Electrical Power and Energy Systems 155 (2024) 109637

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Development and implementation of a MATLAB-based phasor data concentrator for synchrophasor applications

Sugandh Pratap Singh [a,*], Saikat Chakrabarti [a], Devesh Shukla [b], Vladimir Terzija [c]

[a] Electrical Engineering, Indian Institute of Technology, Kanpur, U.P., India
[b] Electrical Engineering, National Institute of Technology, Calicut, Kerala, India
[c] School of Engineering, Newcastle University, UK

ARTICLE INFO

Keywords:
Phasor data concentrator
Synchrophasor measurements
Time-alignment
Wide-area monitoring

ABSTRACT

This work presents the development of MatPDC, an IEEE Std. C37.118.2-2011 compliant Phasor Data Concentrator (PDC) implemented in the Matlab environment. MatPDC enables the integration of wide-area monitoring, protection and control system synchrophasor data into the Matlab platform, offering real-time access to synchronized data streams from multiple Phasor Measurement Units (PMUs). The key features of MatPDC include real-time data access, data integrity verification, data aggregation and alignment, compliant data communication, and latency calculation. These features facilitate efficient processing and analysis of synchronized data, reducing complexity and improving latency between data sources and applications. The development of MatPDC addresses the need for accessing and analyzing real-time synchrophasor data within the Matlab environment, providing researchers and engineers with a powerful tool for power system analysis and experimentation. By developing the PDC and synchrophasor applications on the same platform, MatPDC reduces complexity and latency between data sources and applications. It offers real-time access to synchronized data streams, ensuring data integrity through verification mechanisms and aggregating the data into a unified dataset based on time tags. The implementation of MatPDC opens up opportunities for researchers to work with real-time synchrophasor data. It facilitates the development of advanced algorithms, real-time simulations, and the verification of control strategies. The integration and evaluation of MatPDC demonstrate its effectiveness and potential in power system analysis, providing researchers and engineers with a valuable tool for their research and development activities.

1. Introduction

Wide Area Monitoring, Protection and Control (WAMPAC) system is a crucial component of modern power systems that could ensure enhanced reliability and stability [1]. This system employs PMUs to obtain synchronized measurements of the power system's voltage, current, and phase angle at a high rate. These measurements are then transmitted to the PDC, which processes the data and provides real-time information on the power system's operating state. With the aid of WAMPAC system, the power system's dynamic behavior can be quickly and accurately obtained and power system operators can promptly identify any deviations from the normal operating conditions and take corrective actions to prevent cascading failures or blackouts [2].

The basic function of a PDC is to collect the synchrophasor data from different locations of the power system, aggregate it into a time-aligned data set, and further communicate it to local storage or another PDC [3]. Once the data has been collected and processed, PDCs can transmit it to other systems or applications for further analysis and control. This can include control systems that make real-time decisions based on the synchrophasor data, as well as data storage and analysis systems that can be used for post-event analysis and system planning. PDCs play a critical role in modern power system operation by providing accurate and reliable synchrophasor data-set of large geographical area that can be used for a wide range of applications, from system monitoring and control to advanced analysis and planning. They can be considered as one of the technological enablers of Smart Grid concepts in HV/MV power networks.

PMUs and PDCs in WAMPAC systems provide several notable advantages. Firstly, it significantly enhances grid visibility, offering power system operators a comprehensive real-time view of the grid's dynamic behavior. This enhanced visibility enables the early detection of disturbances, faults, and anomalies, enabling operators to respond

---

* Corresponding author.
E-mail addresses: sugandhgcet@gmail.com, sugandh@iitk.ac.in (S.P. Singh).

https://doi.org/10.1016/j.ijepes.2023.109637
Received 7 August 2023; Received in revised form 12 October 2023; Accepted 31 October 2023

promptly and effectively. PMUs and PDCs also play a crucial role in transient stability analysis. By utilizing synchronized measurements from PMUs across multiple locations, operators can assess the stability of the power system during dynamic events. This empowers them to make informed decisions and implement preventive measures to maintain grid stability. Moreover, the integration of PMUs and PDCs facilitates the implementation of wide area protection schemes. Through accurate fault detection and localization using synchronized PMU measurements, operators can coordinate protection actions, quickly isolate faulty components, and minimize the disruptive impact of disturbances on the grid.

If the online data from the WAMPAC system is available in a programming environment, it provides the opportunity to develop and test algorithms, controllers, and validate test systems using real-time data. This means that the behavior of algorithms, controllers and proposed control techniques can be evaluated in real-time. By acquiring the WAMPAC system data in a programming environment, it becomes possible to simulate and analyze how algorithms and controllers perform under different conditions. This includes observing their response to changes in power demand or disruptions in the power system network, just as it would happen in real-world scenarios. Using real-time data from the WAMPAC system allows for the assessment of how well algorithms and controllers work in practical power system applications. This information can be used to make improvements and ensure their reliability and effectiveness. Overall, the availability of PMUs data in real-time in a programming environment allows for the testing, evaluation, and improvement of research findings using real-time data. This capability contributes to a better understanding of how these methods behave in real-world situations and supports the development of more effective and reliable solutions for power system challenges.

Integrating online synchrophasor data from PDC in a programming environment poses several challenges. One of the key difficulties is designing and implementing a protocol parser to decode and interpret the data transmitted by IEEE Std. C37.118 protocol compliant devices. Understanding the specifications of standardized protocols such as IEEE C37.118 is essential, especially when dealing with different versions of the protocols. Establishing a reliable and efficient communication channel between PMUs, PDCs, and the programming environment is very important. Researchers need to address issues such as network latency, packet loss, and network congestion to ensure seamless data transmission. Selecting appropriate communication technologies, such as TCP/IP or UDP, and optimizing data transmission are critical considerations.

PMU data is typically transmitted in binary format, requiring researchers to design algorithms to decode and extract relevant information from the encoded TCP packet. This involves handling various IEEE Std. C37.118 compliant data frames and accurately interpreting voltage, current, frequency, and phase angle measurements. Real-time data from PMUs and PDCs is time-sensitive, necessitating careful handling in the programming environment. Researchers must implement mechanisms to capture and process data in near real-time, considering the data rates and synchronization requirements.

Data corruption during transmission or device malfunctions are common challenges. Error detection and correction mechanisms need to implement to identify and handle corrupt data. Robust algorithms and error handling techniques, such as checksum verification or redundancy checks, help ensure data integrity. For validation and ensuring the quality of received synchrophasor data, it is necessary to perform data validation of received data. Scalability and performance considerations arise as the number of PMUs and data streams increase. Researchers need to design systems that can handle large volumes of data, ensure efficient data storage and retrieval, and optimize algorithms for processing speed. For integrating online data from PMUs and PDCs in a programming environment requires expertise in protocols, networking, data handling, error detection, and real-time processing. Researchers must address these challenges to effectively utilize real-time data for their studies and applications.

Researchers often prefer to work with offline/archive measurements because acquiring real-time or online data from PMUs can be complicated. This can be due to factors such as the need for specialized hardware and software, complex network configurations, and potential errors or limitations in the data acquired in real-time. When researchers use offline/archival measurements, they are working with data that has already been collected and stored, which makes it easy to access and process. However, because offline data is not collected in real-time, it may not fully capture the real-time dynamics and behavior of the power system. This can limit the accuracy and relevance of research findings.

The data exchanged by IEEE C37.118 communication protocol-based devices is in a machine-readable format that can be challenging for researchers to interpret. To conduct their research, researchers must first establish a connection with a PMU device over Ethernet using TCP/IP or UDP protocols. Then, they must use a protocol parser to translate the machine-readable IEEE C37.118 protocol data into a human-readable form. Synchrophasor applications such as dynamic line parameter estimation, dynamic load estimation [4], fault location [5,6], estimation of the dominant inter-area oscillation mode [7], steady-state and transient performance monitoring of WAMS [8,9], online inertia calculation [10] and sub-synchronous oscillation monitoring [11] require a time-aligned data-set of a section of the electricity grid for real-time analysis. To test these applications in real-time, it would be beneficial to have access to real-time synchronized data from PMUs in a programmable environment.

The Grid Protection Alliance (GPA) [12] came up with the “Time-Series Framework” in order to facilitate the management of synchrophasor data. Incorporating the Time-Series Library, GPA is then implemented an open-source phasor data concentrator in the programming environments of $.NET$, $JAVA$, and C^{++}. The openPDC is capable of distributing data online and to any historian. But to get the data into the programming environment, viz., MATLAB and Python, we again need an IEEE Std. C37.118.2-2011 protocol parser to perform online studies. The utilization of the OpenPDC infrastructure for the extraction and management of PMU data is also discussed in [13]. Additionally, it is utilized for the purpose of monitoring load dynamics using synchronized WAMS data.

In [14], the author created a synchrophasor-based library in the Matlab environment that acts as an IEEE Std. C37.118.2-2011 protocol parser, allowing for the receipt and processing of data into a human-readable form, and communication with another PDC. However, this library can only connect to one PMU/PDC. For multiple PMUs, the library uses openPDC as the data concentrator, which can introduce additional delay due to the latency of openPDC, the communication channel latency between openPDC and Matlab, and the latency of the protocol parser developed in the Matlab environment. Therefore, the overall latency can be affected by these factors when using this library for synchrophasor data analysis.

In [15], the authors have developed an open-source real-time data mediator named BabelFish within the LabVIEW development environment. BabelFish functions as an IEEE C37.118 protocol parser, but it is designed to connect to a single PMU/PDC only. It lacks the capability to directly connect multiple PMUs to LabVIEW; it requires a PDC between PMUs and BabelFish if data from more than one PMUs needs to be parsed. Additionally, BabelFish does not possess the capability to time-align multiple PMU data streams, which is essential for WAMPAC system monitoring.

In this work, an IEEE Std. C37.118.2-2011 compliant PDC has been developed in the Matlab environment. This is an important step towards accessing the online synchrophasor data set of a geographical area of a power system into the MATLAB environment. This MatPDC is capable of connecting with multiple PMUs, synchronizing the synchrophasor data received from these sources to a common time tag, and communicating the data to local storage or another higher-level

PDC. The MatPDC also has the ability to provide real-time synchronized data from PMUs to other synchrophasor applications. By developing both the PDC and synchrophasor applications on the same platform, complexity is reduced and latency between synchrophasor data sources and applications is improved. This PDC will enable researchers and engineers to more easily develop and test their research findings in real-time.

The MatPDC allows the utilization of online synchronized data of WAMPAC system within the MATLAB environment for analysis and research purposes . By integrating MatPDC, researchers gain the ability to work with real-time synchrophasor data, enabling them to study and evaluate the behavior of the power system in a more accurate and dynamic manner. This MatPDC opens up new opportunities for developing advanced algorithms, conducting real-time simulations, and verifying control strategies in more effective manner.

2. Key features of developed MatPDC

The developed MatPDC offers several key features for efficient processing of synchrophasor data from WAMPAC systems:

1. Real-time access to synchronized data: The MatPDC enables real-time access to time-synchronized data streams from multiple PMUs. It parses these data streams into a human-readable format, allowing for easy interpretation and analysis.
2. Data integrity verification: The MatPDC ensures the integrity of the received data stream by performing cyclic redundancy checks. It also validates the data using data flags and time quality tags, ensuring the accuracy and reliability of the synchronized data.
3. Data aggregation and alignment: The MatPDC aggregates the received synchronized data streams and aligns them into a unified dataset based on the time tags associated with each PMU's data stream. This aggregation and alignment process allows for coherent analysis and correlation of the synchronized data.
4. Compliant data communication: The PDC encapsulates the time-aligned data stream into a format that complies with the IEEE Std. C37.118.2-2011 standard. This formatted data stream can be transmitted as an output stream to other PDCs.
5. Latency calculation: The PDC measures the time elapsed between the arrival of the first data set to the PDC and the transmission of the output stream. The PDC processing time for the PDC is within 2 ms to 2+ s [3].

3. PDC architecture

A synchrophasor data collection network is a system that acquires and processes high-speed measurements of the power system in real-time to provide a better understanding of the system's dynamic behavior. In this network, PMUs are placed at different substations to acquire the measurements. The PMUs collect data at a very high rate, typically at 50–60 samples per second, and transmit it to the PDC in real-time.

The PDC is responsible for collecting all PMU data and aggregating it in a time-aligned manner. The time-aligned processed data at the local PDC is transmitted to other applications for local monitoring, archival, visualization, control, and protection [16]. These applications can use the synchronized phasor data to perform real-time analysis and monitoring of the power system. Furthermore, the local PDC transmits its time-aligned data to the control center PDC. The PDC at the control center aggregates various local PDC data with a common time stamp and sends it to other higher-level PDCs for further analysis, archival, and monitoring. The PDCs at different levels of the hierarchy can be located at different geographical locations and can communicate with each other over a wide area network.

3.1. Communication framework of WAMPAC

Fig. 1 represents the architecture of WAMPAC communication systems. In the architecture, PMUs are strategically placed within the power system and acquire synchronized phasor measurements of voltage, current, and phase angle. The PMUs communicate the synchronized phasor measurements to the PDCs over a dedicated communication network. The communication network of a synchrophasor system typically follows a client–server model, where the server is the source of data and the client receives the data. The server can be a PDC or a PMU, while the client can be a PDC or an application. In this model, the client initiates the connection and sends commands to regulate the flow of data. The server responds to these commands and sends the requested data.

The actual connections between the client and server can be TCP, UDP, or a combination of both. TCP connections are the easiest to establish, where the client sends a connection request to the server. Commands and data are then transmitted and received across the connection. TCP connections have the advantage of being aware of the connection state, enabling quick resolution of communication issues. UDP connections, on the other hand, are faster and have lower overhead than TCP. However, UDP connections are not aware of the connection state and do not provide error correction or retransmission of lost messages.

3.2. Message framework

The message framework [17] of a PDC is a critical aspect of synchrophasor systems that enables efficient communication and processing of synchrophasor data. The message framework defines the structure and format of the data messages that are transmitted and received by the PDC. These messages contain important information, such as time stamps, device identification, and data quality indicators, along with the actual synchrophasor data.

The IEEE Std. C37.118.2-2011 defines four types of message frames, i.e. configuration, command, data, and header, which are transmitted over the communication channel from the transmitter to the receiver. Each message frame is composed of several fields, including SYNC, FRAMESIZE, IDCODE, SOC, FRACSEC, and CHK. These fields are common to all message frames and provide important information about the message, such as synchronization, message size, device identification, and time stamps.

4. MatPDC architecture

The basic layout of the MatPDC has been described in Fig. 2. The MatPDC uses TCP connection because of its simplicity and industrial acceptability as standard for substation automation and control. The TCP is a client–server connection through which commands, data, and information are exchanged. The simplicity of this sort of connection is one of its primary benefits. Additionally, the status of the connection is communicated to both ends of the connection, which enables quick and accurate diagnosis of any issues with communication. First, the connection block initializes each server device's IPs, ports, and IDs (PMU/PDC). After initialization, TCP objects are created for each PMU using the IPs and ports. The capabilities of Matlab's instrument control toolbox are utilized to generate the TCP objects. The connection between the MatPDC and remote server devices opens a communication channel for data transfer and exchange.

The CFG and HDR blocks handle the configuration and header frames, respectively. When the configuration frame is received, the version of the configuration frame is checked, i.e., CFG-1, CFG-2, and CFG-3. After that, CFG frames are parsed using their respective frame version formats. Similarly, the header block receives and stores the header frames.

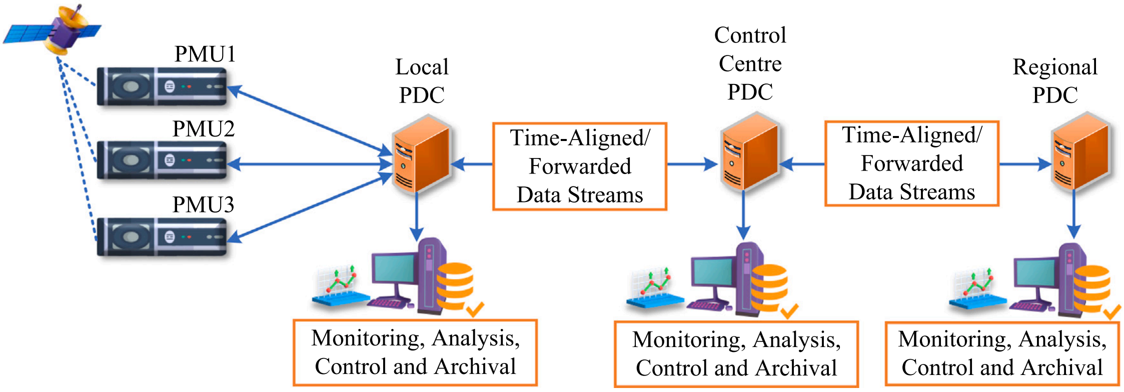


Fig. 1. WAMPAC system communication architecture.

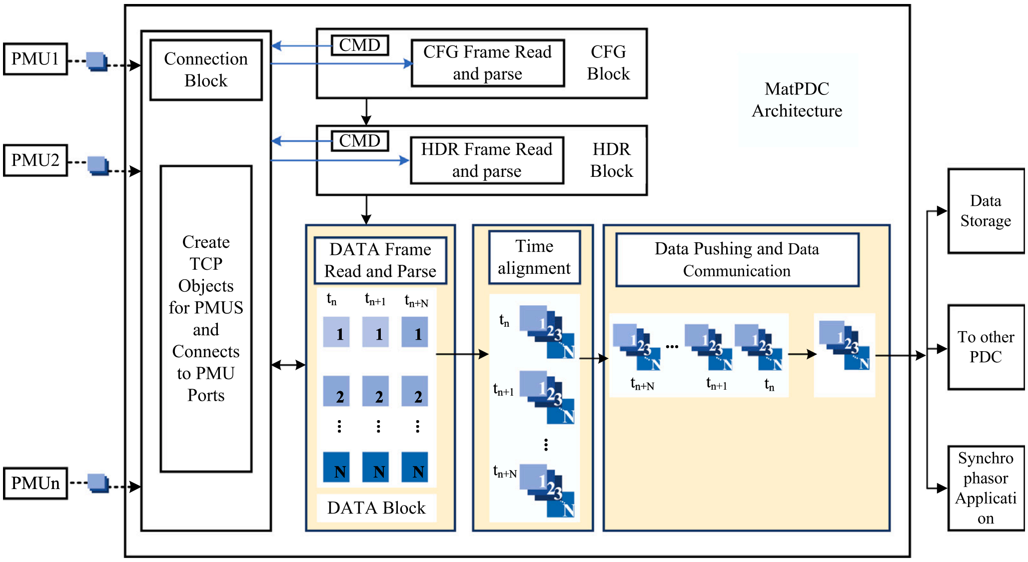


Fig. 2. The Architecture of proposed MatPDC.

The DATA block is responsible for collecting, parsing, and storing the data frames into the data buffer. When a data frame is available at the TCP socket, it is read and then parsed into a human-readable form using the respective PMU's configuration frame. After parsing, the data frame is stored in the data buffer based on its arrival timestamp. The time alignment block aggregates the data based on its arrival timestamp and then pushes the data serially to the synchrophasor application. The MatPDC also has a mechanism for storing data in a relational database for future analysis purposes. If data needs to be transmitted to another PDC, it is converted into TCP data packets and transmitted using TCP/IP.

5. MatPDC functionality

The MatPDC is designed to receive data packets from multiple PMUs located at different locations within a WAMPAC system using the TCP/IP protocol. It then aggregates the collected data into a time aligned data set. By aggregating the data, the MatPDC makes the wide-area measurements available for power system monitoring, protection, and control purposes. This enables operators and control systems to have a comprehensive view of the power system's behavior and make informed decisions to enhance system stability, detect abnormalities, and implement effective protection measures.

The MatPDC initiates a TCP connection with each PMU device, and the TCP objects are created using the IP addresses and ports of the PMUs. Once the connection is established, the MatPDC can receive data packets from each PMU. The received data packets are then interpreted and parsed by the MatPDC to extract the relevant information such as time stamps, synchrophasor data, device identification, and data quality indicators. This process involves the use of the message framework defined in IEEE Std C37.118.2-2011, which describes the structure and format of synchrophasor data messages.

After parsing the data packets, the MatPDC can select the desired synchronized data for further real-time analysis. The time aligned data can be directly fed into applications like state estimation, power system stability analysis, and fault detection developed in MATLAB/Simulink environments. Because the real-time data is available in the MATLAB workspace, there is no need for other toolboxes or software for integration; the data can be called directly into MATLAB/Simulink-based applications. The selected data can also be stored in a relational database for future analysis purposes.

Furthermore, the MatPDC can also transmit the selected data to another PDC over Ethernet. This is done by converting the data into TCP data packets and transmitting them over the TCP/IP protocol. This enables the MatPDC to communicate with other PDCs and share synchronized data between them. In addition to receiving and transmitting data, the MatPDC also performs real-time integrity validation of synchronized data packets. This means that the MatPDC checks the quality and validity of the received data packets to ensure that they are accurate and reliable. This is done using techniques such as data validation checks using data flags and data redundancy checks using CRC.

5.1. Connection establishment

The MatPDC connects to the server devices using the TCP/IP protocol over Ethernet in commanded mode. The TCP/IP protocol is a reliable, connection-oriented protocol that guarantees the delivery of data packets to their intended destination. The protocol is based on the concept of a client–server architecture, where a client requests services or resources from a server, and the server provides these services in response to the client's requests.

TCP/IP protocol is commonly used in industrial settings for its reliability, security, and error-checking mechanisms. It allows for communication between devices across networks and ensures that data is transmitted and received accurately and efficiently.

The MATLAB Instrument Control Toolbox provides functions and tools that allow users to connect to devices using TCP/IP client communication. The toolbox creates TCP/IP objects that can be used to communicate with remote hosts, such as PMUs and PDCs, through network socket communication.

5.2. Integrity evaluation

It is important to check the integrity of the received frames to make sure that the data is correct and reliable. The CHK value in the last two bytes of every frame is used for this purpose. The CRC-CCITT error detection method [18] is employed to compute the CHK value, which is then sent as a CHK field in the last two bytes of the frame.

When a frame arrives at the MatPDC, a new CHK value is calculated using the same CRC-CCITT algorithm. The CHK value that was received in the frame and the CHK value that was calculated are compared. If both values match, the frame is considered valid, and the data is decoded for further processing. However, if the values do not match, it indicates that the frame has been compromised during transmission, and the MatPDC discards the frame.

5.3. Frame parsing/decoding

The SYNC field is a critical component of every frame transmitted by the PMU. It contains information about the type of frame and the version of the configuration frame. The first byte of the SYNC field is a synchronization byte that is always set to 0xAA, followed by the first four bits (0–3 bits) of the second byte, which defines the configuration frame version, and bits 4–6 represent the frame type.

The frame type is used to identify whether the frame contains data, configuration information, or header information. The CFG frame is used to configure the PMU's measurement parameters, while the DATA frame contains the actual measurement data. When a frame is received by the MatPDC, it is decoded based on the type of frame (DATA, CFG, or HDR) and stored for later use. The version of the configuration frame is used to determine the format of the CFG frame and how it should be parsed.

5.4. Configuration frame interpretation

In the commanded mode, requests for configuration frames are made by the client to the server. The configuration frames received from each connected server device (PMU or PDC) are parsed and interpreted by the MatPDC. The second byte of the SYNC field value of each frame is used to identify the configuration frame type and version. The SYNC field value AA21 hex represents configuration-1 with version-1, while AA31 hex represents configuration-2 with version-1. If the SYNC field value is AA52 hex, it represents configuration-3 with version-2.

After parsing the configuration frames, the MatPDC generates an output configuration frame that contains the information of all the connected PMUs/PDCs. This output configuration frame is communicated from MatPDC to other PDCs when requested, after converting it to a C37.118 compliant TCP packet. This allows other PDCs to obtain information about the configuration of the devices connected to MatPDC, which is necessary for synchronizing and analyzing data from multiple devices.

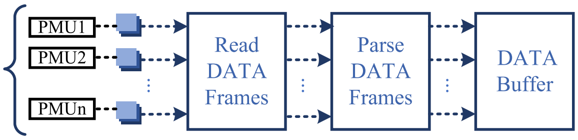


Fig. 3. Multiple PMUs data frame processing.

5.5. Header frame interpretation

First, the request for the header frame is made by the MatPDC by sending the *send HDR* command to the PMU/PDC as a TCP packet. The server device acknowledges the request and responds with the header frame of the device. The header frame is read and identified by the SYNC field of the frame. If the SYNC field is AA11 hex, it represents the received frame is a header frame. Its principal content is in ASCII format and is often used for PMU setup and verification. It has information about things like the data sources, scalability, algorithms, and filters.

5.6. DATA frame

The MatPDC first sends a command to initiate continuous data transmission from the PMUs/PDCs. The server devices acknowledge the command and start transmitting data frames. The MatPDC continuously read data frames from the TCP socket. If multiple PMUs/PDCs are connected, the MatPDC reads data frames from each TCP socket corresponding to the connected server devices. The Fig. 3 represents how data frames are decoded.

The received data frames are decoded using the configuration frames, which provide information about the PMUs/PDCs and the data formats. The decoder function identifies the SYNC field value of the data frame, which is AA01 hex. Then it extracts the relevant data fields, including phasor values, frequency, and ROCOF, among others.

The parsed data frames are stored in a separate data buffer for each PMU/PDC. The buffer depth is a user-configurable parameter, and all PMU data buffers have the same depth. The depth of the buffer determines the number of data frames that can be stored. The data buffer stores the parsed data frames based on the arrival timestamp.

5.7. Data aggregation and alignment

Time alignment [19] is an essential process in wide-area measurement systems to ensure that data from different PMUs located in a large power system are synchronized in time. The Fig. 4 represents the data aggregation and time alignment of synchrophasor data obtained from multiple PMUs. For data aggregation with time, the MatPDC creates the PMU data buffer for each PMU. The depth of the PMU data buffer is n, which means it stores n rows of PMU data frames. If the reporting interval of the PMU is t_{Int}, then the buffer history length is $T_{len} = n.t_{Int}$, where n is buffer depth. The latency of PDC depends on the buffer history length. When the PMUs start transmitting data frames, the MatPDC reads and stores them in the corresponding PMU data buffer based on their arrival timestamp [20]. The timestamp of the first received data frame serves as the reference point for relative time. The waiting time is the time interval between the first data frame's timestamp and the nth received data frame (where n is the buffer depth).

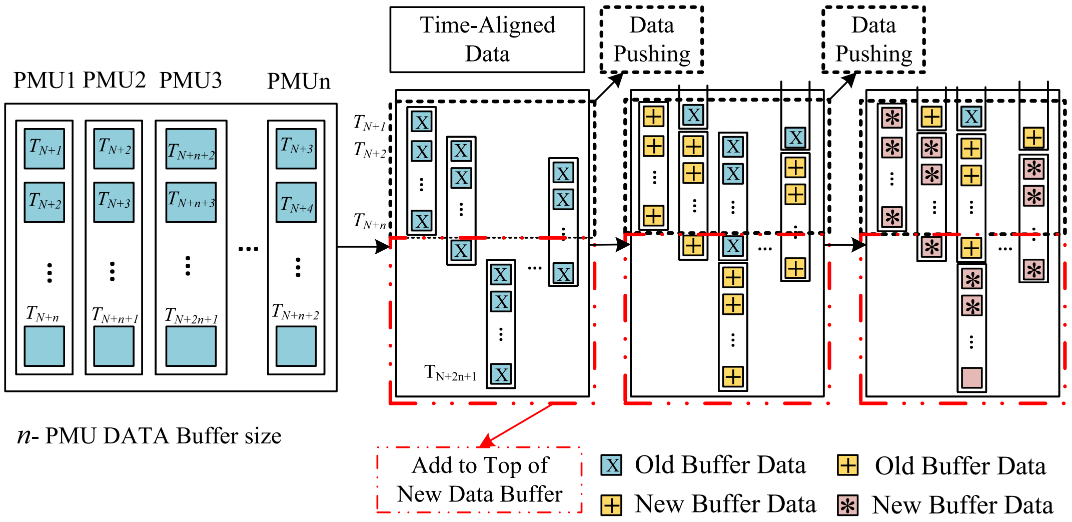


Fig. 4. Data Aggregation with Time Alignment.

Once the waiting time has elapsed, the MatPDC starts processing the received data frames for time alignment. The data frames from all PMUs' data buffers are read and put into a time-aligned fashion. The data frames coming within the waiting time are used for time alignment, while missing data frames are represented as not a number (NAN).

By performing time alignment, the MatPDC ensures that data from different PMUs are synchronized, which is essential for accurate analysis and interpretation of phasor data from multiple sources.

5.8. Data pushing and communication

Data pushing is the process of transferring time-aligned buffer data from the MatPDC to another destination. This is done after the MatPDC has received data frames from all connected PMUs and has performed time alignment. The data from time-aligned data buffer is then pushed serially for further communication. The MatPDC can transmit data to three different destinations:

- Synchrophasor applications: The MatPDC, operating within the Matlab environment, has the capability to provide time-synchronized data from a large wide-area measurement system to synchrophasor-based applications developed in the Matlab environment. This data is presented in a human-readable format, which allows for easy integration and analysis without the need for additional processing steps. These synchrophasor-based applications can utilize the received data to simulate and analyze the real-time behavior of the power system.
- Relational Database: The MatPDC has the capability to store the time-aligned data in a relational database, providing a centralized and structured storage solution. By storing the data in a database, it becomes readily accessible for future analysis and utilization in various applications. MATLAB has a fully developed Database Toolbox that allows data to be stored in various relational databases, including Microsoft SQL Server, SQLite, PostgreSQL, MySQL, and more. Any relational database supported by the MATLAB Database Toolbox can be integrated with MatPDC. The MatPDC's ability to store time-aligned data in a relational database facilitates system monitoring, advanced analysis, and future planning and expansion of power systems.
- Higher-level PDC: The MatPDC has the capability to transmit the time-aligned data from all connected PMUs to a higher-level PDC. Prior to transmission, the configuration frames are converted into a format compliant with the IEEE Std. C37.118 standard. These converted frames are then sent as TCP packet streams.

6. MatPDC performance evaluation

The evaluation of MatPDC's performance is being carried out on a Windows 10 machine with an Intel Core i7-2600 CPU, 8 GB of RAM, and MATLAB 2023A. Three distinct test cases are considered, each involving unique hardware and software configurations of PMUs.

- Case 1: In this case, five hardware PMUs are deployed at different locations within the IIT Kanpur campus, connected to the actual power system. The objective is to evaluate the performance of MatPDC in a real-world scenario using these PMUs. The MatPDC receives data from the PMUs and ensures time alignment to a common data set. The PMUs capture the real-time behavior of the power system, and the MatPDC processes and consolidates the synchronized data, which can be utilized for analysis, monitoring, and control purposes.
- Case 2: In this case, a larger WAMPAC system is developed for analysis purposes. The IIT Kanpur distribution system is modeled using Hypersim software. To evaluate the performance of MatPDC more comprehensively, ten software/virtual PMUs are placed at different buses within the simulation model. This setup enables a thorough assessment of MatPDC's behavior in a simulated environment with an increased number of PMUs. The case provides processing capability and performance of MatPDC when dealing with a larger WAMPAC system configuration.
- Case 3: In this case, the IEEE 39 bus system is modeled using Real-Time Digital Simulator (RTDS), and 24 software/virtual PMUs are strategically placed at different buses within the system. The purpose of this case is to evaluate the scalability and performance of MatPDC in a larger power system with an increased number of PMUs. By simulating a larger-scale power system and incorporating multiple PMUs, the case aims to assess how well MatPDC can handle the higher volume of synchronized measurements, ensuring reliable data collection, time alignment, and efficient processing. This evaluation provides insights into the capabilities and limitations of MatPDC when dealing with larger power systems and a higher density of PMUs.

Case 1: Five hardware PMUs

In this case, the performance of MatPDC is evaluated in a scenario where five hardware PMUs are connected to it. These PMUs are

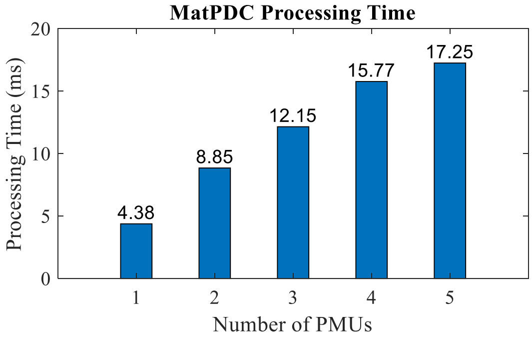


Fig. 5. Representation of MatPDC processing time when five hardware PMUs are connected to it. The processing time refers to the time it takes to perform tasks such as data collection, validation, and synchronization on the synchronized synchrophasor data received from PMUs.

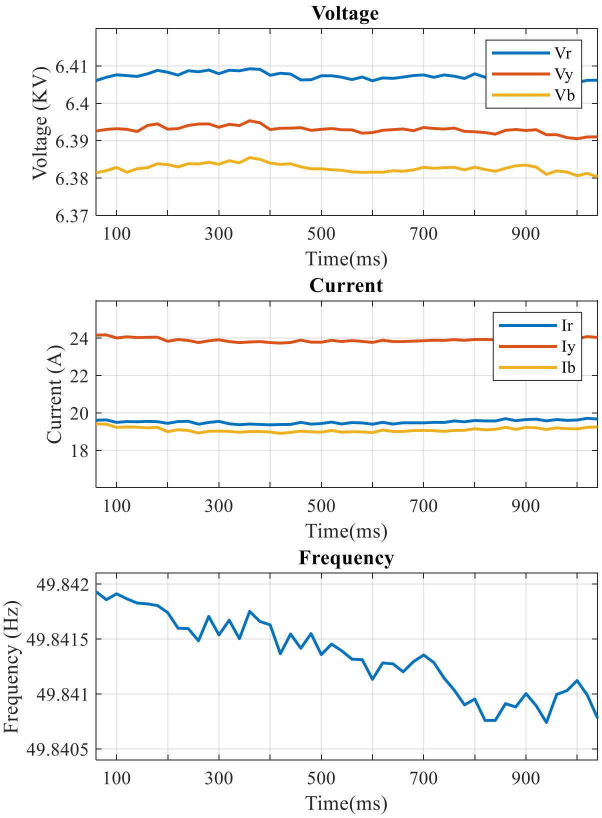


Fig. 6. Online monitoring of the data from the PMU at substation 8.

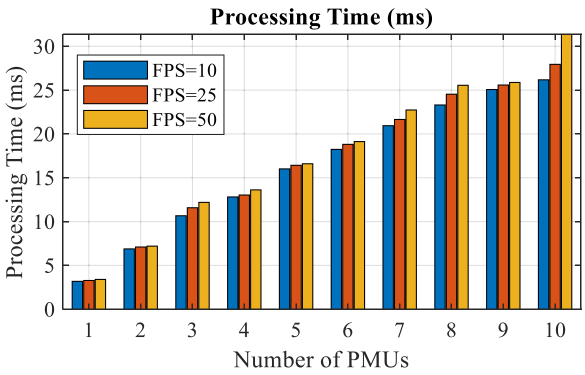


Fig. 7. Processing time of MatPDC when connected PMUs are 1 to 10 PMUs.

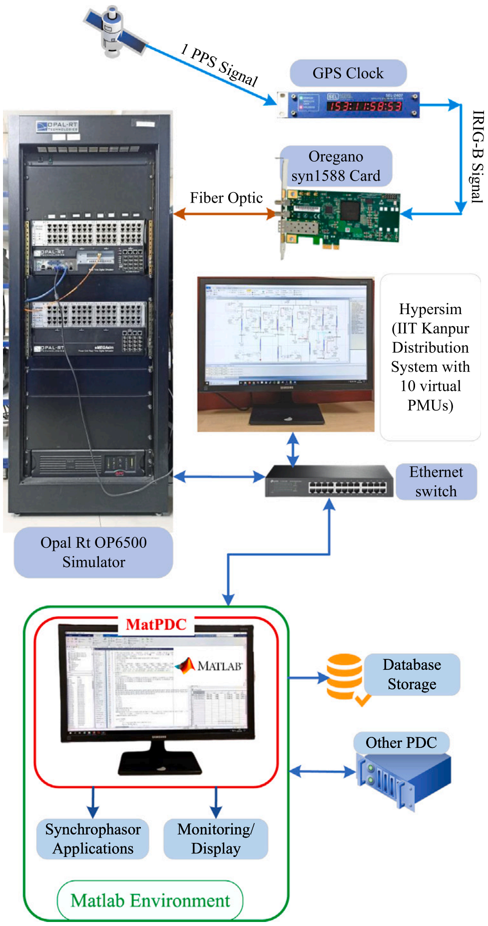


Fig. 8. Laboratory setup of case 2.

strategically positioned at different locations within IIT Kanpur campus and communicate with the MatPDC using the Ethernet. The hardware setup includes the Arbiter 1133 A PMU from Arbiter System, two Q-PMU-9 units from Qualitrol, and two PMUs from SEL, namely, SEL 421 and SEL 451. Each PMU's data frame consists of 12 phasors, 13 analog values, and four digital values, providing a comprehensive set of measurements.

The MatPDC receives the data frames from the hardware PMUs and performs a precise time alignment among the measurements. This time alignment process ensures that the measurements from different PMUs, which may have slight timing variations due to communication delays, are synchronized accurately. Once the time alignment is achieved, the MatPDC is capable of pushing the time-aligned buffer data to various destinations as per the specified requirements.

Fig. 5 shows the processing time of the MatPDC in milliseconds for a 50 FPS reporting rate. It can be observed that as the number of PMUs connected to the MatPDC increases, the processing time also increases. This is expected as the MatPDC has to process more data frames, perform time alignment for each of them, and push the time-aligned buffer data to the different destinations. This evaluation aims to assess the performance of MatPDC in effectively handling multiple hardware PMUs, processing their measurements, and delivering the time-aligned data to the designated destinations.

To illustrate the monitoring capabilities of the MatPDC, online data was collected from one of the PMUs located at substation 8, and the results were presented in Fig. 6. This particular PMU has a data frame

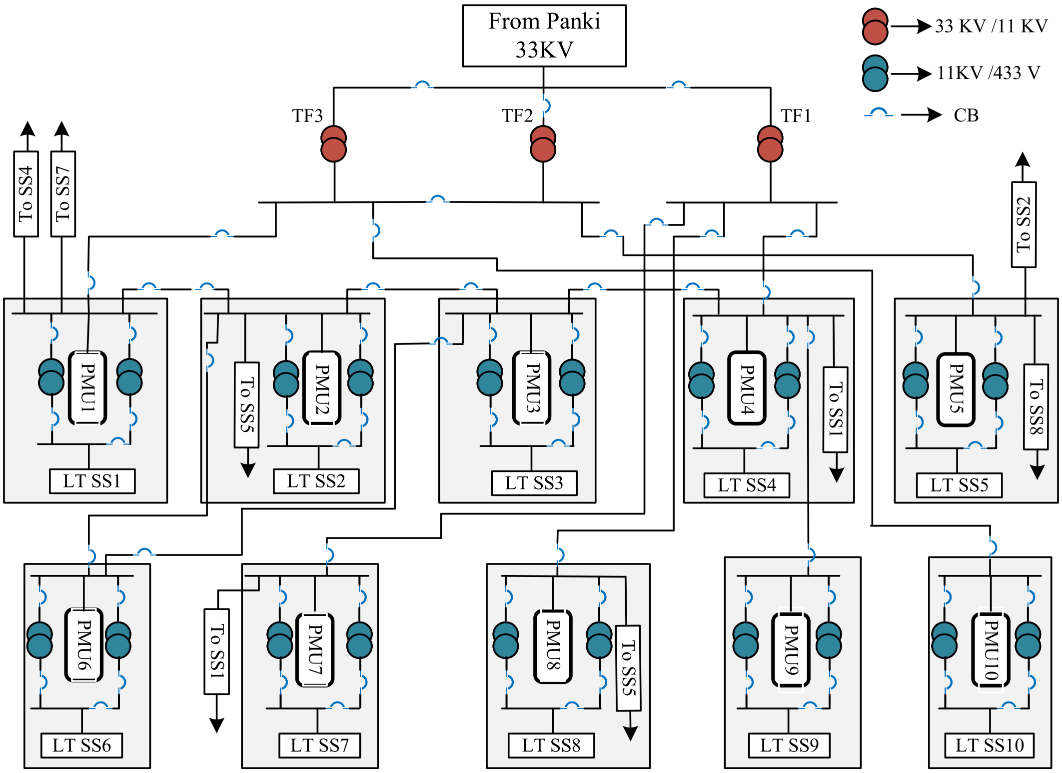


Fig. 9. SLD of IIT Kanpur distribution system with 10 Software PMU locations.

size of 138 bytes and reports measurements at a rate of 50 frames per second, with a reporting interval of 20 ms.

In Fig. 6, the time axis represents the milliseconds of the actual timestamp, specifically January 19, 2023, at 6:07:46.100 UTC. The plot shows the voltage magnitude, voltage angle, and frequency measurements captured by the PMU at substation 8, covering a time duration of one second.

This graphical representation of the PMU's data provides a visual understanding of the measured parameters and their variations over time. It serves as a demonstration of the MatPDC's ability to collect, process, and present real-time measurements for effective monitoring and analysis of the power system.

Case 2: IIT Kanpur system with 10 PMUs

In this case, the IIT Kanpur distribution system was simulated in the Hypersim environment, and the performance of the MatPDC was evaluated using 10 PMUs placed at different buses within the simulation model. The Single line diagram (SLD) of the IIT Kanpur system shown in Fig. 9. Fig. 8 shows the lab setup for this case. The OPAL-RT OP5600 real-time simulator, along with the Oregano syn1588 card, was used to provide accurate timestamp synchronization with GPS using the SEL 2407 satellite clock. Each PMU at the 10 substations of the IIT Kanpur distribution system sends a data frame of 138 bytes, which includes 12 phasors, 13 analog values, frequency, frequency deviation, and four digital values in the TCP packet format, and reports at a rate of 10 frames per second.

Fig. 7 illustrates the processing time of the MatPDC as the number of connected PMUs increases, up to a maximum of 10 PMUs. The processing time refers to the amount of time required by the MatPDC to complete tasks such as data collection, validation, and synchronization on synchronized synchrophasor data received from PMUs. To offer a clearer understanding and capture the variation in MatPDC's processing time, the evaluation begins with a single PMU connected to the MatPDC. The processing time is calculated for different reporting rates of 10, 25, and 50 frames per second.

Subsequently, one PMU connection is added at a time to the MatPDC, and the processing time is measured until all 10 PMUs are connected. This stepwise approach provides insights into the incremental impact on the MatPDC's processing time as the number of PMUs increases.

By analyzing the processing time under varying PMU configurations and reporting rates, this evaluation aims to assess the efficiency of the MatPDC in handling multiple PMU connections. The results obtained from Fig. 7 provide valuable information on how the MatPDC's processing time is affected as the system complexity increases. Furthermore, MatPDC's processing speed is also influenced by the processing capability of computer on which it is running.

Case 3: 24 PMUs

A performance evaluation of the MatPDC has been conducted in this case using 24 software PMUs, which were developed in RSCAD and simulated in RTDS with a reporting rate of 60 frames per second (FPS). RTDS is a real-time power system simulation platform that allows for complex simulations of power systems.

Fig. 10 shows the practical implementation for this case. In the RTDS, each GTNET card can simulate a maximum of 24 PMUs, and the PMUs send four analog values, two phasor values, and one digital value. To synchronize the software PMUs with GPS, the GTSYNC card in the RTDS is utilized. The SEL 2407 GPS clock receives a 1 pulse per second (PPS) input from the GPS antenna and generates an IRIG-B signal. IRIG-B is a time code format used for synchronizing devices to a common time reference.

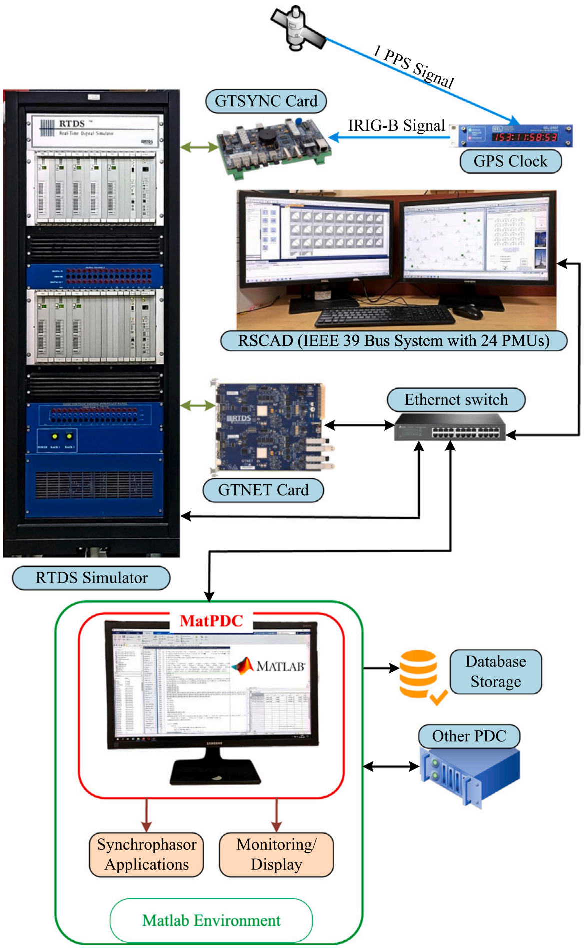


Fig. 10. Laboratory setup for case 3.

Table 1
Processing time when up to 24 PMUs with 60 FPS connected to MatPDC.

Processing time of MatPDC in case 3			
Connected PMUs	Time (ms)	Connected PMUs	Time (ms)
1	3.23	13	35.9
2	7.25	14	38.07
3	13.0	15	40.5
4	15.5	16	43.6
5	18.1	17	45.50
6	20.5	18	48.15
7	23.1	19	50.8
8	23.6	20	53.05
9	26.0	21	55.08
10	28.36	22	58.25
11	30.9	23	60.30
12	33.25	24	63.56

Table 1 provides the processing time required by the MatPDC for different numbers of connected PMUs. When only one PMU is connected, the MatPDC takes 3.23 ms (ms) for processing and time alignment of the data frames. When two PMUs are connected, the processing time increases to 7.25 ms. Similarly, for the case of 24 PMUs, the processing and time alignment of data frames take 63.56 ms.

This table presents valuable information regarding the processing time of the MatPDC as the number of connected PMUs increases. It demonstrates the impact of PMU quantity on the MatPDC's performance, providing insights into the scalability and efficiency of the system. These results aid in understanding the capabilities of the MatPDC and help optimize its performance in handling larger numbers of PMUs within a WAMPAC system setup.

The RTDS system's GTNET card has a maximum simulation capacity of 24 PMUs. MatPDC has been tested with this limitation, but it is important to note that it can potentially integrate more than 24 PMUs if they are available. However, it is crucial to consider that integrating additional PMUs will impact MatPDC's processing time. This is because MatPDC relies on the processing capability of the host PC it runs on, and handling a larger number of PMU data will increase the computational load, potentially leading to longer processing times.

7. Online monitoring application

The online monitoring application, developed in Matlab, demonstrates the MatPDC's monitoring capabilities by providing real-time information of various parameters for all connected PMU devices. Users can easily track the number of connected PMUs, select a specific PMU using the "Select PMU" option, and monitor its associated parameters.

The application features phasor representations of voltages and currents, along with graphical representations of the online frequency. These features allow users to visualize variations of these parameters over time. An indicator is incorporated to indicate the GPS synchronization status of each PMU. The indicator turns green when the PMU is synchronized to the UTC source and red when it is unsynchronized.

To visualize the timing of measurements, the application consistently displays the UTC time of the selected PMU. Additionally, it provides the RMS values of voltage and current for each phase, allowing users to monitor the magnitudes of these quantities.

The dashboard of monitoring application is shown in Fig. 11. The "Connected Devices" field illustrates the PMUs that are currently connected to the MatPDC. In this particular scenario, two PMUs are connected: one is at the Substation 33 KV, and the other is at the Substation 8. The indicators in the application show the synchronization status of these PMUs. The green color of the LED indicates that both PMUs are synchronized to the UTC timing source.

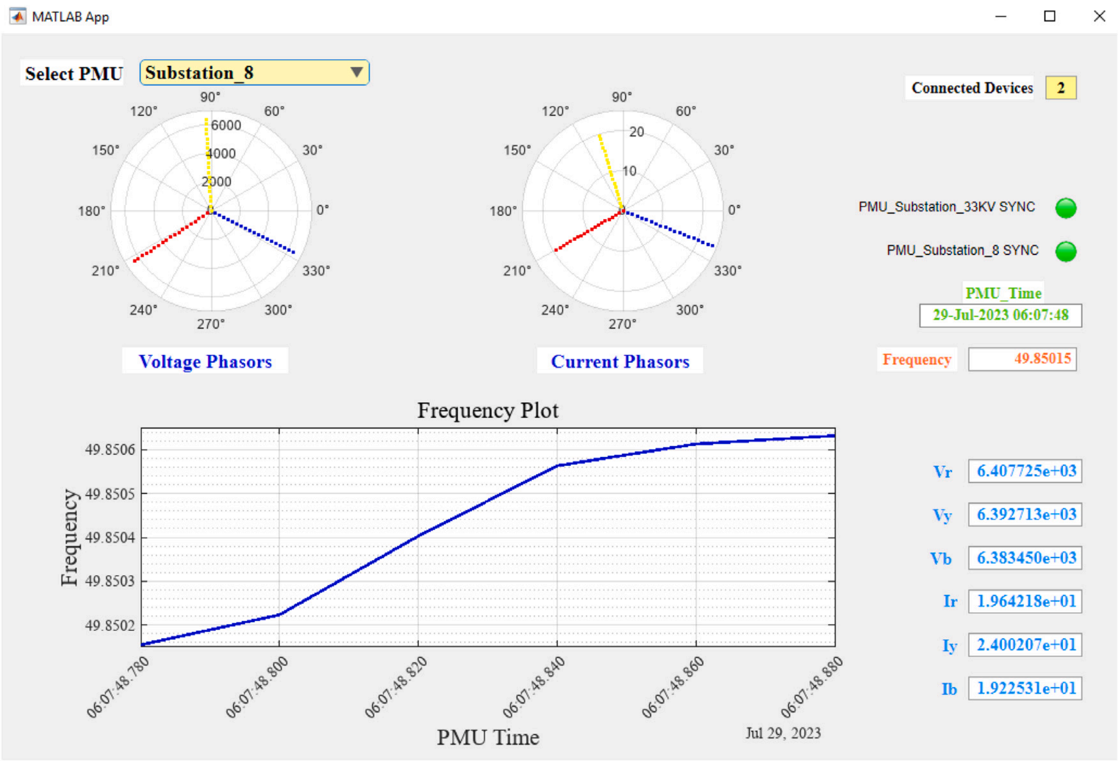
In the "Select PMU" field in Fig. 11, the PMU of the 33 KV substation is currently selected. The "Voltage and Current Phasor" section displays the online rotation of phasors for the 33 KV substation. The "Frequency" plot shows the variations in frequency over time. The "PMU Time" field provides the time stamp of measurements recorded by the selected PMU.

This online monitoring application serves to collect, process, and present real-time measurements for effective monitoring and analysis of the power system.

8. Conclusion

The development of the MatPDC was driven by the need to access and utilize online synchrophasor data in the MATLAB environment for power system analysis and research. By having a PDC compliant with the IEEE Std. C37.118.2-2011, researchers and engineers can now seamlessly integrate synchrophasor data into their MATLAB-based applications. This development reduces complexity and latency, and enhances the overall efficiency of working with real-time synchrophasor data. The MatPDC offers several key features that enable efficient processing of data from wide-area measurement systems. It provides real-time access to synchronized data streams from multiple PMUs, ensuring data integrity through verification checks and validation.

Overall, the MatPDC serves as a valuable tool for researchers and engineers, enabling them to work with real-time synchrophasor data in the MATLAB environment. Its key features and successful performance in test cases highlight its significance in power system analysis, monitoring, and control, paving the way for the development of advanced algorithms and the improvement of power system operations.



Select PMU: Substation_8

Connected Devices: 2

PMU_Substation_33KV SYNC ●

PMU_Substation_8 SYNC ●

PMU_Time: 29-Jul-2023 06:07:48

Frequency: 49.85015

Vr: 6.407725e+03

Vy: 6.392713e+03

Vb: 6.383450e+03

Ir: 1.964218e+01

Iy: 2.400207e+01

Ib: 1.922531e+01

Fig. 11. Online monitoring of the parameters of the PMUs connected to the MatPDC.

CRediT authorship contribution statement

Sugandh Pratap Singh: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Visualization, Investigation, Validation. **Saikat Chakrabarti:** Supervision, Project administration, Writing – review & editing. **Devesh Shukla:** Visualization, Investigation, Validation. **Vladimir Terzija:** Supervision, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Terzija V, Valverde G, Cai D, Regulski P, Madani V, Fitch J, et al. Wide-area monitoring, protection, and control of future electric power networks. Proc IEEE 2010;99(1):80–93.

[2] Chakrabarti S, Kyriakides E, Bi T, Cai D, Terzija V. Measurements get together. IEEE Power Energy Mag 2008;7(1):41–9.

[3] IEEE standard for synchrophasor data transfer for power systems. IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005), 2011, p. 1–53.

[4] Nechifor A, Albu M, Hair R, Terzija V. An enterprise platform for wide area monitoring of load dynamics using synchronized measurement technology. IFAC-PapersOnLine 2016;49(27):85–90, IFAC Workshop on Control of Transmission and Distribution Smart Grids CTDSG 2016.

[5] Preston G, Radojević Z, Kim C-H, Terzija V. New settings-free fault location algorithm based on synchronised sampling. IET Gener Transm Distrib 2011;5(3):376–83.

[6] Cai D, Regulski P, Osborne M, Terzija V. Wide area inter-area oscillation monitoring using fast nonlinear estimation algorithm. IEEE Trans Smart Grid 2013;4(3):1721–31.

[7] Cai D, Regulski P, Osborne M, Terzija V. Wide area inter-area oscillation monitoring using fast nonlinear estimation algorithm. IEEE Trans Smart Grid 2013;4(3):1721–31.

[8] Nechifor A, Regulski P, Cai D, Terzija V. Development of a flexible laboratory testing platform for assessing steady-state and transient performance of WAMS. In: 2011 IEEE international workshop on applied measurements for power systems (AMPS). 2011, p. 62–7.

[9] Terzija VV. Adaptive underfrequency load shedding based on the magnitude of the disturbance estimation. IEEE Trans Power Syst 2006;21(3):1260–6.

[10] Sun M, Feng Y, Wall P, Azizi S, Yu J, Terzija V. On-line power system inertia calculation using wide area measurements. Int J Electr Power Energy Syst 2019;109:325–31.

[11] Wall P, Dattaray P, Jin Z, Mohapatra P, Yu J, Wilson D, et al. Deployment and demonstration of wide area monitoring system in power system of Great Britain. J Mod Power Syst Clean Energy 2016;4(3):506–18.

[12] Grid Protection Alliance, https://www.gridprotectionalliance.org/phasor-PDC.html, Accessed: Nov. 20. 2022.

[13] Nechifor A, Albu M, Hair R, Terzija V. An enterprise platform for wide area monitoring of load dynamics using synchronized measurement technology. IFAC-PapersOnLine 2016;49(27):85–90, IFAC Workshop on Control of Transmission and Distribution Smart Grids CTDSG 2016.

[14] Naglic, et al. Synchro-measurement application development framework: An IEEE standard C37.118.2-2011 supported MATLAB library. IEEE Trans Instrum Meas 2018;67(8):1804–14.

[15] Almas MS, Vanfretti L, Baudette M. BabelFish—Tools for IEEE C37. 118.2-compliant real-time synchrophasor data mediation. SoftwareX 2017;6:209–16.

[16] IEEE guide for phasor data concentrator requirements for power system protection, control, and monitoring. IEEE Std C37.244-2013, 2013, p. 1–65.

[17] IEEE standard for synchrophasors for power systems. IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995), 2006, p. 1–65.

[18] Peterson WW, Brown DT. Cyclic codes for error detection. Proc IRE 1961;49(1):228–35.

[19] IEEE standard for phasor data concentrators for power systems. IEEE Std C37.247-2019, 2019, p. 1–44.

[20] Zhu K, Rahimi S, Nordström L, Zhang B. Design phasor data concentrator as adaptive delay buffer for wide-area damping control. Electr Power Syst Res 2015;127:22–31.