

Software and Hardware to Replace a Sound Level Meter

R. Navarrete, M. F. Rocha, L. Salvador V., L. Orozco, C. A. Calva, M. R. Osnaya, M. I. Rocha G., E. Andrade, and C. Solis

Abstract—In the field of acoustics, noise investigations involving regulated measurements, this requires sophisticated and expensive equipment and obtaining data sets for subsequent analysis becomes very laborious and complicated. We used a virtual instrument computer (PC) and appropriate software to simulate a sound level meter. This becomes a huge advantage in the design, cost and versatility of the instrument, since its functions and parameters can be easily redefined by changes in the programmed instructions. Furthermore, and this method avoids a redesign of a physical circuit. Virtual instrumentation is an indispensable tool to study, measure and evaluate complex conditions, such as acoustic spaces. In this work, the design and implementation of a sound level pressure meter in a virtual instrumentation scheme is presented. The system was implemented with Matlab and instrumentation software. The implemented noise measurement system presents a friendly interface to the user via a GUI and reduces time and costs in the determination of environmental noise measurements quite reliably.

Index Terms—Noise pollution, virtual instrumentation, SLM_AC

I. INTRODUCTION

The measurements and assessments of noise require sophisticated and expensive equipment that can be replaced by virtual instrumentation applications developed according to specific needs.

In large cities, then noise is a serious problem of pollution, called environmental noise pollution that affects human physiology and security in our daily lives.

The cost of professional equipment for the assessment of environmental noise (Sonometro or SLM), which is a device for measuring sound pressure level, can rise to thousands of dollars and that is, most of the time, outside the budget of people interested in making a noise study. There are other lower cost options available in the market, but sacrifice accuracy and precision. Fortunately today the modern student can make use of software, hardware and PC, for studying this field at very low cost.

A virtual instrument is a software application that simulates the functional aspects of the real instrument, based on a PC, in this case. The environment, the design of an

instrumentation system is to make a graphical representation of the physical instrument, designing control flags (buttons) to the logical to physical instrument.

There is now software for implementing specialized programs in this field, such as: LabVIEW; Agilent-VEE (antes HP-VEE); Cyber Tools; Beta Instruments Manager; LabWindows/CVI; HPVEE; and MATLAB. These programs are capable of supporting the 3 aspects or functions of a conventional instrumentation system: 1. Data Acquisition (sound card of the PC, data acquisition Toolbox) 2. Analysis of signals (PDS Toolbox) 3. User interface GUIDE).

This paper presents a study and evaluation of environmental noise through quick, practical and acceptable results.

II. MATERIALS AND METHODS

The system that was used as virtual instrument computer (PC) and software to simulate a sound level meter has the following functions:

a) The sound level meter (sonometro) or SLM_AC (Sound Level Meter and C-weighted), it is an instrumentation system virtual that measure the sound pressure. Its general features considered for the design are: 1) Measurement of sound pressure level, SPL. 2) Weighting A, C, flat or linear. 3) Fast and slow time weighting. 4) Meter Class 2 or 3.

b) A measurement system performs three basic functions: 1. Signal acquisition; 2. The processing data; and 3.- the presentation or transmission of data. That in general is achieved by the *hardware* of data acquisition, signal conditioning, computer and instrumentation *software*. Fig. 1, shows the elements of a measurement system.

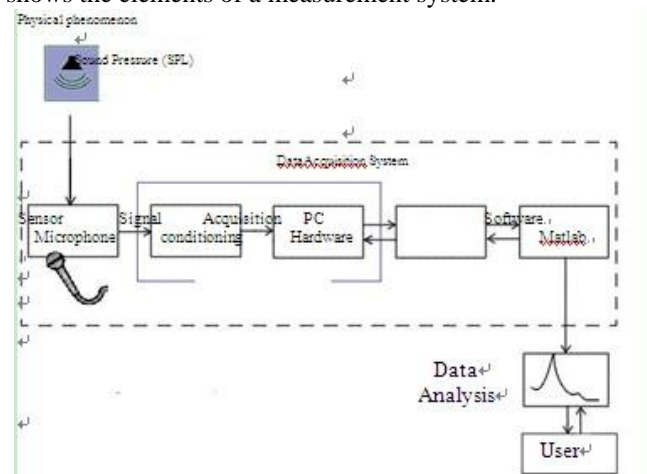


Fig. 1. Diagram of the measurement of sound pressure on a PC.

The functions and virtual design SLM_AC measurement system were carried out in Windows Vista.

A sensor type was selected according to the acquisition card, the PC hardware and software requirements of the

Manuscript received July 21, 2012; revised September 2, 2012. This work was supported in part by the Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica and PIFI of Mexico City.

R. Navarrete, M. F. Rocha, L. Salvador V., L. Orozco, C. A. Calva, and M. R. Osnaya are with the Escuela Superior de Ingeniería Mecánica y Eléctrica, Instituto Politécnico Nacional, México (e-mail: mrocha@ipn.mx).

M. I. Rocha G. is with the G. F. O., Departamento de Ingeniería Electrónica, Universidad Politécnica. Valencia, España (e-mail: marocga@doctor.upv.es).

E. Andrade and C. Solis are with the Department of Physics, Universidad Nacional Autónoma de México (e-mail: andrade@fisica.unam.mx).

signal to be measured.

A sound pressure signal with a frequency range of 20 to 20000 Hz was measured. The sampling rate according to the Nyquist, criterion must be at least 40 kHz. Therefore, a sampling rate of 44.1 kHz was used.

A notebook with Windows Vista Home Premium operating system, AMD Turion 64 X2 Mobile Technology 2 GHz processor, 3 GB RAM, 32 bits MATLAB instrumentation Software, were employed.

A variable sound pressure was measured with a SLM. A microphone was used as sensor; the sound signal is generated and converted into an electrical signal (analog) through the microphone. Different types of microphone were used, with features such as precision, accuracy and sensitivity. The selected microphones depended on the accuracy and correctness. A free field condenser microphone was selected, for its flat response in the frequency range 20 to 20000 Hz. In addition, a sound card installed in the computer was used.

The steps and the set of instructions for performing the system functions in MATLAB, were: 1. Creating an object of the device, 2. Configuring of the acquisition channels, 3. Setting the object properties. The sampling rate, the acquisition time and the number of bits for the quantization and coding of the signal Fig. 2 and Fig. 3 were also set. The *SampleRate* property, $F_s = 44.1$ kHz, the *SamplesPerTrigger* property, which defines the number of samples N to acquire depending on the time of acquisition and the *BitsPerSample* ownership, which allows modifying the number of bits that will be quantized and coded.

Diagram of the MATLAB data acquisition.

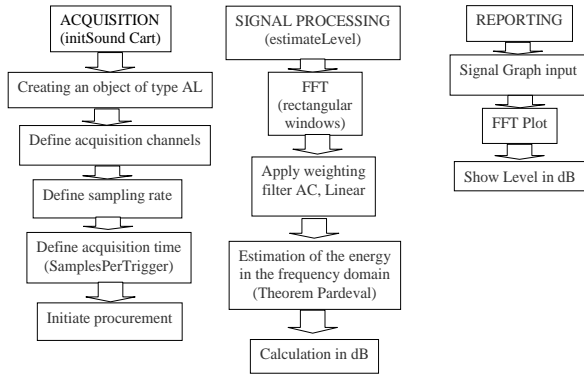


Fig. 2. Schematics of the measurement system functions.

Sample Rate is used to specify the rate of conversion of the analog signal to digital. The input subsystems (AI) and output (AO) analog have a finite number of valid sampling rates. The *propinfo* function (MATLAB) was used, to ensure the selection of sampling rate card which supports the PC

Samples Per Trigger specifies the number of samples to acquire for each analog input channel group member for each trigger that occurs.

Samples Per Trigger function was used to specify the number of samples acquired per channel and for each trigger. The choice is made regarding the slow response or Fast. Radix FFT algorithm-2 was used. The number of samples to process must be power of 2.

Bits Per Sample indicates the number of bits that the sound card uses and the amplitude of the samples. 16 bits were used. Each sample is represented by a number between 0 and \pm

32768.

The model proposed by Douglas R. Lanman was employed for the implementation of the virtual sound level meter. The FFT algorithm was used to obtain the frequency spectrum of a sample window and Parseval's theorem to calculate the average energy of the signal in the frequency domain (Fig. 3).

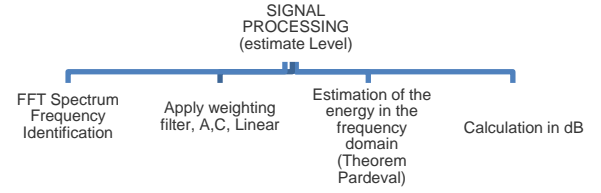


Fig. 3. General diagram of digital signal processing in the SLM.

The frequency spectrum was determined using the Discrete Fourier Transform (DFT). The Function DTFT (MATLAB) is defined by the following expression:

$$X[k] = \sum_{n=0}^{N-1} X[n] e^{\frac{j 2 \pi n k}{N}} \quad (1)$$

where N is the number of samples the *discrete-time Fourier transform* (DTFT) is sampled to convert it into a discrete signal. The above expression produces a vector $X[k]$ expressed in terms of the discrete variable k . From the definition of the DFT is known that each DFT interval is $1/(NT_s)$. To express the vector $X[k]$ in terms of frequency is necessary to determine the frequency of each sample $X[k]$ of the DFT using the equation $\Delta f = 1/NT_s = f_s/N$.

$$k = \left[\frac{N}{2}, \dots, N-1, 0, 1, 2, \dots, \frac{N}{2}-1 \right] \quad (2)$$

$$f_k = \left[-\frac{\frac{N}{2}}{NT_s}, \dots, -\frac{1}{NT_s}, 0, \frac{1}{NT_s}, \frac{2}{NT_s}, \dots, \frac{\frac{N}{2}-1}{NT_s} \right] = k \Delta f$$

f_s is the sampling frequency, N is the length of the input sequence (samples acquired by trigger) and Δf is the frequency resolution.

The use of Radix FFT algorithm-2 for the evaluation of the DFT provides real time acquisition due to the reduction of operations required to calculate the DFT. To implement this algorithm are analyzed rectangular windows of data streams with a length power of 2 ($N=2^m$):

TABLE I: THE NUMBER OF SAMPLES TO BE ACQUIRED (SAMPLES PER TRIGGER)

Type of Respons	Acquisition Time Required	SamplesPerTrigger $N = \text{Time} * F_s$	Nextpow2(N) $\rightarrow 2^h > 2$ power of 2 samples N	Acquisition Time used $= N / F_s$
Slow	1 s	44100	$2^{16} = 65536$	1.486 s
Fast	0.125 s	5513	$2^{13} = 8192$	0.185 s

The number of samples to acquire (*Samples Per Trigger*) according to the type of response is not a power of 2, which is used *nextpow2* MATLAB function, which identifies the number of samples a power of 2, which is necessary to acquire, to apply the algorithm of the FFT Radix-2.

Table I show that the window size for slow response (slow) is 65536 samples, 1.486s, and for the rapid response (fast) is 8192 samples or 0.185s.

The implementation of the weighting filters A and C properties are used $\log AB = \log A + \log B$ and $\log A^n = n \log A$, are:

$$A(f) = \frac{A_1^2 F_4^4 A_2^2 f^8}{(f^2 + F_1^2)^2 (f^2 + F_4^2)^2 (f^2 + F_2^2) (f^2 + F_3^2)} \quad (3)$$

$$C(f) = \frac{A_1 F_4^4 f^4}{(f^2 + F_1^2)^2 (f^2 + F_4^2)^2} \quad (4)$$

where $W_A = 10 \log A(f)$ y $W_C = 10 \log C(f)$

$A_1=1.007152$, $A_2=1.249936$, $F_1=20.598997$ Hz,
 $F_2=107.65265$ Hz, $F_3=737.86223$ Hz, $F_4=12194.217$ Hz

From Equation (2) it is possible to determine the frequencies f_k , corresponding to the magnitudes, $X[k]$, of the FFT. These are used to evaluate the coefficients of the weighting filters using the previous equations.

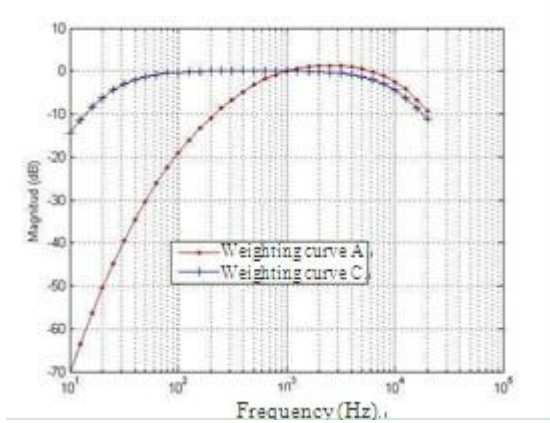


Fig. 4. Weighting filters A y C.

Samples of the FFT weighting filter applied are:

$$\begin{aligned} X_A[k] &= A[f_k] X[k] \\ X_C[k] &= C[f_k] X[k] \quad \text{para } f_k = k \Delta f \end{aligned} \quad (5)$$

The magnitude values in dB which must pass the line weighting curves A and C are presented in Fig. 4. This Fig also displays the values obtained using equations (3) and (4) to draw the curves, by calculating the difference between the values obtained and the reference values according to NOM-AA-59-1978[9]. It is shown that the error is minimal and is due to the implementation of the filters. Although the variation is small, it is important to keep in mind, since it forms part of the total error of the virtual instrument.

The calculation of the sound level in dB is achieved after obtaining the weighted frequency spectrum, A or C, integrating the total energy of the signal over intervals of rapid response or slow response, is for determine the level of the signal in dB (linear response, or in any weight) Fig. 4.

The direct calculation of the signal energy, defined as the sum of the squares of the magnitudes of the samples in the time domain, would perform the inverse FFT, increasing the number and complexity of the calculations to be performed. This problem can be avoided by applying Parseval's relation and to estimate the energy of the signal in the frequency domain.

Parseval's relation is given by:

$$\varepsilon_x = \sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2 \quad (6)$$

Samples of the input signal have real values in the time

domain. Therefore, according to the properties of the DFT, the frequency spectrum is symmetrical around sample $N/2$.

This allows calculation of the energy of the signal using only the first $N/2 + 1$ samples $X[k]$. Were evaluated with the corresponding weighting filter ($X_A[k]$, $X_C[k]$ or linear) as follows:

$$\varepsilon_x \approx \frac{2}{N} \sum_{k=0}^{N/2} |X_{Pond}[k]|^2 \quad \text{para } x[n] \in \mathbb{R} \quad (7)$$

The average instantaneous signal energy should be considered by interval observation of the data stream:

$$\varepsilon_x \approx \frac{1}{NT_s} \sum_{k=0}^{N/2} |X_{Pond}[k]|^2 \quad (8)$$

T_s is the sampling period of the signal, NT_s the duration of the data window (0.185s and 1.486s for slow and fast response, respectively).

The calculation of the level in dB of the energy of the signal requires a standard reference level of 0.000204 dynes/cm². It is difficult to know which input voltage is equal to the reference pressure to solve it. The signal level in dB is obtained by:

$$\text{Nivel en dB} = 10 \log_{10} \left(\frac{\varepsilon_x}{\varepsilon_{ref}} \right) = 10 \log_{10}(\varepsilon_x) - 10 \log_{10}(\varepsilon_{ref}) \quad (9)$$

$10 \log_{10}(\varepsilon_{ref})$ is a constant, so it is possible to express the signal level in dB as follows:

$$\text{Nivel en dB} = 10 \log_{10}(\varepsilon_x) + C \quad (10)$$

The calibration constant, C , can be determined experimentally.

III. RESULTS AND DISCUSSION

The GUI (GUIDE, Graphical User Interface Development Environment) of MATLAB (Fig. 5) was employed for the results presentation.

Before acquiring data, it is required to select the type of temporal response (slow or fast). Data acquisition is started / stopped by clicking on the button ON/OFF. The interface also lets you select the type of frequency weighting used, that is, weighting A, C or linear. To change the conditions of signal processing, it is needed to first stop the acquisition (OFF), and select the new parameters (response and weighting).

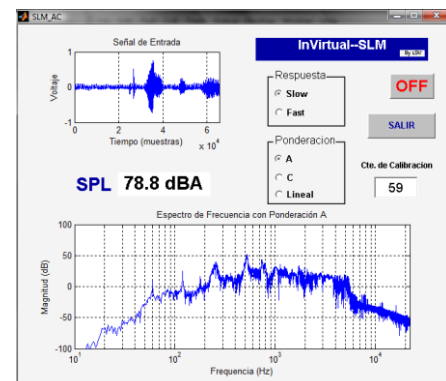


Fig. 5. User Interface virtual sound level meter, the instrument allows the choice of response and weighting.

The output window displays: 1. the input signal in the top left; 2. the weighted frequency spectrum A, C or linear and 3. The estimation of the level in dB (Fig. 6).



Fig. 6. Data displayed to the user.

It is advisable to use a larger data stream and which provides greater resolution in frequency discrimination. This enhances the functionality and usefulness of the system. The use of fast response (low resolution windows of 8192 samples) or slow response (high resolution, 65536 samples per window) is crucial as it may incur significant errors at low frequencies

The curve of relative error for weighting C shows that the error does not exceed 5 dB for most frequencies.

The accuracy is given by the value of the least significant digit of the digital word which represents the value in dB. In slow response, the accuracy is 0.1 dB, fast response, precision is 1 dB, and the variation of the last digit is so fast that no meaning to read; therefore, the accuracy is lower.

The frequency response of the microphone used significantly influences when looking for accuracy, because the calibration method used, compensates for a single point. If the sensor does not respond well to the entire frequency spectrum, the correction is needed across the spectrum.

According to tests carried out, the implementation and evaluation of SLM_AC reaches a sound level meter type or class 3, so its use is limited for surveys and approximate measurements.

It is advisable to use a larger data stream and which thus provides greater resolution in frequency discrimination, thus enhancing the functionality and usefulness of the system. The use of fast response (low resolution windows of 8192 samples) or slow response (high resolution, 65536 samples per window) is crucial as it may incur significant errors at low frequencies

The curve of relative error for weighting C shows that the error does not exceed 5 dB for most frequencies.

The accuracy is given by the value of the least significant digit of the digital word which represents the value in dB. In slow response, the accuracy is 0.1 dB, fast response, precision is 1 dB, and the variation of the last digit is so fast that no meaning to read, therefore, the accuracy is lower.

The frequency response of the microphone used significantly influences when looking for accuracy, because the calibration method used, compensates for a single point. If the sensor does not respond well to the entire frequency spectrum, the correction is needed across the spectrum

According to tests carried out accurately, implementation

and evaluation SLM_AC reaches a sound level meter type or class 3, so that its use as it stands is limited to employment for surveys and measurements approximate.

IV. CONCLUSIONS

Virtual instrumentation is an indispensable tool to study, measure and evaluate complex conditions, such as acoustic spaces.

The system that was use as virtual instrument computer (PC) and software to simulate a sound level meter, becomes a huge advantage in the design, cost and versatility of the instrument, since its functions and parameters can be easily redefined by changes in the programmed instructions.

The implemented noise measurement system presents a friendly interface to the user via a GUI and reduces time and costs in the determination of environmental noise measurements quite reliably.

REFERENCES

- [1] Choosing a Sound Level Meter. [Online]. Available: <http://www.eie.fceia.unr.edu.ar/~acustica/biblio/sonometr.htm>
- [2] The SLMMeasuring Sound, Bruel and Kjaer. [Online]. Available: http://www.conama.cl/investigacion_info/temas_ambientales/Ruido/sonometros.htm
- [3] Brüel and Kjaer, "Modular precision sound level meter-type 2231 with BZ7110," *Product Details*.
- [4] Brüel and Kjaer, "2237A controller - integrating sound level meter," *Product Details*.
- [5] Brüel and Kjaer, "Mediator 2238 sound level meter with basic program BZ7126," *Product Details*.
- [6] Ono Sokki, "Integrating sound level meters LA-5120/5110/2110," *Product Data*.
- [7] Cesva, "Sonómetro integrador de precisión SC-20," *Product Details*.
- [8] Miyara, Federico. Sound levels. Virtual Library of acoustic and electroacoustic Laboratory, National University of Rosario, Argentina, 2000, April 2000. [Online]. Available: <http://www.eie.fceia.unr.edu.ar/~acustica>.
- [9] Mexican Standard NMX-AA-059-1978. Acoustics-Precision Sound Level Meters. April 2000.



Miguel F. Rocha obtained his Degree in Electrical Engineering with specialization in Instrumentation and Control by ESIME-IPN, Mexico; He has studied Semester "II-00" of the Master in Mathematics Education Level by the Center for Research and Advanced Studies. He earned his Master of Arts with a concentration in Educational Anthropology at the University of Tepeyac. He is a graduate of the Diploma in Training and Upgrade for the New Model Teacher Education (IPN-ANUIES). Graduate Project Management in Educational Innovation at the Polytechnic University of Cataluña, Spain. He has attended various courses specific purpose, as well as education and training in educational areas, technical and administrative. In ESIME-Zacatenco México served as: professor, researcher and supervisor; Assistant Director of Educational Services and Social Integration; Deputy Director of Outreach and Academic Support; Academic Head of Automation and Control Engineering; Deputy Academic Academic Department of Control Engineering and Automation; Supervisory Control Theory Course I; and the College of Computing technical served as teacher.