Place, Clifton Park, NY 12065.

[14] K. Martin and J. Carroll, “Phasing in the technology,” *IEEE Power and Energy Magazine*, vol. 6, no. 5, pp. 24–33, September-October 2008.

[15] B. Kasztenny, W. Premerlani, and M. Adamiak, “Synchrophasor Algorithm Allowing Seamless Integration with Today’s Relays,” *IET 9th International Conference on Developments in Power System Protection*, pp. 724–729, March 2008.

[16] eMEGAsim PowerGrid Real-Time Digital Hardware in the Loop Simulator — Opal RT, [Online]. Available: http://www.opal-rt.com/.

[17] eMEGAsim Brochure, [Online]. Available: http://www.opal-rt.com/.

[18] Simulink® version 7.7. The MathWorks Inc., 2011.

[19] K. Yocum, E. Eade, J. Degesys, D. Becker, J. Chase, and A. Vahdat, “Toward scaling network emulation using topology partitioning,” in *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*, oct. 2003, pp. 242 – 245.

[20] M. Chenine, L. Vanfretti, S. Bengtsson, and L. Nordström, “Implementation of an experimental wide-area monitoring platform for development of synchronized phasor measurement applications,” in *IEEE Power and Energy Society General Meeting, 2011.*, july 2011, pp. 1 –8.

[21] Kalkitech Smart Grid Solutions, [Online]. Available: http://www.kalkitech.com/.

[22] The Open Source Phasor Data Concentrator, [Online]. Available: http://openpdc.codeplex.com/.

[23] The MathWorks, Inc., “Instrument control toolbox 2.11 data sheet,” 2011.

[24] PMU Connection Tester, [Online]. Available: http://pmuconnectiontester.codeplex.com/.

Ahmad T. Al-Hammouri is an Assistant Professor in the Department of Network Engineering and Security at Jordan University of Science and Technology. He received the B.S. degree with first-class honors in Electrical Engineering from Jordan University of Science and Technology, Irbid, Jordan, in 2001; and the M.S. and the Ph.D. degrees in Computer Engineering from Case Western Reserve University, Cleveland, Ohio, in 2004 and 2008, respectively. He has held research positions at Case Western Reserve University’s Netlab. During the Summer of 2011, he was a Visiting Research Associate with the Department of Industrial Information and Control Systems, KTH—Royal Institute of Technology, Stockholm, Sweden. His Research interests are in the areas of cyber-physical systems, smart power grid, Internet congestion control, and middleware for real-time sense-and-respond systems.

Lars Nordström (M’06) received the M.Sc. degree in electrical engineering and the Ph.D. degree in industrial control systems, both from KTH—The Royal Institute of Technology, Stockholm, Sweden. He is an Associate Professor in Power System Management and Director of the Swedish Centre of Excellence in Electric Power Engineering, an industry-university research center, located at KTH. His area of research is power systems management and related information exchange, including application of decision theory on information system architectures and the application of ICT to power system problems. Dr. Nordström is chairman of IEC TC57’s Swedish committee, and a member of the IEC’s Strategic group on Smart grids as well as National Cigre SC D2 representative for Sweden.

Moustafa Chenine (S’07) received his B.Sc. with a major in Computer Information Systems from The American College of Greece in 2004. He then joined KTH — The Royal Institute of Technology where he worked towards his M.Sc. (2006) in Information Systems and Licentiate of Engineering in Electrical Engineering (2009). He is a PhD Student at the department of Industrial Information and Control Systems at KTH, doing research in the field of ICT Infrastructure optimizations for Smart Grid systems, specifically focusing on interoperability, security and performance aspects of Wide Area Monitoring and Control Systems.

Luigi Vanfretti (M’10) is an Assistant Professor at the Electric Power Systems Division, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden. He received the Electrical Engineering Degree from *Universidad de San Carlos de Guatemala* in June 2005. He was a visiting researcher at the Department of Electronics and Electrical Engineering of The University of Glasgow, Scotland in 2005. He received his M.S. in 2007 and Ph.D. in 2009, both in Electric Power Engineering, from Rensselaer Polytechnic Institute. His research interests are modeling, dynamics, stability and control of power systems; applications of PMU data, and open source software for power system engineering education.

Nicholas Honeth (S’10) received the M.Sc. degree in computer science from Chalmers University of Technology, Gothenburg, Sweden and the B.Sc, degree in electrical and computer engineering from the University of Cape Town, Republic of South Africa. He is currently a Ph.D. student at the department of Industrial Information and Control Systems at KTH - The Royal institute of Technology, Stockholm, Sweden. His chief interests are in intelligent control systems for electrical distribution networks.. . .

Rujiroj Leelaruji (S ’09) received his B.Sc. degree in electrical engineering from Sirindhorn International Institute of Technology (SIIT), Thailand, in 2004, and the M.Sc. degree in electric power engineering from Royal Institute of technology (KTH), Stockholm, Sweden, in 2007. He is currently a Ph.D. student at the Royal Institute of Technology.