

## Article

# Online System for Power Quality Operational Data Management in Frequency Monitoring using *Python* and *Grafana*

Jose-María Sierra-Fernández<sup>1</sup>, Olivia Florencias-Oliveros<sup>1</sup>, Manuel-Jesús Espinosa-Gavira<sup>1</sup>, Juan-José González-de-la Rosa<sup>1,\*</sup>, Agustín Agüera-Pérez<sup>1</sup>, and José-Carlos Palomares-Salas<sup>1</sup>.

<sup>1</sup> 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; josemaria.sierra@uca.es (J.M.S.F.); olivia.florencias@uca.es (O.F.O.); manuel.espinosa@uca.es (M.J.E.G.); juanjose.delarosa@uca.es (J.J.G.D.R.); agustin.aguera@uca.es (A.A.P.); josecarlos.palomares@uca.es (J.C.P.S.)

\* Correspondence: juanjose.delarosa@uca.es

**Abstract:** This article proposes a measurement solution designed to monitor instantaneous frequency in power systems. It uses a data acquisition module and a GPS receiver for time stamping. A program in Python takes care of receiving the data, calculating the frequency, and finally transferring the measurement results to a database. The frequency is calculated with two different methods, which are compared in the article. The stored data is visualized using the *Grafana* platform, thus demonstrating its potential for comparing scientific data. The system as a whole constitutes an efficient low cost solution as a data acquisition system.

**Keywords:** big data; data acquisition; data visualization; data exchange; dashboard ; frequency stability; *Grafana* lab; Power Quality; GPS reference; frequency measurement.

## 1. Introduction

Static and dynamic characterisation of power systems' is essential in order to assess the quality of the electrical supply (Power Quality, PQ). One of the most critical parameters is the frequency. It exerts a decisive influence on electrical machines as they require precise tuning, especially those operating under synchronization, being deviations from the nominal value inadmissible.

The network parameters oscillate within a range. So, it's important to know if these parameters are taking tolerable values or they are outside the limits so they could turn dangerous. The nominal European frequency (50 Hz) is far unstable, since the delivered signal is not perfectly sinusoidal. The admissible limits are  $50 \pm 0.1$  Hz for systems with synchronous connection and  $50 \pm 2$  Hz for isolated networks [1].

Frequency shifts depend on a number of external factors. Common causes are lightning strike, connection and disconnection of huge loads, non-linear loads, and groundings or short circuits in transmission lines [2]. Some well-known solutions have been developed to track the frequency, based on multiple procedures, for instance Zero Crossing (ZC) methods, Quadratic Forms (QF), Demodulation, Discrete (Fast) Fourier Transform (DFFT) with Phase Compensation (PC), etc. [3]. However, in the modern smart(er) grid there are myriads of distributed resources and consequently, being independent but interrelated, points of the network under test have to be synchronized. As a response, new technologies were developed to fulfill requirements, being Phasor Measurement Units (PMUs) the most relevant.

PMUs are used to estimate the magnitude and phase of electrical magnitudes (e.g., voltage or current). Synchronization is provided via GPS or IEEE 1588 Precision Time Protocol (PTP), designed to allow real-time synchronized of several remote points in the grid. PMUs capture samples of the power signal and rebuild the so-called phasor quantity (magnitude and phase), which describes mathematically the waveform under test. The result is a time-synchronized measurement known as synchrophasor.

PMUs are capable of detecting abnormal waveform shapes. For instance, if both supply and demand do not match perfectly, the associated frequency imbalances cause tightness within the grid, which is a potential cause for power outages.

PMUs are also designed to measure the frequency in the power grid. Indeed a typical commercial PMU can report measurements with very high temporal resolution, up to 120 measurements per second. This helps analyzing dynamic events in the grid, fact which is not possible with traditional SCADAs that generate one measurement every 2 or 4 seconds. Therefore, PMUs with enhanced monitoring and control capabilities are considered one of the most important measuring devices in the current and future of power systems.

Besides the high prices (the average overall cost per PMU - cost for procurement, installation and commissioning - ranges from \$40000 to \$180000), in most cases PMUs measure the voltage magnitude and the phase angle, assuming that the frequency is a constant. This hypothesis is precisely disregarded in PMU's indirect measurements, which consist of a mathematical problem, where the measurements are fit to a sine curve. Therefore, when the sinusoidal shape is not ideal, the PMU is unable to fit it exactly. The less sinusoidal the power signal is, the worse the phasor representation will be. This situation is very common in the modern network, where the random distributed and intermittent generation and non-linear loads make it necessary to adopt new measurement strategies in order to complement the former based on PMUs.

With all the above, this work proposes a measurement solution complementary to the PMUs-based in a double sense. On the one hand, it is intended to provide a more accurate measure of the frequency using Allan's variance, contributing in this way to the smart(er) grid characterisation enhancement. On the other hand, the proposed solution is capable of managing the large volume of data (Big Data) that floods the electricity grid. The instrument is based on a GPS receiver-DAQ binomial; the latter driven by a program developed in *Python*. The former provides time stamp aided by the 1 Pulse Per Second (PPS) signal, and with an accuracy of 1 ppm. In this way, it adapts to what is established in the Power Quality measurement regulations [1], obtaining frequency values every 10 s with a temporal resolution better than 20 ms [5-6].

Additionally, the system proposes to visualize the massive volume of data using *Grafana*. Thus, the work shows its versatility and flexibility as well, thus being integrated as a Data Acquisition Framework (DAQ-F), and presenting a low-cost alternative to the functionalities of SCADAs [9]. Commercially named *PyFreqDAQ*, this DAQ-F system incorporates an easy-to-use graphical user interface, allowing a simple control and automation of a huge variety of experimental setups for frequency measurement.

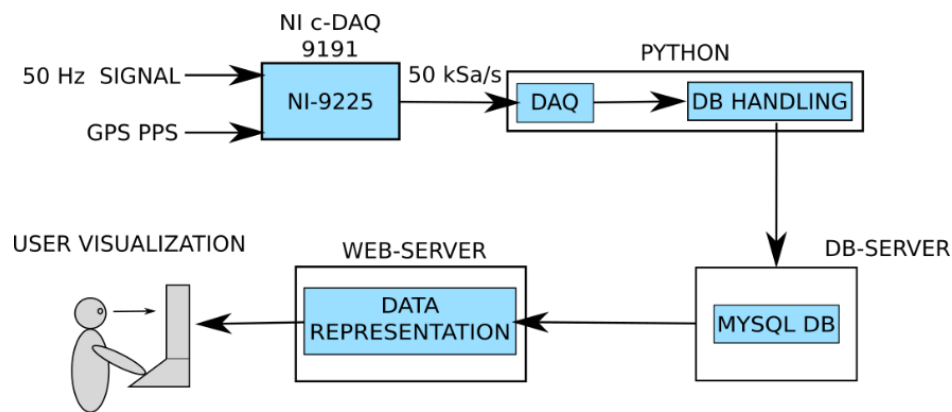
This is the motivation of the work and the general context of the research. The rest of the paper is structured as follows. Section 2 provides details regarding the system description and arrangement. Then, in Section 3 *PyFreqDAQ* operation is detailed along with the frequency calculation. Section 4 displays and explains graphical results of data analysis from *Grafana*. Finally, conclusions are drawn in Section 5.

## 2. System description and arrangement

Hereinafter we describe the distributed measurement system. The hardware essentially consists of a chassis and card from the manufacturer NI<sup>TM</sup> that receives the inputs from the conditioned power grid and the GPS-traceable reference timestamp pulse (Figure 1.). Data is transferred from the DAQ card to the PC via Ethernet. Later, data is processed and results are transferred to the database, using software programmed in *Python*. Using these data, it is possible to draw frequency evolution in the required time period in order to perform the network surveillance [7].

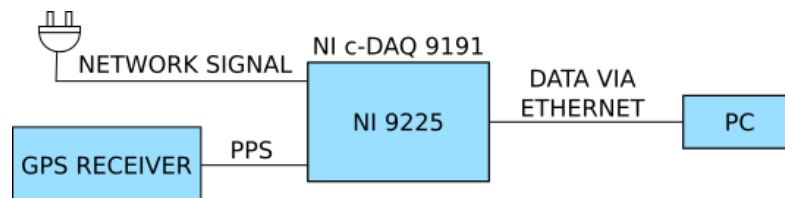
The system allows getting data from multiple sources and places and putting it all in common at the database and the online visualizer. With that, it can be observed and analyzed a large quantity of data of multiple signals collected from different sites, what can be very helpful for the network maintenance.

For a first approach to understand the system operation, it's shown a simple global representation of the system, as well as the interaction between the different elements on Figure 1.



**Figure 1.** System block diagram. The c-DAQ acquires the signals and transfers the information to a software running constantly. It saves information in the DB and then, data can be represented on an online viewer.

*PyFreqDAQ* measurement chain is composed of a GPS receiver, for the synchronization, a plug connected to the network responsible for acquiring the signal, and a DAQ card that receives both signals and sends them to the PC. It can be seen on Figure 2.



**Figure 2.** Scheme of the Measurement chain. Network signal and GPS-PPS signal are acquired with the DAQ card (NI 9225) inside de chassis (c-DAQ-9191). It processes them and sends data to the PC via Ethernet.

Figure 3 shows the connections of the GPS-PPS and the voltage signals to the card NI 9225 (simultaneous, channel-to-channel isolated). This 3-channel module allows up to 300 V<sub>rms</sub>, with a sampling frequency of 50 kS/s and 24 bits. Data is used to calculate frequency later. The card NI 9225 is allocated inside the chassis NI 9191. The chassis controls timing, synchronization and data transfer between I/O modules and an external server.

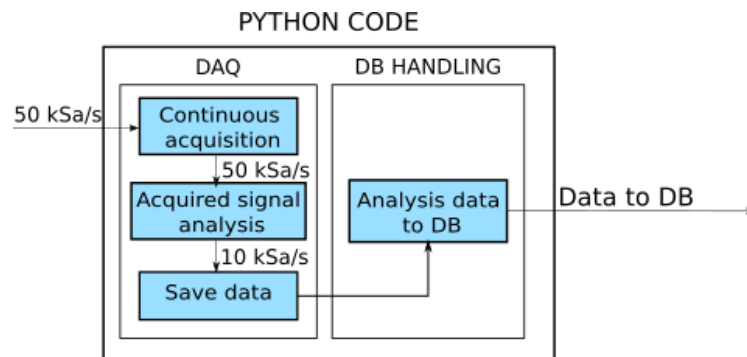


**Figure 3.** The NI 9225 card inside the chassis cDAQ-9191 (1-Slot, Ethernet and 802.11 Wi-Fi compact DAQ chassis) with wiring connections: Yellow wires correspond to the GPS-PPS signal while the black brings the network voltage signal directly by connecting to it with a plug.

The equipment Symmetricom™ TrueTime XL-AK GPS synchronized receiver assures ultra-precise time and frequency delivery. This configurable piece of instrument provides timing outputs within 40 ns of UTC/USNO and frequency outputs accurate to less than  $1 \cdot 10^{-12}$ . It includes 1 pps output that gives a reference for the frequency calculation using the *Allan* variance.

For data processing, calculations as well as sending to the database, two *Python* software projects are constantly running in parallel. The first one is responsible for receiving data transmitted from the NI 9225, calculating frequency

and sending the results to the second one. First project is referred to as “DAQ”; being the second project responsible for receiving data from the DAQ and organizing within the database. This part is referred to as “DB handling”. Figure 4 shows in a simplified way the workflow.



**Figure 4.** Python code operation with the two projects. Data is registered at a rate of 50 kSa/s and received by the first project, DAQ, responsible for acquiring and transferring to the second project at a rate of 10 kS/s. The second project, DB Handling, stores them and manages database.

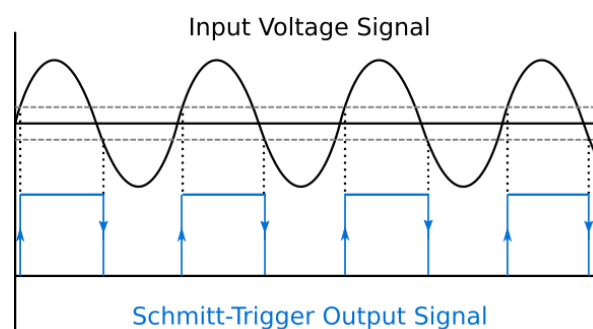
Finally, data is represented on *Grafana*, a well-known multiplatform open-source analytics and interactive visualization web application. It provides charts, graphs, and alerts for the web when connected to supported data sources. As a visualization tool, *Grafana* is a very popular component in monitoring stacks, often used in combination with time series databases and monitoring platforms. In this work it allows the user data surveillance.

### 3. PyFreqDAQ operation. Frequency calculation

The card NI 9225 receives data from two inputs (sampling frequency is 50 kS/s): the voltage line signal by connecting directly to a wall plug, and the GPS PPS signal that allows an absolute time reference. Raw data is then transferred via Ethernet to the software running on the PC.

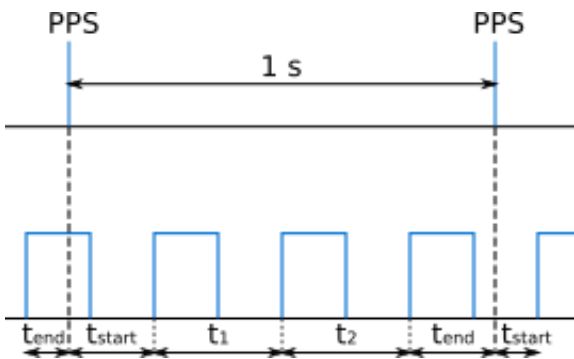
The first of the two projects, referred to as “DAQ” is constantly reading signal data, acquired from the NI 9225. Calculations are done by two ways. The PPS is used for the calculations contemplating the theoretical frequency of 50 Hz, so each theoretical cycle lasts 20 ms.

The voltage line is driven to a Schmitt trigger in order to be conditioned (it returns a square waveform with the same frequency as the input signal). When the input voltage is higher than a chosen upper threshold, the output is high, while when the input is below a lower level, the output is low. If the input voltage is between both thresholds, the output remains in the same previous state (Figure 5). Then, calculations can be done by fragmenting this square waveform in its cycles.



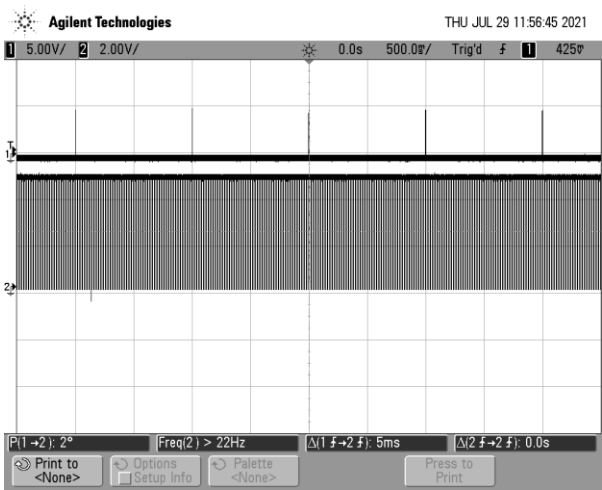
**Figure 5.** Schmitt-trigger application to the input voltage signal. The input voltage signal is drawn as a black sinus and the output signal is the blue square signal. Scales and unit are omitted for simplicity.

Power signal period is measured between two positives Schmitt-trigger signal edges. Between two GPS-PPS pulses, there can be some complete signal periods and also parts of other periods interrupted by the PPS. The sum of all these parts must result 1 s, as seen on the Figure 6.

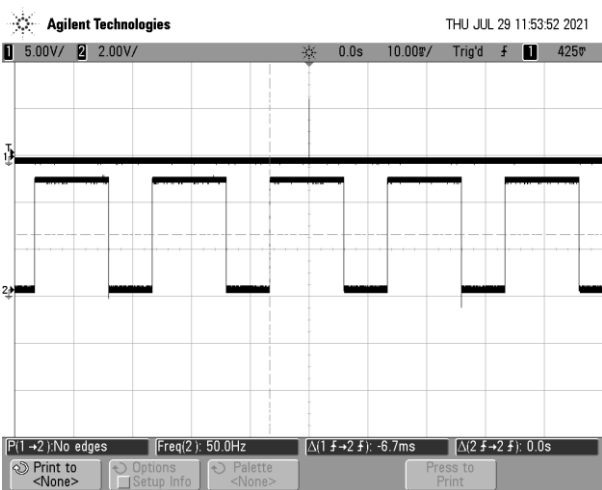


**Figure 6.** Schmitt-trigger output signal fragmented by the PPS pulses. This permits to measure the signal phase and the *Allan Variance*.

For those cycles that contain a GPS PPS signal, it can be measured the signal phase, knowing that each cycle corresponds to 360°. In the next two Figure 7 and Figure 8 can be seen the real PPS and Schmitt-trigger signals, at different sampling rates, from the lab oscilloscope.



**Figure 7.** Schmitt-trigger output signal and PPS pulses. Five PPS pulses can be seen at the top.



**Figure 8.** Schmitt-trigger application to the input voltage signal. At this sample only 1 PPS is seen, but it is clearly observed the square signal from the Smith-trigger.

Using the signal phase, *Allan* variance is calculated. It has exhibited efficient performance in frequency stability characterization of all precision sources [10]. Furthermore, through *Allan* variance, it's calculated frequency, using

frequency deviation from reference frequency. It's used Fractional Frequency Deviation as base of *Allan* variance calculation:

$$y(t) = \frac{f(t) - f_0}{f_0} \quad (1)$$

It depends on the actual frequency and the base frequency,  $f_0$ . In this case,  $f_0$  corresponds to the ideal network frequency of 50 Hz. It's calculated the deviation from the actual frequency respect to  $f_0$ , in a relative or fractional form.

Time deviation is also calculated, starting from the base frequency. At an instant  $t$ , the signal advances a phase of  $\phi(t)$ . If the frequency was equal to the base one:

$$\Phi(t) = 2\pi f_0 t \quad (2)$$

So, if  $\phi(t)$  is divided by  $2\pi f_0$ , and the frequency is ideal, it would result " $t$ ". Then, if we subtract " $t$ " to that fraction, it can be obtained time deviation, for an instant  $t$  but also instant  $t+1$ :

$$x(t) = \frac{\phi(t)}{2\pi f_0} - t \quad (3)$$

As each increment of  $+1$  corresponds to an increment of  $\tau$  in time, it can be determined as fractional time increment, which would be equal to the frequency increment:

$$\begin{aligned} \bar{y}_k &= \frac{1}{\tau} (x_{k+1} - x_k) = \\ &= \frac{1}{\tau} \left( \left( \frac{\phi_{k+1}}{2\pi f_0} - t(k+1) \right) - \left( \frac{\phi_k}{2\pi f_0} - t(k) \right) \right) = \\ &= \frac{1}{\tau} \left( \frac{\phi_{k+1} - \phi_k}{2\pi f_0} - t(k+1) + t(k) \right) = \frac{1}{\tau} \left( \frac{\Delta\phi}{2\pi f_0} - \tau \right) \end{aligned} \quad (4)$$

The phase increment, is computed using the start and ending phases, before and after the PPS signal:

$$\phi_{\text{end}} = 2\pi \cdot \frac{t_{\text{end}}}{t_{\text{start}} + t_{\text{end}}} \quad (5)$$

$$\phi_{\text{start}} = 2\pi \cdot \frac{t_{\text{start}}}{t_{\text{start}} + t_{\text{end}}} \quad (6)$$

And, with them, the phase increment of each second is:

$$\Delta\phi = \phi_{\text{start}} + 2\pi N + \phi_{\text{end}} \quad (7)$$

All in all, the instantaneous frequency at instant  $t$ , for the  $k$ th-measure obeys to the following equation:

$$\frac{\bar{f}_k}{f_0} - 1 = \bar{y}_k; \quad \frac{\bar{f}_k}{f_0} = \bar{y}_k + 1; \quad \bar{f}_k = f_0 (\bar{y}_k + 1) \quad (8)$$

Expression of  $\bar{y}_k$  from Eq. 4 is replaced in (8):

$$\bar{f}_k = f_0 \left( \frac{1}{\tau} \left( \frac{\Delta\phi}{2\pi f_0} - \tau \right) + 1 \right) \quad (9)$$

If  $\tau=1$ , we have:

$$\bar{f}_k = f_0 \left( \frac{1}{1} \left( \frac{\Delta\phi}{2\pi f_0} - 1 \right) + 1 \right) = f_0 \frac{\Delta\phi}{2\pi f_0} = \frac{\Delta\phi}{2\pi} \quad (10)$$

$\Delta\phi$  is then replaced using Eq. 7:

$$\bar{f}_k = \frac{\phi_{\text{start}_{\text{ant}}} + 2\pi N + \phi_{\text{end}}}{2\pi} \quad (10)$$

Separating into simple fractions and replacing  $\phi_{\text{start}}$  and  $\phi_{\text{end}}$  by their expressions in Eq. 5 and Eq. 6, it yields:

$$\bar{f}_k = \frac{t_{\text{start}_{\text{ant}}}}{t_{\text{start}_{\text{ant}}} + t_{\text{end}_{\text{ant}}}} + N + \frac{t_{\text{end}}}{t_{\text{start}} + t_{\text{end}}} \quad (10)$$

As it is observed in Figure 6,  $t_{\text{start}}$  is the time elapsed from the initial PPS to the end of the cycle which is broken by this pulse, and  $t_{\text{end}}$  is the time elapsed from the beginning of the last cycle (considered between the two PPS) and the final PPS.

Frequency is calculated by two methods. Each second, it's calculated following the method just explained, using *Allan* variance and also according to the regulation (UNE 50160), calculating frequency each 10-second period.

Once calculations are done, data results are transferred to the second *Python* project (DB handling). This project is responsible for saving and organizing MySQL database, with the aim of a rapid. *Grafana* gets data directly from the data base and represents it with timescales and other resources so that the user can visualize the evolution of the variables.

### 3. Results

The data base is organized in different tables. If it was necessary, the information could be stored in the same table, but it would lead to millions of rows, what would stress the search engine and would slow down the operations, with the consequent computational cost. To make an idea, putting the focus on the *Allan* frequency, which is the most loading case of the two measurement strategies compared in this work, one result is sent to the database every second, resulting in 86,400 rows generated per day; 604,800 per week; in one month, 18,748,800 and in one year 31,536,000. To solve the above mentioned big data issues, this is programmed directly from the *DB handling*, for both *Allan* and regulated frequencies. As examples, Table 1 and Table 2 show the measurements corresponding to the month of May. Each table preserves its original design in order to show a *Grafana* perspective.

id	absolute_signal_id	time	frequency
1	19997	2021-05-03 08:50:47	50.0099
2	19998	2021-05-03 08:50:48	49.9878
3	19999	2021-05-03 08:50:49	50.0263
4	20000	2021-05-03 08:50:50	50.0701
5	20001	2021-05-03 08:50:51	49.9916
6	20002	2021-05-03 08:50:52	50.0064
7	20003	2021-05-03 08:50:53	49.9766
8	20004	2021-05-03 08:50:54	49.9205
9	20005	2021-05-03 08:50:55	50.0115
10	20006	2021-05-03 08:50:56	50.0458
11	20007	2021-05-03 08:50:57	50.0317
12	20008	2021-05-03 08:50:58	49.972
13	20009	2021-05-03 08:50:59	50.013
14	20010	2021-05-03 08:51:00	49.9837

**Table 1.** *Allan* frequency table for one month, extracted from *Grafana*.

It can be verified that it has been registered one frequency data each second just looking at the "time" column.



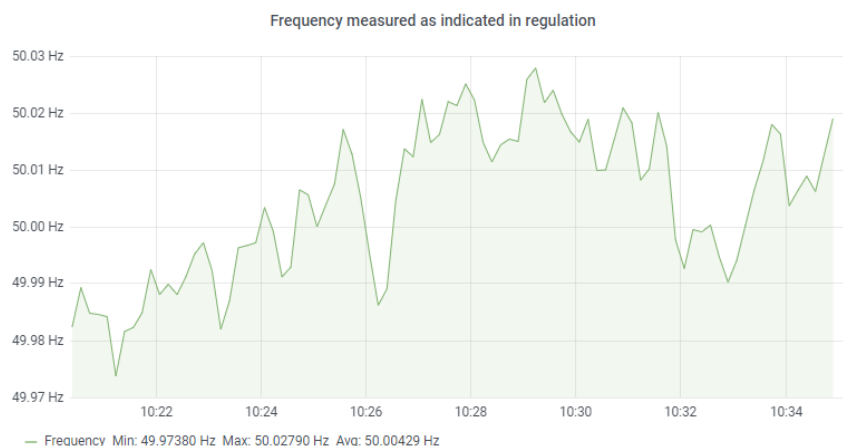
id	absolute_signal_id	time	frequency
28	20287	2021-05-03 08:55:36	49.9691
29	20297	2021-05-03 08:55:46	50.0106
30	20307	2021-05-03 08:55:56	50.0015
31	20317	2021-05-03 08:56:06	49.9994
32	20327	2021-05-03 08:56:16	49.994
33	20343	2021-05-03 08:56:36	50.0048
34	20354	2021-05-03 08:56:56	50.0047
35	20437	2021-05-03 09:43:16	49.9988
36	20447	2021-05-03 09:43:26	49.9955
37	20457	2021-05-03 09:43:36	50.0053
38	20467	2021-05-03 09:43:46	50.003
39	20477	2021-05-03 09:43:56	50.0082
40	20487	2021-05-03 09:44:06	49.9825
41	20497	2021-05-03 09:44:16	50.0116

**Table 2.** Frequency measurements (according to norm 6100-4-30) table extracted from *Grafana*.

It also can be verified that frequency is registered in 10-second periods, as it is stated. In both tables there are four columns, labeled: "id", "absolute\_signal\_id", "time" and "frequency". "Id": primary key of the correspondent table, auto-increment, serves to enumerate the rows of the monthly table. "Absolute\_signal\_id": foreign key from the global\_signal\_list table. "Time": date in which the measurement was taken. Format: YYYY-MM-DD HH:MM:SS. "Frequency": Calculated frequency value in Hz. Following this criterion it is faster to make the queries, that are aligned with *Grafana*. Indeed, *Grafana* uses data stored in tables for making the desired graphics. The graphics are called as "Dashboards" and the users can create as many as desired using available data.

The platform is capable of selecting data from the tables following a searching criterion. For example, in the present work, for the sake of representing the evolution over time of frequency values, it can be selected the table of the month the specific time interval to be represented. In this case, the platform is extracting only the columns of time and frequency from the tables.

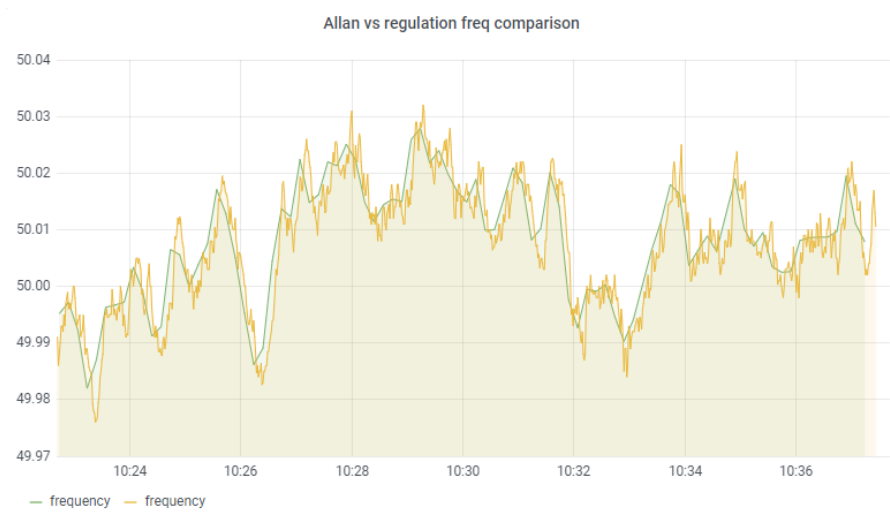
The standard UNE-EN 50160:2011 establishes the limits that the frequency cannot overtake and mentions that the frequency has to be measured in periods of 10 seconds [1]. Figure 9 shown the evolution of frequency during 15 minutes, measured as indicated by the norm.



**Figure 9.** Evolution of the frequency following the norm, with values each 10 seconds, during 15 minutes.



To study the difference between the two methods of calculating frequency, the following dashboard on *Grafana* compares the evolution of both frequencies. It's been chosen a time interval of 15 minutes, as it can be observed in Figure 10.

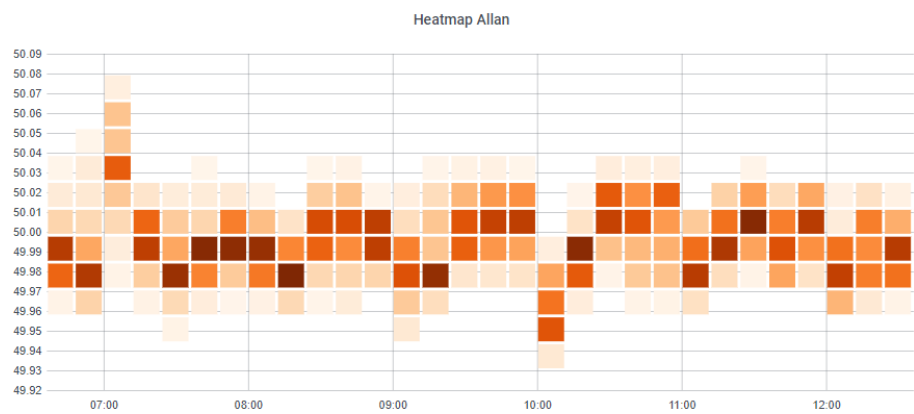


**Figure 10.** Evolution of the frequency following the regulation (green line) and following the algorithm using Allan Variance (orange line).

Looking at Figure 10, it is notorious the difference between the results the two methods. First, the evolution of frequency measured by the regulation method looks softer than the other one. The reason is clear, with the method that establishes the regulation, frequency calculation it's done for periods of 10 seconds, so it's available one measure each 10 seconds. With *Allan* algorithm, frequency is calculated each second, so the number of measures available is 10 times greater so the tendency gets a lot of peaks compared to the other one.

In addition to this, *Allan* frequency is totally synced thanks to the GPS-PPS signal, so it can be stated that this method is more accurate that the one that regulation establishes.

Observing the measures over time, it has been proven that frequency has not exceeded the established limits during at least 10 seconds, in which case, frequency would be out of what current regulation dictates. It's also available at *Grafana* a heat-map showing the distribution of *Allan* frequency through time:



**Figure 11.** Heat-map of *Allan* frequency

The horizontal axis corresponds to timescale. Figure 11 is divided in one-hour intervals. Each interval contains five columns of squares, so each column corresponds to an interval of twelve minutes. The vertical axis corresponds to the frequency scale. As the *Allan* frequency is measured each second and each square in column corresponds to twelve minutes, for each column we have 720 measures. For indicate the distribution of these measurements, in each column there are various squares of different colors. The darker the square, the more measurements have been recorded within the considered frequency interval. It can be observed that data is continuously fluctuating around 50 Hz. Assuming the

rest of measures as outliers, it can be seen that the fluctuations occur between frequencies of 49.95 Hz and 50.03 Hz, approximately.

## 5. Conclusions

A system for automatically measuring and observing the evolution of network frequency has been developed. This system is based on open-source software and permits to visualize temporal evolution of network frequency and allows evaluating if these values are between the allowed limits.

Frequency has been calculated by two ways. First, following what establishes the regulation IEC 61000-4-30 and secondly, with an algorithm based on the *Allan* variance.

They are multiple the benefits proved of Allan method in front of regulation method: better accuracy and less error, faster calculations, more measures available and it's used the whole signal.

The joint solution of Python and *Grafana* for receiving and processing data, doing calculations and visualizing data turned out to be very interesting and effective. The operation is smooth and meets the expectation and necessary requirements.

Using Python for programming with the DAQ card and making all the DB handling has shown more flexibility in front of other options like NI LABview. Python is a much more powerful calculation language than Labview so allows making calculations and required operations with a simpler programming and less computational load.

*Grafana* makes too much simpler data visualization. To visualize data in a web environment using another platform would be complex and it would take a long time to make it work. With this platform, choosing a few parameters and connecting it to the database, data can be visualized. It's a simple work that can do low-skilled staff and takes very little time.

**Author Contributions:** JMSF and MJEG programmed *Phyton* codes. OFO and JJGDR developed the whole paper organization and and were responsible for experiments design. AAP and JCPS analyzed and interpreted graphs in the frame of comparison the two methods of frequency measurement.

**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.

**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. IEC 61000-4-30:2015+AMD1:21 CSV. Electromagnetic compatibility (EMC) . Part 4-30: Testing and measurement techniques – Power quality measurement methods.
2. A. Moreno-Muñoz, J. A. Sánchez, J. J. G. de La Rosa, and J. J. Luna, "Application of smart sensors to the measurement of power quality," in Conference Record - IEEE Instrumentation and Measurement Technology Conference, 2008, pp. 218–222.
3. D. W. P. Thomas and M. S. Woolfson, "Evaluation of frequency tracking methods," IEEE Transactions on Power Delivery, vol. 16, no. 3, pp. 367–371, July 2001.
4. US Department of Energy, "Factors Affecting PMU Installation Costs," 2014.
5. G. L. Kusic, W. E. McGahey, and M. Lehtonen, "Measurement of power system phase differences by means of GPS timing," in 10th International Conference - 2016 Electric Power Quality and Supply Reliability, PQ 2016, Proceedings, Oct. 2016, pp. 297–300.
6. M. S. Golsorkhi, M. Savaghebi, D. D. C. Lu, J. M. Guerrero, and J. C. Vasquez, "A GPS-based control framework for accurate current sharing and power quality improvement in microgrids," IEEE Transactions on Power Electronics, vol. 32, no. 7, pp. 5675–5687, July 2017.

- 
7. J. Luping, Y. Huiping, and L. Linshu, "A new system for frequency monitoring and fault analysis," 2007. doi: 10.1109/ICPST.2006.321708.
  8. S. Penkov, A. Taneva, and R. Kazala, "Supervisory and Data Acquisition project with LoRa," Oct. 2020. doi: 10.1109/ICAI50593.2020.9311338.
  9. S. J. Weber, "PyMoDAQ: An open-source Python-based software for modular data acquisition," Review of Scientific Instruments, vol. 92, no. 4, p. 045104, Apr. 2021, doi: 10.1063/5.0032116.
  10. F. L. Walls and D. W. Allan, "Measurements of Frequency Stability," Proceedings of the IEEE, vol. 74, no. 1, pp. 162–168, 1986, doi: 10.1109/PROC.1986.13429.