Measurement 44 (2011) 702–707

Contents lists available at ScienceDirect

Measurement

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

A real time FFT-based impedance meter with bias compensation

Alfredo Accattatis *, Giovanni Saggio, Franco Giannini

Department of Electronic Engineering, University of Rome Tor Vergata, via del Politecnico 1, 00133 Rome, Italy

ARTICLE INFO

Article history:
Received 28 January 2010
Received in revised form 13 December 2010
Accepted 7 January 2011
Available online 14 January 2011

Keywords:
Analog–digital conversion
Data acquisition
Digital–analog conversion
Impedance
Transforms
Fourier series

ABSTRACT

A real time FFT-based impedance meter is here presented. It is realized simply by a standard PC in addition to a low cost two channels hardware device, and is based on a cost-effective algorithm to meaningfully reduce the known problem of the bias introduced by the relative phase and amplitude error between channels. The impedance meter exploits the implementation of an ad hoc cost-effective algorithm and a synchronous sampling allows eliminating the leakage error. The aliasing and the spectral interference are eliminated by means of an anti-aliasing filter and the use of very low distortion sinusoidal signals. Automatic measurements can be managed too, capturing different measures at different test frequencies and displaying them graphically on the PC screen. To validate the measured results comparative test were performed with respect to certified and calibrated commercially available multimeters known with good accuracies.

Finally, this novel impedance meter integrates a home-made software, previously reported [1] capable to implement real time virtual instruments.

1. Introduction

This paper relates to the development of a software running on a standard PC Windows-based equipped with a low cost two channels data acquisition board, with the interesting characteristic of being able to measure impedances with high accuracy, at lower cost compared with analogous commercially available meters. It is based on the Fast Fourier Transform (FFT) and a method to shrink bias errors. Fig. 1 is a simple schematization of the acquisition side of the meter: the DUT (Device Under Test), in series with a known reference resistor, is supplied by a sinusoidal generator digitally synthesized via software.

A well established method to measure impedance values commonly utilizes the "auto balancing bridge" [2,3]. It has got an elegant inner simplicity, but it is based on a recursive algorithm which can sometimes suffer from instability problems.

This is because, in order to minimize errors and to converge to a solution, it uses the Least Mean Square (LMS) algorithm for which as many samples as possible are needed to assure convergence with a reasonable error, allowing only up to four measurements per second on a modern PC; this latter point was experimentally measured on a modern core-duo personal computer. On the other hand, LMS does not need to compute an FFT, avoiding computational burden.

We propose a novel FFT-based measurement method capable to overcome the mentioned problems. FFT-based methods are typically affected by leakage errors, harmonic interferences and aliasing, but our system can accurately solve and virtually eliminate these problems, as further proved. For both the "auto balancing bridge" method and for our system, the hardware introduces phase and amplitude errors, between the two channels, which propagate through FFT lowering the accuracy of the measure. But these drawbacks are here avoided too, so we compute the correct value of the impedance with a very low cost hardware, obtaining excellent results with a small number of samples, and allowing computing up to twenty measurements per second while maintaining a standard accuracy. Last but not least, time and frequency domain representation of the input signals and the possibility to execute automatic measurements are made available.

* Corresponding author. Tel.: +39 335 8773070.
E-mail address: accattatis@ing.uniroma2.it (A. Accattatis).

0263-2241/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.measurement.2011.01.008

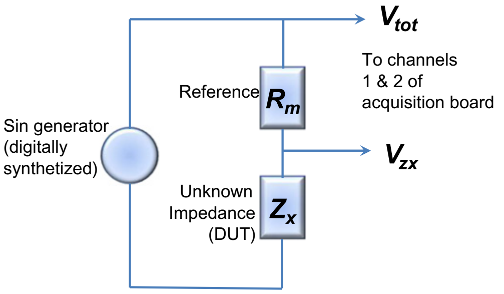

Fig. 1. Adopted scheme to measure the unknown impedance. The voltage is generated via software and applied via the output of the acquisition board; the voltages across the two resistors are fed back to the acquisition board via the input channel and stored in memory.

2. Least Mean Square (LMS) method

Literature reports several error-minimization criteria, and the Widrow–Hoff least square approximation is the most widely known and used. Basically it consists of a function that minimizes the sum of the squared errors to estimate two variables W_1 e W_2, being recursively computed as the coefficient of the adaptive linear combiner, so to obtain the wanted Z_x as [4]:

$$Z_x = R_m W_1 + J(R_m W_2) \tag{1}$$

where R_m is the reference resistor and J stands for the imaginary Z_x part.

The error E_k at the k-iteration of an N-point buffer is computed as:

$$E_k = V_{rmk} - V_{zxk} \tag{2}$$

where V_{rmk} is the voltage across the reference resistor at the k_{th} instant and V_{zxk} the voltage across the unknown impedance at the same instant. In particular, V_{rmk} is computed as difference between V_{tot} and V_{zxk} (Fig. 1). When $E_k = 0$ the bridge is virtually balanced; the LMS proceeds applying a steepest-descendent algorithm to find the "W" coefficient:

$$W_{k+1} = W_k + 2\mu_0 E_k X_k \tag{3}$$

where μ_0 is a convergence factor and X_k is the vector of the samples (V_{rmk} and V_{zxk}) at k_{th} instant. Virtual instruments using LMS basically need only to acquire the samples of signals, compute the real and imaginary part of the impedance by means of Eqs. (1)–(3), and then compute the phase between the voltage and the current flowing through the impedance to determine the nature of the impedance.

3. FFT impedance meter

LMS can suffer from instability and cancellation error if applied with very low level input signal. In addition, it needs at least three times the amount of samples needed by the method we propose to obtain a reasonable level of accuracy, as stated in the Section 1.

Our proposal is therefore to compute the unknown magnitude and phase of impedance not recurring to the commonly applied (1)–(3) equations, but implementing a simpler method, so always referring to Fig. 1, by calculating the voltage across the reference resistor, and finally applying the simple and well known relationship:

$$|Z_x| = (|V_{zx}|/|V_{rm}|)R_m \tag{4}$$

where $|V_{rm}|$ is the magnitude of the voltage across the reference resistor, $|V_{zx}|$ is the magnitude of the voltage across the unknown impedance Z_x and R_m is the value of the reference resistor. To compute the phase (in degree), and hence get a complete knowledge of the Z_x, we prefer to get the magnitude and relative phase of the two voltages exploiting the information provided by the FFT, plus a method to reduce drastically the main cause of bias errors between the two channels. Moreover we choose the frequency of the test voltage and sampling frequency to achieve synchronous sampling, thus avoiding the *leakage error*.

Although the FFT involves more computational burden, it is not such a problem since a modern PCs allow to compute an FFT in a few milliseconds (less of 5 ms on a modern Intel dual core microprocessor T9300 2.5 GHz, with a 4096 points buffer); moreover the use of 80 bit floating points reduce drastically the amplitude of the rounding error.

4. Material and methods

The method we propose is capable to correctly measure impedances with high accuracy but being based on a simple low-cost commercially available acquisition board. For simplicity we realized a home-made prototype using a standard PCM2902 (USB A/D–D/A converter 16 bit up to 96 kHz of frequency sampling) plus a double operational amplifier LM358 to get an high input impedance. Obviously, it is necessary an appropriate choice of the system parameters, such as frequency sampling of the A/D converter, number of acquired samples for each data buffer, and the frequency of the generated test signal. The choice of these parameters is made with the aim to reduce the FFT errors at the minimum possible level together with an appropriate transformation of the acquired signals.

In the following Section 4.1 we will see how we can select the system parameters to avoid most of the FFT errors, in Section 4.2 we report how to manipulate the input signals to minimize the phase error between channels, and in Section 4.3 it is shown how to compensate the relative amplitude error.

4.1. FFT issues

Well known issues using FFT are:

- *Aliasing.*
- *Leakage errors* (*Spectral Leakage*).
- *Harmonic Interference* (*HI*).

Aliasing was avoided simply using the standard anti-aliasing filter (normally present in each acquisition board); moreover, the frequency of the test signal was chosen well below the Nyquist frequency. We start observing that many acquisition board allows to set a frequency sampling well beyond the predefined standard frequency, but leaving unchanged the anti-aliasing filter. If we choose a frequency sampling of 81,920 Hz, the anti-aliasing filter remains tuned at about 20,000 Hz. So, if we limit the frequency of test signal within this latter range, we assure an optimal behavior of the anti-aliasing filter, that is, we are working in the oversampling region.

Leakage error is virtually eliminated making some assumptions.

First of all, consider that the resolution of the FFT for the selected system parameters is:

$$\mathrm{FFT}_{res} = F_s/points = 81920/8192 = 10\ \mathrm{Hz} \tag{5}$$

Secondly, if we assume for the frequency of the test signal an integer multiple of FFT_{res} ($F_t = n\mathrm{FFT}_{res}$, where n = positive integer, for example $n = 100$ means 1000 Hz for the chosen values in (5)) we reach the condition known as synchronous sampling, avoiding completely the leakage error. Moreover, we apply a digital inverted notch IIR filter to the (sinusoidal) signal acquired (with cutoff frequency equal to F_t) so further preventing *leakage error* and *Harmonic Interference*.

Last but not least, the usage of the same clock to drive the both A/D and the D/A converter and also to acquire and generate the signals, further reduce the leakage error.

4.2. Hardware phase error

The two voltages V_{rm} and V_{zx} are acquired from two different channels of the acquisition board (Fig. 1). Many acquisition boards use only one A/D converter for all the supported channels, by means of a hardware multiplexer. In many cases this means that the two channels have a phase differences of *at least* one sample. This is the case, for instance, of the acquisition board we used in our test system. A relative delay of one sample means a delay equal to T_s i.e. the period of the sampling frequency. That is a delay of a number of degrees proportional to F_t; with F_t equal to 1000 Hz, the delay introduced by a one sample delay is about 8.7°. So, given for example, an impedance to be measured with a magnitude of 100 Ω and a phase angle of φ = 50°, we obtain a real part of 64.27 Ω (computed as $|Z|\cos(\varphi)$, where $|Z|$ = 100 and φ = 50). A variation of 8.7°, that is a phase angle φ = 58.7°, means a real part of 51.95 Ω, a variation of about 20% of the computed value. The first step to eliminate this phase problems can be realized by implementing a simplified scheme where the samples flow freely on a channel, while in the other they are accumulated in a software FIFO queue with the possibility to choose via software the amount of (discrete) delay (see Fig. 2). In fact, the delay amount can only be an integer multiple of the sampling period T_s. After the delay, an inverted notch IIR digital filter is applied, with cutoff frequency equal to the test frequency, to guarantee a high purity of the test signal.

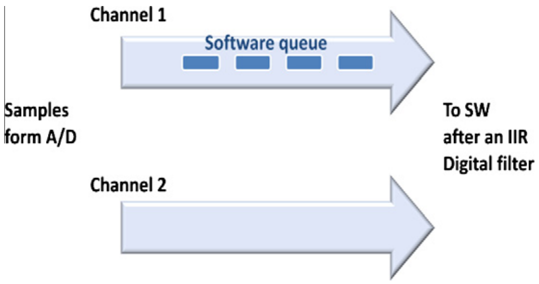

Fig. 2. FIFO queue simplified scheme.

After the selection of an adequate delay, normally only one sample, a little residual phase error between channels still remains and cannot be eliminated by means of a discrete delay. In fact, with this simple method we can only delay a channel of an nT_s quantity, n being a positive integer. Real systems show residual delays of some fraction of degree. These are not easy to compensate and are probably due to internal amplifier and tolerance of reactive components, plus parasitic PCB. Moreover, we are dealing with discrete time signals, i.e. with sequence of samples, and not with relationship for which we can change the phase angles simply changing the value of some variables. The idea was then to modify the sequence of one channel, expressing it as the sum of two components, one sinusoidal and the other co-sinusoidal, by means of the following standard relationship:

$$\sin(\omega t + \varphi) = \cos(\varphi)\sin(\omega t) + \sin(\varphi)\cos(\omega t) \tag{6}$$

Our discrete sampled signal (say V_k) can be written as:

$$\begin{aligned} V_k &= A\sin(\omega t_k + \varphi) \\ &= A[\cos(\varphi)\sin(\omega t_k) + \sin(\varphi)\cos(\omega t_k)] \end{aligned} \tag{7}$$

where "*A*" is the peak amplitude of the signal. To obtain from a sinusoidal waveform a co-sinusoidal one, that is, express a sinusoidal signal as the sum of a sinusoidal and co-sinusoidal component, we can apply a Hilbert transformation, in our case implemented by means of a FIR filter computed as indicated in [5]. So, expressing a discrete time signal as the sum of two discrete signals as in (7) we can change its phase arbitrarily and continuously simply varying the values of the angle φ. In fact, if we consider the data represented in computer memory as in a vector, say *data*[*i*] with $i = 1\ldots N$, N is the number of samples, assuming it as a sinusoidal signal, and applying to it the Hilbert transformation, we obtain a new set of data, say *datacos*[*i*], that is the co-sinusoidal representation of the same signal. Then, we can write the following expression, using (7), for true discrete signals:

$$V_k(\varphi, k) = \cos(\varphi)data[k] + \sin(\varphi)datacos[k] \tag{8}$$

in which it is rather clear that if $\varphi = 0$ the signal is identical to the original one (*data*[*k*]), otherwise it would be again the same signal, but with a phase angle of φ with respect to the original, being φ a continuous variable. In this way we can implement easily a mechanism that compensate with a good level of accuracy the phase difference between the two channels, as indicated in Fig. 3. To do this, we simply apply the same signal to both channels and computing the phase difference between channels. Then, expressing the signals as in (8) we have realized automatic routines that compute the necessary φ angle to reduce at minimum level the phase angle between channels, subtracting (or adding) the computed φ corrective-angle to one channel, each time we start a new set of measurements.

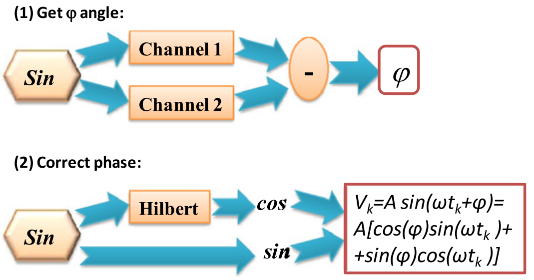

Fig. 3. The scheme used to compute the residual phase shift between channels (1) and the method used to fix it (2).

4.3. Amplitude error

With similar consideration as in the previous paragraph, we can observe that the amplitude of the signals applied on the two channels (supposing to test the acquisition board with the same signal on both channels) is slightly different. While evaluating the impedance by means of Eq. (4), this generates an error of the same relative amplitude of the phase error (20% or more). To compensate the different gain between channels, and with the reasonable assumption that the amplitude error varies linearly along the full range of amplitude, we can apply the following relationship to compute the corrective factor:

$$A_c = V_{ppch1}/V_{ppch2} \tag{9}$$

where A_c is the corrective factor, V_{ppch1} and V_{ppch2} the peak to peak amplitude (in Volt) of the two signals.

5. Visual Analyser

The developed low cost real time FFT-based impedance meter is not simply a standalone instrument. In fact it is now part of a complex freeware solution software tool developed by the authors and recommended as a sophisticated virtual measurements laboratory for researchers and students. It is called "Visual Analyser" [1] (VA from now on), free-of-charge, downloadable from the website http://www.sillanumsoft.org, based on a standard commercial multi-channel data acquisition board or home-made hardware as interface with the external world. VA exploits the power of modern PC's to achieve very interesting performances, including a large set of instruments such as a spectrum analyser, a waveform generator, an oscilloscope, a frequency meter, a true RMS voltmeter, and many other signal processing tools. VA is a tool completely developed by the authors in C++, without using predefined library, fully object oriented and multi-threaded. It is composed by over 300,000 lines of code and is constantly growing.

To get real time performance, the software architecture is based on seven prioritized threads in order to achieve a high level of parallelism (for instance allowing all the simulated instruments to run simultaneously). The basic idea was to use a thread to read a buffer of samples filled by the acquisition board hardware. That is, a thread waits at a semaphore up to the completion of the buffer. The buffer dimension is user-defined, and depends upon the capabilities of the PC hardware and/or from the trade-off between speed of execution and frequency resolution. The time between two consecutive buffers is used by the virtual instruments to compute their function (for example apply FFT and display the spectrum).

To implement the impedance meter in VA (see Fig. 4) we have first reproduced, as a reference, the results of other authors [2,3] whose work implements real time LMS impedance meters, an algorithm already described in a previous paragraph.

We then developed our proposed solution in order to achieve better performances and a higher level of measurements per second; the latter is also useful for dynamic characterization of transducers and sensors, during their real time application [6,7]. The number of measurements per second is necessarily a trade-off between accuracy and speed of execution. Each buffer is acquired according to the sampling frequency; for example with a frequency sampling of 40,960 Hz and a buffer of 4096 points, we get 40,960/4096 = 10 buffer per second, while with 40,960 points we get one buffer per second. That is, in the first case we get a higher number of measurements but lower accuracy, vice versa for the second example, due to the different number of points used by the FFT algorithm.

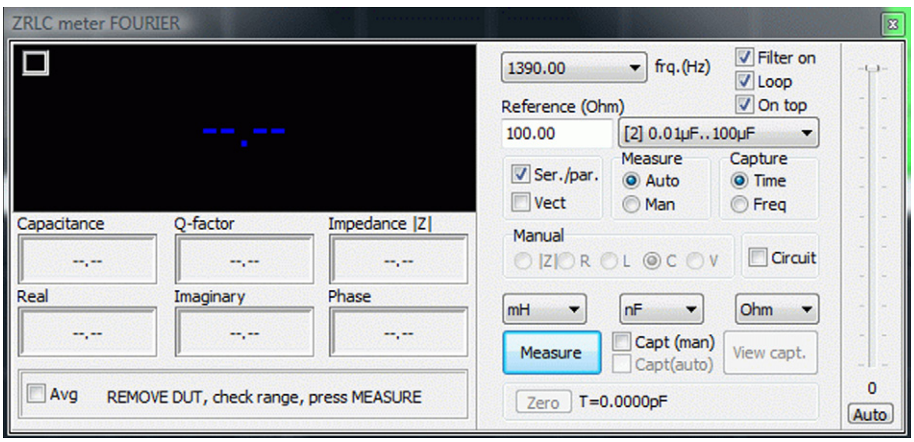

Fig. 4. The ZRLC meter window of Visual Analyser.

6. Hardware considerations

The hardware used by some authors comprises specific PC-acquisition board (for example from the National Instruments acquisition board family) and *LabView* environment, allowing easy writing of virtual instrumentation software. Other authors use even a standard PC soundcard as acquisition device plus proprietary software [3]. This is because a standard PC soundcard is actually a true acquisition board, since it is provided with a double 16 bit A/D and D/A converter (usually two input and output channel) plus a DSP and an internal hardware mixer (interfaced from the *Windows Mixer Utility* or with self-developed software by means of a compiler and direct API calls). The input channels are the line-in and mic-in connectors and the outputs are the speaker out connectors. Unfortunately a soundcard is not provided by the manufacturer of a complete metrological characterization; on the contrary a professional device is normally characterized by means of uncertainty parameters (for example "*B*" type) [8], allowing the user to evaluate the uncertainty of measured values. Considering the costs of professional hardware and the uncertainty determination problems using a standard soundcard (or low cost off the shelf acquisition board), our solution is to project and build a home-made hardware, leaving the computation of the uncertainty parameters for a future development of our measurement system. Nevertheless, we tested our system comparing it with commercial instruments (Section 7) with known uncertainty.

Table 1
Measured values.

Component	*Agilent* 34405A *Digimess* RLC200	VA (10/3)
100 pF	0.102 nF	99.7/ 99.8 pF
1000 pF	1.03 nF	1.0090/1.0080 nF
10 nF	10.1 nF	10.223/10.124 nF
100 nF	0.100 μF	99.66/99.90 nF
220 nF	0.221 μF	218.73/220.18 nF
1 μF	1.02 μF	1.0288/1.0252 μF
100 μF	101 μF	100.280/100.100 μF
1000 μF	1005 μF	990.33/998.29 μF
10 μH	9.96 μH	9.967/10.012 μH
100 μH	97.1 μH	97.22/97.12 μH
1 mH	997 μH	996.90/990.50 μH
4.7 mH	4.67 mH	4.652/4.701 mH
47 mH	47.2 mH	47.12/47.01 mH
10 Ω	10.016 Ω	9.876/10.010 Ω
100 Ω	102.433 Ω	103.876/102.920 Ω
1000 Ω	1.0030 kΩ	999.56/999.36 Ω
10 kΩ	10.015 kΩ	10.021/10.017 kΩ
100 kΩ	0.10086 MΩ	100.018/100.798 kΩ
1 MΩ	1.0059 MΩ	1.0010/1.0038 MΩ

Table 2
System parameters.

Frequency sampling	81,920 Hz
Samples per buffer	8192/16,384 points
A/D	16 bit
Test frequency	1000 Hz

7. Results

We completely tested our impedance meter to verify the correctness of the results, comparing them with the ones provided by commercial meters with a known uncertainty, supported by calibration documents (Tables 1 and 2). The instruments we selected as references were the *Agilent digital multimeter* 34405*A*, 5.5 *digit*, capable to measure resistance and capacitance, and the *Digimess RLC*200 *impedance meter*, 3.5 *digit* for the inductance measurements. Also the value of the reference resistor (Fig. 1) was measured again with the *Agilent digital multimeter* (the accuracy of this measurement is crucial for the global accuracy of the impedance meter). As previously mentioned in Section 4, the hardware we used for testing VA was a home-made prototype of an acquisition board using

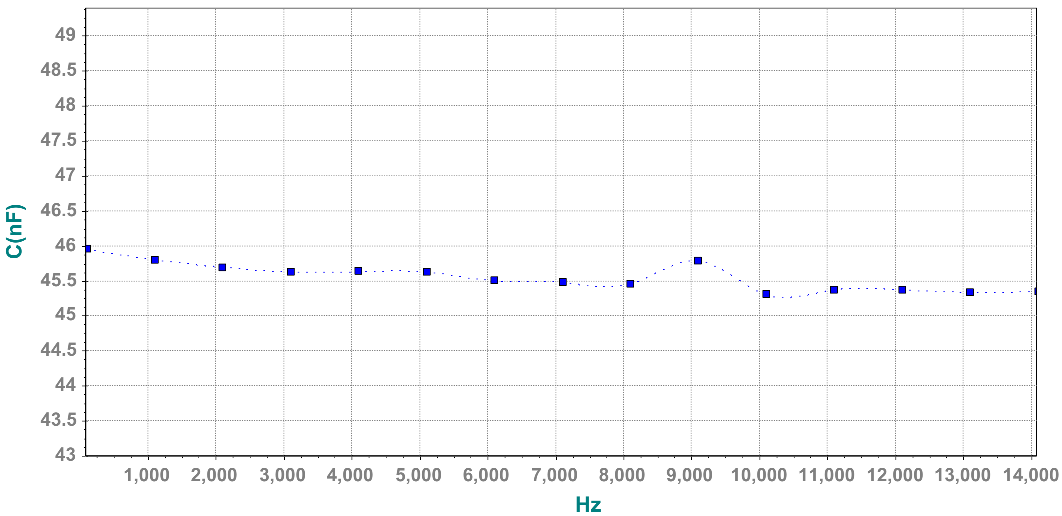

Fig. 5. Measurement of a capacitor (47 nF) using a measurement frequency sweep from 100 Hz to 14 kHz exploiting an advanced feature of VA; each data point is the average of five measurements.

a standard PCM2902 plus a double operational amplifier LM358 to get an high input impedance. It is rather clear that the cost of the hardware used is very low, thus demonstrating that it is possible to reach good performances even in these conditions. The software routines were written in C++, and a new window was added to the VA.

In the first column of Table 1, the type of components used and their values as from the manufacturers are reported; in the second column are the measured resistor and capacitor values obtained by the *Agilent digital multimeter*, and with *Digimess impedance meter* for inductances; the third column is for the values measured with VA, respectively for ten and three measurements per second. The parameters selected for the system are reported in Table 2. The PC used was a core 2 duo T9300 2.50 GHz with 4 Gb of RAM and Windows Seven Ultimate edition.

As further test, we present an example of automatic measurement of a capacitance (Fig. 5). We tested a standard 47 nF capacitor, using a feature of VA that allows to repeat the measurement incrementing the test frequency with a user-defined step. Moreover, for each test frequency it is possible to average a predefined number of measurements (five in this simple case).

8. Conclusions

We propose a novel system capable of very good performance in measuring impedance values, based on a very low cost set-up. The method for reducing the significant sources of uncertainty is presented and implemented in a PC-based software.

The use of software written in pure C++, freely downloadable, and not dependent from any software package producer (for example *LabView*) is another important advantage of our work; moreover the possibility to set-up an automatic measurement bench with data log (for example automatically repeating the measure at different test frequencies) should be of interest for many research laboratories.

All the previous considerations and realizations can be the support for further development on uncertainty evaluations possible by means of the so called "*A*" and "*B*" type [8]. In particular, computing the measurement uncertainty by means of the A/D and D/A converter "*B*" type parameter (gain error, noise, offset, quantization, time-jitter, integral and differential linearity) provided by the manufactures and also the possibility to computes the "*A*" type uncertainty by means of a real time statistical analysis; then it calculates the global uncertainty ("*A*" plus "*B*") by applying the law of propagation of the uncertainty [8,9] and a proper coverage factor. It should also be possible to confirm the results with an embedded Monte Carlo analysis [10]. Notice that, as stated in [1], the uncertainty introduced by the algorithms (rounding error) is negligible because of the 80 bit floating point variables used everywhere in VA, and so we should concentrate only upon the hardware acquisition board.

References

[1] A. Accattatis, M. Salmeri, G. Rabottino, A. Mencattini, R. Lojacono, Visual analyser: a sophisticated virtual measurements laboratory for students, Sixteenth IMEKO TC4 Symposium, Florence, Italy, 22–24 September 2008.
[2] M. Dutta, A. Rakshit, S.N. Bhattacharyya, Development and study of an automatic AC bridge for impedance measurement, IEEE Transactions on Instrumentation and Measurement 50 (5) (2001).
[3] G. Steber, An LMS impedance bridge, QEX September/October 2005.
[4] B. Widrow, S.D. Stearns, Adaptive Signal Processing, Prentice Hall, 5 March 1985.
[5] Rashid Ansari, IIR discrete-time Hilbert transformers, IEEE Transactions on Acoustic, Speech, Signal Processing, vol. ASSP-35(8), August 1987.
[6] G. Orengo, L. Giovannini, G. Latessa, G. Saggio, F. Giannini, "Characterization of piezoresistive sensors for goniometric glove in hand prostheses" Wireless Vitae09 – wireless communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, Aalborg (Denmark), 17–20 May 2009, pp. 684–687.
[7] G. Saggio, G. Latessa, S. Bocchetti, F. Brunetti, A. Reale, F. Giannini, "Sensor's design and characterization for biomedical applications" ISSBB, Società Italiana di Biomedicina e Biotecnologia Spaziale, in: Fourth Congresso Nazionale: "Un mondo senza gravità", Santa Margherita Ligure, 31 Marzo – 2 Aprile 2009.
[8] GUM, Guide to the expression of uncertainty in measurement, UNI CEI ENV, July 2000.
[9] G. Betta, C. Liguori, A. Pietrosanto, Propagation of uncertainty in a discrete Fourier transform algorithm, Elsevier Measurement 27 (2000) 231–239.
[10] Evaluation of measurement data – Supplement 1 to the GUM – Propagation of distributions using a Monte Carlo method, 2008.