Software Sound Synthesiser as a Source of Power Quality Waveforms

J.P. Braun, I.S. Burnett, V.J. Gosbell

School of Electrical, Computer and Telecommunications Engineering, University of Wollongong
jbraun@ieee.org

Abstract

The testing of power quality in the laboratory requires waveform sources to emulate the various disturbances present in electrical networks. Often arbitrary waveform generators are used to recreate these disturbances. The main limitation of this technique is the periodicity of their output signals. To circumvent this limitation, this paper proposes the usage of the Csound sound synthesiser software package as a source of power quality waveforms. This permits the creation of real time waveforms that are accurate and of long duration. It is also shown that this approach improves the testing of the conformance of power quality analysers to existing standards.

1. INTRODUCTION

Over the last few years, the topic of power quality (PQ) has been the subject of increased interest from regulators, power utilities, and consumers of electrical energy as well as from manufacturers of electric and electronic equipment. This has led to multiple studies and investigations, which have resulted in a need to recreate power quality disturbances in a laboratory environment [1].

Power quality testing in the a laboratory is mainly concerned with equipment susceptibility, emission levels, evaluation of mitigation techniques as well as the testing and calibration of power quality instruments. A lesser-known activity is the verification of the conformance of power quality recorders to PQ standards from IEEE or IEC [2-4].

This type of testing differs from the previous one in the requirements placed on the waveforms to be created. The additional constraints include determinism, long duration and flexibility. The large number of disturbances encompassed by the standards dictates both determinism and flexibility while the concept of measurement aggregation [4] calls for longer duration. These characteristics also facilitate the "black box" testing nature of a power quality analyser where the details of its implementation are generally not known to the evaluator.

First this paper reviews the various techniques used for the generation of power quality waveforms and is followed by a short introduction to electronic and computer music. This sets the foundation of why a musical synthesiser could be a solution to these extended requirements. Then it presents an implementation of a PQ generator with the CSound software musical synthesiser as well as some examples of the type of waveforms generated. Finally some initial tests on a real instrument are presented.

2. GENERATION OF PQ WAVEFORMS

Several techniques are used to generate the various waveforms associated with PQ. The most common are the use of electrical circuitry, laboratory waveform generators, arbitrary wave generators (AWG) and real time direct waveform synthesisers. The choice of the appropriate technique is dictated by the nature of the test to be undertaken as well by a trade-off between cost and performances. Generally the waveform parameters such as the frequency content and periodicity determine the generation technique.

Electrical or electronic components are generally used for the generation of simple harmonics waveforms or high frequency switching such as lightning transients. Circuits containing diodes, thryristors or triacs can generate rectified, phase-controlled or burst waveforms with harmonic and sub-harmonic content. Such techniques are used in the testing of electrical meters [5]. The discharge of capacitors into the circuit under investigation is another technique used in lightning pulse or fast transients generators. But in general, these waveform generation techniques lack flexibility.

Synthesising the waveform rather than distorting the grid waveform can increase the flexibility and accuracy. This is done either with analogue function generators and modulators or with arbitrary waveform generators where the waveform is computed and stored in a circular memory before to being "stepped through" and fed into a DAC. The generated waveforms are then applied to the load through an amplifier although several variants are possible [6,8]. The main advantages of this system are simplicity, flexibility and low computing power requirement. However, the main limitation is that all waveforms are periodical and this restricts the testing to continuous PQ disturbances such as harmonics, unbalance and flicker.

To remove the limitation of periodicity requires the direct synthesis technique, where data are produced in real time. This permits the generation of most PQ disturbance, but the computing requirements can become demanding and expensive [9].

3. ELECTRONIC & COMPUTER MUSIC

3.1 A brief history of computer music

Electronic music can be traced back to the end of WW2 and one of its aims has been to extend the range of orchestral instruments available to musicians by mean of sound synthesis [10]. In those early days, the creation and processing of electrical oscillations were achieved with analogue circuits including oscillators, modulators, filters as well as some non-linear circuits. These modules were intercommoned through patch panel that allowed flexibility of reconfiguration.

The advent of computers in the early sixties offered new possibilities. In early applications their role was limited to those of sequencers for analogue studios. However, their computational power allowed for more complex and rapidly changing waveforms that could not be achieved by analogue synthesisers.

The first musical software, Music 1, was completed in 1957 at Bell Laboratories [12]. This software had a long line of successors, which culminated in today's most popular software synthesis program CSound, designed at MIT in 1985. This program can today perform in real time on a high end PC and is continuously being improved.

3.2 The CSound software synthesiser

Since a synthesiser must create a signal representing the sound pressure waveform whose sampling rate is generally 44.1kHz for an audio application two fundamental problems arise [12]:

I. The large amount of data required in specifying the pressure waveform.
II. The need for a simple language to describe a complex sound sequence.

To solve the data problem, CSound exploits the fact that many sounds have highly repetitive components which can be recreated with lookup tables. This usage of tables removes the need for the computing of each point in real time. The available computing power is used for modulation, envelope shaping or other processing activity. For real time performance, the time scale factor must be less than one [12]:

$$\text{Time scale} = \frac{\text{Time to compute samples of a sound}}{\text{Duration of the Sound}}$$

The description of the a sound sequence in CSound is by a collection of building blocks called unit generators. These unit generators are assembled into a flowchart and stored in a file called the Orchestra file, in analogy with the description of the instruments making up an orchestra.

Figure 1 shows some of the CSound unit generators used in the design of the PQ waveform generator.

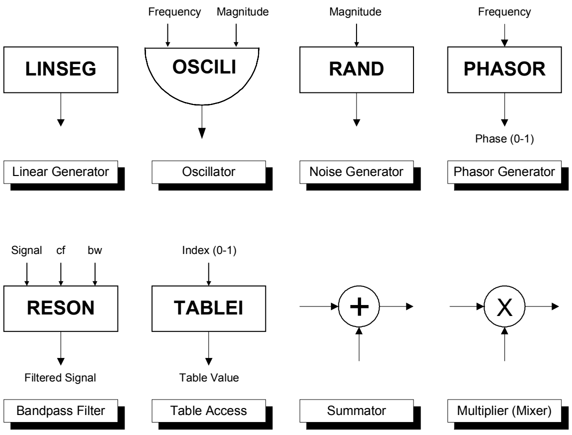

Figure 1 – CSound unit generators (Opcodes)

LINSEG	Generate a straight line or a series of line segments between specified points.
OSCILI	Oscillator whose waveform is contained in a lookup table.
RAND	White noise generator
PHASOR	Produce a normalised (0-1) moving phase value.
RESON	Second order band pass filter. Centre frequency and bandwidth are controllable.
TABLEI	Access to data table and with linear interpolation between points.

In addition to the Orchestra file there is a Score file that contains information about when and how the instruments in the Orchestra file are being played. The Score file contains one or many start and duration times for each instrument as well as parameters such as pitch, envelope duration are also passed to the instruments.

As shown in Figure 2, CSound compiles these two files and starts the generation of the sound described in these files. The output stream can either be stored in a file (WAV, AIFF, IRCAM or raw binary) or directed directly to a DAC. The data resolution can be chosen between 8 to 24 bits.

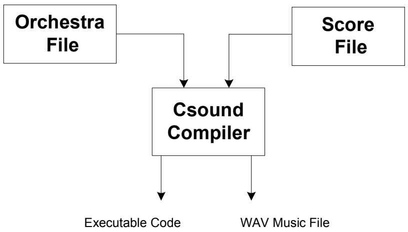

Figure 2 – CSound Orchestra and Score Files

As PQ waveforms are relatively simple, in comparison to music, they are well suited for an implementation in CSound.

4. PQ GENERATOR IMPLEMENTATION

Power quality disturbances are divided between continuous and discrete. The former are harmonics, interharmonics, unbalance and flicker, and the latter are interruptions, sags, swells and transients.

The various opcodes described in the previous section are sufficient for the creation of power quality waveforms through additive and subtractive synthesis techniques [13]. The former is a technique where a sound is made up of the addition of several signals while the later is a technique were a signal has a portion of its spectrum removed by filtering. Amplitude and frequency modulations are also used. Figure 3 shows the overall structure of the generator.

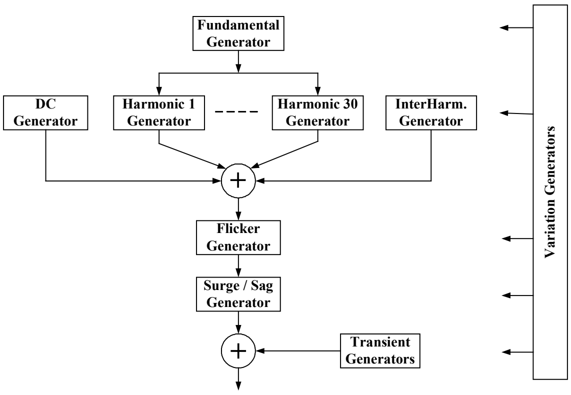

Figure 3 – Block Diagram of CSound PQ Generator

Most continuous disturbances are well suited for a look up table structure due to their periodicity. This is especially true for harmonics whose number could be large especially in a three-phase system.

The modelling of continuous disturbances with the exception of flicker can be achieved by a collection of harmonics with varying magnitude and phase [14] which are then summed up with a DC component and interharmonics. The flicker is an amplitude modulation of the composite signal just described.

Sags and swells are modelled by the amplitude modulation of the continuous waveform with the shape of the sags or swells. Transients and noise are simply signals added to the complete waveform.

4.1 Fundamental and harmonics

Figure 4 shows the opcodes used for the generation of the fundamental and its harmonics. The frequency, stated in the score file, is combined with another variable representing the instantaneous frequency variation. This sum is then converted to the instantaneous phase (0-1) of the fundamental by a PHASOR opcode. To this the phase step of a sag is added thus creating the instantaneous phase of the fundamental. Multiplying that fundamental phase by the rank of a given harmonic and then adding both its phase at time zero and its instantaneous phase variation combines then into its instantaneous phase. The phase of the fundamental and each harmonic is converted to a waveform by the TABLEI opcode. This function is actually the look up table containing the points of a sinusoidal wave.

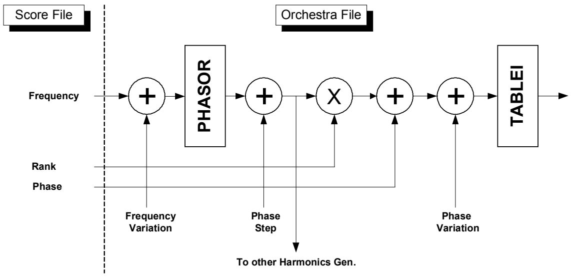

Figure 4 – Generation of fundamental and harmonics

4.2 The voltage waveform

The individual harmonic waveforms are summed up with a DC component as well as the interharmonics. However, before this summation, each component of the voltage waveform is scaled and may be amplitude modulated. Thus each component of the waveform can have its phase and magnitude controlled separately. Figure 5 shows how other disturbances are added to this composite waveform.

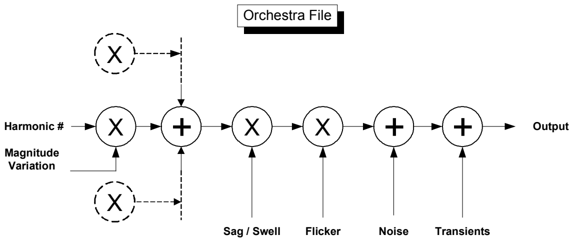

Figure 5 – Flicker, noise & discrete disturbances

Flicker, sag and swell are disturbances affecting the magnitude of the amplitude of the waveform and are implemented as multipliers (modulators). However, their respective modulation signal is different. In the case of the flicker, a continuous signal, periodic or random, is used. A variation generator (see 4.3) has been used to this end.

For the creation of sags and swells, discrete rectangular pulses are used to decrease or increase the magnitude of the waveform. However, an optional tail has also been implemented in order to test the conformance to [4] of the hysteresis present in the sag threshold of a power quality analyser. Two types of tail are available: flat and triangular. In the first case, the sag terminates by a short voltage plateau while in the second case the voltage returns to the pre sag level in a linear fashion. All of these parameters can be defined in the Score file.

Transients and noise are signals added to the complete waveforms. Individual generators create the various types of transients while a variation generator generates noise. Again all parameters are defined in the Score file.

4.3 The variation generator

The various variations of frequency, amplitude, phase and flicker are implemented with the generator shown in Figure 6.

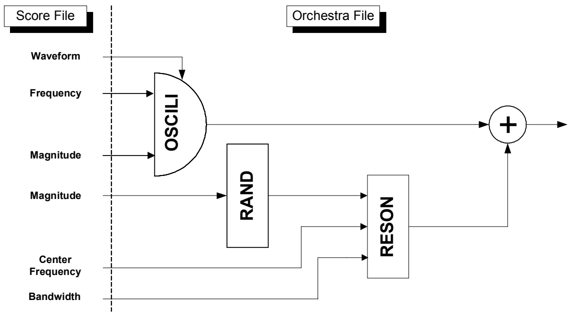

Figure 6 – Variation generator

All the variations can be either periodical or random. The OSCILI opcode is an oscillator whose waveform is stored in a table. Three waveforms have been selected: sinusoidal, square and triangular. On the other hand, the RAND opcode is a white noise generator whose output a is filtered by RESON, a two poles, band pass filter with a centre frequency and bandwidth controllable by the user. Both signals can be generated separately or simultaneously.

5 RESULTS & INITIAL APPLICATIONS

Figure 7 to 10 show some example of waveforms.

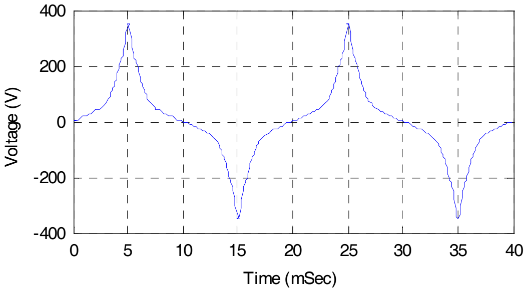

Figure 7 – IEC Composite harmonic waveform

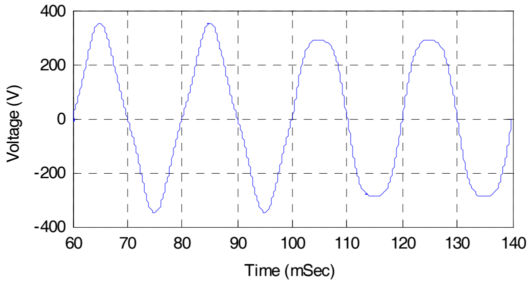

Figure 8 –180 Deg. phase step in 3rd harmonic

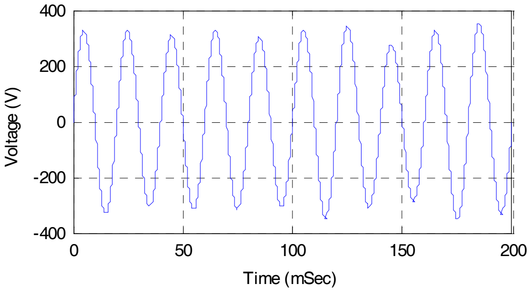

Figure 9 – Flicker (Random envelop)

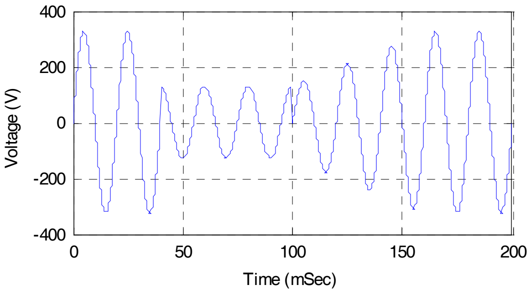

Figure 10 – Sag (90 deg. phase jump) with triangular tail end

Figure 7 shows an example of a 50 Hz IEC test waveform [15] containing harmonics to the 30th rank. This waveform is extremely rich in harmonics whose phase are either 0 or 180 degrees. Figure 8 shows a case where the harmonic content is 10% of third harmonic, but containing a periodical phase change of 180 degrees. This has been achieved with a variation generator controlling the phase of that harmonic. This shows the flexibility offered by the independent control of each harmonic parameter. Figure 9 is an example of flicker by a random generator. The centre frequency of the generator is 15Hz and a bandwidth of 8Hz. The amplitude of the 50 Hz signal is modified by plus or minus 7%. Figure 10 shows a sag down to 40% for a duration of three cycles and a phase jump of 90 degrees. The tail of the sag lasts also three cycles and is of triangular shape.

5.1 Accuracy of waveforms

One of the limitations of waveform generation with pre-stored waveform points is the interpolation accuracy between two points. CSound uses tables of size 2^n, with n being selected by the user, as well as linear interpolation. Thus it is important to choose a value of n that brings the error below the quantisation error [16]. This is especially true when several harmonics are combined.

To evaluate the actual accuracy of a given waveform, the following test system has been designed. The data generated by CSound are stored in file and imported into Matlab. The CSound data (16 bits) are compared with an identical waveform generated by Matlab, but whose data type is double precision (64 bits). Figure 11 illustrates this.

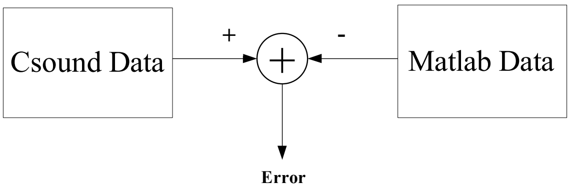

Figure 11 – Testing of CSound data accuracy

Figure 12 shows the error for a 50 Hz waveform containing no harmonics. The error is within the quantisation error of 0.01V. (Full scale +/- 350V and 16 bits of resolution). The sampling rate is 10240 samples per second and the lookup table is 1024 points long. Thus no interpolation is required as the lookup table increment is an integer [16].

$$\text{increment} = \frac{\text{(freq)(table length)}}{\text{(sampling rate)}} = \frac{50 \times 1024}{10240} = 5$$

Changing the increment to 4.5 when the sampling rate is set to 11377 samples per second does not degrade the maximum error. However, with an increment equal to pi, the table length needs to be doubled to maintain the same maximum error.

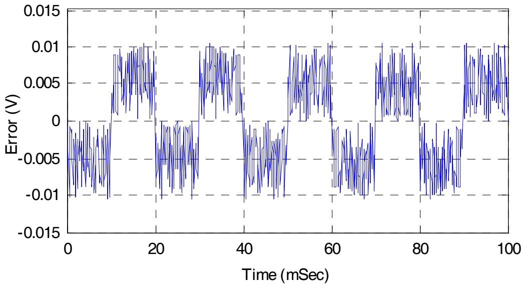

Figure 12 – Testing of CSound data accuracy

5.2 Initial applications

While a complete system is still under development, the waveform generator has been tested in conjunction with a commercial power quality analyser from CHK Wireless Pty. Ltd. In this initial test, an audio output of a PC has been connected directly to the front-end circuitry of a PM30 analyser. This permitted to make some initial testing of the waveform generator.

5.3 Change in harmonic level

In this first test, a pure 50 Hz waveform is first sent to the analyser. After 30 seconds, 20% of third harmonic is added to the waveform. This increased rms voltage triggers a waveform capture of six cycles. Figure 13 shows the distorted waveform.

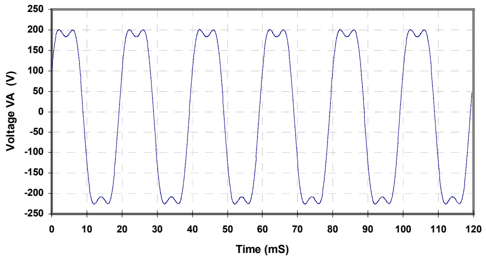

Figure 13 – Captured waveform

Such as test would have been difficult with an arbitrary waveform generator as it would not be capable to generate a single event on a continuous sine wave.

5.4 Sag testing

In this second test, again the “waveform” capture has been enabled and the sag threshold set for 0.9 pu of the nominal voltage. The CSound generates a single sag lasting two cycles and with a phase jump of 45 degrees and with no tail. Figure 14 shows the captured waveform. Again the usage of an arbitrary waveform generator would have not permitted such a test as a periodical sag would have rapidly saturated the PM30's memory.

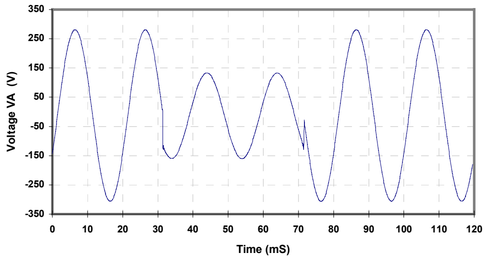

Figure 14 – Sag waveform recorded by the PM30

5.5 Hardware requirements

As these tests were part of a preliminary testing of the generator, it has been found that ordinary sound cards on PC boards have several limitations:

- High level of noise
- Unstable codec clock frequency
- No DC path

A sound card of higher quality can resolve the first two issues. The last may require a hardware modification of the CODEC section of a sound card. This task could be combined with the design of a power amplifier capable of drive directly a power quality analyser.

6. CONCLUSIONS

This paper has shown that the CSound sound synthesiser software can be used as power quality waveform generator in a PQ test system. The waveforms that have been generated with CSound demonstrate that this freeware is capable of synthesising real time waveforms that are both accurate and of long duration. Initial testing with a commercial PQ analyser has also demonstrated that these features enable tests that would be difficult to reproduce with an arbitrary waveform generator. Thus waveforms generated by CSound can improve laboratory testing of power quality and greatly facilitate the testing and evaluation of PQ generators.

7. REFERENCES

[1] Ward J., "Power quality laboratory testing", IEEE Power Engineering Review, February 2002, p13-15

[2] IEC-61000-4-7, "Testing and measurement techniques - General guide on harmonics and inter-harmonics measurements and instrumentation, for power supply systems and equipment connected thereto", July 1991

[3] IEC-61000-4-15, "Testing and measurement techniques - Flickermeter – Functional and design specifications", November 1997

[4] IEC-61000-4-30, "Testing and measurement techniques - Power quality measurement methods", draft issue July 2001

[5] AS1284.5 " Electricity metering – General purpose electronic watthour meters", January 2000, p39-43

[6] V.J. Gosbell, B.S.P. Perera, P. Cooper, A. Jalilan, "A 10 kVA load power quality testing facility", Proceedings of the 8th ICHQP conference, October 1998, p249-254

[7] D. Robinson, S. Perera, V. Gosbell, "Waveform generator for load susceptibility testing", AUPEC 99, September 1999

[8] Evelyn Matheson, Annette von Jouanne, Alan Wallace, "A remotely operated power quality test platform based on a 120 kVA fully programmable three-phase source", Conference record of the 2001 Industry Applications Conference, Vol 2, p1188-1195

[9] E. Acha, O. Anaya-Lara, J. Parle, M. Madrigal, "Real-time simulator for power quality disturbance applications", Proceedings of the 9th International Conference on Harmonics and Quality of Power ICHQP 2000, October 2000, p763-768

[10] A. Douglas, "Electronic music production", Pitman Publishing, 1973

[11] R. Boulanger, "The Csound book", The MIT press, 2000

[12] M. Mathews, "The technology of computer music", The MIT Press, 1969

[13] C. Dodge, T. Jerse, "Computer music", 2nd ed., Schirmer Books, 1997

[14] A. Mansoor, W.M. Grady, A. Chowdhury, M.J. Samotyj, "An investigation of harmonics attenuation and diversity among distributed single-phase power electronic loads", IEEE Transaction on Power Delivery, Vol 10, No 1, January 1995, p467-473

[15] IEC-61000-3-2, "Limits for harmonic current emissions (equipment input current less than or equal to 16A per phase)", 1995

[16] R. Moore, "Table lookup noise for sinusoidal digital oscillators", Computer Music Journal, April 1977, p26-29