 doi:10.20944/preprints202203.0051.v1

Data Descriptor

Statistical Data Set and Data Acquisition System for Monitoring The Voltage and Frequency of The Electrical Network in An Environment Based On *Python* and *Grafana*

Javier Fernández-Morales[1], Juan-José González-de-la Rosa[1,*], José-María Sierra-Fernández[1], Manuel-Jesús Espinosa-Gavira[1], Olivia Florencias-Oliveros[1], Agustín Agüera-Pérez[1], and José-Carlos Palomares-Salas[1], Paula Remigio Carmona[1].

[1] Department of Automation Engineering, Electronics, Architecture and Computers Networks. Research Group PAIDI-TIC-168, University of Cádiz, Higher-Polytechnic School of Algeciras, E-11202 Algeciras, Spain; javier.fernandezmorales@uca.es (J.F.M.); juanjose.delarosa@uca.es (J.J.G.D.R.); olivia.florencias@uca.es (O.F.O.); manuel.espinosa@uca.es (M.J.E.G.); josemaria.sierra@uca.es (J.M.S.F.); agustin.aguera@uca.es (A.A.P.); josecarlos.palomares@uca.es (J.C.P.S.); paula.remigio@uca.es(P.R.C)

* Correspondence: juanjose.delarosa@uca.es

Abstract: This article presents a unique set of voltage and current data from a public building and acquired using a hybrid measurement solution that combines *Python™* and *Grafana™*. The transversal purpose consists of contributing to the community with a vision of the quality of the supply more oriented to the monitoring of the state of the network, providing a more realistic vision, which allows a better understanding, and the adoption of the best decisions to achieve the efficient energy management and thus optimize the operation and maintenance of power systems. The work focuses on higher order statistical estimators that, combined with exploratory data analysis techniques, improve the characterization of the shape of the stress signal. These techniques and data, together with the acquisition and monitoring system, present a unique combination in the line of low-cost measurement solutions. It also incorporates the underlying benefit of the contribution to industrial benchmarking. The paper also includes a computational comparison between *Python™* and *LabVIEW™* to elicit the performance of the measurement solution.

Dataset: https://doi.org/10.7910/DVN/EGI7X1

Dataset License: University of Cádiz and Research Unit- PAIDI-TIC-168.

Keywords: Grid frequency; *Grafana™*; Higher-order statistics, LabVIEW™; Low-cost instrument; Network-attached storage; Power Quality, *Python™*; Statistical Signal Processing; Voltage monitoring

1. Summary

Voltage and frequency are the two basic magnitudes that characterize the quality of the power supplied to consumers and the operation regime of a power system [1]. The parameters to be measured by current instrumentation consist basically of the amplitude and shape, as an alteration in the latter leads to a variation in the former. However, the situation is far from fulfilling the current demands from domestic customers and stakeholders in general.

Indeed, in the modern and smarter electrical grid there are numerous distributed resources that, although a priori independent, exert mutual influence, originating a degradation of the supplied voltage. So far, the distributed non-linear loads and intermittent energy sources in the electrical network have elicited that an historical collection of power quality (PQ) data is even more necessary for the correct interpretation of the network state, in order to make the necessary compensation and to forecast not only demand but also possible network state degradations that obey to a seasonal behavior or could be triggered

 doi:10.20944/preprints202203.0051.v1

by certain unexpected causes. For example, micro grids gather DG systems, like inverters in photovoltaic panels, wind turbines, chargers of electrical vehicles, etc., have direct consequences on the power line and in the subsequent quality, as well [2,3,4,5].

The quality of the electricity can only be provided simultaneously by manufacturers, suppliers, providers and users of electrical energy. Thus, maintaining satisfactory quality is a joint responsibility of the producer, supplier and the user. The influence abilities and responsibilities are in the same order [6].

Recording PQ data for a longer period (days or months) allows tracking the power consumption of the devices and systems.

The complex and highly dynamic electricity sector brings to light a twofold perspective. On the one hand, expensive analyzers are used in the industry, which are connected occasionally in seasonal measurement campaigns. On the other hand, there is a growing tendency to develop domestic energy quality indicators that help a novice user to understand if the supply that arrives is adequate. In both scenarios, the underlying idea is to demystify the interpretation of power quality analysis. Indeed, both scenarios incorporate measures regulated in accordance with the EN 50160 standard, always deployed occasionally. Therefore, with a better understanding of the evolution of the quality of supply, better decisions can be made regarding energy efficiency and the maintenance of facilities. Measurements and analyses beyond the EN 50160 are needed.

There are multitude types of power quality problems, although most of them are related to voltage variations. Examples of this kind of problems are: voltage lag (voltage values are out of the established limits), frequency surge (frequency significantly different to the nominal) and the insertion of harmonics into the power line, or voltage sine waves that are multiples of 50 Hz [7].

All in all, the purpose of this paper is to provide a comprehensive and unique data set associated to long term PQ monitoring that accounts for a comprehensive network status and truly avoid the data management mess, achieving more efficiently downstream analytics. The paper is structured as follows. The following section summarizes the materials and methods. Then, data description and records are described in the third section. Section 4 performs a comparison between performance of LabVIEW™ and Python™ as DAQ software. Finally, conclusions are drawn in Section 5.

2. Materials and Methods

The voltage data is acquired through a chassis and card from the manufacturer NI™. This equipment receives the voltage from the electrical network and a pulsating signal (1 PPS) from a GPS reference; the latter for purposes of temporary stamping of the measures, as explained in [8]. The data is then transferred to the computer via Ethernet, where a program developed in Python carries out the corresponding modifications and stores the results in the database and on a network-attached storage.

Frequency measurements are carried out according two different procedures. On the one hand, a standardized method is used in accordance with the standard [9], storing frequency data every 10 seconds. On the other hand, a specific method, in which the frequency and its associated uncertainty are calculated every second using the method based on the *Allan*'s variance. This method and its benefits over the traditional one, are explained in [10].

In addition to the frequency, other PQ parameters are calculated: variance, kurtosis, skewness, V_{rms}, total harmonic distortion (THD) and the power quality index (PQI), developed in [10]. These calculations are done following different time computation windows: each one cycle of 50 Hz, each 10 cycles of 50 Hz and each two seconds.

3. Data Description and Records

Once the required calculations are done, the next step is to have all data available in a database for further consultation. For those PQ parameters that are calculated in less

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

than a second, it has been proven that it is not feasible to send results directly to the database, since it ends up saturating it. Thus, they first are saved as MATLAB™ files (.mat), stored in a NAS. It has been chosen this type of file due to its high compression rate.

For these cases, it's calculated average, maximum and minimum of all the parameters registered in one second and the results are sent to the database. Nevertheless, for the analysis that records data only each two seconds, this process it's not necessary and data is sent directly from the Python program to the database. Each analysis has its own table on the database and they are created monthly. An example of this tables can be seen on Figure 1.

id	absolute_signal_id	t_fin	variance	skewness	kurtosis	pq_index	rms	thd	frequency
1	27036523	2022-01-01 00:00:00	0.514999	0.000951766	-1.50077	0.0167242	233.424	0.103112	49.994
2	27036525	2022-01-01 00:00:02	0.515402	0.00106657	-1.50077	0.0172412	233.515	0.102949	49.9921
3	27036527	2022-01-01 00:00:04	0.515336	0.00115377	-1.50092	0.017413	233.501	0.102453	49.99
4	27036529	2022-01-01 00:00:06	0.515106	0.00111	-1.50094	0.017157	233.449	0.102185	49.988
5	27036531	2022-01-01 00:00:08	0.515276	0.000938948	-1.50096	0.0171711	233.487	0.103368	49.992
6	27036533	2022-01-01 00:00:10	0.5156	0.000786635	-1.50089	0.0172814	233.56	0.103289	49.995
7	27036535	2022-01-01 00:00:12	0.51559	0.000771492	-1.50089	0.0172542	233.558	0.104094	49.996
8	27036537	2022-01-01 00:00:14	0.515332	0.000719862	-1.50097	0.0170174	233.5	0.103732	49.997
9	27036539	2022-01-01 00:00:16	0.515519	0.000716094	-1.50078	0.0170178	233.542	0.104763	50
10	27036541	2022-01-01 00:00:18	0.515712	0.000641562	-1.50069	0.0170437	233.586	0.104891	50.003
11	27036543	2022-01-01 00:00:20	0.51553	0.000755744	-1.50071	0.0169999	233.545	0.103644	49.999
12	27036545	2022-01-01 00:00:22	0.515973	0.000748121	-1.50069	0.0174153	233.645	0.104832	50.002
13	27036547	2022-01-01 00:00:24	0.516102	0.00060159	-1.50058	0.0172848	233.674	0.104484	50.004
14	27036549	2022-01-01 00:00:26	0.516016	0.000695511	-1.50062	0.0173325	233.655	0.104124	50
15	27036551	2022-01-01 00:00:28	0.516073	0.000772415	-1.50053	0.0173779	233.668	0.103838	49.997
16	27036553	2022-01-01 00:00:30	0.515898	0.00073723	-1.50065	0.0172823	233.628	0.103927	49.997
17	27036555	2022-01-01 00:00:32	0.516352	0.000760574	-1.50064	0.0177517	233.731	0.104382	49.997

Figure 1. An example of a table capture corresponding to a two-second scan. Measurement end time is found in the column "t_fin".

Figure 2. RMS values for the analysis of two seconds.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

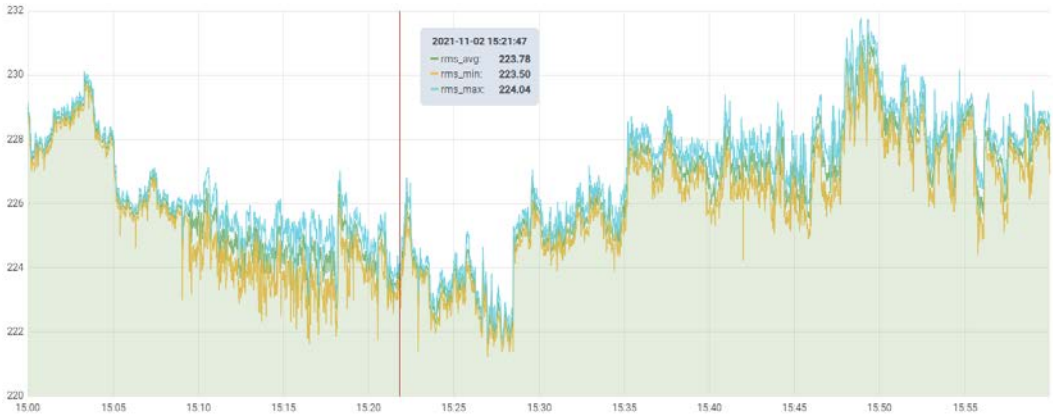

Figure 3. RMS values (minimum, average and maximum) for the analysis of one cycle.

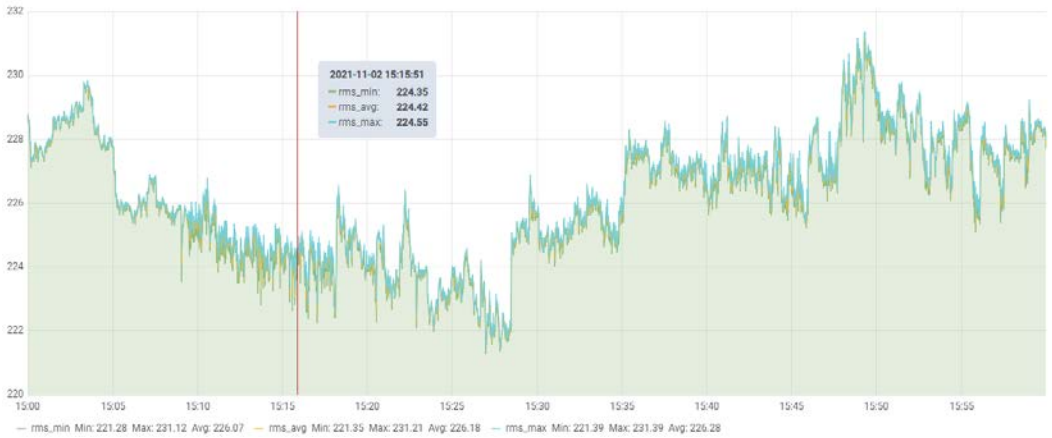

Figure 4. RMS values (minimum, average and maximum) for the analysis of 10 cycles.

Data is then available for consultation (Figures 2-4). It has been used *Grafana*™ and MATLAB™ to represent and assess data. There are multiple analyses and for each type, the PQ parameters are calculated. So, for example RMS is computed each two seconds, one cycle, 10 cycles, and so on. It can be observed in the three analyses that RMS voltage is oscillating between 222 and 232 V, approximately, near to 230V, the nominal value for RMS voltage according to the UNE 50160 standard. For the frequency it is possible to compare directly the values obtained following the method based on *Allan* variance and the traditional one, which follows the [9] standard:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

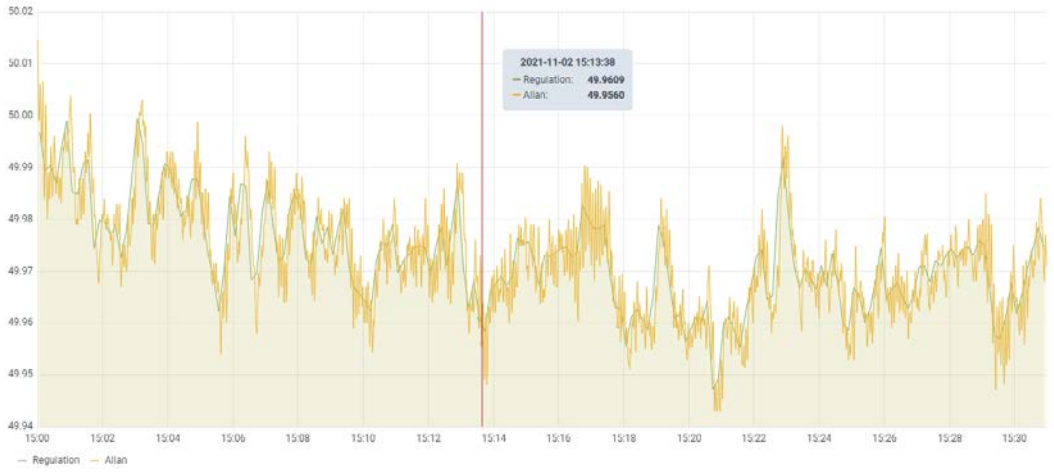

Figure 5. Frequency timeline following the regulation (green line) and following the algorithm using *Allan* Variance (orange line). Values in Hz.

It's observed that the tendence of the frequency based in *Allan* variance has a lot of peaks compared to the tendence of the traditional one. It's logical, since with *Allan* variance method the frequency is calculated every second, while in the traditional method it is calculated each 10 seconds, so the tendency in this last case looks softer. By representing the values of this kind of PQ parameters along time, it's possible to detect anomalies easily, what can be useful to detect and prevent disturbances in the power network. An example of unusual values for the skewness of the signal can be seen on Figure 6, where two peeks were detected out of the tolerable values margin:

Figure 6. Skewness: One-cycle measurements averaged each second.

This kind of disturbances can damage sensible equipment, especially if they are repeated over time. Figure 7 shows another studied PQ parameter, the kurtosis of the voltage signal:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

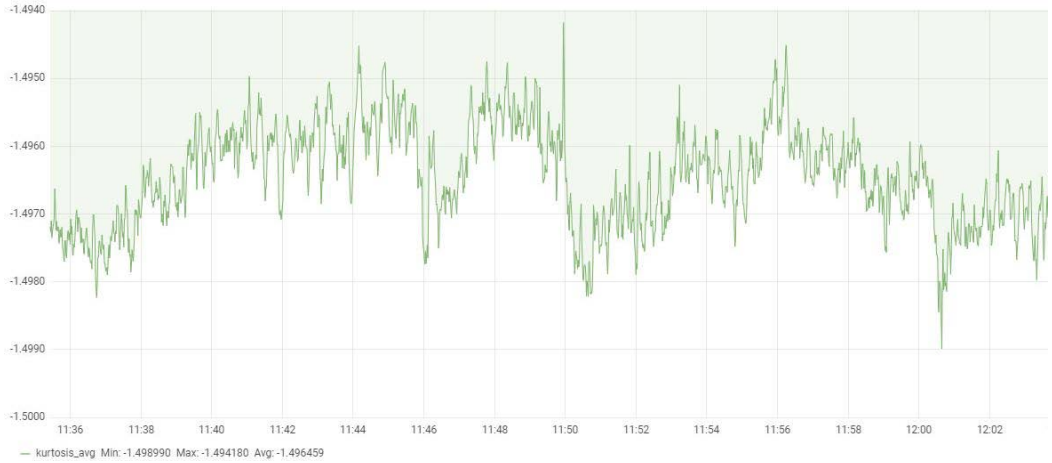

Figure 7. Kurtosis: One-cycle measurements averaged each second. They show the peakedness of the time-series, i.e., the impulsiveness of the measurements.

It's observed that values range between -1.4980 and -1.4950, approximately, so it's a very stable parameter, at least in the period shown in the figure. As a complement to the former plots, MATLAB™ offers even more possibilities for data analysis. For example, Figure 8 shows distribution of the frequency using the tool *Boxplot*:

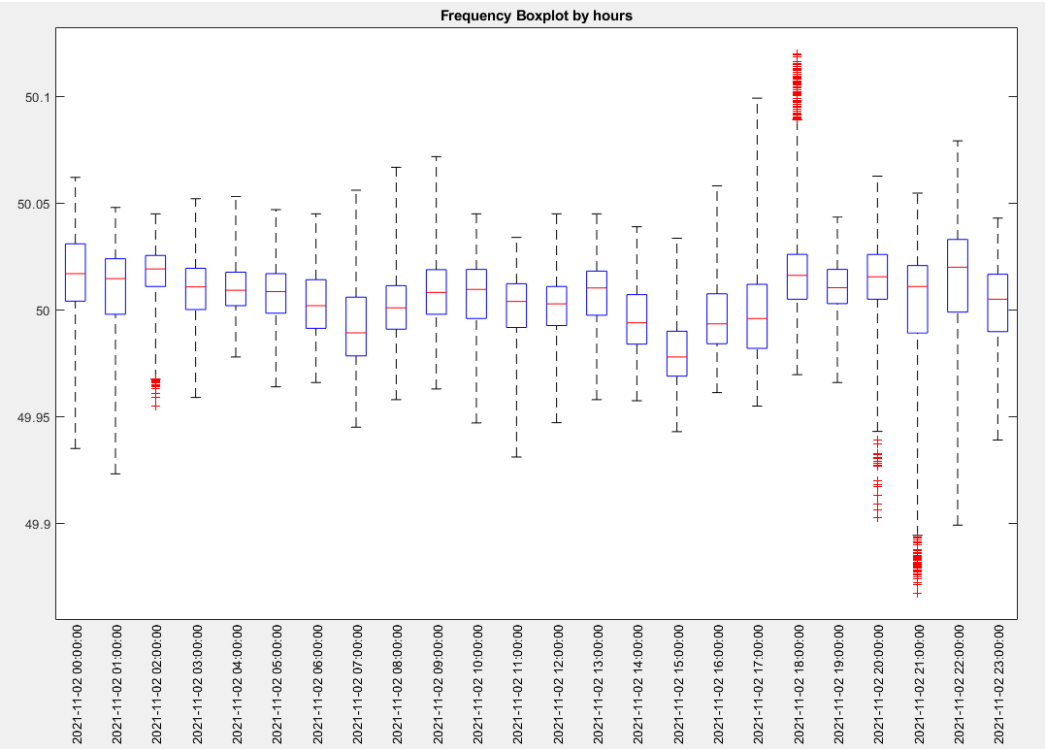

Figure 8. Boxplot of frequency values of a day, grouped by hours.

This type of graphic groups data in boxes. There is one box for each hour of the day. The red line inside the box indicates the median of data that belongs to that time interval. The superior and inferior limits of the boxes represent, respectively, the percentiles 75 and 25 of data. The whiskers of the boxes represent those data out of the percentiles that are not considered atypical, while the red crosses show the anomalous values. More details about this type of representations for RMS Voltage can be seen at [11].

It's observed that data is fluctuating around 50 Hz, the nominal value of frequency following the norm UNE 50160. Another possibility that gives MATLAB™ over Grafana™

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

is to front PQ parameters, for finding relationships between them. For example, the THD vs. the frequency has been represented in Figure 9 for all data recorded in July 2021:

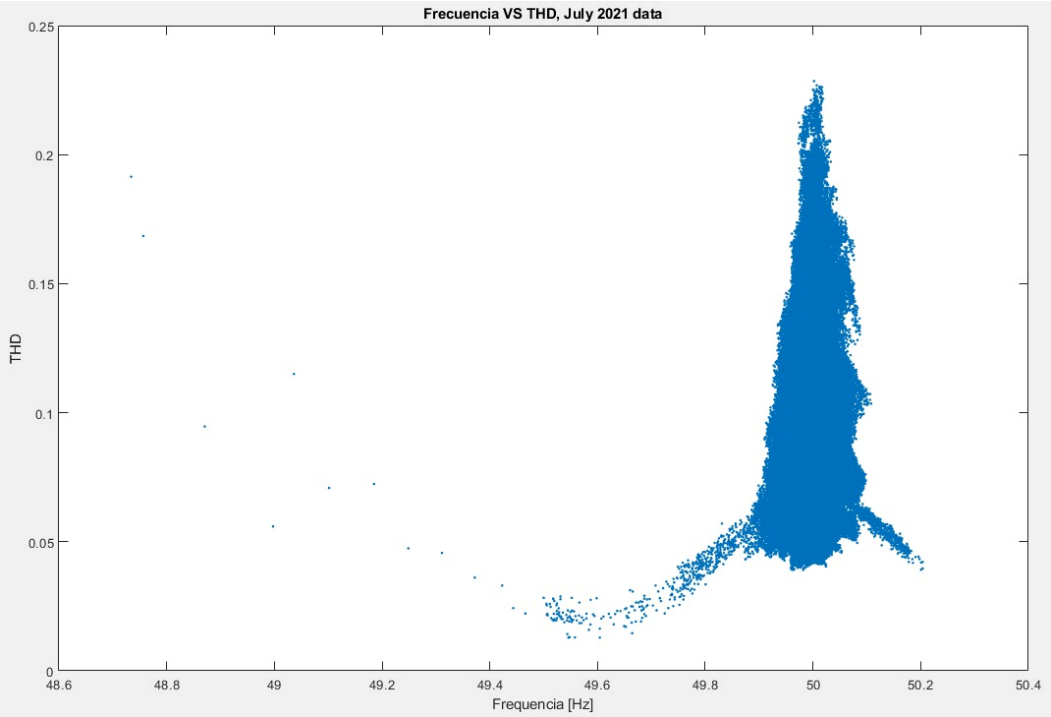

Figure 9. THD vs Frequency plot.

For a better analysis, Figure 10 shows the same plot but discarding the outliers:

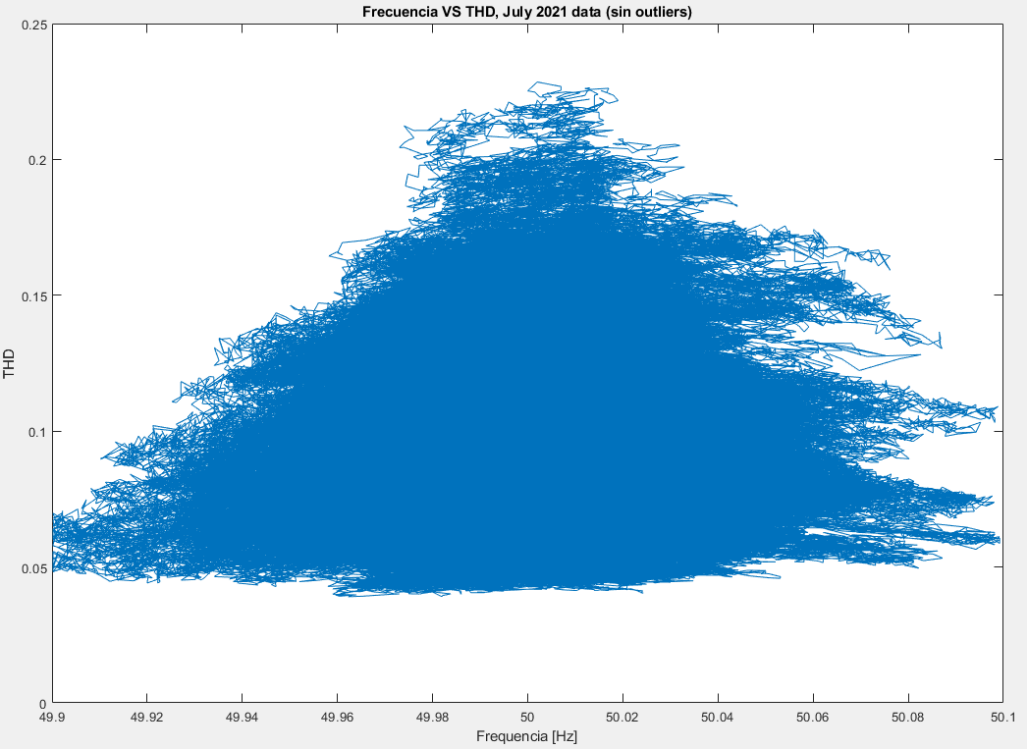

Figure 10. THD vs Frequency plot zoomed without outliers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

It can be appreciated that the figure that form the points is approximately symmetric and its vertical symmetry axis is situated on 50 Hz. The figure has a triangle form: low values of THD correspond to notable variations of frequency, while for a THD of around 0.2, frequency variations are not significant.

4. LabVIEW vs. Python

4.1. Preliminary comparison

There have been studied three program topologies for making a comparison between developing the system in Python or LabVIEW. In the first topology, there is a single program responsible of receiving ACQ-card data and saving it. The second scenario is composed of a program that receives data and connects with a data queue (a data structure which ensures synchronism), working simultaneously with another program that reads the data queue and saves all data. The third topology is similar to the second one, but it introduces a data analysis step among acquisition and data save. This causes the computation load grow.

The three topologies are studied taking 10 kSa/s, but there are two options: accumulate 10k points per acquisition (1 s) or getting 100k points (10 s). So, in total, there have been performed 6 tests: for the three topologies, they have been tested the two available options.

All situations are considered both executed in runtimes and compiled, for LabVIEW and Python. Time is measured since data is received from the acquisition card until code finishes saving all the files. It's also measured CPU % and RAM memory consumption.

Tests have been performed in a PC running Windows™ 10 Enterprise, using a DAQ-card NI-9215 in a chassis c-DAQ9191 (both from National Instruments, NI™). The chassis is connected to a router via RJ45 wire, while the computer is connected to the router over Wi-Fi. Computer has 12GBDDR3 RAM Intel Core i7 4510U @ 2.00GHz and SSD system drive. Data have been saved over HDD drive.

4.1.1. Time

The six tests described previously have been made for validating the comparison. They have been named from A to F in the charts. The results of the study of time taken for data process in each test are presented below:

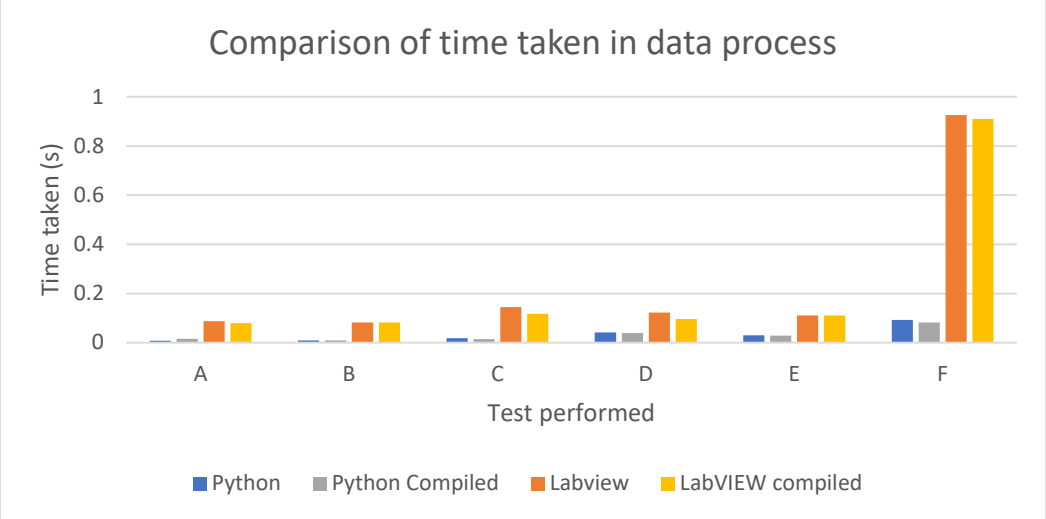

Figure 11. Comparison of time taken in data process for A-F tests.

In the figure it can be seen that for all the six cases, Python performs the whole process between acquisition to data save in less time, so the computation time available for

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

the rest of the required actions is bigger. This brings up a positive point for Python, as the more computation time is saved, the more complex can be the implemented code.

Although we try to carry out the test with the computer dedicated solely to the task, Windows cannot dedicate itself completely to it, so variability in the measured times is appreciated.

4.1.2. CPU

According to system requirements, starting with CPU load, it can be seen a similar CPU usage for both systems.

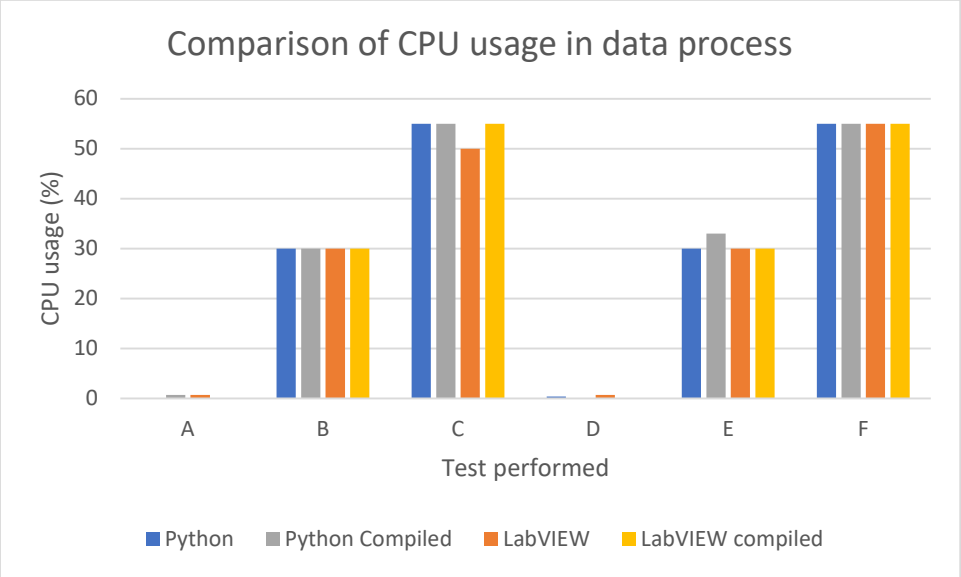

Figure 12. Comparison of CPU usage in data process for Python and LabVIEW.

As indicated before, the considered system has 4 virtual kernels. Maximum performance has been pursued in these tests, so no limits of CPU load have been set, in order to compare maximum computation speed. CPU usage can be dramatically reduced by increasing computation time, introducing delays. Around 30% and 50% of CPU usage implies respectively 1 and 2 kernels dedicated to the processes, checking queues, any time as possible. However, introducing delays invalidates the comparison of maximum computation speed.

4.1.3. Memory

LabVIEW, and specifically, the compiled version, shows the best memory performance, as seen in the next figure:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

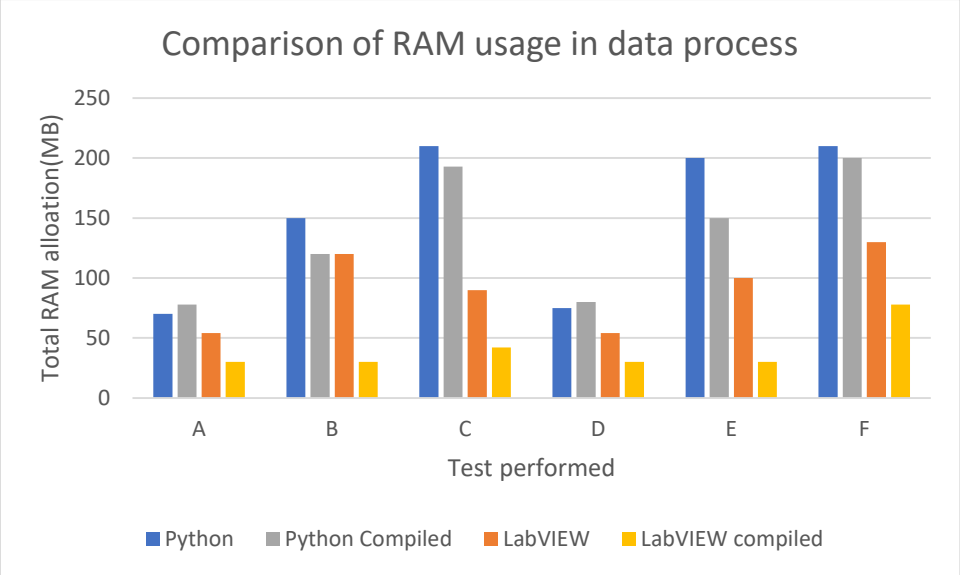

Figure 13. Comparison of RAM usage in data process for Python and LabVIEW

Python reaches over 200 MB of RAM usage in some tests, while LabVIEW, especially the compiled version, can make the whole process using only around 50 MB. If it's required to increase the computation speed, it will need more memory.

4.1.4. Preliminary conclusions

As a general comparison, looking at the memory allocation, LabVIEW shows a better performance, but computation times are bigger there, so Python has a better performance on this point. Anyway, CPU usage and computation times for both Python and LabVIEW are inside the ranges allowed by this application.

In order to examine now CPU performance, same studies would be done, but now including a delay of 10 ms in codes, while waiting data from queues. Acquisition creates a delay, so the test which makes the acquisition and save in the same program will not be studied now.

4.2. Study after adding a delay to the code

After including the delay, data process time has increased to a greater extent than the delay introduced, due to it is introduced in some codes working in parallel. It increases a bit for lower delays values, and really does not change for higher ones. However, CPU usage have been reduced enormously. As long as the processing time keeps in an acceptable range (in any case below 0.2s for 1s and below 1s for 10s data) delay time will remain in 10 ms, in order to optimize CPU performance.

4.2.1. Time

Even with the increase in processing times, all of them remain under 0.2s (much lower for Python) for queue acquisition and 1s analysis. Times are higher for 10s analysis, being around 0.2 s for Python and 1 s for LabVIEW.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

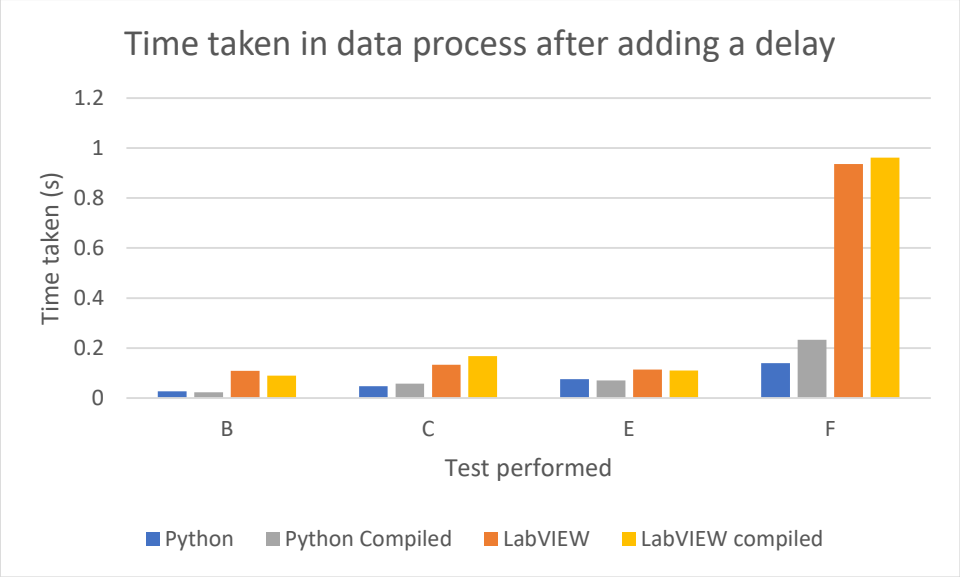

Figure 14. Processing times after introducing delays in the code.

In order to see the difference in time between introducing or not a delay, the difference between these two times is shown as follows:

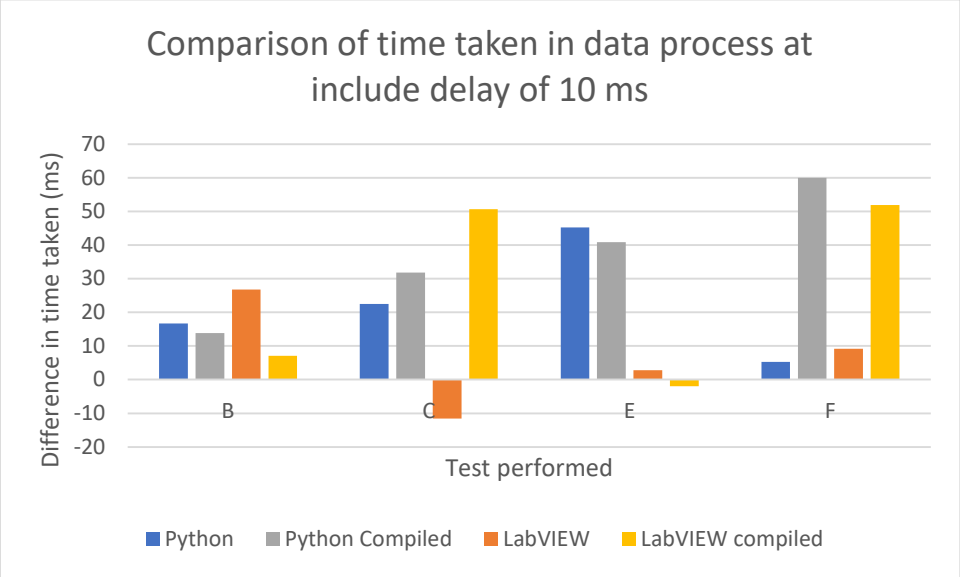

Figure 15. Comparison of processing times with and without delays in the code.

Although introduced delays have been of only 10 ms, averaged computation times have increased in some situations by as much as 60 ms, due to those delays influence also codes working in parallel.

Some situations get a faster processing after adding the delay, for example for LabVIEW in test C and LabVIEW compiled in test E. The times shown are averaged. Although introducing delays almost always increases processing times, CPU gets more relaxed, what makes the system work with a better performance.

As mentioned before, variability in time taken by any of the systems to perform the required tasks is observed, due to Windows cannot be exclusively dedicated to one task. That explains also time variations in averaged time taken, even the reduction of time taken of some of the codes. This can be explained too due the reduction of CPU load. The more resources available, the more the computer can change among processes.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 March 2022 doi:10.20944/preprints202203.0051.v1

4.2.2. CPU

In relation with CPU usage, it can be seen now an enormous reduction.

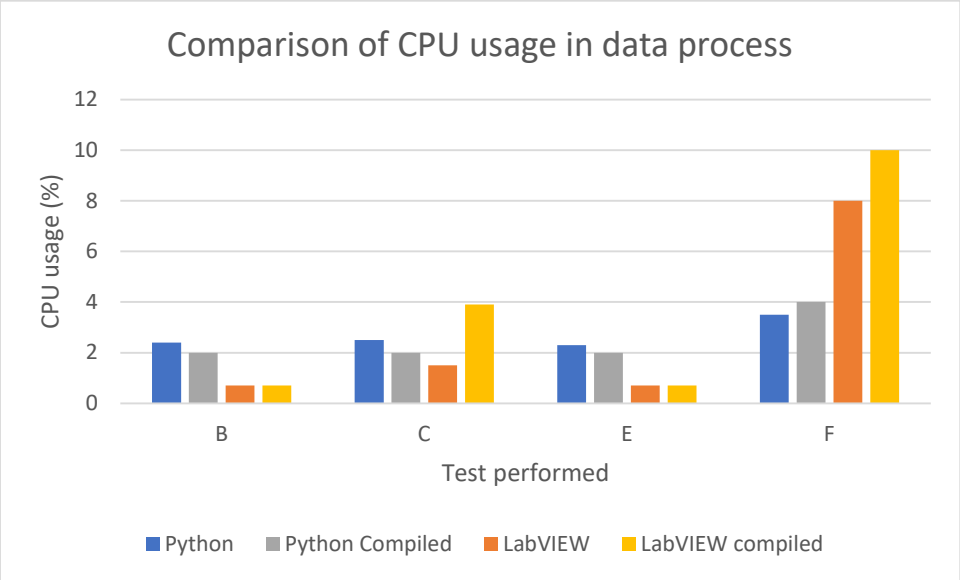

Figure 16. CPU usage in data processing after adding a delay in the code.

In all situations CPU usage is lower that 4% of its max usage, except LabVIEW in the 100k points Analysis, which stays at 10%. Even so, this value is much lower than before, and does not imply a huge system load.

It is interesting to see that Python requires more CPU usage for short vectors, and lower CPU load for large data vectors.

4.3. Results of the comparison

Given all the data analyzed, similar tasks can be performed with similar results with both Python and LabVIEW.

Python is a text-based programming language, designed as a based language with many modules which increase the functionalities, as numeric calculation, graphical interfaces, or web servers. Community as well as companies create and support many modules. It is free to use, except some modules.

LabVIEW is a graphical language, oriented to fast develop of graphical interfaces, and industrial solutions. It simplifies connection to acquisition and interaction hardware, and it has many modules with ready to use solutions. It is a licensed software.

LabVIEW allows to build solutions faster (special mention to graphical interfaces) than Python (in fact, systems for power quality monitoring have been developed there, as in [12]). However, Python enables the development of more complex solutions, as more options are available to support Python programming.

In relation with tested performance, with a proper programming, max CPU usage is similar: a bit higher for high vectors in LabVIEW and a bit higher for low vectors in Python, but anyways, low CPU loads. Code compilation could increase or reduce CPU usage, depending on situation.

In relation with RAM memory allocation, in general code compilation reduces memory needs, except for direct acquisition in Python. Python needs more memory, but no more than around 200MB in the worst situation, 5 times bigger that the one that takes LabVIEW. Still, Python memory demand is affordable for most of the systems, and should only be considered for memory-constrained systems.

In relation with time needed for all calculations, with low CPU loads, Python is up to 4 times faster than LabVIEW performing the same process (without delays, it's up to 10 times faster).

So, in conclusion, Python is faster and its usage of CPU is lower for large data vectors than LabVIEW. On the other hand, Python needs more RAM allocation and a bit more complex programming.

For the developed system, it's been used Python, since the RAM requirement doesn't suppose a problem and high speed is required, in order to perform a large number of tasks in very short periods of time.

5. Conclusions

The dataset provides information about the electrical grid, which it's variable on time and location despite its nominal values and tolerances are established by norm. Studying the characteristics of the grid makes possible to carry out the static and dynamic characterization of the power system, in order to assess the quality of the electricity supply (Power Quality, PQ).

One of the most critical parameters is frequency. It exerts a decisive influence on electrical machines. For example, on filters, which require very precise tuning, significant frequency deviations from its nominal value would result unacceptable. In this paper, frequency has been monitored following two methods: the traditional one, established by the norm UNE EN 61000-4-30:2015, in which the measures are taken every 10 seconds [9]; and a new proposed method, based on Allan variance, in which it's possible to take a frequency measure each one second, so it's possible to track it in more detail.

Various statistical parameters are also calculated, as the skewness, kurtosis, variance, THD, etc. They are calculated by some different analysis, calculating them every cycle of 50 Hz, every 10 cycles and every 2 seconds.

It's important to stress that since it's deposited data in the database every one, two or ten seconds, during each day, the monthly tables have a huge number of rows, so when they are exported, the resulting files are of a very large size. That's why only one month's data has been appended in the dataset.

Finally, data presented in this data descriptor can be used to evaluate the Power Quality, which depends largely on distributed generators and non-linear loads as well as the different operating conditions during the day. Due to their high variability, probabilistic studies are inevitable. The study of the PQ helps to identify the worst performing areas and facilitate the development of appropriate mitigating solutions.

Author Contributions: JMSF and MJEG programmed *Phyton* codes. JFM and JJGDR developed the whole paper organization and were responsible for experiments design and equipment operation. JFM generated, analyzed and interpreted graphs. AAP, OFO and JCPS contributed to interpretation. PRC did bibliography searching.

Funding: This research is funded by the Spanish Ministry of Science and Education through the project PID2019-108953RB-C21; has been co-financed by the European Union under the 2014-2020 ERDF Operational Program. Also, funding for frequency monitoring comes from the Andalusian-FEDER project FEDER-UCA18-108516 (Intelligent Techniques for visualization and data compression of PQ data in the smart grid).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The open data are fully available at https://doi.org/10.7910/DVN/EGI7X1.

Acknowledgments: The authors express their gratitude to the Spanish Ministry of Science and Education for funding the research project PID2019-108953RB-C21, entitled "Strategies for Aggregated Generation of Photo-Voltaic Plants-Energy and Meteorological Data" (SAGPV-EMOD). This work has been co-financed by the European Union under the 2014-2020 ERDF Operational Program and

by the Department of Economy, Knowledge, Business and University of the Regional Government of Andalusia.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] V. Mitrović and M. Mijalković, "Power quality data logger with internet access," *19th International Symposium on Power Electronics, Ee 2017*, vol. 2017-December, pp. 1–6, Dec. 2017, doi: 10.1109/PEE.2017.8171702.

[2] O. Florencias-Oliveros, J.-J. González-de-la-Rosa, A. Agüera-Pérez, J.-C. Palomares-Salas, "Power quality event dynamics characterization via 2D trajectories using deviations of higher-order statistics", Measurement, vol. 125, pp. 350-359, 2018.

[3] S. Abdelrahman, H. Liao, J. Yu and J. V. Milanovic, "Probabilistic assessment of the impact of distributed generation and non-linear load on harmonic propagation in power networks," 2014 Power Systems Computation Conference, 2014, pp. 1-7, doi: 10.1109/PSCC.2014.7038371.

[4] J. Rens, J. de Kock, W. van Wyk and J. van Zyl, "The effect of real network phase disturbances on the calculation of IEC 61000-4-30 parameters," 2014 16th International Conference on Harmonics and Quality of Power (ICHQP), 2014, pp. 303-306, doi: 10.1109/ICHQP.2014.6842886.

[5] M. de Apraiz, J. Barros, R. I. Diego, J. J. Gutiérrez, K. Redondo and I. Azcarate, "Detection and analysis of rapid voltage changes in power system networks," 2014 IEEE International Workshop on Applied Measurements for Power Systems Proceedings (AMPS), 2014, pp. 1-6, doi: 10.1109/AMPS.2014.6947712.

[6] N. Gourov, P. Tzvetkov, G. Milushev and V. Vassilev, "Remote monitoring of the electrical power quality," 11th International Conference on Electrical Power Quality and Utilisation, 2011, pp. 1-5, doi: 10.1109/EPQU.2011.6128842.

[7] "Open Power Quality" https://openpowerquality.org/docs/intro-power-quality.html; last visited january 12th, 2022.

[8] J.-J. González-de-la-Rosa, A. Moreno-Muñoz, I. Lloret, V. Pallarés, M. Liñán, "Characterisation of frequency instability and frequency offset using instruments with incomplete data sheets", Measurement, vol 39(7), pp. 664-673, 2006.

[9] IEC 61000-4-30:2015. *Electromagnetic Compatibility (EMC). Part 4-30: Testing and Measurement Techniques—Power Quality Measurement Methods*. IEC: Geneva, Switzerland, 2015.

[10] J.-M. Sierra-Fernández, O. Florencias-Oliveros; M.-J. Espinosa-Gavira, J.-J. González-de-la-Rosa, A. Agüera-Pérez, J.-C. Palomares-Salas, "Online System for Power Quality Operational Data Management in Frequency Monitoring Using Python and Grafana". *Energies 2021, 14, 8304.*

[11] D. Apetrei, I. Silvas, G. Chicco, R. Neurohr, M. Albu and P. Postolache, "Voltage and frequency time series analysis improvements in transforming data from quality survey into information," 2010 9th International Conference on Environment and Electrical Engineering, 2010, pp. 265-268, doi: 10.1109/EEEIC.2010.5489989.

[12] M. Simić, D. Denić, D. Živanović, D. Taskovski and V. Dimcev, "Generation of the power quality disturbances in LabVIEW software environment," 2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS), 2011, pp. 593-596, doi: 10.1109/TELSKS.2011.6143184.