Noise generator for tinnitus treatment based on look-up tables

Alejandro J. Uriz\(^1\), Pablo Agüero\(^2\), Juan C. Tulli\(^2\), Jorge Castiñeira Moreira\(^1\), Esteban González\(^2\), Roberto Hidalgo\(^2\), Manuel Casadei\(^2\)

\(^1\) CONICET / Laboratorio de Comunicaciones - Facultad de Ingeniería - Universidad Nacional de Mar del Plata
\(^2\) Laboratorio de Comunicaciones - Facultad de Ingeniería - Universidad Nacional de Mar del Plata

E-mail: {ajuriz,casti}@fi.mdp.edu.ar

Abstract. Treatment of tinnitus by means of masking sounds allows to obtain a significant improve of the quality of life of the individual that suffer that condition. In view of that, it is possible to develop noise synthesizers based on random number generators in digital signal processors (DSP), which are used in almost any digital hearing aid devices. DSP architecture have limitations to implement a pseudo random number generator, due to it, the noise statistics can be not as good as expectations. In this paper, a technique to generate additive white gaussian noise (AWGN) or other types of filtered noise using coefficients stored in program memory of the DSP is proposed. Also, an implementation of the technique is carried out on a dsPIC from Microchip\(^\text{®}\). Objective experiments and experimental measurements are performed to analyze the proposed technique.

1. Introduction
The noise is usually considered an undesirable signal which should be removed from the desired signal. However, noise is also an useful tool for the characterization of systems \([1, 2]\). It can be used for instance for the analysis of communication systems \([3, 4]\) or for characterising filters \([1]\).

In recent years, the noise has been used for medical purposes such as treatment of tinnitus \([5, 6, 7]\). In this way, a noise synthesizer is used to mask the tinnitus and, consequently, improve the quality of life of the individual who suffers from this condition \([6, 7]\). Thus, it is convenient to implement a noise generator (based on a pseudorandom numbers generator) in assistive listening devices, which actually are based on a digital signal processor (DSP). In this sense, it is possible to find digital hearing aid devices\([8, 9]\) which implement noise generators based on fractals or more complex noise generators.

One of the limitations found in performing this task is the complexity of the algorithms necessary to synthesize white noise with the appropriate statistical properties. In some cases, white noise must be filtered through a digital filter to obtain the appropriate response to mask tinnitus, which generates an additional computational load to the system, which may not be acceptable because this functionality is usually not the principal task in assistive listening devices. There are papers in the literature \([10, 11]\) where noise generators are described, but those require a DSP exclusively dedicated to their implementation, and they were not developed for tinnitus treatment and the considerations required \([6, 7]\) for this application.
In this work, a technique for generating sequences of pseudo-random numbers in a microcontroller using tables stored in the program memory is presented. The proposal technique improves system performance mainly because filters are not necessary into the DSP to obtain noise with specific features, and furthermore the castings between numbers formats are reduced.

Objective and subjective experiments and, finally, experimental measurements are carried out to analyse the performance of the technique. Moreover, an implementation performed in a Microchip® dsPIC [12] is described.

This paper is organised as follows. In Section 2, the most important aspects of the tinnitus masking techniques are presented. In Section 3, a new method is proposed. In Section 4, the results of the objective and subjective tests are presented and discussed. Finally, the main conclusions are summarised in Section 5.

2. Tinnitus masking using sounds
Tinnitus is a hearing condition that can be very rarely cured, even using medical treatments and surgical methods. In recent years methods to improve the quality of life of individual with tinnitus were developed. One of the most used approaches are methods based on sound therapies, which reduce the perception of tinnitus in the individual. There are various types of therapies based on sounds, in the following subsection one of the most commonly used methods are presented.

2.1. Tinnitus treatment using masking sounds
This kind of therapy can be divided into two subgroups: techniques based on zen music, and based on colored noise [5], [6], [7]. In the first subgroup of therapies techniques based on natural sounds such as the sound of rain or waterfalls, among others, are included. These sounds can be recorded or, as in the case of some high-end assistive devices[8], they may be generated using mathematical algorithms.

The second subgroup includes techniques based on colored noise. These use the spectral masking phenomenon between components[13],[14]. Masking is defined as the process taking place in the auditory system of a person, where a sound of a certain energy can reduce the perception of another sound which is spectrally adjacent to it.

There are three possible situations defined between the tinnitus and the sound generated for masking it. Firstly, if the sound does not change the perception of tinnitus, there is not masking between components. Moreover, if the sound level generated a partial alteration of tinnitus is said that the tinnitus is partially masked by the external stimulus. Finally, in cases where tinnitus is not perceived due to the external stimulus, it is said that first component has been completely masked by the masker. Note that when applying these therapies, care should be taken that the external stimulus not causes damage to the auditory system of the patient.

Therapies based on colored noise attempts to partially or fully mask the tinnitus using synthesised noise. Actually, this technique is included in high-end assistive listening devices[9], where the colored noise is obtained by filtering AWGN. In these devices, the filtering is done by means of an equalizer with up to 16 subbands, which are also divided into four bands located between 100Hz-500Hz, 500Hz-1400Hz, 1400Hz-3500Hz and 3500Hz-8000Hz. In recent years, our research Group has developed a digital hearing aid device, which has been patented [18]. Simultaneously, a tinnitus treatment method has been presented [17]. Thus, in the next section, a method to implement a pseudo-random number generator in the architecture of the DSP used in the developed hearing aid (or another microcontroller) is presented.

3. Proposed Technique
In cases where according the needs of the individual, filtered noise is necessary, this signal is obtained by filtering a white noise source. In these cases, the hearing aid device must be
capable to compute the filter operation, which in several cases requires a considerable amount of resources.

Thus, the main goal of this work is to implement a noise synthesiser in a hearing assistance device, with a substantial reduction of DSP operations. Also, the system must perform simultaneously other functions such as audibility extender, noise reduction, etc. Furthermore, it should be noted that a microcontroller has more program memory (EPROM) than data memory (RAM), which usually is a limitation in this kind of design. Therefore, there is a tendency to store constant data (such as coefficients for windowing, or constants for computing a Fast Fourier Transform (FFT), etc.) in EPROM. In this paper, a pseudo-random number generator which uses coefficients stored in the program memory of the microcontroller is proposed.

The first step of the technique is to generate segments containing pseudo-random values on a PC using a pseudo-random number generator (for example the `rand()` or `randn()` functions of MATLAB®), which is chosen depending on the nature of the noise to be synthesised, additionally the signal could be filtered using MATLAB® according to the needs of the individual. This filtering can be done using the `filter()` function of MATLAB® or a more complex subroutine.

Then, the generated segments of noise are stored in the program memory of the DSP. The coefficients are stored in program memory in a matrix of 256 columns and $M$ rows, which in the implementation in a microcontroller, will be replaced by a table with $256 \times M$ elements. Then, a random pointer, pointed to a specific program memory location (which is generated using the architecture of the microcontroller), can choose which of the rows of the table will be used to synthesise each segment. Consequently, the order of appearance of each row in the synthesised signal will depend on the jump generator.

![Figure 1. Architecture of the proposed table of data to generate noise.](image)

One of the advantages of the proposed method is that it can be implemented in a microcontroller without an architecture dedicated to processing mathematical operations. This is because, if it is desired to synthesise filtered white noise through a specific filter, segments of the desired signal will be stored in the program memory of the microcontroller. Firstly, it avoids noise filtering on chip, according to the needs of the user and, secondly, the casting between numbers formats is also avoided. Thereby, once the microcontroller was programmed, the operation of the proposed noise generator is limited to generate a random integer which defines which row vector of the matrix of coefficients will be selected to synthesise each signal segment.

Then, in this paper it is assumed that vectors of length $N = 256$ are generated to analyse the generator. Then coefficients generated using MATLAB® are stored in a table that contains 256 columns. The number of rows $M$ of the table is adjusted depending on the amount of memory available in the microcontroller. Figure 1 shows a diagram of the proposed implementation. It can be seen that the matrix of coefficients generated contains $N = 256$ columns and $M$ rows. The proposed technique works according to the scheme shown in Figure 2.
For each segment of the signal to synthesise, an integer $0 < j < M$ is randomly generated. Then, this value represents the row of the matrix used for synthesising the frame. This is performed in the microcontroller by means of a pointer to the starting position according to the row ($POS + j \times 256$), where the constant $POS$ represents the initial position of the table in the program memory, then the following 256 elements are used to synthesise the signal.

Note that by increasing the value of $M$ a noise generator with better performance is obtained from the statistical point of view, since when having more values per column, the repetition period of each value is also increased. Another interesting aspect of the proposed technique is that the type of noise of the synthesiser can be modified by simply modifying the values stored in the table. Thus, if for example, $\text{randn()}$ is used instead of using MATLAB® function $\text{rand()}$, the pseudo-random number generator output signal is obtained with Gaussian distribution. Finally, as was described previously, another option to change the features of the output signal (which is appropriated to set the noise features to the specific needs of an individual who suffer tinnitus) is to change the coefficients of the filter used to filter the noise. In the next section the experiments and measurements carried out to analyse the performance of the system are presented.

4. Experiments

With the aim of testing the performance of the developed method, a set of objective and subjective experiments, and finally experimental measurements, were performed. As was previously described, the main goal of this work is to implement a filtered noise generator in a hearing aid device based on a dsPIC33EP256MU806 of Microchip® [12], which has 256 kB of program memory. Previous works [17, 19] establish that only 10% of program memory is required to implement the most important functions of a hearing aid in the chosen dsPIC. Thus, as a design rule, it is established that the proposed pseudo-random generator is implemented using up to 80% of EEPROM, in this manner the last 10% of EEPROM will be reserved to future updates of the software. In these conditions, there are 209715 bytes assigned to implement the
Figure 3. Probability Density Functions (PDF) of noise with gaussian distribution generated using the proposed method for several values of $M$. Also, the reference PDF is shown using a black line.

generator, then, by setting $N = 256$, and 16 bits data, the maximum value of $M$ is 409. A second hearing aid device implemented on a dsPIC33EP512MU806 of Microchip® [12], which has 512 kB of program memory was also included in the experiment, taking into the account the same considerations than for the other device, the maximum value of $M$ is 972.

Based on this limitation, configurations of the proposed technique with $M = 40$, $M = 100$, $M = 400$, $M = 950$ and $M = 10000$ will be used in the experiments.

4.1. Analysis of the obtained Probability Density Functions
To appreciate one of the effects of choosing the value of $M$, Figures 3 and 4 show the Probability Density Function (PDF) of two implementations of the pseudo-random number generator. Figure 3 shows in red colour the PDFs of the system configured as an AWGN generator for the six values of $M$ previously adopted, and in green colour for the function `randn()` of MATLAB®. As a reference, using a black line, the ideal PDF function of a AWGN was also drawn in each figure. It is possible to see that, according to expectations, for bigger values of $M$, the PDF obtained tends to the reference PDF.

In Figure 4 the experiment is repeated, but in this case, the generator was configured for a uniform distribution, and the function `rand()` of MATLAB® was adopted as reference. The observed tendency is similar to the results obtained for the Figure 3.

4.2. Analysis using the Marsaglia toolbox
In this Subsection a set of objective experiments are performed in order to analyze the statistical features of the pseudo-random number generator. For this proposal the Marsaglia toolbox [15], a set of statistical tools, was used. This toolbox is widely used to test pseudo-random number generators and it is disposal in the website of the Florida State University [21]. This tool allows to perform 15 tests, which give as result p-values which are in the range $0 < p < 1$[15].

In this case, the pseudo-random number generator was configured as an uniform distribution noise generator without filtering, and several values of $M$ were used to study the influence of this parameter in the noise signal obtained. The results are obtained for the reference method and for the proposed method with $M = 100, 400, 1000, 10000, 100000, 400000$ and $1000000$ were
obtained. In this case, it was verified that with larger values of $M$, the median value tends to 0.5 and the first and third quartile tends to 0.25 and 0.75, respectively. Then, a large value of $M$ is necessary to ensure a proper performance of the method, which exceed the limit established previously ($M = 972$). But, in order to analyze the feasibility of using the proposed method for tinnitus treatment, in the next subsection a set of experimental measurements are presented.

4.3. Experimental measurements: Implementation on a dsPIC

With the aim of testing the performance of the system, it was implemented in a dsPIC33EP256MU806 [12] of Microchip, which was adopted in the developed hearing aid [17, 19]. The language tool-suite chosen was C30 [22], which has libraries that contain functions appropriated to digital signal processing, then this toolbox can be used to implement the most important functions of a hearing aid device in a very efficient way.

The proposed technique was successfully implemented, in this case segments of noise of $N = 256$ were generated and a sampling frequency of $F_s = 16384$ Hz was used. Then, several experimental measurements were done. First, an acquisition time 15.63 ms was measured for each frame. Consequently, in order to operate the system in real-time, the processing time must be lower than this limit. The measured processing time was 3.32 ms, which is according to expectations. Also, the program and data memory usages, were 79% and 40%, respectively. Finally, direct memory access (DMA) was used to acquire and synthesise data in a more efficient way. In Figure 6 the block diagram of the developed system is shown. It can be seen that the system is composed by two main blocks: a digital signal processor (in this case: dsPIC33EP512MU806) and an audio codec PCM3052 [16], which acquires the analog data, digitalises it and sends it to the DSP. Then, once the information is processed, it is returned to the codec, which by means of a delta-sigma DAC synthesises it. In this experiment the input signal is zero, thus the output signal only contains the noise synthesised.

In order to test the performance of the system, the testing bench shown in Figure 5 was mounted. In this case, the system’s output was measured with a PDF meter [23]. The outputs
of the PDF meter must be analysed using a oscilloscope, which must be configured in XY mode.

For the experiment, a 2Vpp output noise wave was synthesised using two devices: the system under study (with $M = 400$ and $M = 950$) and a computer (using WAVE files). The sampling frequency and the data path of each device were the same (16kHz, 16 bits). The output of each system was compared using the PDF meter, whose output is shown in Figure 7. In the figure is seen that both probability density functions are similar.

Finally, the output of the system and the computer were compared using a spectrum analyser. Results obtained using four filtering configurations are shown in Figs. 8, 9 and 10 where it is

Figure 7. Screen captures of an oscilloscope for the probability density functions measured. Left figure shows the PDF for a WAVE file which contains a signal generated using the randn() MATLAB®. Middle and right figures show the measured PDF obtained the proposed technique for AWGN without filtering with $M = 400$ and $M = 950$, respectively.
Figure 8. Spectrum of output signals using a LPF with $f_C = 1000\,Hz$. Left figure shows the reference (using the `randn()` MATLAB® function). Middle and right figures show the cases where $M = 400$ and $M = 1000$.

Figure 9. Spectrum of output signals using a BPF with $f_{C1} = 1000\,Hz$ and $f_{C2} = 3000\,Hz$. Left figure shows the reference (using `randn()`). While, middle and right figures sketch the spectrum obtained with $M = 400$ and $M = 1000$.

possible to seen that the spectra of the sound generated by means of the computer is similar to the generated using the hearing aid device. In Figure 8 are shown the spectrum of the output signal using a low pass filter (LPF) with $f_C = 1000\,Hz$. In Figure 9 the spectrum of the output signal using a band pass filter (BPF) with $f_{C1} = 1000\,Hz$ and $f_{C2} = 3000\,Hz$. In the last place, in Figure 10 the spectrum of the output signal using a BPF with $f_{C1} = 1000\,Hz$ and $f_{C2} = 2000\,Hz$.

4.4. Subjective experiments
Subjective experiments were done in order to test the performance of the proposed algorithm with values of $M$ in the implementable range ($M < 950$). 15 volunteers (which accepted an informed consent) were asked to listen to the synthesised algorithms in random order. Listeners were asked to judge the similarity between two noises using a 5-point scale, from 1 (noises are distinguishable) to 5 (sounds are identical). The aim of this test is to analyse if a human can perceive a difference between a sound synthesised using the MATLAB `randn()` and `filter()` functions and the proposed technique, in this last case for several values of $M$. Although, 15 volunteers is not the amount of population needed to determine a conclusion, it is possible to obtain a trend. As a further work, more complex subjective experiments with a bigger population will be carried out.
Figure 10. Spectrum of output signals using a BPF with $f_{C1} = 1000\text{Hz}$ and $f_{C2} = 2000\text{Hz}$. Left figure shows the reference (using randn()). Middle and right figures presents $M = 400$ and $M = 1000$.

| Method | $M = 40$ | $M = 100$ | $M = 400$ | $M = 950$ |
|--------|----------|-----------|-----------|-----------|
| MOS-S  | 4.10     | 4.35      | 4.80      | 4.95      |

In Table 1 it is possible to see that the value of the similarity mean opinion scores (MOS-S) is increased with the value of $M$, which is according to expectations. Also, all the MOS-S values obtained in the experiment are bigger than 4. Thus, even in the lower adopted values of $M$, the listeners were not able to distinguish between signals. Thus, although the method only obtains good scores in the Marsaglia test for values of $M > 400000$, human listeners are not able to distinguish the difference between noise synthesized using the randn() function, and the proposed technique (even for values of $M$ in the required interval).

5. Conclusions
In this work, a technique to generate filtered noise by means of a pseudo-random number generator is a microcontroller is presented. The main aim of this method is to simplify the design of noise generators in a DSP and, consequently, it allows help people who suffer tinnitus.

It was demonstrated that the technique allows to successfully generate filtered noise reducing the amount of floating point operations (the most computational expensive task). Also, the method was implemented in a hearing aid device developed by the group. In order to validate the successful operation of the system, a set of subjective and objective test were carried out, and also experimental measurements were performed.

Firstly, objective experiments were developed to analyse and validate the performance of the technique. Then, once theoretical aspects of the technique were validated, the effectiveness of masking tinnitus was analysed to mask tinnitus. In this case, a set of volunteers heard a set of audios with a simulated tinnitus, and also different types of colored noise were added. At a first glance, the proposed technique obtains scores similar to other baseline methods. However, a set of more complex subjective experiments will be necessary to determine if the method can accomplish the expectations.

Finally, a set of experimental measurements were performed in order to analyse the operation of the technique in the hearing aid. In this experiment it was demonstrated that the system
operates in real time, which is essential for this application. As a further work, it is necessary to carry out more extensive subjective experiments, in order to validate the performance of the proposed method for tinnitus treatment. Also, it is desired to study and implement in a DSP a set of most complex noise generators oriented to tinnitus treatment.

Acknowledgement
The authors would like to thank to CONICET and and National University of Mar del Plata for the funding received through projects Secuencias caóticas digitales en procesamiento y encriptado de señales and Análisis de señales destinados a mejorar la calidad de vida, respectively. The authors would like to thank the cooperation of PhD. Luciana de Micco for her assistance in the work.

References
[1] B.M. Oliver and J.M. Cage, Electronic Measurement and Instrumentation. McGraw Hill, 1971.
[2] A. Papoulis, Probability, Random Variables and Stochastic Processes. Mc Graw Hill, 2002.
[3] J.L. Danger, A. Ghazel, E. Boutilhon and H. Laamari, Efficient FPGA Implementation of Gaussian Noise Generator for Communication Channel Emulation. IEEE ICECS. Vol. 1. pp. 366-369. 2000.
[4] A. Ghazel, E. Boutilillan, I.L. Danger, G. Gubril and H. Laamari. Design and Performane Analysis of a High Speed AWGN Communication Channel Emulator. IEEE PACRIM Conference. pp. 374-377. 2001.
[5] J. A. Henry, T. L. Zaugg, P. J. Myers and M. A. Schechter, Using therapeutic sound with progressive audiologic tinnitus management, Trends on amplification, Vol. 12, No. 3, pp.188-209, 2008.
[6] A. J. Schleiencing and R. M. Johnson, Use of masking for tinnitus, The International Tinnitus Journal, Vol.3, No.1., pp. 25-29, 1997.
[7] H. Feldmann, Masking of tinnitus - Historical remarks. In Proceedings III International Tinnitus Seminar, pp. 210-213, 1987.
[8] Widex Inc.http://www.widex.com/, Lyenge, Denmark, 2013.
[9] Siemens Hearing Instruments, Introduction to tinnitus management, http://hearing.siemens.com, 2008.
[10] R.M Rodriguez-Osorio, A.D.C. Urbina, L.H. Ariet, M.C. Ramon and M.G Sanche. A DSP-based impulsive noise generator for QoS and EMC tests in wireless systems, 60th IEEE Conference: Vehicular Technology Conference, 2004. Volume: 6. 2004.
[11] M. Sadasivam, J. Waghohilak and H. Sangjin, A look-up based low-complexity parallel noise generator for particle filter processing. In Proc. Of International Symposium on Signals, Circuits and Systems 2003 (SCS 2003). Vol. 2. pp. 621-624. 2003.
[12] Microchip Inc., dsPIC33EP512MU810 Data Sheet, High Performance 16-bit Digital Signal Controllers. http://www.microchip.com/, 2011.
[13] H. Fletcher, Auditory Patterns, Modern Physics Vol. 12, No. 1, pp. 47-65, 1940.
[14] T. Painter and A. Spianias, Perceptual coding of digital audio. In Proceedings of the IEEE, Vol. 88, No. 4, pp. 451-513, 2000.
[15] G. Marsaglia and W.W. Tsang, Some difficult-to-pass tests of randomness, Journal Statistical Software, Vol. 7(3). 2002.
[16] Texas Instruments,PCM3052A Datasheet, http://www.ti.com/, 2005.
[17] A.J. Uriz, P.D. Agüero, J. Castiñeira Moreira, J.C. Tulli, E.L. González, G. Moscardi and E.E. Sajama. A development and implementation of a tinnitus treatment method. Journal of Physics: Conference Series (JPCS). Londres: IOP SCIENCE. 2013. ISSN: 1742-6596
[18] A.J. Uriz, P.D. Agüero, J. Castiñeira Moreira, J.C. Tulli, E.L. González and F. Denk. Método y Dispositivo de Asistencia auditiva. Patente de Invención - INPI:20130103576 . Buenos Aires, Argentina. 2013.
[19] A.J. Uriz, P.D. Agüero, J. Castiñeira Moreira, J.C. Tulli, E.L. González and R.M Hidalgo. Low complexity noise power estimator for speech enhancement implemented on a dsPIC. Anales del CASE 2014. Buenos Aires, Argentina. Agosto 2014.
[20] G. De Sebastian, J.J. Badaroco and D.G. Postan. Audiologia Práctica, Ed. Médica Panamericana, 1992.
[21] G. Marsaglia, The Marsaglia Random Number CDROM, with The Diehard Battery of Tests of Randomness. Disponible en: www.stat.fsu.edu/pub/diehard. 1995.
[22] Microchip Inc. 16-Bit Lenguage Tools Libraries. http://www.microchip.com/, 2005.
[23] E.L. González. User manual Probability Density Function meter. http://www3.fi.mdp.edu.ar/mediciones/apuntes/ManualPdmetro.pdf