Voltage Phase Angle Measurement System based on a Raspberry Pi Single Board Computer

Marius Otto Kleinbach
Technische Hochschule Köln
Köln, Germany 2023
marius_otto.kleinbach@th-koeln.de

***Abstract*—This paper presents the test results of a Measurement System that compares two 50 Hz voltage signals to determine the phase shift between different points in a power grid. A hardware-near C program on a single board computer (Raspberry Pi) takes time measurements via an hardware interrupt routine in relation to a fixed 1 Hz signal that is received from a GPS satellite. The measurements are uploaded to a cloud every minute. The results can be evaluated from an external computer by a python script. This project proves the feasibility of the system and analyses the accuracy of the time measurements under laboratory conditions.**

***Index Terms*—Hardware interrupt routine, phase measurement unit, Wide area monitoring system**

I. Motivation

The PROGRESSUS project aims to reduce the peak load in distribution grids by 30% by using smart charging infrastructure. To determine how much power each charging station is allowed to retrieve the topology of the distribution grid must be known. Otherwise it would be impossible to know how much electric current flows through which line and the effects of an additional load to the voltage level of each node cannot be calculated.

Since there is often no documentation on the exact topology of a given distribution grid an algorithm was developed that estimates the grid topology based on a matrix of impedances between different nodes in the network. The featured measurement system can act as a Wide Area Measurement System to determine the phase angle between multiple power nodes and therefore the impedance matrix needed for the topology estimation [1].

In the following chapters the functionality of the system is explained as well as the setup for test measurement. These are furthermore analysed to determine the accuracy of the time measurements and show the limitations of the method.

II. Raspberry Pi as a Wide Area Measurement System

Wide area measurement systems (WAMS) bring information to the control center in modern power systems to improve observability for achieving stability and security [2]. A single board computer like the Raspberry Pi can be used to monitor important parameters of a power grid. It can be used to measure the net frequency and amplitude under the condition that the signal is transformed to a voltage level between 0 and 3.3 V that doesn't damage the signal inputs of the device.

To measure the voltage phase angle between two nodes a timer has has to be started at the exact same time on two devices and stopped once the voltage signal crosses zero. The time data of each device is sent to a Dropbox every minute. These time stamps can then be compared to determine the phase angle between the two nodes. The resulting impedance between the measurement points can then be calculated as shown in chapter III-B.

A. Hardware

The entire computing is done on a Raspberry Pi single board computer. The system was tested with the 3B+ model which features a 1.4 GHz processor and 1 GB of RAM and a 4B model with a 1.5 GHz processor and 8 GB of RAM. No performance differences could be observed between the two systems and varied models can be used on different nodes of the same WAMS.

One Raspberry system needs two peripheral circuits to work as part of the measurement system. To transform the 230 V AC signal to a square wave signal below 3,3 V a comparator circuit is used that was developed within the PROGRESSUS project [3].

To ensure that a time measurement is started at the same time on two different devices a reference time signal is needed. To receive an identical time stamp a GPS module (model GNSS 5 click) is connected to the Raspberry Pi. This peripheral circuit automatically connects to a GPS satellite and receives a periodical signal that provides a voltage pulse every second. With identical GPS receivers the fluctuation of this pulse per second (PPS) signal is $\pm 10\,ns$ which is negligible for this application [4]. Each of these two circuits is connected to one general purpose input/output (GPIO) of the Raspberry Pi and all three boards have to have a common ground. Additionally the serial data receive port (RxD) of the Raspberry port can be used to request information about date, time and location from the data transmit port (TxD) of the GPS module. The setup therefore only occupies two to three

GPIOs of the Raspberry Pi which leaves plenty of room for additional peripheral devices.

B. Measurement Software

The code to compare the time signals was written in C. In comparison to other user friendly coding languages like python, C is much more hardware oriented and therefore the code is executed multitudes faster [5].

Every time delay between the peripheral signals and the time stamp taken by the software leads to a systematic measurement error and should therefore be kept as short as possible.

When a flank is detected at the PPS input pin it triggers an hardware interrupt that takes the current system time as a reference time. This enables a second interrupt for the zero crossing signal of the comparator circuit. On the next four zero crossings of the voltage signal a time stamp is taken to calculate the time difference between the reference signal from the GPS module to the zero crossings. This routine is repeated every 30 seconds but could also be conducted every second if necessary for the application.

The results are saved in a comma separated values (CSV) file every minute to avoid data loss during long measurements. The implementation of one method to compare the time stamps of two independent nodes is explained in chapter II-C.

C. Evaluation Software

To calculate the phase shift the CSV files of two nodes have to be compared by a single device. This can be done by declaring one of the single board computers as the main node and sending all time measurements to this central computing unit. The CSV files can then be sent to this device via a local server if the devices are located in the same network or via power-line communication if the devices are still in the range of several hundred meters and not dozens of kilometres apart [6].

Since there is no hierarchy in the nodes and a decentralized solution is being sought a different method was chosen for this project. The CSV data is sent to a dedicated Dropbox cloud. This makes it possible for every device with an internet connection and the right authorization to evaluate the gathered time data. This solution eliminates the need for a master slave hierarchy and the voltage phase angle can be accessed by every device that needs it to determine the network state.

The upload to the cloud is handled by a python script that is called by the main C program via a terminal command. The script is run every minute after the second measurement routine described in chapter II-B. Since there is no time critical code executed in the following 29 seconds there is no risk of the script interfering with the accuracy of the time measurements as described in chapter V.

The actual calculation of the voltage phase angle is conducted by a separate python script that pulls the CSV files from the cloud and calculates the time difference between the corresponding time stamps. The voltage phase angle based on the time difference of two zero crossings in a 50 Hz system is calculated using the formula:

$$\varphi = \Delta t \frac{360°}{20.000 \mu s} \tag{1}$$

Whereby φ is the voltage phase angle in degree and Δt is the time difference in μs.

III. LABORATORY MEASUREMENTS

To validate the functionality of the WAMS using Raspberry Pi single board computers and to verify the demanded precision of the system multiple measurements have been conducted in the smart grid laboratory of the TH Köln.

Evaluations of the voltage phase angle to derive a topology estimation of the local power grid within the PROGRESSUS project have shown that the angle φ has to be measured with an error margin of less than 0.3° to make an accurate estimation of the grid topology. According to formula 1 the measurement therefore has to be precise enough to measure time differences Δt with an accuracy of $\pm 16.6 \, \mu s$ to generate a reliable data foundation for topology estimations.

The different test layouts can be divided into two categories which will be examined in the following subsections III-A and III-B.

A. Precision Assessment of a single Measurement Device

The first test scenario is used to examine the maximum precision of one measurement device with as little interference and additional time delay as possible. The signal source in this layout is therefore not the 230 V 50 Hertz sinus signal of the power grid but a clean 3.3 V 50 Hz square wave signal coming from a R&S HM8150 signal generator. This eliminates the need for the comparator circuit which would also add minor fluctuations in the low µs range [3]. The signal quality was additionally monitored with a Agilent DSO1024A oscilloscope.

As the measurement device a Raspberry Pi model 4B was used. The peripheral GPS module could also be omitted in this scenario, since there is no need to compare the time stamps with a second device. The start of a measurement is triggered by a rising flank of the signal and the four subsequent falling edges each trigger a time measurement in relation to the first rising edge.

The file format stays the same as described in chapter II-B. The difference between the expected and the measured value is evaluated in chapter IV to determine the highest precision that the measurement method is capable of.

B. Angle Measurements with a defined Impedance

The second series of tests was conducted with two phase measurement units (PMU), each consisting of one Raspberry Pi, one comparator circuit and a GPS module. The devices are connected as shown in figure 1.

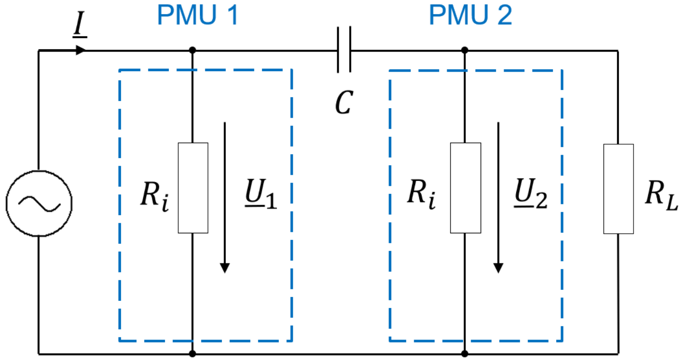

Figure 1. Test circuit for phase voltage angle measurements

The capacitor and the load resistor were chosen with $C = 82\,\mu F$ and $R_L = 750\,\Omega$ so that the phase difference in the designed circuit with a 50 Hz 230 V source result in $\varphi = 2,97°$. The internal resistance of the PMU data input with $Ri = 356,6\,k\Omega$ has been taken into account as well. This test setup represents an exemplary application in the field where the voltage phase angle between two nodes can be monitored to determine the resulting impedance between the two measuring points. To record a series of measurements the C code described in chapter II-B is executed on both devices simultaneously. Via a wireless internet connection the measurement results get uploaded to a Dropbox cloud minutely from where they can be evaluated by a third device even as the test is still running.

IV. Measurement Results

To determine whether the measurement method is precise enough for the topology estimation the results of a series of measurements according to the test layout described in chapter III-A are analysed. Figure 2 the time deviation from 4760 timestamps in an histogram.

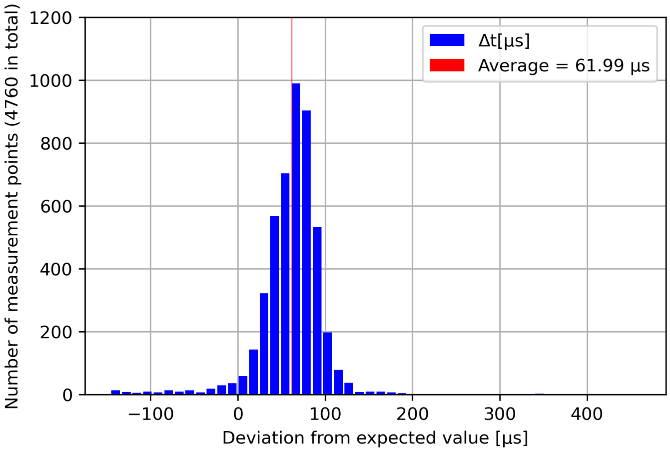

Figure 2. Absolute time deviation from expected value

Most notably the mean value of all data points is not around zero as it would ideally be. There is a constant offset of $62\,\mu s$. In chapter V considerations are made about the origin of this time delay.

Since for the actual deployment of the measurement system two separate signals are compared to each other, a constant offset of each individual PMU can be neglected for the precision analysis. If the first measured time interval is omitted and only the relative differences of the last three measured voltage flanks are considered, the results look much more promising. This can be seen in figure 3.

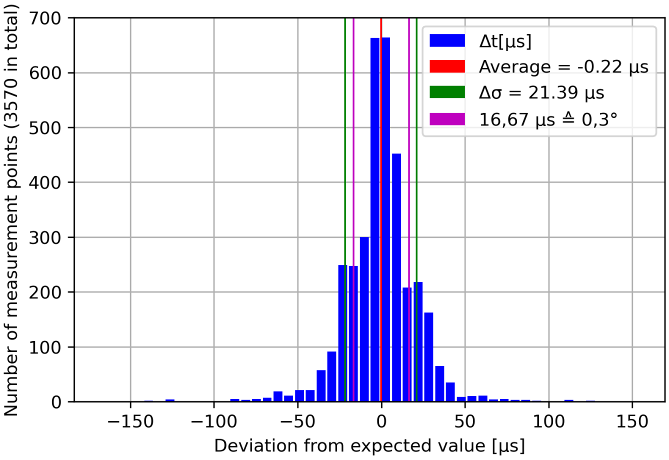

Figure 3. Relative time deviation from expected value

Even though the mean value can be corrected by an offset the standard deviation $\Delta\sigma$ (green) of the values is still higher than the maximum acceptable variation of $16.6\,\mu s$ (purple) which is needed to measure the voltage phase angle with a precision of $\pm 0.3°$.

Since the fluctuation is fairly symmetrical around the mean value it can be smoothed out by a filtering algorithm. This function goes through the list of time values and replaces each one with the mean value of the neighbouring data points. How many neighbouring points are taken into account is defined by the window size of the filter algorithm. Using this method the results can be smoothed out and noise can be suppressed. The larger the filter window size the smaller the spread of the time deviation. This of course leads to relevant fluctuations getting filtered out and switching operations that abruptly change the impedance of the network might be missed. The filter size should therefore be set as small as possible to achieve the desired accuracy for a given application. Figure 4 shows the effect of a filter window size 10 on the deviation from each expected measurement point.

This filtering results in a standard deviation of $\Delta\sigma = 11.7\,\mu s$, which under ideal circumstances would be proficient to deliver results for the voltage phase angle that are accurate enough for the topology estimation.

The same precision could however not be verified based on measurement series conducted on the test layout described in chapter III-B. Possible reasons for this are discussed in chapter V and need to be evaluated further before the

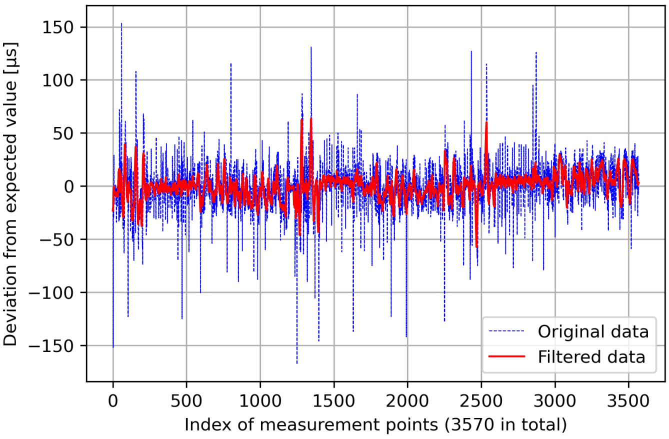

Figure 4. Effects of filtering with a window size of 10

WAMS can be reliably used in the field.

V. Possible Reasons for Measurement Deviations

The evaluated precision measurements show fluctuations of roughly ± 50 μs and a constant offset of 62 μs could be observed. One reason for the initial offset is the runtime of the C code itself. once an interrupt is detected on the PPS signal input a reference time stamp is taken. Since the Raspberry Pi system natively has no real time capabilities, there is no guarantee that this part of the code is executed in a certain amount of time [7].

Other processes running parallel to the measurement software can interfere with the accuracy of the time stamps triggered by an hardware interrupt. The tests were conducted on a system with the Raspbian operation system with a graphical user interface and no other application running but the measurement software. If the system is used not exclusively as a PMU but also to fulfil other purposes the uncertainty of the time measurements will increase. One method to counteract this interference of other application would be to use the Linux “nice” command to prioritise the execution of the C code and alleviate the effects of other less time critical processes taking up processor time in crucial moments.

Another source of error are the inputs for the hardware interrupts of the Raspberry Pi. They show a latency of 6.0 μs with a jitter of 37.7 μs [8].

One more reason for the noise recorded while measuring the 50 Hz signal could be a fluctuation of the signal generator output. The signal was additionally monitored with an oscilloscope as described in chapter III-A but the highest zoom level to capture an entire period only allowed for a display accuracy of ± 50 μs.

The test setup with two devices and a defined impedance between them showed higher variations in the measured time intervals. Reasons for this can be the more complex layout with the additional comparator circuits. There is was also a longer wiring harness involved which is more susceptible for crosstalk from nearby sources of interference. The arguably most import reason for the additional fluctuation is that compared to the evaluated single device setup, here the results of two devices with individual variations were compared which leads to error propagation. Differences between the Raspberry Pi 3B+ and 4B model like the amount of RAM or the processor speed could have also contributed to errors in this time critical measuring method.

VI. Possible Application

A different measuring method with an analogue counter that needs additional hardware but delivers more reliable results which are not dependent on an immediate response time of the Raspberry Pi was also developed as part of the PROGRESSUS project. Compared to this method the presented measurement system was deemed too unreliable to deliver data for a topology estimation. Yet it can still be used as a measurement unit for the voltage signal. As seen in figure 3 the fluctuation of the results is still acceptable if an accuracy of ± 50 μs is precise enough for a given application. In the evaluated test series even without additional filtering 96,8% of the 3570 time intervals recorded were in a margin of error of less than 50 μs which corresponds to an accuracy of the voltage phase angle of ± 0.9°.

The measurement system could therefore be used to conduct decentralized measurements of the distribution network condition to give information about a regional power surplus and shortage in the transmission network or in similar scenarios where the achieved accuracy is sufficient [9].

References

[1] Hotz, C.; Baum, S.; Waffenschmidt, E.; Stadler, I. “Topology Estimation in Low Voltage Grids Using Wallbox Charging Data Recordings” CIRED workshop on E-mobility and power distribution systems, Porto, 2022

[2] Molina-Cabrera, A.; Ríos, M.A.; Besanger, Y.; Hadjsaid, N.; Montoya, O.D. “Latencies in Power Systems: A Database-Based Time-Delay Compensation for Memory Controllers” Electronics 2021, 10, 208.

[3] Schäfer, T. “Entwurf einer Schaltung zur GPS-synchronisierten Spannungsphasenwinkel-messung in großflächigen, elektrischen Netzen” Technische Hochschule Köln, 2021.

[4] AaronFoo “GPS Pulse Per Second Accuracy” fmad.io https://www.fmad.io/blog/accuracy-of-pps-pulse-per-second (accessed Jan. 16, 2023).

[5] Xie, P. “How Slow is Python Compared to C” peter-jp-xie.medium.com https://peter-jp-xie.medium.com/how-slow-is-python-compared-to-c-3795071ce82a (accessed Jan. 16, 2023).

[6] Aalamifar, F.; Hassanein H. S.; Takahara, G. “Viability of Powerline Communication for the Smart Grid” 26th Biennial Symposium on Communications (QBSC), 2012.

[7] Carvalho, A.; Machado, C.; Moraes, F. “Raspberry Pi Performance Analysis in Real-Time Applications with the RT-Preempt Patch” Latin American Robotics Symposium (LARS), 2019.

[8] Stefan, J. “Interrupt response times on Arduino and Raspberry Pi” International Postgraduate School Ljubljana, 2016.

[9] Waffenschmidt, E.; Littau, P.; Pelikan, C. “Dezentrale Messungen des Netzzustandes” Zukünftige Stromnetze für Erneuerbare Energien conference, Berlin, 2018.