16th IMEKO TC4 Symposium
Exploring New Frontiers of Instrumentation and Methods for Electrical and Electronic Measurements
Sept. 22–24, 2008, Florence, Italy

Visual Analyser: a Sophisticated Virtual Measurements Laboratory for Students

Alfredo Accattatis, Marcello Salmeri, Arianna Mencattini,
Giulia Rabottino, Roberto Lojacono

Department of Electronic Engineering, University of Rome "Tor Vergata"
Viale del Politecnico, 1 – 00133 – Roma, Italy
Phone: +39–0672597373, Email: {accattatis,salmeri,mencattini,rabottino,lojacono}@ing.uniroma2.it

Abstract – The paper presents Visual Analyser, a free software tool developed by the research unit and suggested for use as a sophisticated virtual measurements educational laboratory for students. It can use either the sound card provided with all PC's or dedicated hardware as the interface with the external world, and exploits the power of modern PC's to achieve great performance. Visual Analyser includes a large set of instruments including a spectrum analyzer, a waveform generator, an oscilloscope and many other signal processing tools. For its special features Visual Analyser has been adopted by many professional and academic laboratories over the world.

I. Introduction

Due to the cost of devices and software, the setup of a complete didactic laboratory for electronic measurements can be a very expensive undertaking. While the availability of diverse measurement instrumentation is helpful to students, very high–level performance (for example, in the area of frequency response) is rarely needed, and therefore constitutes a waste of resources. Furthermore, it is beneficial for students to be able to use the same instruments at home, as they use in the University laboratory.

The idea could be realized using low cost and ubiquitous hardware, for example a standard Windows–based PC and a sound card as data acquisition board. The only additional hardware could be a standard oscilloscope probe and a simple protection circuit to guard against excessive voltage. The newest soundcards with a 192 kHz sample rate will manage signals with frequencies up to 96 kHz (which is well beyond the audio frequencies). Thus, a suitable program exploiting the power of modern PC's is able to simulate a lot of measurement instruments. The idea of developing such a program, called "Visual Analyser" (VA from now on), was born some years ago [1] in spite of the presence of several other similar solutions in the market, such as: SpectraPlus, from Pioneer Hill Software (formerly SoundTechnology) [2], Real Time Analyzer (RTA), from YMEC software [3], Virtins Soundcard Multi Instruments, from Virtins Technology [4]; and a lot of minor ones such as: Zelscope, Oscillometer, Spectrogram, etc. There are indeed other VA–like programs, but they are shipped with dedicated hardware (such as Vellemann 8220 PCS500 with Pc–Lab2000 software [5] and PoScope [6]), thus they won't be compared with VA.

Although the previously cited programs implement a lot of useful functions, they often lack in many features indispensable for both educational and researching purposes. Therefore, a program self developed can be modified in order to fit the requirements when they occur.

II. Scope of Visual Analyser

We have been developing VA for several different purposes, such as:

1. implement a complete virtual measurement laboratory for students;
2. research activities, involving signal acquisition, elaboration and synthesis;
3. demonstrate, during lessons, concepts like FFT, digital filtering, Nyquist theorem, Cepstrum analysis, cross–correlation, signal synthesis, distortion, and aliasing;
4. applying and test the valuation of uncertainty, as the algorithms used in VA are the same as that used for real instruments;
5. applying and testing various software multi–threading strategies to obtain better real–time performance.

The availability of complete source code allows dynamic adaptation of the program to suit the stated goals (1), (2), (3), and (4). For instance, "Electrical Impedance Spectroscopy" involves the generation and analysis of signals in a range of 20 Hz – 50 kHz. It has been possible to quickly adapt VA to such measurements. The ECG (or EEG) signal analysis is also an interesting field to which VA could be quickly adapted, for example, to compute in real time the spectrum of ECG – RR variations.
As an example of purpose (3), VA has the capability of performing a full real time Digital–Analogue conversion in the oscilloscope function, although sometimes it is not well understood. In other words, we developed an optimized thread able to reconstruct the digitized signal using the Nyquist Theorem. This means that in the "scope window" the signal will always be continuous and well suited to demonstrating the sampling theorem. In fact it is possible to turn the D/A conversion of the signal ON and OFF in real time (during the data acquisition), thus clearly showing the difference between the sampled signal and the reconstructed signal. Furthermore, it is possible, using two channels, to show both the signal in contemporary (sampled) and "original" or "reconstructed" forms, improving the quality of demonstration.
A student could make a lot of different measurements using VA, a probe and a simple protection circuit. For example, VA can determine the frequency response of an audio device. It is possible to generate white noise, inject it into the device, compute the input and output spectrum, and calculate the transfer function in real time, displaying the signal in the frequency and time domains and simultaneously generating the waveform and computing the transfer function.
Another example could be the generation of a carrier with amplitude modulation: the student can demodulate the signal and separate the carrier from the original signal, strictly in real time.
The student could also synthesize a waveform using the tool embedded in VA (Visual Tool), generate and display it simultaneously, and optionally apply various predefined real–time filters (pass–band, low–pass, high–pass, etc.) allowing a true understanding of the spectral analysis and filtering. These are but a few examples of the many measurements that can easily be conducted without the need of expensive professional instruments, at the University laboratories or/and at home.
The calculus of uncertainty (4), although topic of research for many years [9–13], is totally absent from the other programs cited above.
Finally, purpose (5) supports significant optimization of simultaneous multiple measuring instrument use, allowing an easy test of different multi–tasking strategies on a PC rather than using the real instrument.

II. Features of Visual Analyser

VA is a real time software which simulates a set of electronic instruments such as:

- Oscilloscope (dual channel, xy, time division, trigger, Nyquist conversion);
- Spectrum Analyzer with amplitude and phase display (linear, log, lines, bar, octaves band analysis 1/3, 1/6, 1/9, 1/12, 1/24) by means of FFT algorithm;
- Wave–form generator with "custom functions", triangular, square, sinus, white noise (Gaussian and uniform distribution), pink noise and pulse generation (all without aliasing, when applicable);
- Frequency meter (in time and frequency domain) and counter with resolution and simple uncertainty estimation;
- Volt meter with true RMS, peak to peak and mean display;
- Filtering (low pass, high pass, band pass, band reject, notch, "diode", DC removal);
- "Capture" window for analysis and file saving of time, spectrum and phase displays with "triggering" events (level, duration);
- A true "software" digital–analog conversion (for complete signal reconstruction);
- Frequency compensation; you can create/edit a custom frequency response and apply it to the acquired spectrum (for example to compensate non–linearity of a specific device);
- True support for 8/16/24 bit soundcard;
- Sample rate limited only by the capabilities of the available soundcard and/or dedicated device [8];
- THD measurement with simple bias compensation algorithm;
- Cepstrum analysis;
- Cross–correlation, auto–correlation.

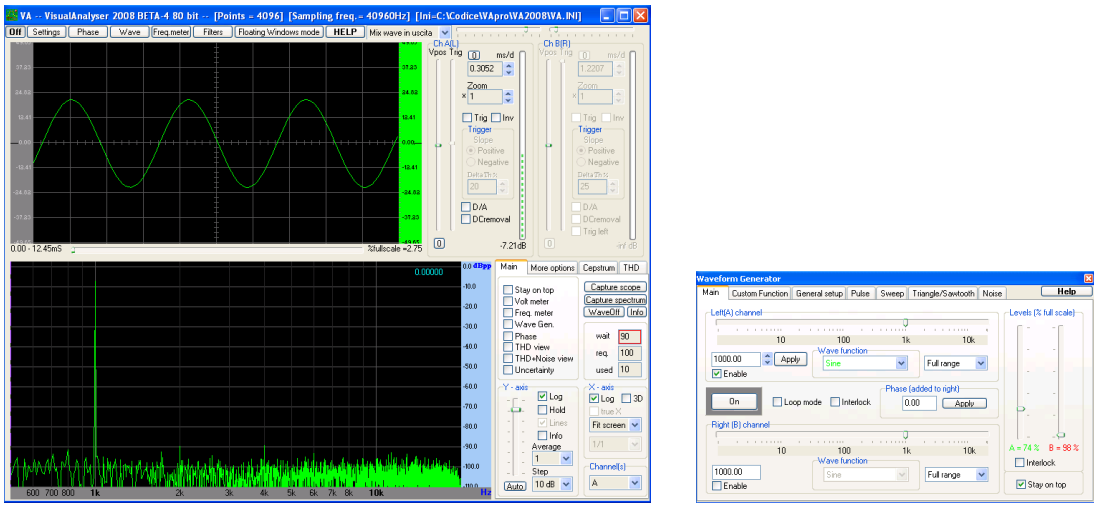

Figure 1. The main window of VA plus wave generator window.

One can notice that many of the listed features are not present in other solutions in the market. Table 1 compares some characteristics of VA with those present in above cited programs.

	FFT	Oscilloscope	Wave generation	Custom wave generation	Frequency meter	D/A conversion	Cepstrum analysis	THD+ Anti bias	Filtering	Phase	Floating Windows	Capture	Uncertainty computation
VA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Spectraplus	✓	✓	✓	≈	≈			≈		✓		✓	
RTA	✓	✓	✓	✓	✓	✓	✓	✓		✓			
Virtins	✓	✓	✓					≈					
ZelScope	✓	✓								✓		✓	
Oscillometer	✓				≈			✓			✓		

Table 1 – Comparison of features of similar software. The ✓ symbol indicates the full presence of the feature, while ≈ indicates its partial presence.

Moreover, despite its appearance, a great merit of VA is its simplicity in using confirmed by thousand users. VA auto–detects the characteristics of the acquisition device, and, after the first start, it is ready to use, simply by clicking the ON button. The main window includes the oscilloscope and spectrum analyzer displays, with standard parameters predefined. One can easily select input source by means of a listbox, and in real time one can adjust the input level by means of a simple scrollbar.
Moreover, one can enable all the other virtual instruments, by clicking the corresponding button on the main bar to activate the waveform generator, frequency meter, AC voltmeter, phase window and filters, plus THD and uncertainty calculus for the FFT algorithm. From the main window, one can quickly modify the Spectrum Analyzer parameters (frequency spanning, log and linear plot, zoom and much more) and that of the oscilloscope (time division, trigger, vertical position, horizontal position). At all times one can select the channel to display (left and/or right) and many other features (as cross–correlation calculus, cepstrum, octave band analysis, etc.) can be selected.

III. Visual Analyser software architecture

Visual Analyser is a Windows based software developed in C++ Builder 2007 (Borland RAD Developer Studio). It is a fully Object Oriented software, real–time and Multi–threaded, and it is composed from over 200.000 lines of code. VA can run also in a Linux based system, using the Wine

library [7]. It has been tested with Ubuntu, Mandriva and Suse Linux distributions with a negligible loss of speed. A new library based on Wine can be used (CodeWeavers crossover Linux 6) [14].
To get real time performance, we developed a software architecture based on seven prioritized threads in order to achieve a high parallelism level (for instance allowing all the simulated instruments to run simultaneously). They are: (1) main VCL thread (user interface thread) (2) thread to acquire samples from the soundcard (3) two threads for D/A conversion (4) thread for the frequency meter (5) thread for the waveform generator (6) thread for D/A conversion in capture window . The basic idea is to use thread (2) to read a buffer of samples that is a power of two in length (e.g. 1024, 2048, 4096, etc.) and use an FFT algorithm to compute the spectrum. The buffer is filled by the sound card hardware. That is, thread (2) waits over 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 default value is 4096 points (16 bit). If we use (for simplicity) a sampling rate of 40960 Hz, we can fill a buffer each 100 ms. A Pentium IV with a clock of 1.6 GHz is able to compute the FFT, as well as to handle the graphical displaying of the oscilloscope and modulus of the spectrum itself, within about 10 ms. In other words, after a buffer has been filled, we need 10 ms to compute the spectrum, and the remaining time (90 ms) is available to compute other useful functions (for example to execute threads 3, 4, 5, 6) while waiting for the next buffer to be filled. By using a bigger buffer dimension, we can improve the frequency resolution; the upper limit depends from the speed of the hardware and operating system. Thread (4) has been created to implement a frequency meter–computing an "extended" FFT by means of zero–padding (up to 20 times the dimension of the real time buffer) to obtain better resolution. It runs in the background with a refresh rate ranging from real time to a few seconds. For instance, a buffer of 4096 samples and a frequency sample rate of 40960 Hz provides a resolution of 10 Hz. By zero padding the buffer to add 4096 x 9 zeros, we obtain 40960 overall points, providing 1 Hz resolution, but with an execution time unsuitable for computation in a single cycle.

IV. Uncertainty estimation and metrological characterization

The overall uncertainty depends essentially on three contributions. The first is related to the acquisition board (i.e. the sound card of the PC); the second from the microprocessor finite word length; the third form uncertainty on input data itself (non considered in this paper).

The first can be derived from the metrological characterization of the board itself. The problem is that a standard PC soundcard is often not characterized metrologically by the manufacturer. For this purpose, we tried to characterize it at least for with regard to the *quantization* and *jitter* error [15]. First of all, we are considering "synchronous sampling", though we are working on a more general version which also includes also asynchronous sampling. Moreover, in order to evaluate the magnitude of the quantization error introduced by a standard ADC, VA can use the standard relationship

$$U_q = V_r \, 2^{-B} \big/ \sqrt{12} \,, \tag{1}$$

where U_q is the standard deviation, B the actual bit number and V_r the input range accepted by the real ADC. This uncertainty has been modelled as uniform random variable zero mean and standard deviation U_q. Then the uncertainty propagates in FFT algorithms as clearly described in [15], and we used the relationship (29) in [15] to estimate it.
Another uncertainty source is the jitter, due to the fact that the sample period might not be a constant parameter. The cause of this behaviour may be noise on the sampling impulse, uncertainty on the Sample and Hold aperture time and clock instability. We can model this phenomenon adding to each ideal sampling instant t_n a random variable uniformly distributed in the interval $-J_\tau, +J_\tau$ with zero mean and variance $J_\tau/3$. This means that each n^{th} sampled value has to be considered as a random variable, and represented as the sum of the ideal value and a zero mean uniformly distributed random variable α_n. In other words, each sampled value is affected by an uncertainty equal to the standard deviation of the variable α_n. If we approximate the original signal *x(t)* in the interval $n\tau - J_\tau, n\tau + J_\tau$ with a linear function the α_n range can be estimated as equal to $-a_n n\tau - J_\tau, a_n + J_\tau$ and finally its standard deviation is

$$U_{j_n} = a_n J_\tau / \sqrt{3}\,, \tag{2}$$

where a_n is the first derivative of *x(t)* in the considered point.

The second contribution is due to the calculus of FFT with a microprocessor, that is, with *rounding* errors due to the microprocessor finite word length [15]. For many other algorithms and functions (FIR/IIR filters, Frequency meter, Voltmeter, THD–meter, Cepstrum, Windowing) the estimation is still under construction. The uncertainty on the modulus of the harmonic has been computed, assuming the real and imaginary part of sampled values are uncorrelated with respect to this kind of uncertainty, by means of the following equation for the multiplication (for the phase and for further details, see (29)–(32) in [15]):

$$U^2_{M\,k}\Big|_m = \frac{1}{N^4 M^2\ k}\left(R^2(k)\sum_{m=0}^{N-1}(x(m)\cos(k\beta_m))^2 U^2_{fl\_m} + I^2(k)\sum_{m=0}^{N-1}(x(m)\sin(k\beta_m))^2 U^2_{fl\_m}\right). \tag{3}$$

Similar equations could be derived for the addition term (see [15] for further details).
The $U^2_{M\,k}\Big|_m$ is the standard deviation of the amplitude of the k^{th} *harmonic*, related to all the multiplications executed in the FFT algorithm; conversely $U^2_{M\,k}\Big|_a$ will be the standard deviation of the addition executed in the same algorithm. The overall uncertainty has been computed as:

$$U = \sqrt{U^2_{M\,k}\Big|_m + U^2_{M\,k}\Big|_a} \tag{4}$$

Because the computations have been made using double–precision numbers (80 bit IEEE floating point with 64 bits of mantissa), the uncertainty introduced by microprocessor finite word length is negligible compared with the other ones (see also [9]). In fact, the calculus executed for 80 bit floating points returns the values summarized in Table 2.

A routine inserted in VA allows the estimation of the computation of the three highlighted contributions. The result is obtained by normalizing the standard deviation by the amplitude of the input signal. The test has been conducted using as input a 1000 Hz sinusoidal signal of about 50% of the maximum input amplitude of the selected sound card input, 16 bit quantization, 4096 points buffer and 40960 Hz of sampling frequency.

Quantization	*Jitter*	*Rounding*
$4.866698 \cdot 10^{-6}$ %	$3.189439 \cdot 10^{-3}$ %	$3.151169 \cdot 10^{-17}$ %

Table 2. Rounding and quantization error expressed as percentage of maximum amplitude; jitter as percentage of sampling period.

Other typical parameters that metrologically characterize a standard measurement instrument are difficult to evaluate due to the lack of documentation provided by soundcard manufacturers. This is obviously due to the fact that a standard soundcard is not normally used to make measurements. On the contrary, using professional hardware, complete documentation is normally provided. Nevertheless, the scope of VA is to be a cost–effective virtual instrumentation for students, and we are not considering the use of VA in conjunction with professional hardware. Finally, VA is provided with a calibration procedure, that permits a calibration in Volts or dB of the spectrum analyzer and/or the oscilloscope, using a sinusoidal input of known amplitude (RMS or peak to peak). The procedure to calibrate VA is very simple; one selects the desired input to calibrate; connects known calibration signal to the input (it must be a sinusoidal signal of known RMS value, or peak to peak); and clicks the “Start Measure” button. VA will start to acquire the signal automatically and apply calibration. Then you can save the calibration in a disk file, so that you can have VA calibrated for multiple input and different soundcards.

V. Conclusions

The reason to undertake the writing of a complex program like VA is to obtain a cost–effective set of virtual measurements instrumentation with full access to source code. The program has been designed to work with only standard sound card, although dedicated hardware can improve the capabilities of VA with virtually no–limits [8] or can simply increase the safety level against PC damage.
In this way, we can adapt VA to many situations arising in academic work and student laboratory experiments. VA is continuously *under construction* thanks to the huge feedback that the Internet communities send to us. The real–time capabilities of VA, due to the strong optimization of the multi–threaded core, is another good test for software to be developed for real measurement instruments. A simple metrological characterization has been provided, and new sophisticate metrological characterization will be added in the near future. In fact, under construction is a standard Monte Carlo analysis to confirm the uncertainty calculus presented in this work, and also compensate for the lack of metrological characterization of the hardware. A "Running error analysis" is also under test as further confirmation and more accurate estimation of microprocessor finite word length.
The program is distributed over the Internet with a freeware license, and it is possible to download it from the SIMPlify.it official site (www.simplify.it/va and www.sillanumsoft.org).

Acknowledgements

The authors would like to thank Scott Willing for his precious support in correcting the paper and for his availability and competence.

References

[1] A. Accattatis, Master Thesis, "Sviluppo di uno strumento virtuale real–time per la generazione analisi ed acquisizione dei segnali".
[2] Web site: http://www.spectraplus.com.
[3] Web site: http://www.ymec.com/products/dssf3e/.
[4] Web site: http://www.virtins.com.
[5] Web site: http://www.velleman.be.
[6] Web site: http://www.poscope.com.
[7] Web site: http://www.winehq.org.
[8] *Nuova Elettronica*, n. 232 and n. 233, September / October 2007 and November / December 2007.
[9] S. Caldara, S. Nuccio, C. Spataro, "Measurement uncertainty estimation of a virtual instrument", *Instrumentation and Measurement Technology Conference*, Baltimore, USA.
[10] H. Haitjema, B. Van Dorp, M. Morel, P. H. J. Schellekens, "Uncertainty estimation by the concept of virtual instruments", *Proceedings of SPIE*, 2001.
[11] D. A. Lampasi, L. Podestà, "A Practical Approach to Evaluate the Measurement Uncertainty of Virtual Instruments", *Instrumentation and Measurement Technology Conference*, Como, Italy, May 2004.
[12] E. Ghiani, N. Locci, C. Muscas, "Auto–Evaluation of the Uncertainty in Virtual Instruments", *IEEE Trans. on Instrumentation and Measurement*, vol. 53, n. 3, June 2004.
[13] M. J. Korczynski, A. Hetman, "A Calculation of Uncertainties in Virtual Instrument", *Instrumentation and Measurement Technology Conference*, Ottawa, Canada, May 2005.
[14] Web site: http://www.codeweavers.com.
[15] G. Betta, C. Liguori, A. Pietrosanto, "Propagation of uncertainty in a discrete Fourier transform algorithm", *Measurement*, vol. 27, pp. 231–239, 2000.
[16] R.I. Becker, N. Morrison, "The errors in FFT estimation of the Fourier transform", *IEEE Trans. Signal Processing*, vol. 44, n. 8, pp. 2073–2077, 1996.
[17] A. V. Oppenheim, R. W. Schafer, "Discrete–time signal processing", Prentice Hall.