



Multi-device software for impedance spectroscopy measurements with stabilization in low and high temperature ranges working under Linux environment

Tomasz K. Pietrzak¹

Received: 10 January 2019 / Revised: 28 February 2019 / Accepted: 28 February 2019 / Published online: 23 March 2019
© The Author(s) 2019

Abstract

Impedance spectroscopy is a powerful experimental technique for complex analyses of electrical response of conducting samples. Experimental setups consist of a wide variety of analyzers and temperature controllers for high and low temperatures, and the experiments may last for days. In this paper, a self-developed controlling software written in C++ and working under GNU Linux that co-operates with many different devices in the laboratory is described. Measurements can be carried out at elevated and low temperatures, using either a furnace or a cryostat, i.e., above 77 K, and to a temperature exceeding 1000 K. The major advantage of this solution is the possibility to freely combine different analyzers and temperature controllers according to actual experimental needs. The possibility to conduct several parallel measurements at one PC makes this software interesting for every well-equipped and intensively used laboratory of impedance spectroscopy.

Keywords Impedance spectroscopy · Acquisition software · Low and high temperature · Multi-device · Linux

Introduction

Impedance spectroscopy is a powerful experimental technique for complex analyses of electrical response of conducting samples. While total electrical conductivity can be easily determined for electronic conductors (i.e., metals and semiconductors) with DC measurements, materials with more complex charge transport require in-depth analyses. This is inevitable in case of pure ionic and mixed electronic-ionic conductors, where mobile ions would be blocked at irreversible electrodes and suppress direct current. In case of polycrystalline ceramics or glass-ceramics, impedance spectroscopy may be used to distinguish charge transport processes related to microstructure of the materials, e.g., in grains or grain boundaries [1]. Finally, with this method ionic and electronic transference numbers may be estimated [2]. While the aforementioned applications are of the interest of the author of this paper, numerous other can be named,

e.g., in electrochemical cells (lithium or sodium batteries), diffusion processes in liquid electrolytes and many other.

Two most popular ways to analyze impedance data should be mentioned. The most classical method is equivalent circuit approach. An electronic circuit is proposed, in which each element would model a physical process. Typically, resistors, inductances and capacitors are used. Advanced processes may require more complex elements, such as constant phase element (CPE), Warburg impedance (W) or elements related to relaxation processes (Cole-Cole, Havriliak–Negami). With such a variety of elements, a vast number of processes can be modeled, such as electronic or ionic conductivity, bulk or microstructure capacitance, diffusion, electrode processes or inductance of experimental setup. Typically, equivalent circuits are unambiguously described using Boukamp notation [3].

An approach that gains a growing popularity is density function of relaxation times (DFRT) method [4]. It allows to determine parameters of relaxation processes (characteristic relaxation times, activation energies of the processes) without knowledge about equivalent circuit. It allows to distinguish two processes (e.g., relaxation in grain and intergrain of ceramics) and to evaluate their contributions to total resistance of the ceramics, also in case when the frequency ranges of responses of these processes are

✉ Tomasz K. Pietrzak
tomasz.pietrzak@pw.edu.pl

¹ Faculty of Physics, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland

overlapped and they cannot be easily separated by other methods [5].

In the field of solid state ionics, impedance spectroscopy measurements are performed in a function of temperature, i.e., a spectrum is collected in isothermal conditions for each selected temperature. Hence temperature dependences of conductivity can be obtained and the value of the activation energy is calculated. The total impedance of the measured sample can strongly vary from single ohms to more than gigaohms. The temperature range depends of the nature of the material. Compounds for solid oxide fuel cells (SOFC) can be investigated at temperature as high as 800 °C (e.g., [6]). Processes inside ionic conductors may be more pronounced at much lower temperature, ranging down to −180 °C (e.g., [7]). Conductivity based on electron hopping shall be studied in a wide temperature range from −180 °C to +500 °C to observe both phonon assisted hopping and variable range hopping (e.g., [8–10]).

Such a variety of experimental expectations results in a necessity to use various devices, including different impedance analyzers and temperature controllers/stabilizers which should co-operate flexibly with each other. Furthermore, automatic control of the measurement by a computer is essential in case of prolonged experiments, lasting for days. Such features can be hardly provided by a software delivered by manufactures of single devices. Because impedance spectroscopy is a major experimental technique at the Solid State Ionics Division of the Faculty of Physics, a development of our own controlling software was a natural solution to meet all the requirements for effective experimental works. For decades, we have been successfully using a software written by Prof. J. Dygas from our faculty [11], including further improvements [12]. However, it was written in Fortran and later in Pascal for single-tasking DOS and in 21st century it became more and more difficult to maintain. While it supported measurements at high and low temperatures, it operated only with Solartron analyzers. Therefore a new computer program named *Kura* written in modern C++ and working under GNU Linux that co-operates with many different devices in our laboratory has recently been created and implemented.

Supported devices

The devices controlled by the software can be divided into two categories: (A) impedance analyzers; (B) temperature controllers. Currently, two impedance analyzers are supported:

- 1) Solartron 1260 – with frequency range from 10 μHz to 32 MHz and impedances up to 100 MΩ. The device is connected through serial interface RS-232.

- 2) Novocontrol Alpha-A – with frequency range from 30 μHz to 20 MHz and impedances up to 1000 TΩ. The device is connected through GPIB interface.

The temperature can be controlled by two setups:

- 1) Eurotherm 2404 coupled with ADAM-4011 thermocouple interface for high-temperature (HT) measurements. These devices provide control of the furnace temperature and measure the real temperature of the sample using a separate thermocouple. They are connected through ADAM-4520 RS-485 to RS-232 adapter. Currently, the devices have been ergonomically mounted inside a chassis of a tube furnace supplied by a Polish manufacturer *Czylok*.
- 2) Lakeshore 331 for low temperature (LT) measurements coupled with a nitrogen-flow cryostat. The device connected through RS-232 interface allows to measure the temperature in two separate points (e.g., by a thermocouple or by Pt resistor). One is situated next to the cold finger, the other next to the sample. The device plays also a role of a power supply for heating coils in the cryostat.

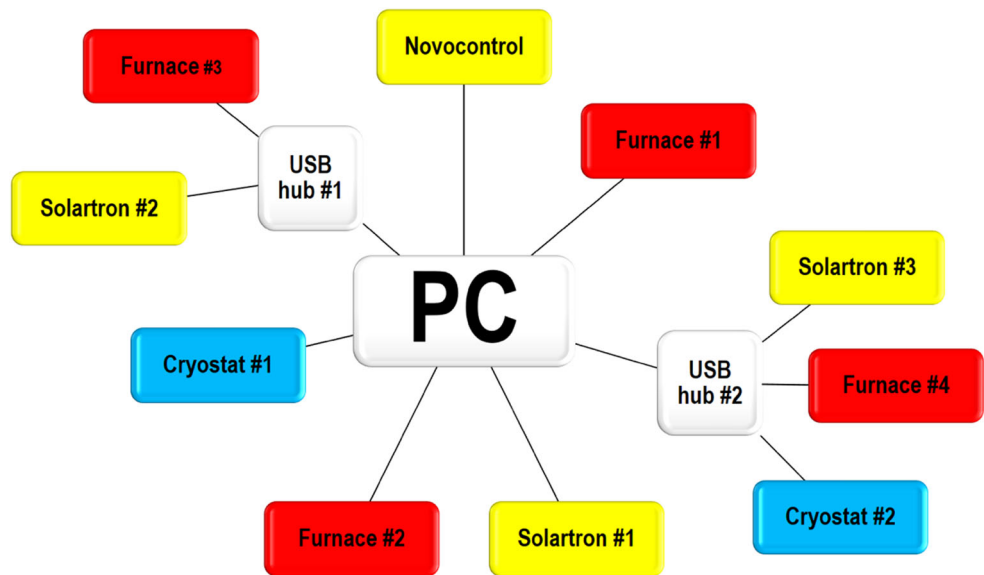
As one can see, the devices use either GPIB or RS-232 interface. While RS-232 is rather deprecated, GPIB usually requires costly cards for PC. These issues have been, however, circumvented. Firstly, RS-232/USB adapters with FTDI chips were bought. These chips allow programmable identification of each adapter. This feature will be described later in this section. Secondly, a Prologix GPIB-USB interface is a cost-efficient alternative to GPIB cards and provides simple communication protocol with controlling software.

The aforementioned devices were implemented due to their availability in the laboratory. However, a support of any other devices may be added to the program by an experienced programmer, if necessary. The drivers are implemented according to their communication handbooks in a form of a C++ class. It inherits from a common abstract class providing a unified programmer interface either for impedance analyzers or temperature controllers.

The principle that stands behind the whole experimental setup is that one can combine any of impedance analyzers with any of temperature controllers, including both HT i LT devices. Therefore all devices are connected to one controlling computer using USB cables and hubs, when necessary. An example of such connections is shown in Fig. 1.

In Linux, a serial port is identified by its name stored in */dev* directory, e.g., */dev/ttyUSB0*. However, this name may change after reboot, so it cannot be permanently assigned to a measurement device. Fortunately, both Prologix and FTDI adapters may store unique ID that can be read by the operating system. In case of the former, a unique number is

Fig. 1 An example of the connections between the controlling computer and various devices. For a particular measurement, necessary devices can be freely chosen



stored by the manufacturer. In case of the latter, the internal ID can be set by additional software, e.g., FT_PROG. Those IDs are easily accessible in `/dev/serial/by-id/` directory (Fig. 2). Therefore the administrator of the program needs to assign unique IDs of the adapters to the devices (e.g., analyzers or temperature controllers) they are connected to in `/opt/kura/devices.list` file. This procedure has to be done by the administrator only once (and repeated each time a new measurement device appears) and no further action is required from everyday users. This configuration file is parsed by the program which automatically combines port names with user-friendly names of the devices.

Controlling software

The software is written in modern C++ (standard C++14) for Linux. This environment provides stable conditions for working of the application, which is essential for impedance measurements which last for days. The software is based on Qt SDK (software development kit), which is a popular, powerful, and open-source widget toolkit for creating graphical user interfaces and hardware communication. While the main interface is based on a flat file configuration and a Linux terminal, appropriate Qt and Qwt classes are used for visualization of the impedance spectrum during

the measurement. Serial communication is done through *Qt Serial Port* class, which provides an easy access to USB/RS-232 terminals.

Nowadays, programs with user-friendly “clickable” interfaces gain growing popularity. If well designed, they are easy to operate by new users. However, poor and not intuitive interfaces make applications difficult to use. The Linux environment is known for a rich collection of efficient batch/console applications. They are especially useful to conduct data processing that is needed repeatedly. Similarly, impedance measurements are usually performed in the same frequency ranges and similar temperature ramps. Therefore, it was decided that for such routines, a console interface of the program based on a flat file configuration would be the most efficient solution.

Configuration files

To start a measurement, three configuration text files are needed:

1. the main configuration file (CFG),
2. frequency ramp configuration (FRQ),
3. temperature ramp configuration (TEMP).

In the main configuration file (Fig. 3), the following parameters are set:

```
administrator@administrator-715krio:/dev$ ls -l /dev/serial/by-id/
total 0
lrwxrwxrwx 1 root root 13 gru 3 12:11 usb-Piecyk_1-if00-port0 -> ../../ttyUSB2
lrwxrwxrwx 1 root root 13 gru 3 12:11 usb-Piecyk_2-if00-port0 -> ../../ttyUSB3
lrwxrwxrwx 1 root root 13 gru 3 12:11 usb-Prologix_Prologix_GPIB-USB_Controller_PXG4I1GN-if00-port0 -> ../../ttyUSB0
lrwxrwxrwx 1 root root 13 gru 3 12:11 usb-Solartron_2-if00-port0 -> ../../ttyUSB1
```

Fig. 2 Unambiguous identification of adapters with fixed unique IDs in Linux console. In this example, four devices are visible: two furnaces (*piecyk* in Polish), a Solartron and Prologix GPIB-USB adapter for communication with a Novocontrol

```

1 # Filename core
2 N35aIS01
3 # Description
4 N35aIS01, Na3Fe2(P04)2F3, 2018-12-03, to 500*C, d = 0.98 mm
5 # Frequency ramp filename
6 ../N33a.frq
7 # Temperature ramp filename
8 N35aIS01.temp
9 # Furnace temperature tolerance
10 0.1
11 # Sample temperature tolerance
12 0.1
13 # Minimum stabilization time (minutes)
14 30
15 # Maximum stabilization time (minutes)
16 90

```

Fig. 3 An example of the main configuration file

1. core of the filename—each recorded spectrum will have a filename composed of this core and an incremental extension starting from 000;
2. human readable description of the sample—one line of text that will be stored at every file that will provide identification for the sample;
3. filename of the frequency ramp configuration (it may include a relative or absolute path);
4. filename of the temperature ramp configuration;
5. furnace temperature tolerance (ΔT_f);
6. sample temperature tolerance (ΔT_s);
7. minimal time of temperature stabilization ($\Delta \tau_{min}$);
8. maximal time of temperature stabilization ($\Delta \tau_{max}$).

The meaning of the last four parameters will be described later in this paper.

Each impedance spectrum is recorded at exactly the same frequencies described in a FRQ file (Fig. 4). Because measurements at high frequencies are quick and scans at low frequencies can last minutes or hours, usually it is convenient to divide full frequency range into smaller ramps, e.g., for high, middle and low frequencies. Each frequency ramp is defined in a single line of the FRQ file.

```

1 #LF» HF» Step»Ampl»Bias»Type
2 1» 1e+6»2» 1.0» 0.0» log
3 1e+3»1e+7»25» 1.0» 0.0» log
4 1» 1e+3»15» 1.0» 0.0» log
5 1e-2»1» 5» 1.0» 0.0» log
6 1» 1e+6»2» 1.0» 0.0» log

```

Fig. 4 An example of the frequency configuration file. Five frequency ramps are set, covering the frequency range from 10 mHz to 10 MHz. The first and the last ramps are the same and cover range from 1 Hz

The following parameters separated with a white character (e.g., a space or a tabulation) must be given:

1. lower frequency limit,
2. upper frequency limit,
3. number of measurements per decade,
4. voltage amplitude,
5. voltage bias,
6. type of a sweep—logarithmic or linear.

The construction of a TEMP file is similar (Fig. 5). Here each line defines a heating or cooling ramp. The order of parameters must be as following:

1. initial temperature (T_1),
2. final temperature (T_2),
3. temperature step,
4. heating/cooling rate.

It is worth to notice that most of these parameters would usually be unchanged (e.g., frequency ramps or stabilization parameters) between measurements of similar samples. In such case FRQ and TEMP configuration files can be shared and stored in a global location, whereas only CFG file needs

to 1 MHz with only 2 points per decade—they are useful to verify the stability of the sample upon long measurement

Fig. 5 An example of the temperature configuration file. Two ramps are set (heating and cooling) with the same measurement step (25 °C) and the heating/cooling rate (5 °C/min)

#	T1	T2	Step	Rate
1	# T1	>> T2	>>	Step>>Rate
2	# Heating ramp:			
3	50.0	>> 500.0	>> 25	>> 5
4	# Cooling ramp:			
5	475.0	>> 25.0	>> 25	>> 5

to be modified (e.g., filename core and sample description) and presumably copied to the current working directory before each measuring procedure. On the contrary, for a measurement with a unique set of parameters, customized configuration (i.e., FRQ, TEMP and CFG files) can be provided for each measurement.

In all configuration files, lines starting with # are treated as comments, which may be useful in memorizing the order of the parameters.

Program initialization

Having prepared the configuration files, the user starts the program in a Linux terminal with a simple command:

```
kura configuration-file.cfg
```

The program will print a welcome message followed by a list of available impedance analyzers and temperature controllers. The user should pick up one analyzer and one temperature controller. Alternatively, there is an option for

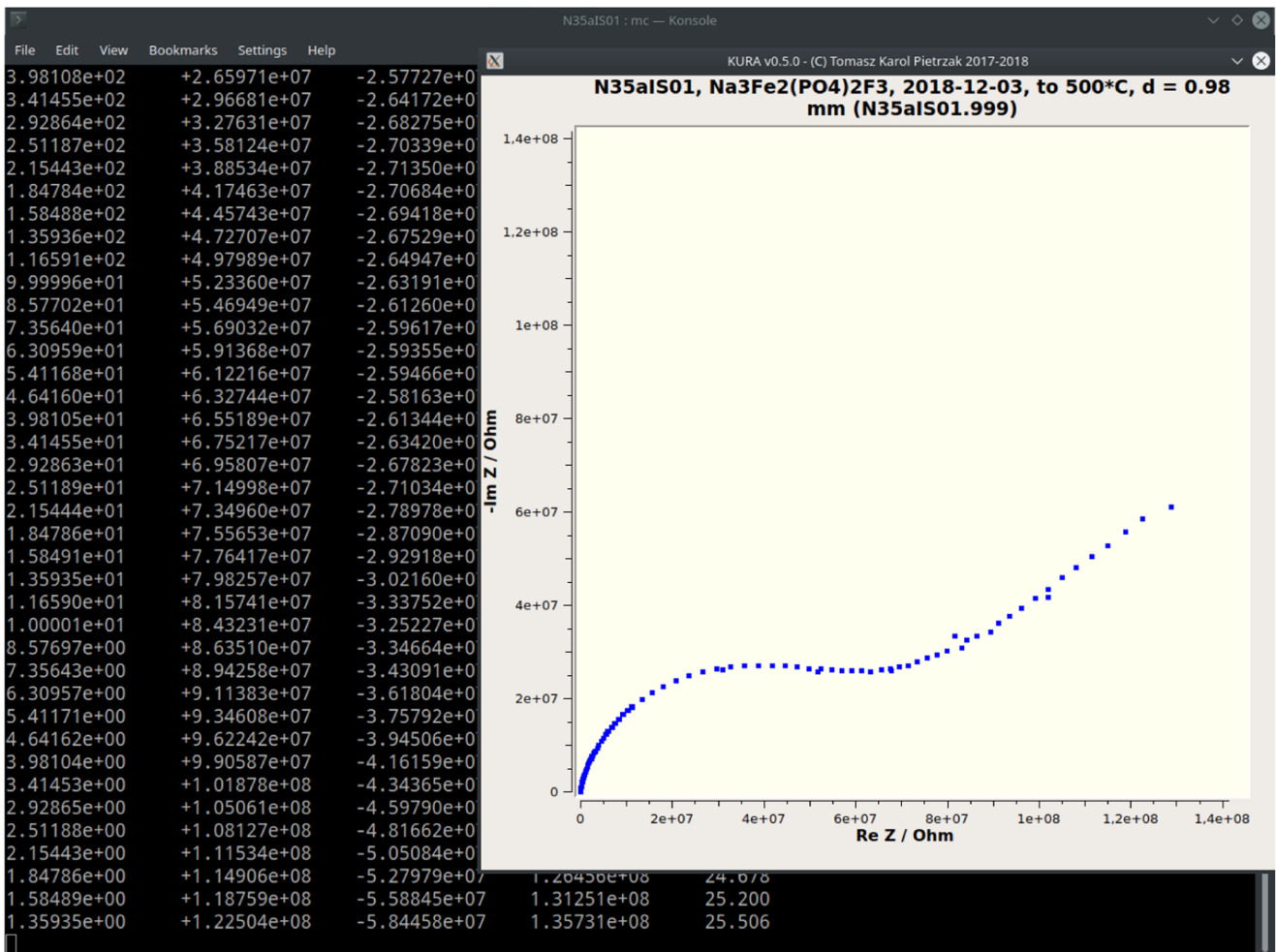


Fig. 6 A screenshot of the program during a measurement of a single impedance spectrum. A live visualization of the data in Nyquist plot is shown. At the same time, numerical values of the frequency, the real

and the imaginary part, the absolute value and the phase of impedance are printed in a terminal in the background

a single impedance spectrum measurement. In such a case, no temperature controller will be used. Please note that several instances of the program may be launched on one computer. In such a case, the software will prevent from selecting devices that have already been assigned to other measurements.

Measurements

After the initialization, the main measurement loop starts according to FRQ and TEMP files. The next measurement temperature T_{sp} is approached with the specified heating (or cooling) rate. Then, the program waits until the real controller temperature T satisfies the stability condition in a 60-s period, i.e., $|T - T_{sp}| \leq \Delta T_f$. However, if the condition has not been satisfied for $\Delta\tau_{max}$, the program will print an error message and move forward. Subsequently, the program waits $\Delta\tau_{min}$ for the stabilization of the temperature measured next to the sample (T_s). Meanwhile, values of T_s from the last five minutes are stored. After $\Delta\tau_{min}$, the program waits for the sample stability condition (i.e., $\max(T_s) - \min(T_s) \leq \Delta T_s$) to be satisfied. In any case, the stabilization stage lasts no longer than $\Delta\tau_{max}$ and eventually the measurement starts.

In the measurement stage, the impedance of the sample is acquired at various frequencies defined in the FRQ file. A preview of the collected data appears in a graphic plot in Nyquist coordinates (Fig. 6) and is updated each time a new point is collected. Meanwhile, numeric values of the frequency, the real and the imaginary part, the absolute value, and the phase of impedance are printed in a terminal in the background. When the spectrum is acquired for all set frequencies, the output text file is saved on the disk.

Besides impedance values (i.e., the frequency, the absolute value of the impedance and the phase angle), the following information is stored: the initial, minimal, and maximal temperature of the sample during measurement, the initial temperature of the furnace, and the description of the sample taken from the CFG file.

After acquisition of the impedance spectrum, the program sets the controller temperature with heating/cooling rate according to the TEMP file.

Termination

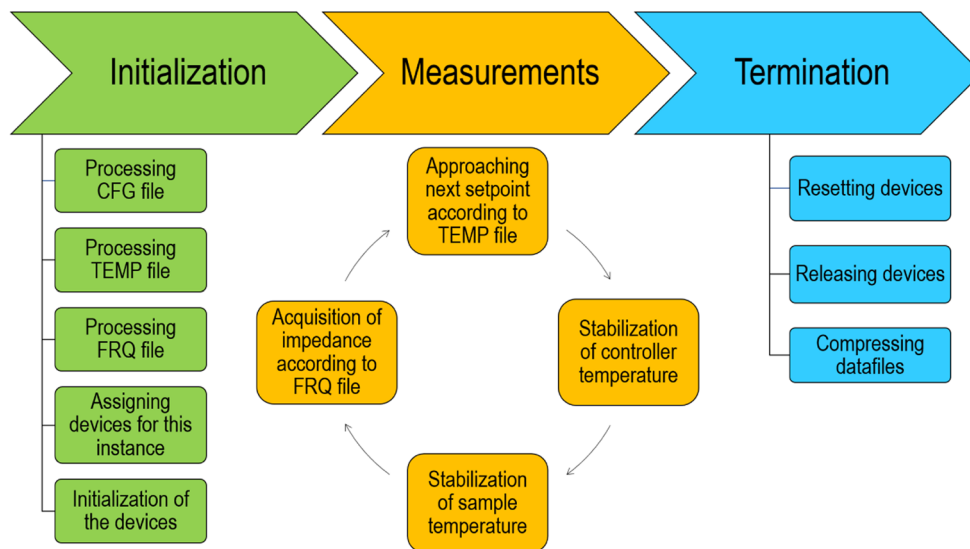
When the impedance has been acquired for all temperatures defined in the TEMP file, the program terminates. The devices that have been assigned to the measurement are released, and now, they become available for any further instance of the program. Usually tens of data files are created by the program for a single sample. Therefore, for the user’s sake, the program prints in the terminal a command for ZIP-ing all the files in a single archive.

The flowchart showing the program execution from initialization through measurements to termination is schematically shown in Fig. 7.

Multiple instances

As it was said before, the main idea standing behind the program is that many analyzers and temperature controllers are connected to the same PC. Therefore, it is expected that many users will run their measurements at the same time. For each measurement, a new instance of the program with its own CFG file has to be run. Obviously, a device must not be used by more than one instance of the program.

Fig. 7 Flowchart of program execution



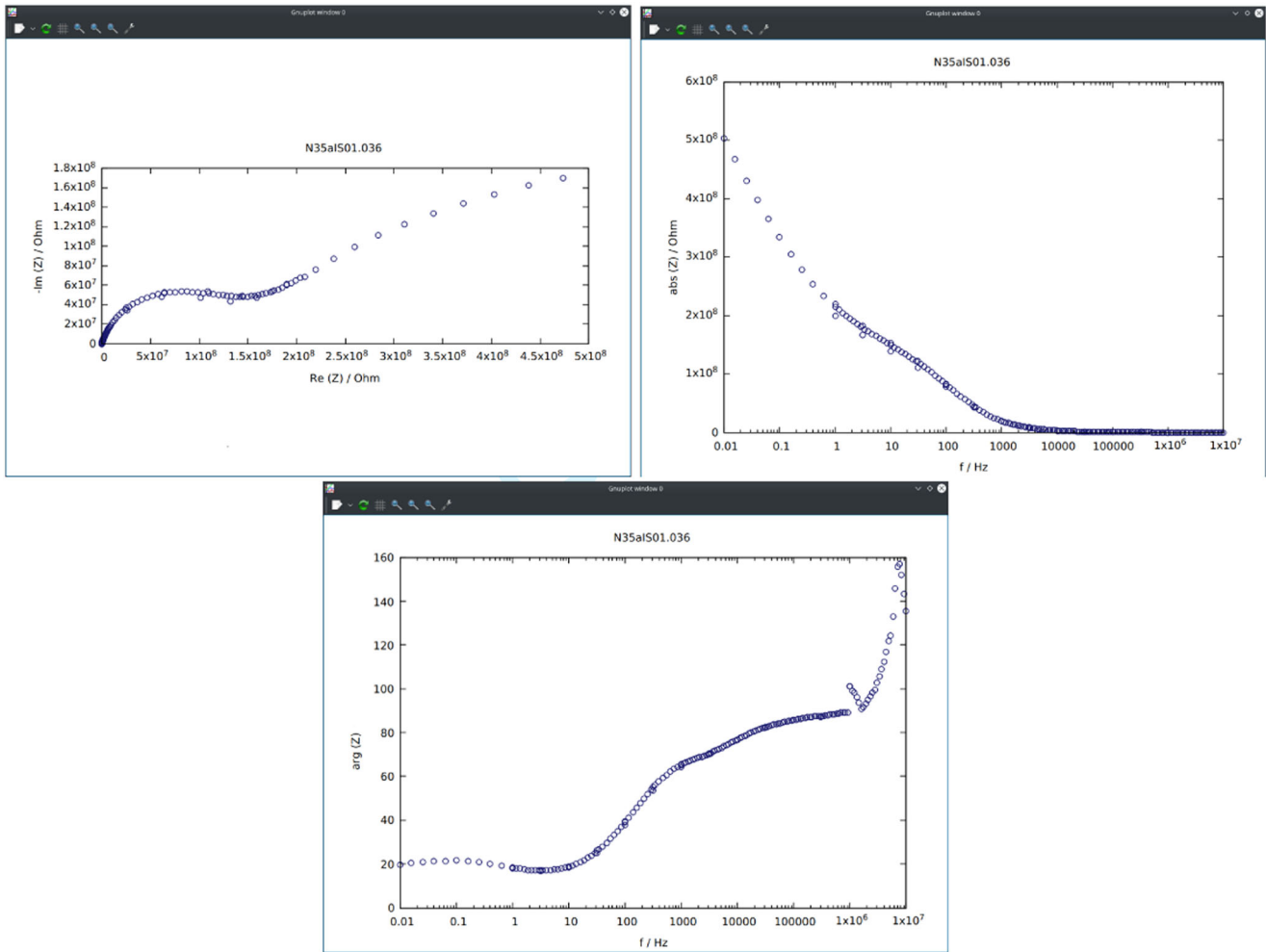


Fig. 8 Immediate plots of collected impedance data in various coordinates, generated by a Bash script adopting *Gnuplot* software

Therefore, the program keeps the current status of the devices in */opt/kura/devices.stat* file and allows the user to select for their experiments only the devices that have not been assigned by other instances of the program.

Data preview

The program displays only the spectrum that is currently being acquired. Obviously, users may want to preview their results during measurement in order to check the quality of the experiment, etc. Unfortunately, it seems that popular software for plotting impedance spectra are available for the Windows platform only. To fill this gap for professional Linux users, a simple and convenient Bash script was written that utilizes popular plotting software—*Gnuplot*—to preview a single impedance spectrum. The script is as short as ca. 75 lines, and to run the script, a user needs to follow its name with a filename of the spectrum in the command line. Besides a Nyquist plot, $|Z|$ and $\arg(Z)$ (the phase) representations are implemented. Examples of such

plots are shown in Fig. 8a–c. The script adopts all useful features of *Gnuplot*, such as plot scaling or export to graphic files.

Conclusions

In this paper, a self-developed software for impedance measurements in a function of temperature was described. At present time, Solartron 1260 and Novocontrol Alpha-A analyzers are supported. Measurements can be carried out at elevated and low temperatures, using either a furnace or a cryostat, i.e., above 77 K, and to a temperature exceeding 1000 K. The major advantage of this solution is the possibility to freely combine different analyzers and temperature controllers according to experimental needs, as well as to conduct simultaneously independent measurements controlled by one PC. Further advantages include non-commercial and open-source Linux environment, cost-efficient GPIB-USB and RS232-USB adapters,

flat file configuration (time effective for routine and repetitive measurements), unambiguous assignment of devices to serial ports, and the possibility to implement support for other devices in the future. The aforementioned features make this software interesting for every well-equipped and intensively used laboratory of impedance spectroscopy.

Any experimentalists who are interested in this solution or wish to deploy it in their laboratories are welcome to contact the author to start co-operation.

Acknowledgements I wish to thank Prof. Jerzy Garbarczyk and Prof. Marek Wasiucionek for discussions on this manuscript. I wish to acknowledge my former B.Sc. students who contributed to the development of the software: Mr. Jan Chelmicki (a driver for Eurotherm 2404), Mr. Jan Olszak (a driver for Lakeshore 331), and Mr. Michał Możejko (management of USB interfaces).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Pietrzak TK, Garbarczyk JE, Wasiucionek M (2018) Stabilization of the delta-Bi₂O₃-like structure down to room temperature by thermal nanocrystallization of bismuth oxide-based glasses. *Solid State Ionics* 323:78–84
- Huggins RA (2002) Simple method to determine electronic and ionic components of the conductivity in mixed conductors a review. *Ionics* 8:300–313
- Boukamp BA (1986) A nonlinear least squares fit procedure for analysis of immittance data of electrochemical systems. *Solid State Ionics* 20:31–44
- Casida ME (1995) Recent Advances in Density Functional Methods, 155–192
- Kazlauskas S, Kežionis A, Šalkus T, Orliukas AF (2013) Electrical properties of YSZ and caSZ single crystals. *Solid State Ionics* 231:37–42
- Borowska-Centkowska A, Leszczynska M, Krok F, Malys M, Wrobel W, Hull S, Abrahams I (2018) Local structure and conductivity behaviour in Bi₇WO_{13.5}. *J Mater Chem A* 6:5407–5418
- Foltyn M, Wasiucionek M, Garbarczyk JE, Nowiński JL, Gierlotka S, Pałosz B (2008) Low-temperature conductivity of composites based on Ag⁺-ion conducting glasses and α-Al₂O₃ matrix, prepared via a high-pressure route. *Solid State Ionics* 179:38–41
- Pietrzak TK, Garbarczyk JE, Wasiucionek M, Gorzkowska I, Nowinski JL, Gierlotka S (2011) Electrical properties vs. microstructure of nanocrystallized V₂O₅-P₂O₅ glasses - An extended temperature range study. *Solid State Ionics* 192:210–214
- Pietrzak TK, Maciaszek M, Nowiński JL, Ślubowska W, Ferrari S, Mustarelli P, Wasiucionek M, Wzorek M, Garbarczyk JE (2012) Electrical properties of V₂O₅ nanomaterials prepared by twin rollers technique. *Solid State Ionics* 225:658–662
- Pietrzak TK, Garbarczyk JE, Wasiucionek M, Nowiński JL (2016) Nanocrystallization in vanadate-phosphate and lithium-iron-vanadate-phosphate glasses. *Physics and Chemistry of Glasses: European Journal of Glass Science and Technology Part B* 57:113–124
- Dygas JR (1986) Ph.D. Thesis, Northwestern University, Evanston
- Dygas JR, Fafilek G, Durakpasa H, Breiter MW (1993) Automated setup for impedance measurements of electrochemical cells with two electrodes. *J Appl Electrochem* 23:553–558

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.