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

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

DOI
10.1109/TIM.2018.2807000

Publication date
2018

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Instrumentation and Measurement

Citation (APA)
Naglic, M., Popov, M., van der Meijden, M. A. M. M., & Terzija, V. (2018). Synchro-Measurement Application Development Framework: An IEEE Standard C37.118.2-2011 Supported MATLAB Library. IEEE Transactions on Instrumentation and Measurement, 67(8), 1-11. https://doi.org/10.1109/TIM.2018.2807000

Synchro-measurement Application Development Framework: an IEEE Standard C37.118.2-2011 Supported MATLAB Library

M. Naglic[1*], Student Member, IEEE
M. Popov[1], Senior Member, IEEE
M.A.M.M. Meijden[1, 2], Member, IEEE
V. Terzija[3], Fellow, IEEE

[1]Delft University of Technology, Faculty of EEMCS, Delft, Netherlands
[2]TenneT TSO, Arnhem, Netherlands
[3]The University of Manchester, School of Electrical and Electronic Engineering, Manchester, UK

[*]**Corresponding author:** Matija Naglic
E-mail: m.naglic@tudelft.nl
Tel.: +31 15 27 88007

Abstract

Electrical power system monitoring, protection, operation, and control schemes are undergoing significant changes towards 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 Std. 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 Std. 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, the 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 Std. C37.118.2-2011, MATLAB, online processing, PMU, WAMPAC**

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 towards advanced Wide Area Monitoring, Protection, and Control (WAMPAC) systems [4]. The WAMPAC system is favourable to ensure efficient, more resilient, and secure operation of EPS through sophisticated utilisation of Smart Grid components like 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 of intelligent electronic devices (IED) or Phasor Measurement Units (PMU), and Phasor Data Concentrators (PDC), connected over ICT infrastructure into a hierarchically organized network [7] (further detailed in section III).

A. *Paper motivation*

Over the recent years IEEE Std. C37.118 [8]–[11] has evolved to one of the widely deployed communication protocol for synchro-measurement data exchange among SMT supported power system components. The typical IEEE Std. 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 pre-processed measurements, typically 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 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]; 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 papers 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 a Synchro-measurement Application Development Framework (SADF), a novel IEEE Std. C37.118.2-2011 supported MATLAB software library used for online receiving and parsing of IEEE Std. 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 IEEE Std. 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:

1. Implementation procedures regarding the missing link between the IEEE Std. C37.118.2-2011 specifications and its implementation to comply with the complete IEEE Std. C37.118.2-2011 requirements. The focus is on (i) a robust data receiver communication technique with integrated fall-back procedures, (ii) 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 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 IEEE Std. C37.118.2-2011 requirements, and performance evaluation by using real-time cyber-physical simulation platform.

C. Paper organization

The rest of the paper is organized as follows: Section II addresses the related work. Section III presents the related background. Section IV presents the Synchro-measurement Application Development Framework, 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 the paper ends up with conclusions.

II. Related work

In the last two decades 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 Std. 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. In [23] and [24] authors implemented IEEE Std. C37.118-2005 for PMU/PDC emulation purposes as an open-source application using C and Python programming language respectively. However, their published work has several limitations including: (i) lack of IEEE Std. C37.118.2-2011 support, (ii) lack of UDP connection protocol support and spontaneous communication mode [23], (iii) 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, 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 one of the most used open-source applications built on top of TimeSeriesFramework 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 IEEE Std. C37.118.2-2011 data stream in LabVIEW programming environment, by using TCP/IP and commanded mode only. The proposed work fills the gap and presents a complete IEEE Std. C37.118.2-2011 supported implementation in MATLAB programming environment. It is important to point out that the proposed SADF library supports both commanded and spontaneous communication mode. In 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 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 Std. 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 of specialized IEDs, connected into a hierarchically organized ICT infrastructure as illustrated in Fig. 1.

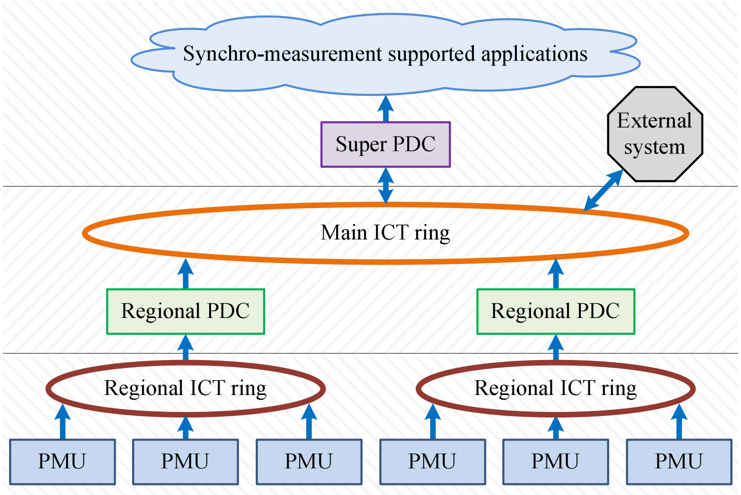


Fig. 1. Synchronized Measurement Technology infrastructure.

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 per second. Additionally, it sends analogue sample measurements (single point-on-wave, RMS of analogue input, peak of analogue signal) and Boolean status words from supported analogue and digital input channels respectively. PMUs send their synchro-measurements over supporting ICT infrastructure [7] to regional PDCs. A PDC works as data contractor, which collects and pre-process 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 pre-process the synchro-measurements received from PMUs and/or PDCs and sends them in single data stream to application data acquisition center and/or to an external SMT supported system.

B. *IEEE Std. C37.118.2-2011 overview*

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

1) Communication infrastructure

The IEEE Std. 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. Primary the standard was designed for RS-232 serial communication, but the past implementations show great success of fast fiber optic supported Internet Protocol (IP) telecommunication networks, utilizing Transmission Control Protocol (TCP/IP) and/or User Datagram Protocol (UDP).

a) Transmission Control Protocol

TCP/IP is connection oriented (one-to-one, 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 arrives to the final destination (application layer), remains intact, and in 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 case of corrupted and out-of-order received TCP/IP packets [27].

b) User Datagram Protocol

In contrast UDP is simpler, connectionless (one-to-one/many, 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 PMU/PDC data source device (further addressed as sender) and takes control of configuration and measurement data exchange. Hereby receiver sends specific data request commands to sender, which afterwards replies with requested data. On the other hand, in 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 Std. 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.

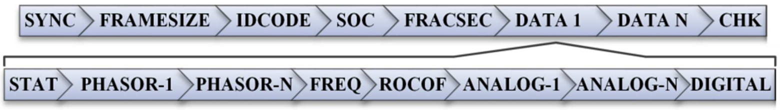


Fig. 2. DATA message frame structure.

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 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 by:

$$TimeStamp = SOC + FRACSEC / TIME\_BASE \tag{1}$$

where *TIME_BASE* integer determines the maximum time resolution in ticks.

Additionally, a FRACSEC contains 8-bit time quality flags to indicate quality of time being measured plus indication of leap second status. Finally, a content of a message (DATA 1 - 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 set of measurements at precisely determined moment of time. Each DATA message contains at least one encapsulated PMU measurement block (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 info 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), rate-of-change-of-frequency (ROCOF), analogue values (ANALOG), and digital Boolean status words (DIGITAL).

c) Configuration message

A configuration (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 analogue 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 analogue values offset, a synchrophasor estimation algorithm window length and group delay, and geographical 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.

e) Command message

A command (CMD) message frame is used exclusively in 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 is transmitted using big endian, i.e. most significant byte first.

In 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 a sender configuration settings. Additionally every minute rollover it sends a CFG-2/CFG-3 frame (1) to inform a receiver about DATA message frames structure currently being reported.

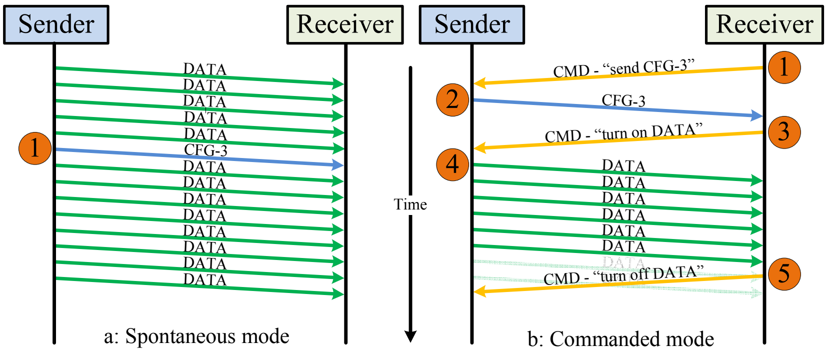


Fig. 3. Communication data exchange mode: a - spontaneous mode, b - commanded mode.

On the other hand, in 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. Further after a receiver successfully receives a CFG frame it sends to a sender a “turn on DATA” (3) command. Afterwards, a sender starts sending DATA (4) frames to a receiver in continuous and equally time spaced intervals, as specified in 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

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 IEEE Std.

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.

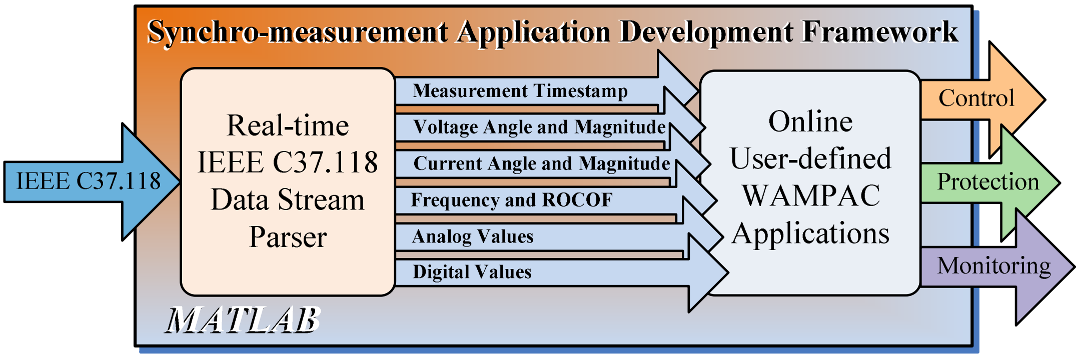

Synchro-measurement Application Development Framework
IEEE C37.118
Real-time IEEE C37.118 Data Stream Parser
Measurement Timestamp
Voltage Angle and Magnitude
Current Angle and Magnitude
Frequency and ROCOF
Analog Values
Digital Values
Online User-defined WAMPAC Applications
Control
Protection
Monitoring
MATLAB

Fig. 4. Synchro-measurement Application Development Framework.

MATLAB programing suite is a powerful engineering computing toolbox, supported by various built-in mathematical, signal processing, and visualization functions. It is cross-platform compatible (Windows, Mac, 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 of 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 Std. 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 WiFi technologies. In both cases 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 commanded and spontaneous communication mode. In 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 second case UDP is supported to facilitate fast and efficient transmission of configuration and measurement data exchange, critical for fast response WAMPAC applications.

(1) Commanded mode

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

First, the SADF sequently sends the CMD frame requests to obtain a 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 seconds 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 info obtained from a HDR or CFG-1 frame. Lastly, in 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 reestablished. Afterwards, the connection establishment procedure with the PMU/PDC is reinitiated. In this way the SADF library does not halt in 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 send 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 info 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 send. Afterwards, 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, the ICT connection and connection establishment procedure is reestablished and reinitiated respectively.

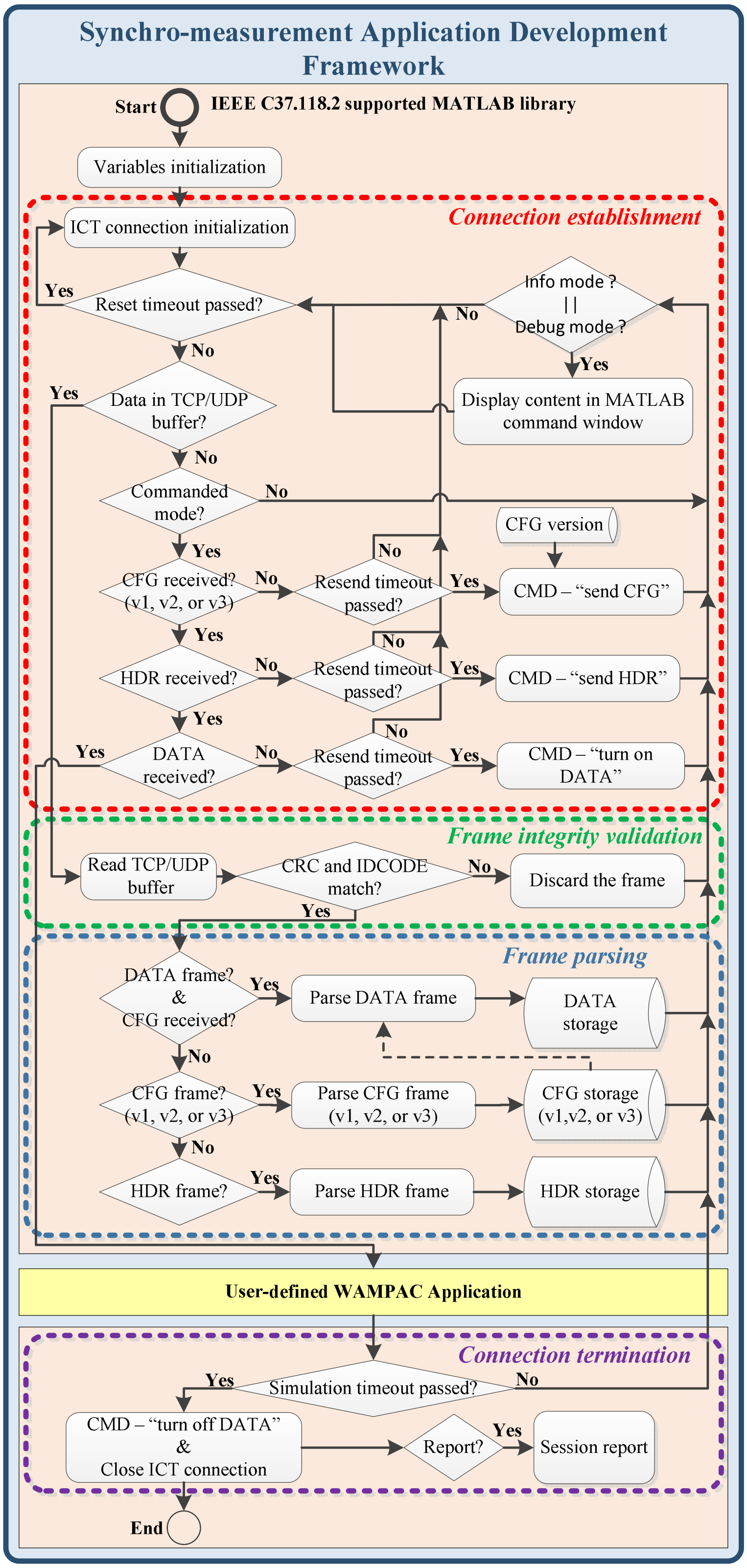


Fig. 5. SADF library structure.

(2) Spontaneous mode

In 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 info 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 determinated 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 following polynomial function

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

with 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 Std. 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, 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 MATLAB workspace using structure array format and by 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:

- *info mode*; only basic info regarding send and received frames with corresponding timestamp is displayed in MATLAB command window.
- *debug mode*; info regarding send and received frames with detailed message frame content in user-friendly format is displayed in MATLAB command window. Additionally, all the raw data exchange between sender and receiver is stored for ease of debugging purposes.
- silent mode (default); no info is displayed in 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 same order as were sent; hence no additional DATA stream post-processing 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 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:

- *leap second status*
- *time accuracy*
- *measurement data validity*
- *time source synchronization status*
- *measurement device operation status*
- *data sorting type*
- *trigger status and reason*
- *configuration change notification*

The standard defines 16-bit integer and 32-bit floating point formats for synchro-measurements (phasors, analogue 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 post-processing measurement adjustments factors related to magnitude or angle offset corrections (in case of phasors and analogue 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 analogue values, frequency, and ROCOF measurements.

b) CFG frame interpretation

The SADF can receive CFG-1, CFG-2, and CFG-3 configuration frame types. Since a 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:

- *number of PMU blocks included in DATA frame*
- *number of phasors, analogue and digital values for each PMU block included*
- *PMU station, phasor, analogue channel naming*
- *digital channel*
- *measurement format, conversation, and offset factors*
- *station geo-location properties*
- *estimation algorithm window length and group delay*
- *configuration change count*
- *DATA transmit rate*
- *nominal system frequency*

c) HDR frame interpretation

The SADF can receive the standard supported HDR frame. Its content is primary in 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 MATLAB command window.

d) CMD frame sending

In 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 Std. C37.118 compliant devices and accuracy of the provided data, a cyber-physical simulation platform has been used.

A. Cyber-physical simulation platform

The cyber-physical simulation platform (Fig. 6) is comprised of 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 analogue 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 fps 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 fps 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 on 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.

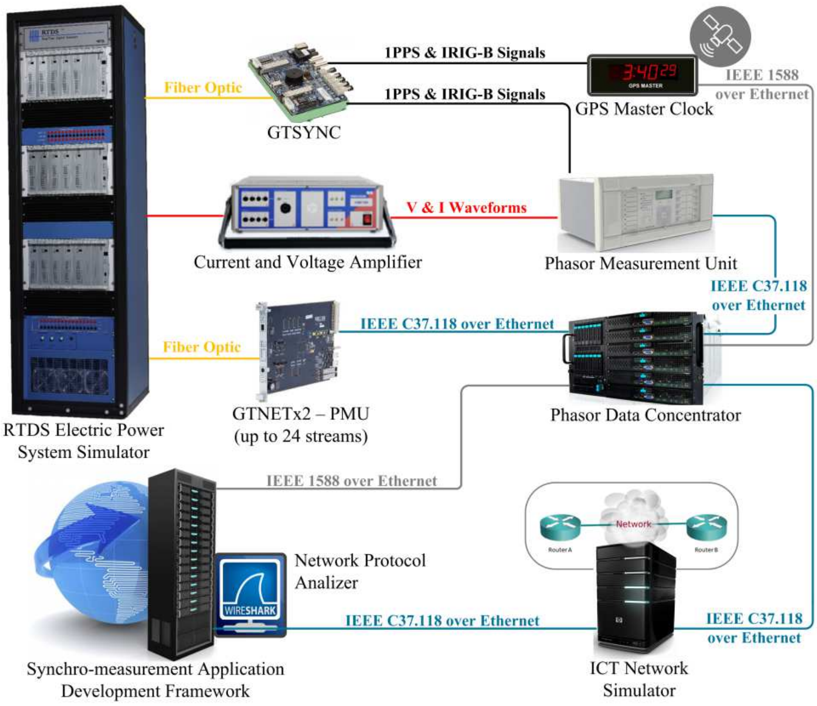

1PPS & IRIG-B Signals
1PPS & IRIG-B Signals
Fiber Optic
GTSYNC
3:40:29
GPS MASTER
GPS Master Clock
IEEE 1588 over Ethernet
V & I Waveforms
Current and Voltage Amplifier
Phasor Measurement Unit
IEEE C37.118 over Ethernet
Fiber Optic
IEEE C37.118 over Ethernet
GTNETx2 – PMU (up to 24 streams)
Phasor Data Concentrator
RTDS Electric Power System Simulator
IEEE 1588 over Ethernet
Router A
Network
Router B
Network Protocol Analizer
WIRESHARK
IEEE C37.118 over Ethernet
IEEE C37.118 over Ethernet
Synchro-measurement Application Development Framework
ICT Network Simulator

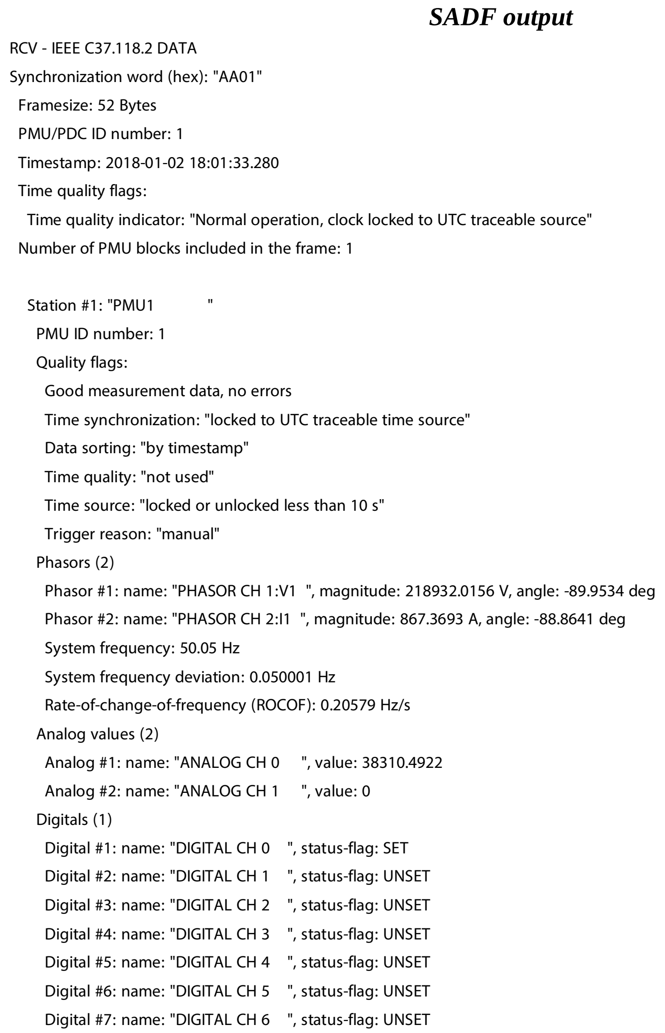
Fig. 6. Cyber-physical simulation platform.

The performance evaluation of proposed MATLAB supported SADF library is performed on a typical office computer (CPU: Intel Xeon E5 1620v2 - 3.70 GHz, memory: 16 GB RAM, OS: Windows 7) with installed Wireshark communication protocol analyser 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, 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.

B. IEEE Std. 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. Afterwards 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 analyser 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 on Fig 7.

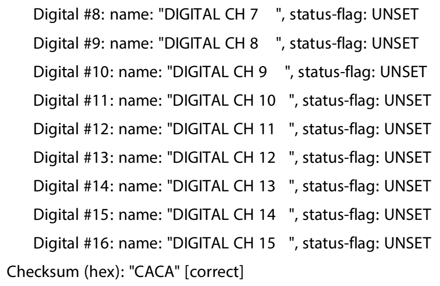
In this way the SADF compliance and interoperability with the standards compliant devices was confirmed.



SADF output

```
RCV - IEEE C37.118.2 DATA
Synchronization word (hex): "AA01"
 Framesize: 52 Bytes
 PMU/PDC ID number: 1
 Timestamp: 2018-01-02 18:01:33.280
 Time quality flags:
  Time quality indicator: "Normal operation, clock locked to UTC traceable source"
 Number of PMU blocks included in the frame: 1

  Station #1: "PMU1            "
   PMU ID number: 1
   Quality flags:
    Good measurement data, no errors
    Time synchronization: "locked to UTC traceable time source"
    Data sorting: "by timestamp"
    Time quality: "not used"
    Time source: "locked or unlocked less than 10 s"
    Trigger reason: "manual"
   Phasors (2)
    Phasor #1: name: "PHASOR CH 1:V1  ", magnitude: 218932.0156 V, angle: -89.9534 deg
    Phasor #2: name: "PHASOR CH 2:I1  ", magnitude: 867.3693 A, angle: -88.8641 deg
    System frequency: 50.05 Hz
    System frequency deviation: 0.050001 Hz
    Rate-of-change-of-frequency (ROCOF): 0.20579 Hz/s
   Analog values (2)
    Analog #1: name: "ANALOG CH 0     ", value: 38310.4922
    Analog #2: name: "ANALOG CH 1     ", value: 0
   Digitals (1)
    Digital #1: name: "DIGITAL CH 0    ", status-flag: SET
    Digital #2: name: "DIGITAL CH 1    ", status-flag: UNSET
    Digital #3: name: "DIGITAL CH 2    ", status-flag: UNSET
    Digital #4: name: "DIGITAL CH 3    ", status-flag: UNSET
    Digital #5: name: "DIGITAL CH 4    ", status-flag: UNSET
    Digital #6: name: "DIGITAL CH 5    ", status-flag: UNSET
    Digital #7: name: "DIGITAL CH 6    ", status-flag: UNSET
```



```
    Digital #8: name: "DIGITAL CH 7    ", status-flag: UNSET
    Digital #9: name: "DIGITAL CH 8    ", status-flag: UNSET
    Digital #10: name: "DIGITAL CH 9    ", status-flag: UNSET
    Digital #11: name: "DIGITAL CH 10   ", status-flag: UNSET
    Digital #12: name: "DIGITAL CH 11   ", status-flag: UNSET
    Digital #13: name: "DIGITAL CH 12   ", status-flag: UNSET
    Digital #14: name: "DIGITAL CH 13   ", status-flag: UNSET
    Digital #15: name: "DIGITAL CH 14   ", status-flag: UNSET
    Digital #16: name: "DIGITAL CH 15   ", status-flag: UNSET
Checksum (hex): "CACA" [correct]
```

Fig. 7. SADF output (in “debug mode”) in case of processed DATA frame.

C. SADF performance evaluation

Synchro-measurement data latency in SMT supported system refers to a time delay between a PMU measurement timestamp and an instant when a PMU measurement data is 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, hence 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 MATLAB workspace. Hereby, processing time of different PMU configuration settings have been evaluated based on [10, Annex C, Table C.1] and the most commonly used configuration scenarios. Table 1. presents the median processing time per DATA, CFG-2, and CFG-3 frame of total 100 received frames of equal type and size.

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	/

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 1. 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, floating point). In this case, a maximum DATA receiving rate that the SADF library is capable of processing is theoretically limited to ~ 1300 fps. However, the SADF library is confirmed to successfully receive 250 fps DATA frame rate (1 block, 12 phasors, 2 analog, 2 digital, 1 freq, 1 ROCOF, integer), which is the maximum data reporting rate of the GTNETx2 virtual 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 Std. 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.

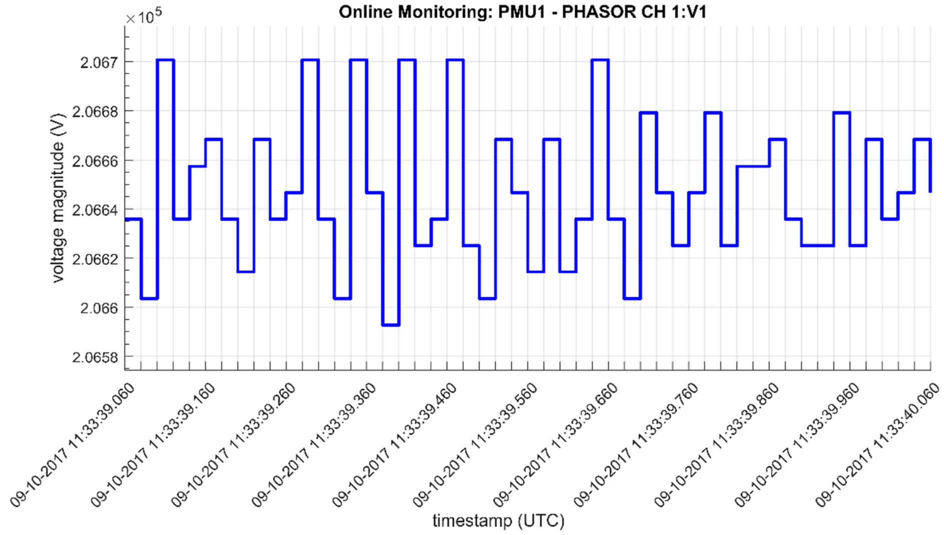


Fig. 8. Example application: Online voltage magnitude monitoring of PMU station ”*PMU1*”, channel “*PHASOR CH 1:V1*”, with 50 fps measurement reporting rate.

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 Synchro-measurement Application Development Framework (SADF) library, used for online receiving, and real-time parsing of IEEE Std. 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 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 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 Std. C37.118 supported devices from their functional and characteristics perspectives.

VII. References

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

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

[3] P. Zhang, F. Li, and N. Bhatt, "*Next-Generation Monitoring, Analysis, and Control for the Future Smart Control Center*," IEEE Transactions on Smart Grid, vol. 1, no. 2, pp. 186–192, Sep. 2010.

[4] V. Terzija, G. Valverde, D. Cai, P. Regulski, V. Madani, J. Fitch, S. Skok, M. M. Begovic, A. Phadke, "*Wide-Area Monitoring, Protection, and Control of Future Electric Power Networks*," Proc. IEEE, vol. 99, no. 1, pp. 80–93, 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: Springer, 2008.

[7] V. Terzija, P. Regulski, L. Kunjumuhammed, B. Pal, G. Burt, I. Abdulhadi, T. Babnik, M. Osborne, W. Hung, "*FlexNet wide area monitoring system*", Proc. IEEE PES General Meeting, pp. 1-7, 2011.

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

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

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

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

[12] A. G. Phadke and B. Kasztenny, "*Synchronized Phasor and Frequency Measurement Under Transient Conditions*," IEEE Transactions on Power Delivery, 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 Transactions on Instrumentation and Measurement, 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*," International Journal of Electrical Power & Energy Systems, 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 Transactions on Instrumentation and Measurement, vol. 61, no. 10, pp. 2653–2664, Oct. 2012.

[16] G. Barchi, D. Macii, and D. Petri, "*Synchrophasor Estimators Accuracy: A Comparative Analysis*," IEEE Transactions on Instrumentation and Measurement, 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 Transactions on Instrumentation and Measurement, 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 Transactions on Power Delivery, vol. 32, no. 1, pp. 33–44, Feb. 2017.

[19] “*IEEE TSS*”, *IEEE Synchrophasor Measurement Test Suite Specification - Version 2*, 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 Transactions on Instrumentation and Measurement, 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 Transactions on Power Delivery, 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 Transactions on Instrumentation and Measurement, 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 Institute of Technology, Bombay, Tech. Rep., May 2012. Available: https://sourceforge.net/projects/iitbpdc/files/Technical%20Document.pdf

[24] S. Sandi, T. Popovic, B. Krstajic, "*Python Implementation of IEEE C37.118 Communication Protocol*", Journal of Electrical Engineering, Faculty of Electrical Engineering, University, Podgorica, Montenegro, Vol. 21, No. 1, pp 108-117, Dec. 2015.

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

[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, 2017.

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

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

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

[30] M. Naglic, I. Tyuryukanov, M. Popov, M. van der Meijden and V. Terzija, "*WAMPAC-ready platform for online evaluation of corrective control algorithms*", Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MedPower 2016), 2016.

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

VIII. Biographies

Matija Naglic received the Uni. Dipl-Ing. degree in electrical engineering from University of Ljubljana, Faculty of Electrical Engineering, Slovenia, in 2012. Currently, he is a doctoral researcher with the Intelligent Electrical Power Grids group at Delft University of Technology, The Netherlands. Before, he has worked as a researcher at the Milan Vidmar Electric Power Research Institute, Slovenia. His research interests are related to real-time monitoring and control of large-scale power systems. He is a Student Member of the IEEE.

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 Delft University of Technology, Delft, The Netherlands, in 2002. In 1997, he was an Academic Visitor at the University of Liverpool, Liverpool, U.K., working in the arc research group on modeling SF6 circuit breakers. His major fields of interest are future power systems, large-scale power system transients, intelligent protection for future power systems, and wide-area monitoring and protection. Prof. Popov is also a member of CIGRE and actively participated in WG C4.502 and WG A2/C4.39. He is IEEE PES Prize Paper Award and IEEE Switchgear Committee Award recipient for 2011 and associate editor of the international journal of electric power and energy systems.

Mart A.M.M. van der Meijden received the M.Sc. degree (cum laude) in electrical engineering from the Eindhoven University of Technology, the Netherlands, in 1981. He is full professor (part-time) with the Department of Electrical Sustainable Energy of the Faculty of Electrical Engineering, Mathematics and Computers Science, Delft University of Technology, since June 2011. His chair and research focus is on Large Scale Sustainable Power Systems. Professor van der Meijden 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. He is leading research programs on intelligent electrical power grids, reliable and large scale integration of renewable (wind, solar) energy sources in the European electrical power systems and advanced grid concepts. Since 2003 he is working with TenneT TSO, Europe's first cross-border grid operator for electricity. He is Manager R&D/Innovation and was responsible for the development of the TenneT long term vision on the electrical transmission system. Professor van der Meijden is member of IEEE, ENTSO-E/RDIC and CIGRE and he has joined and chaired different national and international expert groups.

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. He is 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., where he has been since 2006. From 1997 to 1999, he was an Assistant Professor at the University of Belgrade, Belgrade, Serbia. From 2000 to 2006, he was a senior specialist for switchgear and distribution automation with ABB AG Inc., Ratingen, Germany. 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.