Delft University of Technology

Synchro-Measurement Application Development Framework
An IEEE Standard C37.118.2-2011 Supported MATLAB Library

Naglic, Matija; Popov, Marjan; van der Meijden, Mart A.M.M.; Terzija, Vladimir

Synchro-Measurement Application Development Framework: An IEEE Standard C37.118.2-2011 Supported MATLAB Library

Matija Naglic, *Student Member, IEEE*, Marjan Popov, *Senior Member, IEEE*,
Mart A. M. M. van der Meijden, *Member, IEEE*, and Vladimir Terzija, *Fellow, IEEE*

Abstract—Electrical power system monitoring, protection, operation, and control schemes are undergoing significant changes toward the next generation fully automated, resilient, and self-healing grids. At present, there still exists a lack of available user-friendly tools for the synchronized measurement technology supported application design. This paper presents a synchro-measurement application development framework (SADF) to promote a simplified design and thorough validation of synchro-measurement (IEEE Standard C37.118.2-2011) supported user-defined applications under realistic conditions. The proposed SADF supports online receiving of a phasor measurement unit (PMU) or phasor data concentrator (PDC) provided data stream and enables simultaneous use of processed machine-readable synchro-measurements in advanced user-defined applications. This paper fills the scientific gap between the IEEE Standard C37.118.2-2011 specifications and its implementation by proposing a novel robust communication technique and efficient synchro-measurement data parsing methodology. As a proof of concept, the proposed SADF is implemented as a novel open-source MATLAB library. Combining this library with MATLAB's signal processing and visualization functions allows mastering the design and validation of complex wide-area monitoring, protection, and control applications, as well as PMU/PDC performance and compliance verification. Finally, this paper verifies the proposed library against the standard specifications, assesses its interoperability and performance via a cyber-physical simulation platform, and presents online voltage magnitude monitoring as an example application.

Index Terms—IEEE Standard C37.118.2-2011, MATLAB, online processing, phasor measurement unit (PMU), wide-area monitoring, protection, control (WAMPAC).

M. Naglic and M. Popov are with the Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: m.naglic@tudelft.nl).

M. A. M. M. van der Meijden is with the Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with TenneT TSO, 6812 AR Arnhem, The Netherlands.

V. Terzija is with the School of Electrical and Electronic Engineering, University of Manchester, Manchester M13 9PL, U.K.

I. Introduction

THE smart grid technological advances [1], [2] have created opportunities to design the next generation of monitoring, protection, and control schemes of electrical power system (EPS) [3]. As a result, both research and industry are moving forward toward advanced wide-area monitoring, protection, and control (WAMPAC) systems [4]. The WAMPAC system is favorable to ensure efficient, more resilient, and secure operation of EPS through sophisticated utilization of smart grid components such as intelligent sensors, actuators, and state-of-the-art information and communications technology (ICT). Typical WAMPAC applications include, but are not limited to recording and visualization of dynamic system response, online EPS model validation and state estimation, detection of security and stability margins, wide-area protection (fault/event detection), wide-area control (intentional controlled islanding, wide-area oscillation damping, frequency control), and actions to facilitate maximum stable system loading (dynamic thermal rating, demand-side management) [5].

Typically, WAMPAC system uses the advanced synchronized measurement technology (SMT) as a key building block to deliver time synchronized measurements (synchro-measurements) of electrical quantities from grid wide remote locations [4]–[6]. Supported by precise time synchronization, the SMT comprises intelligent electronic devices (IEDs) or phasor measurement units (PMUs), and phasor data concentrators (PDCs), connected over ICT infrastructure into a hierarchically organized network [7] (further detailed in Section III).

A. Paper Motivation

Over the recent years, IEEE Standard C37.118 [8]–[11] has evolved to one of the widely deployed communication protocols for synchro-measurement data exchange among SMT supported power system components. The typical IEEE Standard C37.118-based data exchange is in machine-readable format (binary representation) to efficiently transfer the data between electronic devices [8], [11]. The machine-readable format is very difficult to interpret for humans. Researchers often need to deal with the lack of available user-friendly tools

for online synchro-measurement data acquisition to design and validate online grid-saving applications. This is one of the reasons why many researchers prefer to work on preprocessed measurements, typically the comma-separated value format and stored by a PDC, primary designed for offline analysis. However, by taking this approach, the measurements are available in the user-defined applications with some processing delay with respect to the write/read operations to/from a file or database. This can have a negative impact on the performance of the closed-loop control WAMPAC applications, which require a fast response on the present system conditions, and can consequently endanger secure EPS operation. The proposed synchro-measurement application development framework (SADF) enables a seamless integration between the SMT supported power grid and advanced user-defined WAMPAC applications.

Additionally, many of the available synchro-measurement-based literature papers make use of conventional software simulated bus measurements rather than actual PMU provided synchro-measurements. It is important to note that actual PMU provided synchro-measurements are affected by transients [12] and system-imposed high-frequency components [13], instrument transformers errors [14], and PMU analog filtering, sampling, and windowing [15]–[18]. Therefore, actual PMU-provided synchro-measurements can dramatically differ from typical bus measurements, obtained from conventional software simulations. As a result, the grid-saving algorithms' performance may negatively differ in case actual synchro-measurements are utilized and can, in worst case scenario, lead to complete EPS black-out. This paper enables a simplified and online use of synchro-measurements in advanced user-defined WAMPAC applications to facilitate applications' extensive validation under realistic conditions in an isolated and flexible simulation environment.

B. Paper Contributions

This paper proposes an SADF, a novel IEEE Standard C37.118.2-2011 supported MATLAB software library used for online receiving and parsing of the IEEE Standard C37.118-2005 and C37.118.2-2011 specified messages. The new library can be used for a simplified design and validation of advanced WAMPAC applications under realistic conditions. Certainly, the library can be utilized as a part of the PMU/PDC metrological characterization and the IEEE Standard C37.118 compliance validation [19], as a tool to receive and compare the "true" synchro-measurements (produced by a reference PMU) with the estimated synchro-measurements produced by a PMU under test. In short, the main contributions of this paper include the following.

1) Implementation procedures regarding the missing link between the IEEE Standard C37.118.2-2011 specifications and its implementation to comply with the complete IEEE Standard C37.118.2-2011 requirements. The focus is on the following.
   a) Robust data receiver communication technique with integrated fallback procedures.
   b) Efficient online receiving and parsing methodology of the encapsulated machine-readable configuration and data frames.
2) Open-source available implementation of the proposed SADF as a new software library for the MATLAB programming environment. The stress is put on the real-time and simplified use of the synchro-measurements in user-defined applications.
3) Detail analysis of the SADF library compliance with the IEEE Standard C37.118.2-2011 requirements, and performance evaluation using real-time cyber-physical simulation platform.

C. Paper Organization

The rest of this paper is organized as follows. Section II addresses the related work. Section III presents the related background. Section IV presents the SADF, its library structure, communication, and parsing methodology. Section V presents the SADF compliance, interoperability, performance evaluation, and demonstrates the utilized cyber-physical simulation platform and example application. Finally, in Section VI, this paper ends up with conclusion.

II. RELATED WORK

Over the recent years, the SMT solutions and applications have gained increased attention from research and industry. A significant amount of work has been done with respect to PMU synchrophasor and frequency estimation technique [20] to meet and exceed the IEEE Standard requirements [10], [21]. The remaining challenge is to design a computationally efficient, fast response, and accurate PMU estimation algorithm for off-nominal system frequency conditions. An OpenPMU project [22] was introduced as an open-source platform for development of new PMU instruments. With the development of SMT supported devices, there was a parallel development of sophisticated applications in monitoring, protection, and control [5], utilizing advantages of synchro-measurements. Despite the evident interest of research community for synchro-measurement supported applications, there still exists a lack of available user-friendly tools for simplified and online WAMPAC application design as well as PMU/PDC performance and compliance verification. Pandit and Khandeparkar [23] and Sandi *et al.* [24] implemented the IEEE Standard C37.118-2005 for PMU/PDC emulation purposes as an open-source application using C and Python programming languages, respectively. However, their published work has several limitations that include: 1) the lack of the IEEE Standard C37.118.2-2011 support; 2) the lack of user datagram protocol (UDP) connection protocol support and spontaneous communication mode [23]; and 3) the detailed performance evaluation is missing. The Grid Protection Alliance [25] developed a "Time-Series Framework" for managing of synchro-measurement data and implemented as an open-source library in .NET, Java, and C++ programming environments. Despite the big potential, the library has not been vastly utilized in research, mainly due to proficient programming required. The PMU connection tester and OpenPDC [25] are currently being
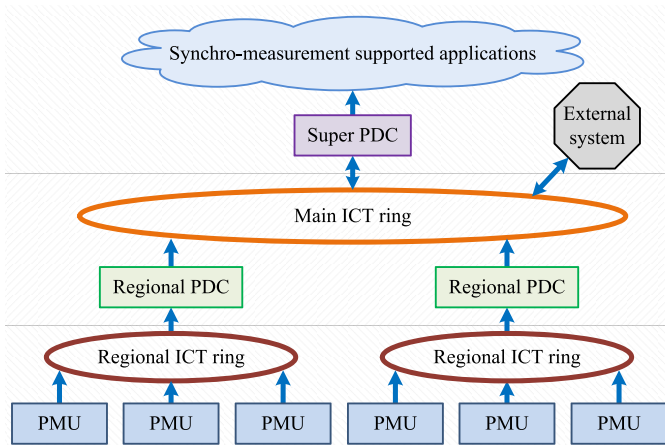
Fig. 1.   SMT infrastructure.

one of the most used open-source applications built on the top of time-series framework and designed for PMU connectivity testing and gathering of synchro-measurement streams, respectively. For the first time, a LabVIEW-based synchrophasor software development toolkit (named as BableFish) was recently developed [26] to online receive the IEEE Standard C37.118.2-2011 data stream in the LabVIEW programming environment, by using transmission control protocol/internet protocol (TCP/IP) and commanded mode only. The proposed work fills the gap and presents a complete IEEE Standard C37.118.2-2011 supported implementation in the MATLAB programming environment. It is important to point out that the proposed SADF library supports both the commanded and spontaneous communication modes. In the first case, one can select either TCP/IP or UDP, or a combination of both [TCP/IP for sending/receiving command (CMD), header (HDR), configuration (CFG-1, CFG-2, CFG-3) frames, and UDP for sending/receiving DATA frames] for configuration and measurement data exchange. In the second case, UDP is supported to facilitate fast and efficient transmission of configuration and measurement data exchange, critical for fast response WAMPAC applications. By using the proposed SADF library, one can perform a detail IEEE Standard C37.118.2-2011 compliance and performance verification of PMU/PDC device under test.

III. BACKGROUND

A. Synchronized Measurement Technology Architecture

Supported by precise time synchronization, the SMT comprises specialized IEDs, connected into a hierarchically organized ICT infrastructure as illustrated in Fig. 1.

The hierarchically bottom consists of geographically dispersed PMUs, typically located at EPS substations. A PMU provides measurements of voltage and current magnitude and angle (called as synchrophasor) per phase and positive, negative, and zero-sequence components; system frequency; and rate of change of frequency (ROCOF) at precisely determined intervals, typically 50/60 times/s. Additionally, it sends analog sample measurements (single point-on-wave, rms of

analog input, and peak of analog signal) and Boolean status words from supported analog and digital input channels, respectively. PMUs send their synchro-measurements over supporting ICT infrastructure [7] to regional PDCs. A PDC works as data concentrator, which collects and preprocesses the synchro-measurements from numerous PMUs and/or PDCs. Typically, it performs quality checks, time aligns, and merges synchro-measurements into a single data stream, further sent to a hierarchically higher level PDC (super PDC). Similarly, a super PDC collects and preprocesses the synchro-measurements received from PMUs and/or PDCs and sends them in a single data stream to application data acquisition center and/or to an external SMT supported system.

B. IEEE Standard C37.118.2-2011 Overview

Over the recent years, the IEEE Standard C37.118 family has evolved as one of widely deployed among SMT supported power system components. First introduced in 2005, the IEEE Standard C37.118-2005 [8] specifies both the measurement and communication requirements. In 2011, the IEEE Standard C37.118-2005 was improved to cover transient performance requirements and split into two parts. The first part, the IEEE Standard C37.118.1-2011 [9], focuses only on the PMU measurement and performance requirements. In 2014, only this part was further updated (IEEE Standard C37.118.1a-2014 [10]) with modified and relaxed PMU performance requirements. Next, the second part IEEE Standard C37.118.2-2011 [11] specifies communication requirements for real-time data exchange among measurement, concentration, and application devices. It is important to note that the IEEE Standard C37.118.2-2011 is an extended version, which is backward compatible with the communication part of its predecessor, the IEEE Standard C37.118-2005. This paper focuses on the implementation issues regarding the latest IEEE Standard C37.118.2-2011 version, therefore, only the communication part is further addressed.

1) Communication Infrastructure: The IEEE Standard C37.118.2-2011 defines synchro-measurement messaging framework (application layer) requirements, including message types, use, contents, and data formats. It does not put any restrictions on utilized communication medium and transport protocol, since they are typically application dependent. Primarily, the standard was designed for RS-232 serial communication, but the past implementations show great success of fast fiber optic supported IP telecommunication networks, utilizing TCP/IP and/or UDP.

a) Transmission control protocol: TCP/IP is connection-oriented (one-to-one and bidirectional) data exchange protocol and is suited for applications that require high reliability of data transfer; meanwhile, transmission time is relatively less critical. The TCP/IP stack guarantees that data arrive to the final destination (application layer), remain intact, and in the same order in which it was sent. As a consequence, a continuous PMU/PDC measurement data stream utilizing TCP/IP may experience retransmission interruptions in the case of corrupted and out-of-order received TCP/IP packets [27].
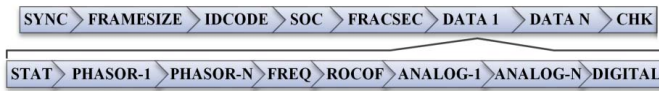
Fig. 2. DATA message frame structure.

b) User datagram protocol: In contrast, UDP is simpler connectionless (one-to-one/many and unidirectional) data exchange protocol and is suited for applications that must meet fast and efficient transmission of data. UDP does not guarantee arrival of the packets to the final destination and has no inherent delivery order as all packets are independent of each other. If data ordering and recovery are required, it has to be managed by application layer on server and client side [28]. UDP is often preferable during SMT application operational period, since data transmission is faster (smaller overhead size) and continuous compared to TCP/IP.

2) Communication Mode and Type: Synchro-measurement data exchange is client–server oriented, using either commanded or spontaneous mode of communication. In the commanded mode (bidirectional server–client communication), a hierarchy higher level PDC or application data acquisition (further addressed as receiver) initiates a connection with a PMU/PDC data source device (further addressed as sender) and takes control of configuration and measurement data exchange. Hereby, a receiver sends specific data request commands to a sender, which afterward replies with requested data. On the other hand, in the spontaneous mode (unidirectional server–client communication), configuration and measurement data exchange is initiated by a sender in a continuous way without any negotiation with a receiver.

Moreover, synchro-measurement data exchange can be either unicast (between one sender and one receiver), multicast (between one sender and group of receivers, via one data stream), or broadcast (between one sender and all network nodes). Typically, unicast is utilized for one domain network, while multicast is suitable for synchro-measurement data exchange between different synchro-measurement domains. Broadcast is utilized only in special cases since it is resource expensive.

3) Frame Structure and Message Types: The IEEE Standard C37.118.2-2011 defines basic frame structure and four types of messages for synchro-measurement configuration and data transmission. These are *data, configuration, header,* and *command*. Typically, the first three message types are sent from a sender as response on received command message, initiated by a receiver.

a) Frame structure: All four message types start and end with the predefined header and footer, respectively, forming together a message frame. Header and footer contain a set of parameters, referred as fields and illustrated in Fig. 2.

Each message frame first starts with a SYNC (2-byte) field, used to identify message frame type (bits 6-4), and utilized synchrophasor standard version number (bits 3-0). It continues with a FRAMESIZE (2-byte), defining total frame size in bytes. Further, an IDCODE (2-byte) field defines a unique identification number of a frame originating

device or destination device in the case of command frame. Next, a second-of-century (SOC) (4-byte) and a fraction-of-a-second (FRACSEC) (4-byte) fields together define measurement timestamp or transmission time for the non-DATA frames. Hence, the timestamp is given as

$$\text{TimeStamp} = \text{SOC} + \text{FRACSEC}/\text{TIME\_BASE} \qquad (1)$$

where TIME_BASE integer determines the maximum time resolution in ticks.

Additionally, a FRACSEC contains 8-bit time quality flags to indicate the quality of time being measured plus indication of leap second status. Finally, a content of a message (DATA1 - N) field is encapsulated and terminated with a CHK (2-byte) field, containing cyclic redundancy check (CRC) for transmission error detecting.

b) Data message: Data (DATA) message frames are primarily used during synchro-measurement data exchange and contain a set of measurements at precisely determined moment of time. Each DATA message contains at least one encapsulated PMU measurement block (a set of measurements and control flags of a single PMU measurement stream). A PMU device can send only one (its own) encapsulated measurement block, while a PDC device typically sends multiple encapsulated measurement blocks in a single DATA frame from hierarchically underlying PMUs and/or PDCs. It is important to note that a DATA frame content cannot be interpret without information obtained from a configuration message frame beforehand.

As illustrated (Fig. 2), a single measurement block (DATA-1) includes time and measurement quality information (STAT), estimated synchrophasors (PHASOR), system frequency (FREQ), ROCOF, analog values (ANALOG), and digital Boolean status words (DIGITAL).

c) Configuration message: A CFG message frame is used to inform a receiver about a sender-related capabilities, metadata, and configuration structure of DATA frames being reported. Typically, it is sent during connection initialization period and before first DATA is sent to permit the receiver to properly separate and interpret measurement identities of DATA frames. The standard defines three types of configuration frames: CFG-1, CFG-2, and CFG-3 (optional).

A CFG-1 denotes a PMU/PDC full reporting capability and is used in exceptional occasions. Further, a CFG-2 and CFG-3 indicate a PMU/PDC configuration structure of DATA frame currently being reported. It includes station identification number and name, phasor and analog total number and channel names, measurements reported formats and scaling factors, default digital status positions, reporting rate, nominal system frequency, and CFG change number identification. Additionally, a CFG-3 introduces variable frame length, metadata related to phasor and analog values offset, a synchrophasor estimation algorithm window length and group delay, and station geolocation properties.

d) Header message: A header (HDR) message frame is primary used to inform one about a PMU installation location, measurement properties, filtering applied, reporting rate, and other user-defined metadata. Unique about the HDR message frame is that its content is in human-readable ASCII format.
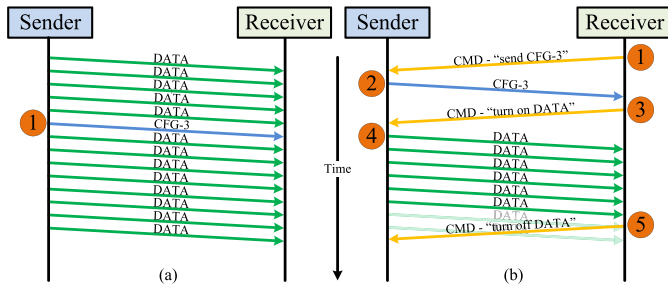
Fig. 3. Communication data exchange mode. (a) Spontaneous mode. (b) Commanded mode.

Fig. 4. SADF.

e) Command message: A command (CMD) message frame is used exclusively in the commanded mode to send instruction from a receiver to a sender. Typically, after a sender receives the instruction, it responds with a requested action. Commands which a receiver can send to a sender are "send CFG-1," "send CFG-2," "send CFG-3," "send HDR," "turn ON DATA," and "turn OFF DATA."

4) Typical Data Exchange: All message frame types (except HDR) are transmitted in machine-readable format (binary representation) to efficiently transfer data between IEDs. Data transmission of a message frame always starts with a SYNC and ends with a CHK field (Fig. 2). Data are transmitted using big endian, i.e., the most significant byte first.

In the spontaneous communication mode, a sender typically sends DATA frames to a receiver continuously, as illustrated in Fig. 3(a). DATA frames are sent with a reporting rate as specified in sender configuration settings. Additionally, for every minute rollover it sends a CFG-2/CFG-3 frame (1) to inform a receiver about DATA message frames structure currently being reported.

On the other hand, in the commanded communication mode [Fig. 3(b)], a receiver typically starts with sending a "send CFG-2" or "send CFG-3" (1) CMD message frame to a sender. After a sender receives a command, it responses with a requested CFG (2) message frame. Furthermore, after a receiver successfully receives a CFG frame, it sends to a sender a "turn ON DATA" (3) command. Afterward, a sender starts sending DATA (4) frames to a receiver in continuous and equally time-spaced intervals, as specified in the device configuration (reporting rate). Finally, in case, data exchange should be terminated, a receiver sends a "turn OFF DATA" (5) command to a sender, which stops sending data frames immediately after receiving of the command.

IV. SYNCHRO-MEASUREMENT APPLICATION DEVELOPMENT FRAMEWORK

SADF is a novel MATLAB-supported library to facilitate simplified design and online validation of advanced closed-loop control WAMPAC applications under realistic conditions.

The proposed SADF enables a seamless integration between the SMT supported electric power system and synchro-measurement-supported user-defined applications. This is done by online receiving and parsing of the IEEE Standard
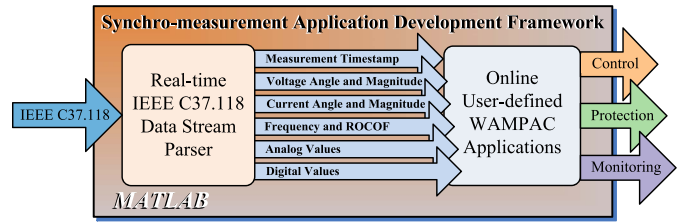
C37.118-2005 and C37.118.2-2011 specified machine-readable messages into a human-readable MATLAB format. Combining this library with MATLAB's signal processing and visualization functions allows mastering the design and validation of complex WAMAPC applications, as illustrated in Fig. 4.

MATLAB programming suite is a powerful engineering computing toolbox, supported by various built-in mathematical, signal processing, and visualization functions. It is a cross-platform compatible (Windows, Mac, and Unix) and can be interfaced with programs written in C, C++, Java, Fortran, and Python. MATLAB has been a de facto programming language for research worldwide and has an extended online community support [29].

A. Library Structure

The proposed SADF library comprises developed functions to online receive and in real-time validate, parse, interpret, and store synchro-measurements into a human-friendly MATLAB workspace format, making it available for the user-defined WAMPAC applications (Fig. 5). It includes a robust communication technique with fallback procedures for data exchange establishment, and an efficient processing methodology for online parsing of the IEEE Standard C37.118-2005 and C37.118.2-2011 defined messages.

1) Connection Establishment: The SADF can establish a TCP/IP or UDP unicast connection with a PMU or PDC, typically by utilizing a host platform-supported wired Ethernet or wireless Wi-Fi technologies. In both cases, the MATLAB instrument control toolbox Java language-supported socket interfaces are utilized for data exchange.

a) Communication mode: It is important to point out that the proposed SADF library supports both the commanded and spontaneous communication modes. In the first case, one can select either TCP/IP or UDP, or a combination of both (TCP/IP for sending/receiving CMD, HDR, CFG-1, CFG-2, CFG-3 frames, and UDP for sending/receiving DATA frames) for configuration and measurement data exchange. In the second case, UDP is supported to facilitate fast and efficient transmission of configuration and measurement data exchange, critical for fast response WAMPAC applications.

(i) Commanded mode: In the commanded mode (Fig. 5), the SADF initiates a connection establishment procedure to obtain a PMU/PDC configuration and initiate measurement data transmission. This procedure is divided into multiple steps with an implemented fallback algorithm to cover
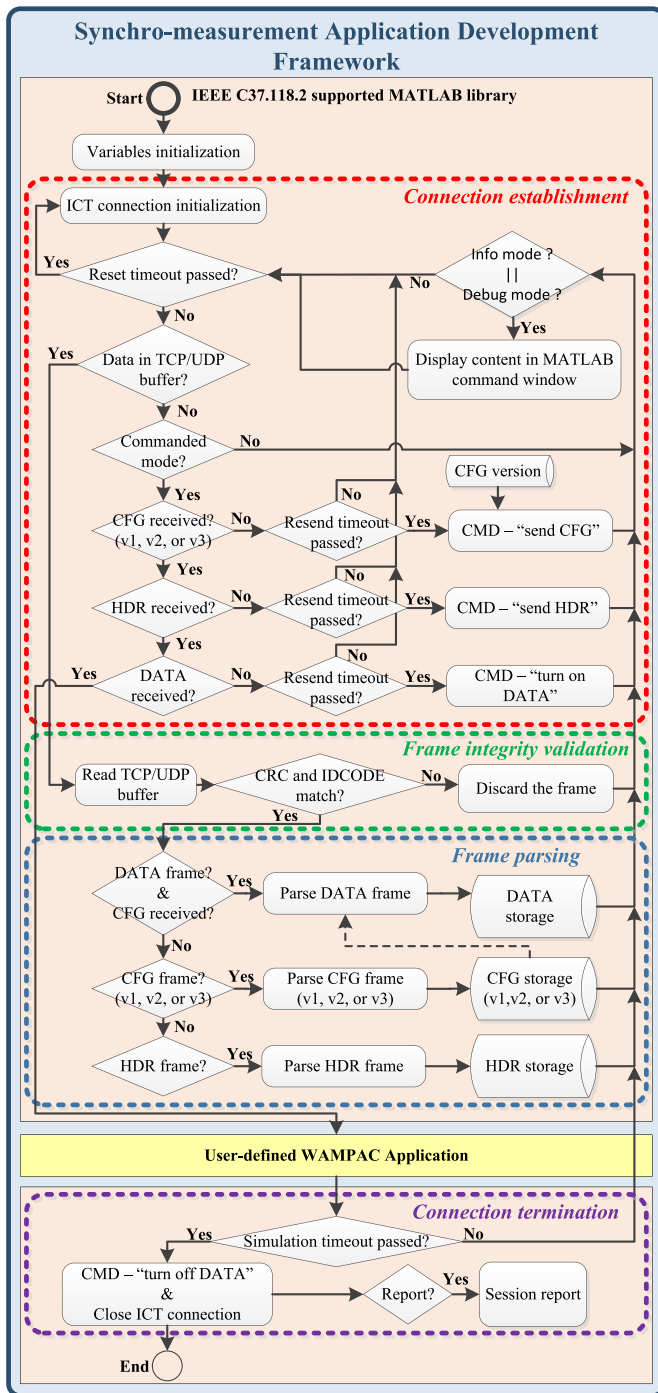
Fig. 5.   SADF library structure.

situations, when a PMU/PDC does not support requested CMD frame, or ICT interruptions.

First, the SADF sequentially sends the CMD frame requests to obtain an HDR, following CFG-1, CFG-2, or CFG-3 frame from a PMU/PDC. It may happen that a PMU/PDC does not support the requested CMD frame type. In this case, the PMU/PDC typically responds with a CFG-2 frame or does not respond at all. It may also happen that due to ICT interruptions, the PMU/PDC does not receive the sent CMD frame request. In all the cases, the specific CMD request is resend every 2 s and up to 5 times. In case the expected

response from PMU/PDC is still not received, the SADF will automatically continue with a next step of the connection establishment procedure, since crucial DATA exchange can be still established, even without information obtained from an HDR or CFG-1 frame. Lastly, in the case of unreceived response on sent "send CFG-2/CFG-3" or "turn ON DATA" (if applicable) CMD frame request, the ICT connection with the PMU/PDC is re-established. Afterward, the connection establishment procedure with the PMU/PDC is reinitiated. In this way, the SADF library does not halt in the case of unexpected events.

It may also happen that a PMU/PDC starts sending DATA frames immediately after the establishment of an ICT connection or before the SADF requests a DATA exchange. In this case, a "turn OFF DATA" CMD frame request is immediately sent to the PMU/PDC in order to terminate the existing DATA exchange and proceed with a normal connection establishment procedure. Any received DATA frame is flagged and discarded, since it cannot be interpret without information obtained from a CFG-2/CFG-3 frame beforehand.

Finally, after the SADF successfully receives at least a CFG-2/CFG-3 frame, a "turn ON DATA" command is sent. Afterward, an online processing of received DATA frames is automatically initiated.

In case, data exchange between the SADF and PMU/PDC stops for any reason for more than 10 s, and the ICT connection and connection establishment procedure is re-established and reinitiated, respectively.

(ii) Spontaneous mode: In the spontaneous mode, the SADF opens a user-defined UDP port and waits for a PMU/PDC to initiate frame exchange. First, the SADF waits to receive a CFG-2/CFG-3 frame, which is typically sent by a PMU/PDC each minute rollover. During this waiting period any received DATA frame is flagged and discarded, since it cannot be interpret without information obtained from a CFG-2/CFG-3 frame beforehand. After a successful receiving of a CFG-2/CFG-3 frame, an online processing of data frames is automatically initiated.

2) Frame Integrity Validation: The SADF performs two-step frame validation on all received frames to detect data transmission errors and verify a message frame source.

First, a CHK field value of any received message frame is compared against a newly determined CRC from a received entire frame block. The CRC is calculated based on a CRC-CCITT error detection scheme and implemented in C-code for processing speed increase. The CRC-CCITT is computed by the following polynomial function:

$$g(x) = x^{16} + x^{12} + x^5 + 1 \tag{2}$$

with a seed value of 0xFFFF and no final mask [10]. In case the compared CRC values do not match, then the frame certainly contains bit error, hence it is flagged and discarded.

Finally, the IDCODE field of the successfully received frame is compared against the user-defined IDCODE in the SADF configuration file. In case, the IDCODE values do not match, then the received frame is flagged and discarded.

3) Frame Parsing: The SADF automatically identifies and parses in real time the IEEE Standard C37.118-2005 and

C37.118.2-2011 supported configuration and measurement message frames of any reporting rate provided.

Received frame type identification is based on a SYNC field, unique to each frame type. First, a SYNC value of a received frame is compared against an identification code of DATA, HDR, CFG-1, CFG-2, and CFG-3 message frame. After a positive match, the message frame content is read using adequate parsing functions. A content of each message frame is stored into the MATLAB workspace using structure array format and using a variable name corresponding the frame type for ease of use. Individual frame content identities are stored into child structure arrays with self-explanatory names.

To get an insight into the standard specified configuration and measurement data exchange, the SADF library supports three modes of operation.

1) *Info mode*: Only the basic information regarding send and received frames with corresponding timestamp is displayed in the MATLAB command window.
2) *Debug mode*: Information regarding send and received frames with detailed message frame content in user-friendly format is displayed in the MATLAB command window. In addition, all the raw data exchange between the sender and the receiver is stored for ease of debugging purposes.
3) *Silent mode (default)*: No information is displayed in the MATLAB command window. This mode is typically used during normal operation for processing speed increase.

Lastly, the SADF end-of-session report optionally displays in MATLAB command window a summary regarding successfully received, duplicated, out-of-order, and corrupted message frames, all grouped by frames type.

a) DATA frame interpretation: In case, TCP/IP is used for data transmission, the protocol itself guarantees that all message frames arrive in the same order as were sent; hence, no additional DATA stream postprocessing is required. On the other hand, when UDP is used, the SADF takes control over out-of-order received, duplicated, or missing data frames. In the first case, the SADF time aligns received DATA frames, and stores measurements in right time sequence. In the case of duplicated or missing DATA frames, the measurements are discarded or stored with unrepresentable (NaN) values, respectively.

The SADF extracts from each received DATA frame an encapsulated PMU measurement block; time and measurement quality information, which are condensed in 8-bit FRACSEC time quality flags and 2-bytes STAT field respectably. These include the following:

1) leap second status;
2) time accuracy;
3) measurement data validity;
4) time source synchronization status;
5) measurement device operation status;
6) data sorting type;
7) trigger status and reason; and
8) configuration change notification.

The standard defines 16-bit integer and 32-bit floating point formats for synchro-measurements (phasors, analog values, frequency, and ROCOF) data exchange. Most often a 16-bit integer format is used to facilitate efficient transmission, but requires additional multiplication with a scaling factor. Phasor measurements can be transmitted in rectangular (real and imaginary components) or polar (magnitude and angle) format. Additionally, CFG-3 message frame contains postprocessing measurement adjustment factors related to magnitude or angle offset corrections (in the case of phasors and analog values). To facilitate ease of use, the SADF automatically converts synchro-measurement data of any format into a user-friendly double format. Hereby, phasor measurements are stored in the polar format; the magnitude is in engineering units and the angle is in degrees, all with applied correction factors (if applicable). Similar applies to the analog values, frequency, and ROCOF measurements.

b) CFG frame interpretation: The SADF can receive CFG-1, CFG-2, and CFG-3 configuration frame types. Since PMU/PDC configuration settings can change over time, the SADF supports receiving of multiple CFG frames within a single session. Hereby, only CFG frames containing newer structure as already received one are to be stored and used for further DATA frame processing. The remaining (duplicated) CFG frames are flagged and discarded. A CFG-3 frame is parsed to obtain the following:

1) the number of PMU blocks included in the DATA frame;
2) the number of phasors, analog and digital values for each PMU block included;
3) PMU station, phasor, analog channel naming;
4) digital channel;
5) measurement format, conversation, and offset factors;
6) station geolocation properties;
7) estimation algorithm window length and group delay;
8) configuration change count;
9) DATA transmit rate; and
10) nominal system frequency.

c) HDR frame interpretation: The SADF can receive the standard supported HDR frame. Its content is primary in the ASCII format and is typically used during PMU setup and verification purposes. The SADF supports receiving of multiple HDR frames within one session. By using "debug mode," its content can be displayed in the MATLAB command window.

d) CMD frame sending: In the commanded mode, an individual CMD frame content is sent (written) to the TCP/IP/UDP Java socket in an asynchronous mode. In this way, the MATLAB command line is not blocked for the duration of Java socket execution. Consequently, the user-defined application has more time available for its own operation (execution).

V. COMPLIANCE AND PERFORMANCE EVALUATION

The SADF library is primary designed for use in real-time SMT supported environment, facilitating an application online operation; therefore, the library real-time operation is crucial. In order to evaluate the SADF library performance as well interoperability with the IEEE Standard C37.118 compliant devices and accuracy of the provided data, a cyber-physical simulation platform has been used.
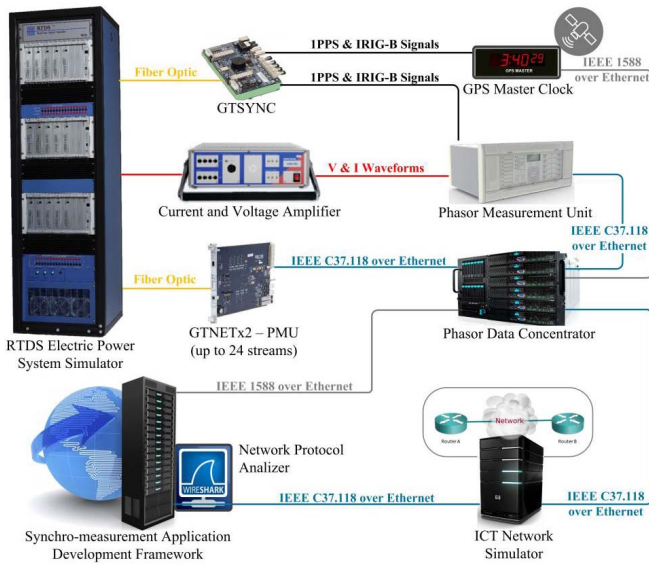
Fig. 6.   Cyber-physical simulation platform.

A. Cyber-Physical Simulation Platform

The cyber-physical simulation platform (Fig. 6) is comprised of the SMT supported power system model and associated ICT infrastructure [30]. A simple 50-Hz nominal system frequency power system model is run in the RTDS power system digital simulator to generate test signals, fed to the PMU input channels. A GTAO card is used to provide low-level analog current and voltage waveforms, first fed to an OMICRON CMS156 amplifier and further to an external ALSTOM P847 PMU (of class P with 2-cycle window length and 50-frames/s reporting rate). Additionally, a GTNETx2 card is used to simulate up to 24 independent PMU (of class P with 2-cycle window length and 50-frames/s reporting rate) streams. All the PMUs are first connected to the SEL-5073 PDC and OpenPDC, and further over the ICT network emulator to the SADF, as presented in Fig. 6. A WANem ICT network emulator is utilized to emulate communication delay, and jitter; data corruption, loss, and miss ordering of the PDC sent data messages.

The performance evaluation of the proposed MATLAB-supported SADF library is performed on a typical office computer (CPU: Intel Xeon E5 1620v2 - 3.70 GHz and memory: 16 GB RAM, OS: Windows 7) with installed Wireshark communication protocol analyzer and MATLAB-R2017a programming environment. The MATLAB is executed in real-time priority. All the platform components are precisely time synchronized using a GE-RT430 grand master clock, which provides the time accuracy within 100 ns (peak), with the time drift less than 0.1 ppm in case an external GPS reference signal is lost. Moreover, IRIG-B protocol-based timestamp and 1PPS time signals are fed to the GTSYNC card (used to synchronize RTDS PMUs) and the external PMU. Additionally, the IEEE 1588 Precise Time Protocol (PTP) is used to time synchronize over Ethernet the PDCs and PC running the SADF by using the Oregano Systems syn1588 PCIe card and Greyware Automation Domain Time II Client, respectively.

Fig. 7.   SADF output (in "debug mode") in the case of processed DATA frame.

B. IEEE Standard C37.118.2-2011 Compliance Verification

In order to develop a reliable WAMPAC application, it is prudent to confirm the SADF compliance with the standard specifications and SADF interoperability between the standard supported devices as well accuracy of provided data. First, the SADF has been connected directly with the external PMU to verify the typical data exchange during TCP/IP and UDP-commanded mode operation. Afterward the SADF was connected with the PDC, which was further connected with the external PMU and RTDS provided PMUs. Hereby, the SADF was tested to confirm UDP, TCP/IP, and mixed UDP&TCP/IP commanded mode as well UDP spontaneous mode of operation. Additionally, the output of the Wireshark network protocol analyzer and SADF library in "debug mode" have been compared to identify any inconsistencies in configuration and synchro-measurement data exchange. The SADF "debug mode" provides parsing of machine-readable message frames into extended human-readable format with interoperated information from the standard supported frames, as indicated in Fig. 7.

In this way, the SADF compliance and interoperability with the standards compliant devices was confirmed.

C. SADF Performance Evaluation

Synchro-measurement data latency in SMT supported system refers to a time delay between a PMU measurement

TABLE I
PERFORMANCE EVALUATION

| PMU/PDC configuration	Processing time [ms] | length (bytes)		
	DATA	CFG-2	CFG-3
1 block, 2 phasors, 1 freq, 1 ROCOF, floating point	0.77 | 42	1.64 | 94	1.88 | 152
1 block, 2 phasors, 1 freq, 1 ROCOF, integer	0.78 | 30	1.62 | 94	1.87 | 152
1 block, 12 phasors, 1 freq, 1 ROCOF, integer	1.09 | 70	2.11 | 294	2.63 | 442
1 block, 12 phasors, 2 analog, 2 digital, 1 freq, 1 ROCOF, integer	1.12 | 78	2.54 | 854	2.88 | 1044
2 blocks, 24 phasors, 4 analog, 4 digital, 2 freq, 2 ROCOF, integer	1.53 | 140	3.51 | 1684	/
4 blocks, 48 phasors, 8 analog, 8 digital, 4 freq, 4 ROCOF, integer	2.41 | 264	5.53 | 3344	/
24 blocks, 48 phasors, 24 freq, 24 ROCOF, integer	3.62 | 640	7.29 | 1704	/
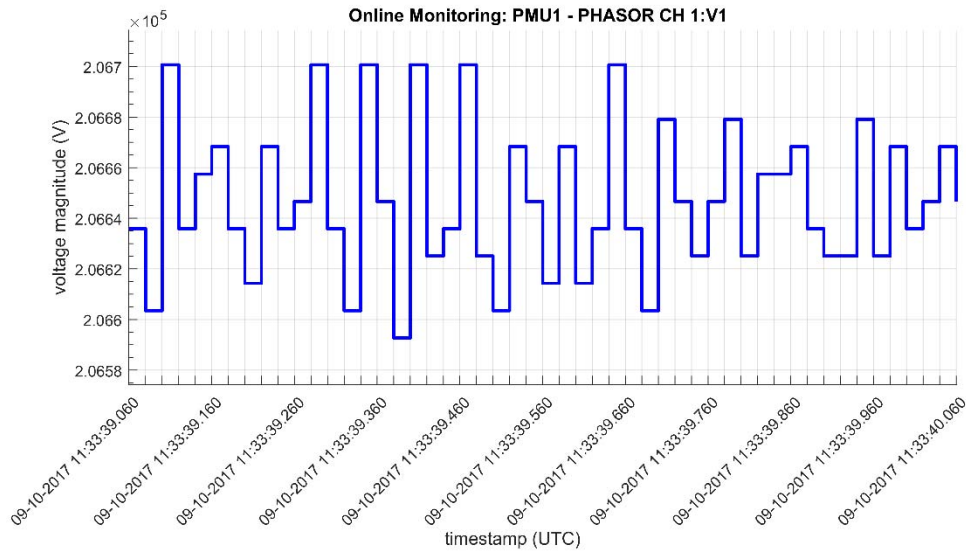
Fig. 8. Example application: Online voltage magnitude monitoring of the PMU station "*PMU1*," channel "*PHASOR CH 1:V1*," with 50-frames/s measurement reporting rate.

timestamp and an instant when a PMU measurement data are available in an end-user application [31]. It depends on different intermediate factors, such as PMU synchro-measurement estimation and processing performance, ICT capabilities, PDC processing performance, and application data acquisition performance [11]. The synchro-measurement data latency is a crucial factor for any WAMPAC system. It may affect the SMT system ability to provide real-time synchro-measurement data access fast enough to meet the WAMPAC application requirements [31].

Since synchro-measurement data latency is a variable system property and it depends on the utilized PMUs, PDCs, and supporting ICT network capabilities and conditions, only the SADF performance is further investigated.

The SADF performance evaluation is based on a processing time required to read individual message frame from a TCP/IP or UDP input buffer, parse its content, and store it into the MATLAB workspace. Hereby, processing time of different PMU configuration settings has been evaluated based on [10, Annex C, Table C.1] and the most commonly used configuration scenarios. Table I presents the median processing time per DATA, CFG-2, and

CFG-3 frame of total 100 received frames of equal type and size.

It can be summarized that the processing time depends on a frame type and sender configuration, which typically increases with the frame length (larger configuration). Furthermore, a DATA frame measurement format has direct impact on frame size and practically negligible effect on frame processing time. The evaluation of the time required to process a CFG-3 frame in case of multiple PMU blocks included in the PDC configuration has not been performed due to the PDC limitations. Additionally, the results presented in Table I contain latency introduced by unavoidable data reading from TCP/IP/UDP buffer of MATLAB Instrument Control Toolbox to MATLAB Workspace. To illustrate, the average time to read a single DATA or CFG-2/CFG-3 message frame takes around 0.31 or 0.78 ms, respectively.

Based on the performance evaluation, it takes ~ 0.77 ms to process a received DATA frame (1 block, 2 phasors, 1 freq, 1 ROCOF, and floating point). In this case, a maximum DATA receiving rate that the SADF library is capable of processing is theoretically limited to $\sim$1300 frames/s. However, the SADF library is confirmed to successfully

receive 250-frames/s DATA frame rate (1 block, 12 phasors, 2 analog, 2 digital, 1 freq, 1 ROCOF, and integer), which is the maximum data reporting rate of the RTDS PMUs. Due to the lack of suitable open-source tools, the performance benchmarking of the proposed SADF library was not possible. The available open-source implementations are either specific software dependent (LabVIEW), lack of the IEEE Standard C37.118.2-2011 support, or are primarily not designed for synchro-measurement data acquisition purpose.

D. Example Application: Online Voltage Magnitude Monitoring

The main purpose of the proposed SADF is to enable a simplified design and facilitate an extensive online validation of emerging synchro-measurement-supported monitoring, protection, and control applications under realistic conditions. As a proof of the concept, a simple online voltage magnitude monitoring (Fig. 8) is implemented in MATLAB. The application is automatically updated every time a new PMU measurement becomes available. This application is available as a part of the open-source SADF library.

VI. Conclusion

In spite of evident interest of research community for synchro-measurement-supported applications in monitoring, protection, and control, there still exists a lack of available user-friendly tools for application design. This paper presented a novel MATLAB-supported SADF library, used for online receiving, and real-time parsing of IEEE Standard C37.118-2005 and C37.118.2-2011 machine-readable measurements into a user-friendly format. Moreover, a new robust data receiver communication technique and an efficient parsing methodology of the standard defined messages are proposed and implemented in the MATLAB programming environment as a proof of concept. A detailed performance evaluation of the proposed MATLAB-supported SADF library has shown its practical merits in research community, although a complete SADF implementation in C, or similar programming languages would be suitable for use in high-performance industrial environment. It was shown that the MATLAB-supported SADF library is particularly suitable for simplified development and thorough validation of wide-area monitoring applications. From the perspective of the online disturbance detection and oscillation monitoring, MATLAB offers a broad range of advanced mathematical and signal processing functions, which can be applied for these important purposes. This level of flexibility opens opportunities for development of advanced wide-area control and protection applications, which as such present a challenge today. The research team is now focused on the development of new applications targeting detection of cascading events and next generation of blackout prevention-oriented system integrity protection schemes. Besides, the SADF library is shown to be an efficient tool for training/teaching activities for exploring/validating IEEE Standard C37.118 supported devices from their functional and characteristics perspectives.

References

[1] C. W. Gellings, M. Samotyj, and B. Howe, "The future's smart delivery system [electric power supply]," *IEEE Power Energy Mag.*, vol. 2, no. 5, pp. 40–48, Sep./Oct. 2004.

[2] (May 2009). *The Smart Grid: An Introduction.* Accessed: Dec. 20, 2017. [Online]. Available: https://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE_SG_Book_Single_Pages.pdf

[3] P. Zhang, F. Li, and N. Bhatt, "Next-generation monitoring, analysis, and control for the future smart control center," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 186–192, Sep. 2010.

[4] V. Terzija *et al.*, "Wide-area monitoring, protection, and control of future electric power networks," *Proc. IEEE*, vol. 99, no. 1, pp. 80–93, Jan. 2011.

[5] F. Aminifar, M. Fotuhi-Firuzabad, A. Safdarian, A. Davoudi, and M. Shahidehpour, "Synchrophasor measurement technology in power systems: Panorama and state-of-the-art," *IEEE Access*, vol. 2, pp. 1607–1628, 2014.

[6] A. Phadke and J. Thorp, *Synchronized Phasor Measurements and their Applications.* Cham, Switzerland: Springer, 2008.

[7] V. Terzija *et al.*, "FlexNet wide area monitoring system," in *Proc. IEEE PES Gen. Meeting*, Jul. 2011, pp. 1–7.

[8] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Standard C37.118-2005, 2005.

[9] *IEEE Standard for Synchrophasor Measurements for Power Systems*, IEEE Standard C37.118.1-2011 (Revision of IEEE Standard C37.118-2005), 2011.

[10] *IEEE Standard for Synchrophasor Measurements for Power Systems—Amendment 1: Modification of Selected Performance Requirements*, IEEE Standard C37.118.1a, 2014.

[11] *IEEE Standard for Synchrophasor Data Transfer for Power Systems*, IEEE Standard C37.118.2-2011 (Revision of IEEE Standard C37.118-2005), 2011.

[12] A. G. Phadke and B. Kasztenny, "Synchronized phasor and frequency measurement under transient conditions," *IEEE Trans. Power Del.*, vol. 24, no. 1, pp. 89–95, Jan. 2009.

[13] D. Macii, D. Fontanelli, G. Barchi, and D. Petri, "Impact of acquisition wideband noise on synchrophasor measurements: A design perspective," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 10, pp. 2244–2253, Oct. 2016.

[14] J. Warichet, T. Sezi, and J.-C. Maun, "Considerations about synchrophasors measurement in dynamic system conditions," *Int. J. Elect. Power Energy Syst.*, vol. 31, no. 9, pp. 452–464, Oct. 2009.

[15] D. Macii, D. Petri, and A. Zorat, "Accuracy analysis and enhancement of DFT-based synchrophasor estimators in off-nominal conditions," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 10, pp. 2653–2664, Oct. 2012.

[16] G. Barchi, D. Macii, and D. Petri, "Synchrophasor estimators accuracy: A comparative analysis," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 5, pp. 963–973, May 2013.

[17] A. J. Roscoe, "Exploring the relative performance of frequency-tracking and fixed-filter phasor measurement unit algorithms under C37.118 test procedures, the effects of interharmonics, and initial attempts at merging P-class response with M-class filtering," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 8, pp. 2140–2153, Aug. 2013.

[18] S. Maharjan, J. C.-H. Peng, J. E. Martinez, W. Xiao, P.-H. Huang, and J. L. Kirtley, "Improved sample value adjustment for synchrophasor estimation at off-nominal power system conditions," *IEEE Trans. Power Del.*, vol. 32, no. 1, pp. 33–44, Feb. 2017.

[19] *IEEE Synchrophasor Measurement Test Suite Specification—Version 2*, IEEE Std C37.118.1a-2014.1 TSS, 2015.

[20] C. Thilakarathne, L. Meegahapola, and N. Fernando, "Improved synchrophasor models for power system dynamic stability evaluation based on IEEE C37.118.1 reference architecture," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 11, pp. 2937–2947, Nov. 2017.

[21] K. E. Martin, "Synchrophasor measurements under the IEEE Standard C37.118.1-2011 with amendment C37.118.1a," *IEEE Trans. Power Del.*, vol. 30, no. 3, pp. 1514–1522, Jun. 2015.

[22] D. M. Laverty, R. J. Best, P. Brogan, I. Al Khatib, L. Vanfretti, and D. J. Morrow, "The OpenPMU platform for open-source phasor measurements," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 4, pp. 701–709, Apr. 2013.

[23] N. Pandit and K. Khandeparkar, "Design and implementation of IEEE C37.118 based phasor data concentrator & PMU simulator for wide area measurement system," Indian Inst. Technol., Mumbai, India, Tech. Rep., May 2012. [Online]. Available: https://sourceforge.net/projects/iitbpdc/files/iPDC%20-%20Technical%20Report.pdf

[24] S. Sandi, T. Popovic, and B. Krstajic, "Python implementation of IEEE C37.118 communication protocol," *J. Elect. Eng.*, vol. 21, no. 1, pp. 108–117, Dec. 2015.

[25] (Dec. 2017). *Grid Protection Alliance*. Accessed: Dec. 20, 2017. [Online]. Available: https://www.gridprotectionalliance.org/

[26] M. S. Almas, L. Vanfretti, and M. Baudette, "BabelFish—Tools for IEEE C37.118.2-compliant real-time synchrophasor data mediation," *SoftwareX*, vol. 6, pp. 209–216, Dec. 2017.

[27] W. Stevens and G. Wright, *TCP/IP Illustrated: The Protocols*, vol. 1. Reading MA, USA: Addison-Wesley, 1994.

[28] W. Stevens and G. Wright, *TCP/IP Illustrated: The Implementation*, vol. 2. Reading, MA, USA: Addison-Wesley, 1995.

[29] (Dec. 2017). *MATLAB Central*. Accessed: Dec. 20, 2017. [Online]. Available: http://www.mathworks.com/

[30] M. Naglic, I. Tyuryukanov, M. Popov, M. van der Meijden, and V. Terzija, "WAMPAC-ready platform for online evaluation of corrective control algorithms," in *Proc. Medit. Conf. Power Generat., Transmiss., Distrib. Energy Convers. (MedPower)*, Nov. 2016, pp. 1–6.

[31] *IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, Monitoring*, IEEE Standard C37.224, 2013.

Mart A. M. M. van der Meijden (M'10) received the M.Sc. degree (*cum laude*) in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 1981.

He is leading research programs on intelligent electrical power grids, reliable, and large-scale integration of renewable (wind and solar) energy sources in the European electrical power systems and advanced grid concepts. He has more than 30 years of working experience in the field of process automation and the transmission and the distribution of gas, district heating, and electricity. Since 2003, he has been with TenneT TSO, Arnhem, The Netherlands, Europe's first cross-border grid operator for electricity, where he is the Manager of Research and Development/Innovation and was responsible for the development of the TenneT long-term vision on the electrical transmission system. He has been a Full Professor (part-time) with the Department of Electrical Sustainable Energy, Faculty of Electrical Engineering, Mathematics and Computers Science, Delft University of Technology, Delft, The Netherlands, since 2011. His current research interests include large-scale sustainable power systems.

Prof. van der Meijden is a member of ENTSO-E/RDIC and CIGRE. He has joined and chaired different national and international expert groups.

Matija Naglic (S'15) received the Uni. Dipl.-Ing. degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2012.

He was a Researcher with the Milan Vidmar Electric Power Research Institute, Ljubljana, Slovenia. He is currently a Doctoral Researcher with the Intelligent Electrical Power Grids Group, Delft University of Technology, Delft, The Netherlands. His research interests include real-time monitoring and control of large-scale power systems.

Marjan Popov (M'95–SM'03) received the Dipl.-Ing. degree from the University of Skopje, Skopje, Macedonia, in 1993, and the Ph.D. degree in electrical power engineering from the Delft University of Technology, Delft, The Netherlands, in 2002.

In 1997, he was an Academic Visitor with the ARC Research Group, University of Liverpool, Liverpool, U.K., where he was involved in modeling SF6 circuit breakers. His current research interests include future power systems, large-scale power system transients, intelligent protection for future power systems, and wide-area monitoring and protection.

Prof. Popov is a member of CIGRE. He has actively participated in WG C4.502 and WG A2/C4.39. He was a recipient of the IEEE PES Prize Paper Award and the IEEE Switchgear Committee Award in 2011. He is an Associate Editor of the *International Journal of Electric Power and Energy Systems*.

Vladimir Terzija (M'95–SM'00–F'16) was born in Donji Baraci (former Yugoslavia). He received the Dipl.-Ing., M.Sc., and Ph.D. degrees in electrical engineering from the University of Belgrade, Belgrade, Serbia, in 1988, 1993, and 1997, respectively.

From 1997 to 1999, he was an Assistant Professor with the University of Belgrade. From 2000 to 2006, he was a Senior Specialist for switchgear and distribution automation with ABB AG Inc., Ratingen, Germany. He has been the Engineering and Physical Science Research Council Chair Professor in Power System Engineering with the School of Electrical and Electronic Engineering, The University of Manchester, Manchester, U.K., since 2006. His current research interests include smart grid application of intelligent methods to power system monitoring, control, and protection, wide-area monitoring, protection, and control, switchgear and fast transient processes, and digital signal processing applications in power systems.